CONSTRUCTIONISM 2020

The University of Dublin

Trinity College Dublin

IRELAND

"26-29 May"

Constructionism 2020

26th to 29th of May 2020 in Dublin, Ireland

Proceedings of the 2020 Constructionism Conference

Edited by Brendan Tangney, Jake Rowan Byrne and Carina Girvan

ISBN 978-1-911566-09-0

Foreword

Little did we imagine, back in early 2019, when we issued the call for submissions for the conference, how prescient our words would turn out to be. Back then we wrote that "*We certainly are "living in interesting times". Times which require all of us working in education to do everything we possibly can to provide everyone with the best possible education to help them navigate and construct the "brave new world" in which they will live."*

The brave new world to which we referred was one in which information technology - in all its manifestations, AI, robotics, IoT etc. - would be a significant driver in changing how we lived, worked and educated young people. The call for papers sought contributions from our community of practitioner-researchers as to how Constructionism could contribute more fully to teaching & learning in this time of change.

The advent of the Covid-19 pandemic has of course changed the landscape in a radical way. As we write the pandemic is an unfolding tragedy, varying only in degree, across all parts of the world and of course the face-to-face gathering in Dublin had to be cancelled. Nevertheless the conference participants were adamant that a proceedings be produced.

This volume includes 44 full papers and each of the keynote speakers also produced a paper elaborating on the themes they would have covered in their talks. There is a strong element of practitioner-researcher in the community and this is reflected in abstracts for panels, demonstrations, workshops and posters which feature in this volume. However these abstracts cannot really do justice to the rich diversity of learning experiences they describe or substitute for the hands-on experiences that would have occurred.

The call for submissions sought to extend the Constructionist dialogue beyond its traditional base of STEM (and coding in particular) and, while the number of such submissions in the proceedings is modest it does include submissions on art, music, drama, social science, civics and geography. But it is the element of dialogue which suffered most from the conference gathering not taking place. We are sure that if the conference had gone ahead the keynote presentations, and the conference chairs, would have provoked a "lively conversation" on how the community sees itself and that at least some of the major research challenges facing the field would have been debated, thus helping to shape the research direction of Constructionism going forward.

The Covid-19 pandemic has brought about change at a rate we could not have imagined. In our domain of teaching & learning schools shut and to the best of their abilities moved on-line. Technology, which has not to date led to the widescale re-imaging of education which at least some in the Constructionist community have long argued for, overnight became central to the way in which teaching and learning takes place. Teachers, many of whom belonged to the "late majority" of technology adopters, have availed of the myriad of professional development opportunities which have sprung up in response to the move to on-line and are embracing the use of technology on a scale which would have been unimaginable a few months previously.

This transition to on-line has of course not been smooth. The inequalities of the digital divide have been shown in stark relief and far too many students, and their families, have been caught on the wrong side of that divide. It is very difficult to learn, or study for major exams, in a bedroom you share with two siblings, using only a mobile phone and a costly data package!

Furthermore the adoption of technology for on-line teaching has in many cases followed a substitution paradigm with the traditional "chalk and talk" paradigm now taking place on a different medium. In many cases even synchronous classes proved a bridge too far with technology being used to disseminate lessons and collect homework.

While these observations are drawn largely from the Irish experience and are based on (well informed) anecdotal evidence we expect that the situation is not untypical of what is happening in many places and will in due course be backed-up by a more rigorous research evidence base.

A phrase which is commonly brandied about at present is the “new normal” which refers to how all aspects of society will operate as we await a vaccine for the virus, attempt to recover from the economic shock and avail of the opportunity to learn from the experience and re-imagine how things can be done better for everyone, and for the planet, going forward. Surely education must be central to this re-imaging process and the challenges for the Constructionist community, as alluded to in the call for conference submissions, are to outline what role our pedagogy could play in the “new normal” and to endeavour to make sure those ideas make the difficult transition from a minority of innovative places of learning to the mainstream.

This set of proceedings will sit in the archive of Constructionist conferences, going back to Paris in 2010, and in the wider Constructionist literature as a check-point reflecting the thinking and activity of the community just prior to the pandemic. It should, at the very least, make for interesting reading in the years to come when we look back and reflect on the shape of the educational new normal and the role which Constructionism plays in it.

As the Irish poet, W.B. Yeats, put it, writing in a different context, *“All is changed, changed utterly.”* It remains to be seen what sort of *“terrible beauty is born”*.

We would like to thank: the Programme Committee for reviewing all the submissions received, Jane O’Hara for administrative and planning support, the “three wise men”, and all the authors who took time to revise their submissions when there was a lot else going on!

Conference Co-chairs

Brendan Tangney, The University of Dublin, Trinity College Dublin

Jake Byrne, The University of Dublin, Trinity College Dublin

Carina Girvan, Cardiff University

Dublin 26th May 2020

Programme Committee

Aibhín	Bray	The University of Dublin, Trinity College Dublin
Deirdre	Butler	Dublin City University
James	Clayson	The American University of Paris
Cornelia	Connolly	National University of Ireland Galway
Valentina	Dagiene	Vilnius university Institute of Mathematics
Gerald	Futschek	Vienna University of Technology
Keith	Johnston	The University of Dublin, Trinity College Dublin
Marianthi	Grizioti	National and Kapodistrian University of Athens
Tony	Hall	National University of Ireland, Galway
Brian	Harvey	University of California, Berkeley
Nathan	Holbert	Columbia University
Stephen	Howell	Microsoft Dublin
Yasmin	Kafai	University of Pennsylvania
Ivan	Kalas	Comenius University, Bratislava, Slovakia
Chronis	Kynigos	National and Kapodistrian University of Athens
Kevin	Marshall	Microsoft Dubln
Richard	Noss	UCL Knowledge Lab
Elizabeth	Oldham	The University of Dublin, Trinity College Dublin
Ralf	Romeike	Freie Universität Berlin
Ana Isabel	Sacristán	Cinvestav-IPN
Jenny	Sendova	Institute of Mathematics and Informatics, BAS
Roger	Sipitakiat	Chiang Mai University
Bernadette	Spieler	University Hildesheim
Kevin	Sullivan	The University of Dublin, Trinity College Dublin
Michael	Tan	National Institute of Education
Marta	Turcsanyi-Szabo	Eotvos Lorand Tudomanyegyetem
Michelle	Wilkerson	University of California, Berkeley
Niall	Winters	University of Oxford

Table of Contents

Keynotes

Constructionism – a “partitioning of concerns”

Andrea A. diSessa, *diSessa@berkeley.edu*
Graduate School of Education, University of California, Berkeley, CA, U.S.A.

Abstract

This essay provides a personal view of the current state and continuing advancement of Constructionism. I propose a roughly hewn analytical framework, a “partitioning of concerns,” that can help us evaluate where we are along multiple dimensions and perhaps what might be most helpful to pursue next. The framework encompasses four areas: (1) cognitive principles for optimal learning; (2) affordances of computational representations; (3) activity forms and engagement; (4) cultural processes of large-scale educational innovation. These areas should be further subdivided into scientific vs. practice-oriented evaluations.

Keywords: computational literacy; Constructionism; science and practice; cultural change

Introduction

In 1988 I wrote a chapter called “Knowledge in Pieces.” It had two parts. The first described a then-new theory of intuitive knowledge. The second part discussed the implications of that theory for learning/teaching with computers. This chapter is an update based on the same model: It concerns both science and educational practice. A lot has changed since 1988, so there is a huge amount to say. But, this contribution must be extremely schematic. An elaborated version of this note will appear shortly.

The biggest change from my view in 1988 is that my overall image of what is and should be happening with computers and learning has changed. I now think that the “big picture” is developing a new, technical literacy—in many ways like literacy with text—on the basis of computer representations, including programming. Computational literacy is extensively developed in my book *Changing Minds* (diSessa, 2000), and a brief update appears in diSessa (2018).

Computational literacy and Constructionism have very similar features. Here, I treat them as roughly interchangeable.

“Gaming” in the title refers to our strategic (e.g., scientific), and our heuristic and experience-directed forays into improved learning in a Constructionist framework. “Science” can stand here without elaboration. But, “heuristic and experienced-based” work—what I’ll call *practice*—needs a brief elaboration.

Although developing science has been pervasive in my own approach, I have an abiding respect for practice. Experienced, sensitive, and creative practitioners can regularly outstrip what current science tells us to do concerning learning and instruction. Design of complex technological artifacts, such as airplanes or computer systems for learning, often relies on some basic science. But, it also relies—necessarily, I think—on intuitive design and judgment. We need to assess both the state of our science, and also that of our wider cultural resources and know-how concerning educational design.

I consider four distinct regimes of scientific study concerning Constructionist learning, including an assessment of where we stand in each. I separately consider where we are with respect to excellent practice in each arena. Partly for brevity—but also because I feel I know strengths and limitations better—I use mostly examples from my own immediate community.

The first two areas are both *epistemic reformulations,* attempts at very substantial changes in the subject matter itself—either in choice of topic or in the basic human encoding of that knowledge and, hence, in paths to learning it.

Area 1: Cognitive Simplicity

From the earliest days of Logo, and even before the term "Constructionism" was coined, changing teaching and learning via "cognitive simplicity" was highlighted. A protean example was the idea that a circle might be better conceptualized as what is created by a turtle that goes forward a bit, turns a bit, and repeats. Papert used the term "body syntonic" to mark that this particular conceptualization of a circle drew on strong intuitions and experience with moving our own bodies. More recent and similar avenues of improving learning include the use of agent-based modeling, and "participatory simulations" where learners, themselves, become part of a simulation and use their personal experiences to help themselves learn.

Science Examples:

Constructivism is a powerful heuristic orientation. However, I believe that it is too vague to track learning adequately or to design for it optimally. A major line in my own scientific work has been to improve our ability to understand learning, and therefore use it more generally and more precisely in our instructional design. I have pursued a number of models of different kinds of knowledge and their evolution, which I call collectively "Knowledge in Pieces."

The first and best-known model concerns the intuitive ideas that constitute a core of our experiential common sense about "how things work." These ideas, called *p-prims*, are abundant and are actually a great resource for learning, in a classic constructivist fashion. This model of syntonic knowledge was the basis of my original 1988 paper. I also developed a complementary model of mature scientific concepts, called *coordination classes*. A third cluster of models concerns how people comprehend computer systems. These were the basis for designing and tracking the learning of the computational medium, Boxer, which has been the foundation for my Constructionist experiments since the late 1980s.

The science behind p-prims and these other models has progressed substantially since those early days. Within the last several years, it has become possible to track, almost moment-by-moment, how learning happens, or not. This is a very high-end goal for the science of learning, and it is still difficult. However, advances in theory and empirical methodology put us in a very different position with respect to things like syntonic learning—"cognitive simplicity" in my general rendering—than we were thirty years ago. Kapon and diSessa (2012) tracks a number of students as they work through a well-known and generally successful instructional sequence in physics. We see, based on an analysis of their starting knowledge state, why some students succeeded quickly and easily, why some took a meandering path, and why some simply "did not get it." In diSessa (2017), we did an after-the-fact analysis of how a group of high school students managed to create, *without instruction*, a correct scientific model of thermal equilibration—something that is regarded as a very difficult accomplishment within the subdiscipline of the learning sciences called "conceptual change." We learned how the students managed this feat, but we could also see why it was actually a very specific accomplishment, unlikely to be replicated as a model for how to teach this topic. The success of these students was tantalizing. However, from the science we determined that the pathway these students took would not easily be replicated.

A different application of the same science led to a substantially new model of instructional design that I call "bottom-up curriculum development." The idea is that we can deeply explore students' native knowledge, which might support learning, before even deciding what to teach. In the "Patterns Project," we studied students' intuitive knowledge of what we call "patterns of change and control," which includes, for example, that even young children know and understand (at some level) the phenomenon of "threshold" or "tipping point." Out of this, we developed a curriculum that approaches the "fancy" topic of dynamical systems theory in a way that is quite accessible to disadvantaged populations of eighth-grade students. Indeed, the remarkable achievement in learning about thermal equilibration mentioned above was part of the "bottom-up" curriculum project. Only, to make it work in a larger curriculum and for a broader population of students required instruction with a lot of scaffolding, over a much longer period of time, and careful revision based on micro-tracking of learning successes and failures.

Practice-based Accomplishments:

The "turtle circle" was a provocative early example of using cognitive simplicity in designing learning. Hal Abelson and I extended that idea into a major reformulation of a high school mathematics curriculum (Abelson & diSessa, 1981). However, the construction of that curriculum and its success were largely the product of intuitive design and never had any detailed scientific principles or scrutiny. That remains for the future: I hope my in-progress work on "turtle physics" can do better scientifically than we did with turtle geometry.

Area 2: Re-representation

Changing representation is well known, scientifically and practically, to (often) radically change the accessibility of tasks and ideas. Hindu-Arabic notation changed arithmetic from something only specially trained experts could do (Roman times) to an elementary school topic. Likewise, non-spatial isomorphs of tic-tac-toe make the game categorically unplayable.

When the representational infrastructure of a society changes (i.e., a literacy develops) monumental changes follow. Computational representations may change the intellectual landscape of civilizations at a level comparable to textual literacy.

Science:

I cite only one example. Bruce Sherin, in his dissertation and following work (Sherin, 2001), researched the intellectual competence developed via particular representations. In it, representational and conceptual competence develop together via what I call "dual citizenship" knowledge elements; they are conceptual like p-prims, but also correspond to representational patterns. Sherin studied how students learn physics with algebra or, in contrast, with programming. Somewhat unsurprisingly (but now documented and explained in detail) algebra focuses on things like balance (=, e.g., conservation of energy), but programming is better for learning about change over time (e.g., Newton's laws).

Practice:

In an early project of ours teaching sixth-grade children about motion using programming, we were up against the collective judgment of the field that "vectors" is a difficult topic, even at high school, and is "developmentally inappropriate" for elementary school students. As a matter of fact, it turned out that vectors were completely easy to teach in the form of computational objects (boxes with arrows that can be re-aimed, lengthened, or shortened) that controlled on-screen motion. Part of how this worked relied on student interest in using them as part of "video game" construction. Hence, students had massive, personally meaningful experience with vectors. But we did no micro-tracking of learning, nor even statistical proof of student learning.

Area 3: Activity Forms and Engagement

The "building things" component of Constructionism emphasizes the fact that innovative forms of engagement, many of them computer mediated, can be pivotal in transforming learning. So, we should ask, what is the state-of-the-art in science and practice concerning altering activity structures and long-term engagement? I can be very brief here, for two very different reasons.

Science:

My feeling is that the science of engagement is in a very poor state, particularly concerning designing for excellent engagement. In the longer version of this paper, I will discuss why I believe this is so and what I consider "green shoots" for more productive science in this arena.

Practice:

I can be equally brief with practice. I think the community of Constructionists is so full of good examples of excellent student engagement that it scarcely needs examples, here.

Area 4: The Social Evolution of Computational Literacy

How do successful movements toward societal change (such as toward a computational literacy, or broad adoption of Constructionist learning) get organized? Can we act effectively to instigate and guide movements, or are all such unpredictable and driven by the full complexity of societies and of the times in which movements succeed or fail?

In short, I believe current science here is minimal or non-existent. In the longer version of this note, I will present a case study of Papert's own excellent practice, raising also some issues that beg scientific inquiry.

References

Abelson, H. and diSessa, A. A. (1981). *Turtle Geometry: The computer as a medium for exploring mathematics.* Cambridge, MA: MIT Press.

diSessa, A. A. (2000). *Changing minds: Computers, learning, and literacy*. Cambridge, MA: MIT Press.

diSessa, A. A. (2017). Conceptual change in a microcosm: Comparative analysis of a learning event. *Human Development*, *60*(1), 1-37.

diSessa, A. A. (2018). Computational literacy and "The Big Picture" concerning computers in mathematics education. *Mathematical Thinking and Learning*, *20*(1), 3-31.

Kapon, S. and diSessa, A. A. (2012). Reasoning through instructional analogies. *Cognition and Instruction*, *30*(3), 261-310.

Sherin, B. (2001). A comparison of programming languages and algebraic notation as expressive languages for physics. *International Journal of Computers for Mathematical Learning*, *6*, 1-61.

Contentious Subjects: Constructionism and the Project of Higher Arts Education

Mick Wilson, *xwimic@gu.se*
HDK-Valand, University of Gothenburg, Sweden

Abstract
Based on an analysis of the ways in which constructionism is both implicitly and explicitly operational within contemporary higher arts education, this paper proposes a re-reading of the political stakes of constructionism.

Keywords
Constructionism, higher arts education, the project method, individualism, subject formation, race, social ontology

Introduction: Constructionism *avant la lettre*

> [T]he educational question – how to come and remain in dialogue with the world – is also the question of art... the educational moment appears inside the artistic endeavour, in such a way that art itself can and is allowed to teach. (Biesta, 2017a)

In an often-cited scene, Seymour Papert describes his passing encounter with an art class in a 1960s Massachusetts high school as the matrix out of which his vision to transform the dynamics of the math classroom emerged. Watching students over time work on soap sculptures, Papert "mused about ways in which this was not like a math class. In the math class students are generally given little problems which they solve or don't solve pretty well on the fly." In contrast the students in the art class carved whatever "came from wherever fancy is bred and the project was not done and dropped but continued for many weeks. It allowed time to think, to dream, to gaze, to get a new idea and try it and drop it or persist, time to talk, to see other people's work and their reaction to yours." Papert identified this as a kind of early eureka moment: "I want junior high school math class to be like that. I didn't know exactly what 'that' meant but I knew I wanted it. I didn't even know what to call the idea. For a long time it existed in my head as 'soap-sculpture math.'" (Papert and Harel, 1991)

Decades later and 'soap sculpture math' has emerged as constructionism, a widely successful framework for math and science education, but also for conceiving education more broadly in a learner-centric way. Like all successful frameworks, there has been considerable debate as to what constitutes the real essence of constructionism, especially in its building upon, but also diverging from, the psychological account of human development in Piaget's constructivism. In the first section of this paper, 'A constructionist way of talking,' I attempt to bypass the controversies over what the essence of constructionism is, in favour of a strategy of describing constructionism as a way of talking about, and a way of prioritising certain activities and aspects, of the educational situation.

In the second section of the paper, 'The many projects of higher arts education,' I provide a somewhat compressed and generalized account of a wide spectrum of institutional forms and practices in university level of arts education, in order to support the claim that constructionism is both implicitly and explicitly operational within contemporary higher arts education through the centrality of the project method and through the foregrounding of student agency. Given Papert's myth of origin for constructionism, this should be of little surprise. Indeed, Thompson observes with respect to 'constructivism' in the USA, that it "is more frequently practiced in art education than it is explicitly invoked as an approach to curriculum or pedagogy." (Thompson, 2015, 118)

Referring to Piaget's and Vygotsky's work (rather than the specific inflection of constructivism at work in Papert's 'constructionism') she further notes that references "to constructivist theory are rare in the literature of art education, while implementation of constructivist principles and practices is common."

The parallelism that Thompson identifies will be developed in this paper with specific reference to the centrality of project method in higher arts education. The generality of this account of higher arts education, as with all claims for what is *typical* of a given site in the lifeworld, will require a certain caution. Counter-examples may be easily adduced for each general claim utilized in the description of the *typical*. However, this general account, treated with due caution, can be nonetheless serviceable for the purposes of teasing out the ways in which a political critique of constructionism may be seen in analogy with respect to current internal contestations of higher arts education and questions of the political.

The third section of this paper, 'Individual: On subjects in contention,' will be concerned to tease out the political stakes invoked in the contestation with respect to the pedagogical strategies and discourse of higher arts education. It is proposed, by analogy, that constructionism must take account of similar challenges. It may help to indicate, at the outset, that a key term in the contestation within higher arts education, is the way in which individuation–the construction of the learner as a reflexive self-constituting agent–is being problematized against a wider backdrop of critiques of neoliberal reductions of the social to the simple aggregate of 'individuals' construed as atomized self-actualizing agents. A key resource in developing drawn upon this regard is the work of Gert Biesta, a prominent critic of the 'learnification' of education.

This leads to an account, in the closing section 'Constructing education in fundamentally conflicted worlds,' of the way education implicitly or explicitly invokes social ontology (a term elaborated within the text). The paper then concludes with a proposition for maintaining constructionism as a flexible approach while accepting that some the challenges of the political critique of education require a rethinking of how the terms of social ontology are operative within any educational setting, something that perhaps exceeds the discursive resources of constructionism, but which cannot be simply delegated to philosophers.

A constructionist way of talking

> It is easy enough to formulate simple catchy versions of the idea of constructionism; for example, thinking of it as "learning-by-making." … [We propose] a sense of constructionism much richer and more multifaceted, and very much deeper in its implications, than could be conveyed by any such formula. (Papert and Harel, 1991)

In place of a once-and-for-all definition of constructionism, I have chosen to identify a set of recurrent themes that resonate across constructionist accounts of learning and that constitute what may be described as a widespread, and at times diffuse, constructionist *ethos*. Given the success of constructionist ways of speaking, it should come as no surprise that on occasion these ways of speaking may be internally contradictory. It has been noted often that constructionism may be an avowed approach in the rhetoric of a given teacher or institution, but something that diverges radically from their actual classroom practices. This is not just a challenge within constructionist ways of speaking, it is often the case that an educational discourse operates as a rhetoric that does not describe or inform, but rather obscures, habitual practices of teachers, learners and their institutional settings. However, rather than be drawn into an account of the authentic core or essence of *true* constructionism, it will be sufficient here to simply recognize certain recurrent tropes and themes within the discourse of constructionism. Furthermore, describing constructionism as a way of talking allows that we can recognize that constructionism operates in different contexts as a philosophy of learning; as an epistemology; as a practical framework for elaborating teaching strategies; as a rhetoric of policy and planning; as a descriptive and /or normative account of learning; as a critique of institutional forms and practices; and as an emancipatory project that seeks to promote social betterment through school reform.

Theme, trope and tone

A simple—though hopefully not overly reductive—account of constructionism as a discourse may be provided through identifying a few key themes and central tropes that pertain not just to ways of talking about, but also ways of acting with respect to, the learning-and-teaching situation. One such key thematic within constructionist discourse is that the educational process should be conceived and operated in a learner-centric manner. This means also, that educational strategy is described not from the primacy of content-transmission by a teacher, but rather from the priority of content-making and skills-acquisition realized through learner activity. A central trope, perhaps the primary trope characterizing constructivist discourse, is the learner's active construction and reconstruction of the world, knowledge and insight. Learning is accomplished through active agency in producing stuff rather than through passive attention in the reception of content. Multisensory and embodied experiential hands-on tasks are privileged over the reception and assimilation of pre-formatted and rigorously encoded knowledge productions. Constructionism may also be characterized by the discursive axiom that the acquisition of insight, knowledge, skill, judgement and understanding are not best accessed through rote learning and abstracted memorization tasks nor through passive content acquisition.

A second key thematic is that the constructive work of the learner is especially evident and activated in the building of 'public' artefacts. Such productions are public in the sense that these are externalized and may be apprehended by others. This includes a very wide range of possibilities such as: digital-virtual entities (e.g., a website, a digital portfolio); or material-physical entities (e.g., a built model, a display board combining various images, texts and objects); or events (e.g., a physical behavior programmed into an automated artefact, a performance). Such public artefacts have the capacity to gather people around them and operate in the shared intersubjective spaces beyond the interiority of the learner. The work of making such public artefacts is itself discussed as a driver of learning. These artefacts may operate as a clear demonstrator of learning achieved.

This manifestation of learning through productive action has resonances with, but is markedly different from, the behaviorist commitment to the externally observable as the objective basis for assessing knowledge acquisition. Constructionism does not disavow the saliency of the learner's inner life. Rather, it asserts the vitality of the learner's active (constructive-)encounter with their world and those other agents active there. The meta-reflective work whereby the process of learning is reflected upon and re-constructed by the learner is of key importance. The mental constructions of the learner have more than an evidential corollary in the observable world. There is an important unfolding of the learner's interiority in the self-reflection brought to bear on learning, and in the active reciprocal action of the learner and the learning environment in the orchestration of a public artefact: The learner's self-relations are established and augmented through world-relations. Within some variants of constructionism, that place emphasis on the social construction of meaning, the intersubjective and collective dimensions of these processes are paramount.

Another important dimension of the discourse of constructionism, is that it has a distinctive rhetorical confidence in its own robustness and efficacy. (Again, there is a partial resonance with that older family of pragmatist and later behaviorist discourses with their tone of energetic assertiveness and presumptive self-evidence.) There is a recurrent trope in constructionist discourse, of pointing at things in the world that prove the case: "Look. Here it is. See, it works!" Constructionist discourse is characterized then not just by a set of themes and repeated tropes, but also by a rhetorical style and tone of confident empirical and active affirmation: "Here you are. Look what's possible!"

There are of course fracture lines crisscrossing the wide swathe of constructionist positions and practices. Perhaps most prominent among these is the tensive pull between a constructionism that is focused on the individual learner and various social constructionist accounts that prioritize the intersubjective and trans-subjective dimensions of learning. However, for the moment let us settle for this broad characterization of the constructionist way of talking education as learning, and proceeding on that basis consider the implicit and explicit echoes of these within higher arts education.

Constructing an analogy

In taking these four aspects as descriptive: the theme of learner-centrism; the trope of ‘public’ productive activity; the inward-take of meta-reflection on the learning process; and the tone of self-evidentiary robustness and ‘good common-sense’, I do not wish to obscure the systematic and nuanced philosophical and psychological theorizing that is mobilized within constructionist accounts of learning. These four aspects are taken rather as the most consistent features of a widely dispersed and hugely successful discourse. They also provide the key points of traction between constructionism and higher arts education.

The many projects of higher arts education

The project arc

Within contemporary higher arts education, a key construct is the project, the durational stretch, of an episodic arc of activity proceeding to some terminus, a final production, object or event. This project terminus might comprise a suite of interrelated fabricated and found images and objects, an installation, an exhibition, a temporary public project, performance, a digital artefact or system, some form of public event, or even a soap sculpture. Rather than the course or the module (even those these might superficially appear as the organizational building blocks) the basic unit of practice in higher arts education is essentially the project. The openness, that is to say the relatively unfixed specification of form/content for the student’s artistic production tasks is especially important with respect to the culmination of both bachelor and masters level studies in fine art. This question of what concerns, and by what means the student should advance those concerns, is the key framework of decision-work assigned to the student artist at these culminating moments in formal study. The requirement imposed with respect to the project arc—the process of initiating, naming, describing, contextualizing, developing, and bringing to some form of public resolution a body of work—is that the student should in some way explicate their decision process. These decisions are typically understood to emerge from within the project arc, rather than being pre-set by a teacher or by being comprehensively pre-formulated by the student.

Certain specific parameters may be preset within the brief proposed to the student. This is often the case in the earlier years of undergraduate education. In these earlier stages of the learning process, project briefs may be defined in terms of a prescribed thematic; a space of display; a first audience; a technique; or a request upon the student to demonstrate a research process or to disclose a particular aspect of their artistic process (the choice of sources, choice of historical precedents and references, observational practices etc.); or by the specification of a concrete situation or social encounter as the point of departure or the target of response for the project process. Typically, in setting such limiting frames, there is a relatively open parameterization set within any project brief. Most often the goal is that there is maximal space for the artist-learner to assert their own concerns and priorities within the project framework. While not completely unbounded and not without restrictions, the student-artist’s project brief tends towards an open-structure form of initial scaffolding that can fade from view once work has begun to accumulate along a particular project arc. Very often, in the initial phase of any project, the learner’s task is to specify their own customization of the assigned brief.

This tendency toward customization in project brief might be construed as an emphasis on the *expressive* function of an artistic process. However, this is not necessarily what is at stake. Rather, this relatively light framing of project briefs, the openness of parameters set within assigned tasks, might better be understood as an attempt to prioritize the agency of the artist-learner and the allocation to them of a responsibility to formulate the terms of their “own practice”: What are your concerns? What things in the world interest you? These days it is increasingly common for a student’s art practice to be asked to disclose itself in terms other than those of authenticity and expressivity. For example ‘enquiry’ or ‘social engagement’ might take the place of ‘expression’; and ‘trickery’, ‘parody’ and ‘fabulation’ might take the place of ‘authenticity.’ Nonetheless there is clearly a prioritizing of individual choice and the giving of reasons or rhetorical grounds for that choice.

What remains primary is an emphatic focus on the individuation of the artist-learner as a reflexive agent, who is asked to give an account of their process of decision-making and their authorship – and ultimately of their self-production. In this way, the old-fashioned image of the emotionally fraught, tortured or sensitive artist persona (the genius) has been increasingly modified and supplemented by the emergence of new artistic personae and scripts. These new artistic personae are often predicated upon a critically knotted reflexive artist-subject who struggles with the problems of achieving ethical coherence and political agency. These may be contrasted with the figure of the artist as an emotionally knotted or fragmented subject who struggles with the problems of achieving psychic coherence, psychological stability and intimate communication – the artist as the bearer of misunderstood greatness.

Interestingly, with the widespread adoption of the learning outcomes model of open curriculum specification, increasingly the student-artist is asked to be a doubly reflexive agent: to give account of their decision-making within their process of art making and within their process of learning: What are you trying to encounter or access in this work of art? What was it you were hoping to learn from this way of proceeding?

The teaching instruments that are typically prioritized in contemporary art education include the one-to-one tutorial, the small group tutorial, the group critique session (the 'crit'), the hands-on workshop, and the group meeting for project planning. Lectures and seminars, those more formal teaching instruments that characterize education in the broader swathe of university disciplines, are also used within arts education. However, there is a tendency to prioritize the solicitation of themes, content and ideas from the learner rather than to push prescribed mandatory curriculum content or canonical exemplars. Nonetheless, there are some moments of explicit 'content push', often in the form of theoretical or historical surveys, training workshops and other tried and trusted formats such as the reading seminar, the exhibition visit, and the visiting artist's/curator's talk.

A dizzying economy of ideas

Art schools–both smaller stand-alone institutions and art departments incorporated within larger universities–are typically subject to waves of intellectual and aesthetic fashions. This is not so much a matter anymore of aesthetic and intellectual convulsions wrought in the manner of the avant-garde formations of the early- and mid-twentieth century (such as surrealism, abstract-expressionism, or minimalism). Today, these waves of mobilization within art school settings tend to occur more often in terms of wider cultural formations of shifting intellectual, aesthetic, and political concerns and sensibilities.

These waves can by identified by naming broad political projects and intellectual traditions (such as anti-racism, anti-fascism, feminism, trans-activism, new materialism, accelerationism, Afro-pessimism, the decolonial, climate crisis activism and so forth). Sometimes these mobilizations are martialed in terms of the engagement with specific author-names (such as Mouffe, Deleuze & Guattari, Spivak, Harvey, Rancière, Agamben, Butler, Haraway, Braidotti, Barard, Stiegler, Glissant, Moten & Harney, and Ahmed, to name only an arbitrary few.) In turn these broad tendencies are interwoven with the far longer lists of artists, curators and art theorists active within the expanded field of contemporary art (too many to mention here.) The wider political projects, the author names and the various contemporary art players with their affiliation to various interdisciplinary endeavors, indicate something of the wealth of citation in play across the contemporary art field. This circulation of ideas can become quite dizzying, and has on occasion solicited charges of dilettantism.

This economy of ideas, sensibilities and citations provide analytical concepts, motivating frameworks and rhetorical sources for different practices and divergent project undertakings. Their circulation within the art school milieu is informed by their wider circulation in the culture at large. Additionally, the art school may often act as a key site for the wider activation, dissemination and translation of these ideas beyond their primary site of production. The art school as a milieu of intellectual and aesthetic production is typically inflected by the different interests and commitments of specific teaching faculty and students, so that there is significant variation in the mix of references mobilized across different institutions, and different national contexts.

The ethical and political inflection of these different intellectual and aesthetic traditions and trends is important. There is usually a fairly high degree of declared ethical or political intention and positioning with respect to the theories and sensibilities mobilized. The general tenor of these intellectual and political currents is left-ish and liberal, predominantly so but not exclusively so.

Sub-domains: Teaching art education and educating art teachers

Both the education for specialist arts teachers and the education in art pedagogies for general primary education teachers, are often accomplished within the same broad institutional matrix as the education of artists. Nonetheless, there is a tendency to make a very sharp distinction between the artist-learner and the learner-teacher; and for the faculty charged with these tasks to establish clear boundaries and divisions between their respective territories. Often this is correlated with the degree of formal specification and regulation by the state of the competencies and the service education experience required in the education of primary and secondary teachers. This may be contrasted with the relatively open, unfixed and highly variable requirements that delimit the education of artists not working toward a specifically pedagogical qualification. Unsurprisingly, it also correlates with status contests and the struggle as to which sub-domain of higher arts education attaches to itself the greater artistic credential, or secures its claim to establish the benchmark of artistic and educational legitimacy. The ensuing academic turf wars are for the most part unremarkable. Although, one interesting consequence of this structural tension is a divergence in the ways learning and teaching are reflected upon in each sub-domain, and may in part account for the relative absence of constructionist themes within the discourse of artist-teachers focused on artist-learners.

The structural dynamics generate a significant disconnect between arts teacher education and artist education programs. While the degree of divergence is substantially impacted by different national regulatory frameworks, there have emerged significant differences in: (i) modes of pedagogical reflection; (ii) styles of argumentation with respect to learning and teaching; and (iii) core conceptions of what constitutes artistic practice and its specificity vis-à-vis other disciplines and subjects. These two sub-domains then—the education of artists and the education of teachers for the arts—may be housed within the same building on campus, but manage to live in quite different, and jealously guarded worlds as testament to our fragile narcissism of small differences,

There are three caveats that need to be appended to this broad characterization. The paths of these two sub-domains increasingly cross because of: (i) the rise of audit culture and new steering mechanisms in higher education; (ii) the convergence of critical theoretical frameworks within these sub-domains, so that for example, Dewey or Freire of Rancière or hooks or Haraway or Biesta may show up in the shared citational networks of these sub-domains; and (iii) the educational turn within artistic practice beyond the formal institutions of education.

The terms of an analogy

The themes that mark a convergence between the discursive repertoire of constructionism and of arts education include: the centrality of learner’s agency; the positing of the teacher as active facilitator, but not as the prescriptive bearer-of-content charged with the task of knowledge transmission; the aesthesis of embodied activity in concrete situated material production (as opposed to merely symbolic manipulations in the single register of the representational); the rhetorical construction and re-construction of meaning in the student’s own discursive recuperation of their project work and their own learning process; the durational arc of a holistic and constructive project as against episodes of discrete fragmented rote learning and memorization tasks; the use of immanent critical assessment processes and emergent criteria that are specific to the student project arc, rather than externalized pre-fixed assessment criteria.

Individual: On subjects in contention

From individual freedom to the double bind

In the previous sections the focus moved from an account of constructionism to its implicit and explicit role as a *modus operandi* within higher arts education. The next phase of the discussion will be to propose that the consideration of current discussions within higher arts education might enable a way to rethink the political stakes of constructionism.

The move from the account of constructionism to an account of higher arts education was specifically focused on the centrality of the project as the essential unit of practice. It was asserted, in passing, that the project format, in conjunction with the widespread adoption of the learning outcomes model of open curriculum specification, positions the student-artist as a doubly reflexive agent. The student-artist is required to give account of their decision-making with respect to both their artistic concerns and their learning goals. In rhetorical terms, the student-artist is addressed by the educational interrogative as an agential-subject that both gives account of itself and produces itself.

The historical development within higher arts education of an orientation to the artist-learner's self-determination in artistic choices and learning goals has been broadly conceived as progressive, emancipatory and democratic in tendency, espousing the free individual against the demands and restrictions of the system. This is not so surprising given the rhetoric of freedom of expression in the development of modern art's critical frameworks over the course of the twentieth century and given the paradigmatic role of originality, individuality and autonomy as value terms. The fundamental correlation of unfettered artistic freedom with open and free societies, a correlation installed at the very heart of liberalism's political imaginary, remains one of the great propaganda success stories of the 'Cultural Cold War.' (Cockroft, 1974; Guibault, 1983; Iber, 2015; Stonor Saunders, 1999) It has had a profound impact on the development of higher arts education. So much so, that the broader crisis of liberal democracy and the undermined faith in individual freedom as foundational to societal freedom, have generated a widespread re-assessment of the centrality of individuality in the practice of higher arts education.

The neoliberal collapse of the idea of a free society into the reductive image of the free market has problematized this master equation of freedom of artistic expression with freedom *par excellence;* and of individual freedom with societal freedom. The great chain of equivalences that links from individualism to freedom of expression, to freedom of choice, to free markets, and on to the freedom to 'maximize human potential' has produced something of a double-bind. This double-bind becomes apparent in the ways in which individual freedoms have been transposed into totally individualized responsibilities by appearing to legitimize the systemic removal of social protections. This double-bind becomes apparent in the cooption of individual expression by the attention economies of social media platforms and the impoverishment of social relations as these become currency of self-esteem (how many *likes*?) and data assets to be exploited. The double bind is manifest in the corporate appropriation of the "artistic critique" of liberal capitalism so that counter-culture demands for personal authenticity, conviviality, emotional connection and creativity become the values espoused for promoting employee productivity and augmenting manager efficiency. (Boltanski and Chiapello, 2005) The double bind is that the unfettered promotion of individual freedom and individual choice produces an unfreedom as pressing as the iron cage of modern bureaucracy. Unbinding the individual from collective obligation and social restriction ties the individual to a solitary life of self-as-enterprise, abandoned to absolute self- responsibility as the sole proprietor of that property that is oneself. Released from external obligations through the guarantee of self-ownership, one can then freely choose to alienate one's personal data as the price of access to an online *community*.

The champions of free-market and neoliberal ideologies have attached a particular importance to both a methodological individualism in the framing of social policy and a rhetoric of entrepreneurial individualism in economic policy, giving renewed centrality to figure of the heroic individual within the social imaginary. The critique of liberal individualism is a staple of conservatism from the

Heritage Foundation's repudiation of "a worldview of individualism, expressivism, and secularism" (Ceaser, 2008) to the Roger Scruton's declaration that "the principal enemy of conservatism" is "liberalism with all its attending trappings of individual autonomy". (Scruton, 2001, 5) There is a long tradition of critique of liberal individualism from the left that also finds expression in the critique of contemporary art.

Writing with respect to some of the problems attendant on contemporary artists attempting to participate within activist movements, one commentator has observed that "many artists often have no idea how to actually work with others or how to begin to break out of regimes of value linked to cultural capital." Furthermore, this is specifically linked to the way that artists are "[t]rained to operate as hyper-individuals in a competitive and brand-oriented set of institutional and market hierarchies." (Kelley, 2013, 53) Drawing upon Hito Steyerl's image of "post-Fordist … conveyor belts", Kelley describes the production of the artist "as an arch opportunist, devoid of social solidarity and without political consciousness beyond their narrow desire for exposure and success." Writing with reference to Sweden, the UK and USA Lindström asserts that higher art education encourages "an individualistic understanding of the artist, based on certain romantic notions of the particularity and autonomy of the art world" and the central importance of the construction of the artist identity and the burden of self-steering imposed upon the artist-learner within the educational process. (Lindström, 2016, 22) Singerman, in his history of higher arts education in the USA, writes of the MFA system in terms of "the cruelty of current art training," where the work of individuation in positions the graduating artist in a way that "psychologises and personalises" any career failure as an artist that may ensue. (Singerman, 1999, 211)

This hyper-individualised figure of the artist, described by Kelley and others as marked by a collaborative deficit and a peculiarly intense form of individualization, may be taken here as a shorthand for the problematising of the centrality of individualism in the practice of higher arts education. It is not necessary to propose or adopt the figure as the literal truth of art education, that would seem a considerable overreach. However, the hyper-individual serves to disclose a widely discussed problematic within higher arts education, which is the way in which an educational practice of soliciting critical and reflexive production may engender a particularly atomized individualist, especially when operating within the wider historical frames of neoliberal social policy and social media regimes of mandatory self-disclosure.

A further complication of this heightened individualization is the way in which it intersects on art school campuses (and elsewhere of course) with an older politics of difference and rights activism (anti-colonialism, civil rights, anti-war, feminism, anti-racism, gay liberation, 'rights to the city') in newer formations of identity politics and identitarianism, within a climate of moral panic about personal freedoms. Often this can lead to identity becoming treated as a matter of possession and so subject to rights of ownership by some, a matter of the possessive individual's owning a property in 'self.' This is not to propose an equivalence between the left's identity politics and the far right's identitarianism (ethno-nationalism and white supremacism), but to identify a problematic reduction of identity terms to the terms of possessive individualism. The logical extension of this is the proposition that one, as an individual, has the absolute right to determine how others relate to oneself: "I" assert—and on occasion receive institutional validation of— "my" right to prescribe how others must address and enter into social relation with "me".

It is a strange twist within our neoliberal condition that the old Thatcherite disavowal–"There is no such thing as society!", a claim for methodological individualism, taken out of context–returns in this way. The claim that there is no foundational social relation beyond the mere interactions of sovereign individuals re-appears as the seemingly radical and progressive demand for institutional protection and the guarantee that "I" should be able to determine unilaterally my position in social relations. This is not to be confused with the right of free association. This is an historically new claim for one's sovereignty in defining how others should socially orient towards one. Such is the strange fate of identity becoming translated into the terms of individualized property.

Becoming subject

> The real educational work, as I will argue, is precisely not about facilitating expression but about bringing children and young people into dialogue with the world. It is about turning them towards the world and about arousing their desire for wanting to be in the world and with the world, and not just with themselves. It is there that their expressions can 'encounter' the world –material and social–and that such encounters can provide starting points for exploring what it might mean to exist in the world in a grown-up way, that is, 'in the world without occupying the centre of the world.' (Biesta, 2017a, 37)

The problematic figure of the hyper-individual has prompted a wide range of different responses from within higher arts education, that do not simply demand the abandoning of the project method. These include the turn to group process, collaborative and social practice, and to the outright rejection of paradigms of individual expression. However, the problem of hyper-individualization is not necessarily addressed by the appeal to collectivity and group work. In as much as the group process may often simply provide a theatre of operations for various individualized performances, which through the intersubjective encounters of the already individuated members of the group merely orchestrates the regime of hyper-individualism. The group project, like the social media platform, can become a theatre of sociability and conviviality in the service of self-production and self-marketing/self-projection. The appeal to group process, is an appeal to the idea of constructing a milieu within which the formation of the individual as an ethical subject takes place. This is an ethical subject that is not only adapted to, and but actively embracing the conditions of being one-among-many-beings in a world that is not constructed in exclusively ego-centric terms.

Recently, the educator and educational theorist, Prof. Gert Biesta, known for his wide-ranging critique of constructionism and what he terms the ‘learnification’ of education, has provided a remarkable intervention into the debate on art education with his (2017a) *Letting Art Teach*. Biesta’s scholarly work has constructed an extraordinary arc from his early critique of learner-centrism, to a plea for re-valuing teaching as an art of the uncertain asserting the ‘beautiful risk of education’ as against its instrumentalized service and measure with reference to economic, psychological, sociological and other external rationalities. (Biesta, 2006, 2010, 2017b, and 2019). His most recent work makes an important and provocative challenge to what he characterizes as the overproduction of research, or what he describes pithily as: “too much research in education and too much belief in the value of research.” (Heimans and Biesta, 2020, 105; see also Biesta, 2020.)

Biesta proposes a tripartite construction of education in three domains of purpose: qualification, socialization and subjectification. Qualification indicates that dimension of education that pertains to functional capacities, to knowledge, skills and dispositions. Socialization is about orientation in the world and refers to coming into relation with the pre-existing organization of the world, the histories, practices, and cultures that provide the pre-established horizons of world-ordering into which each generation arrives as ‘new comers’ (*pace* Arendt.) Under the last heading of ‘subjectification’ Biesta presents an alternate account of individuation in educational process, without seeming to invoke the absoluteness of individualism. Biesta argues that within the educational process, properly conceived, we are called into the world and:

> …we encounter the question of what it means to exist as subject – as subject of our own actions, our own intentions, and our own responsibility – and not just as the object of what others are inclined to or would like to decide about our lives. It is crucial to see, however, that to exist as subject does not mean that we just do what we want to do without ever considering what our actions mean for and do to the opportunities for others to act as well…. To exist as subject…means to exist in dialogue with the world… (Biesta, 2017a, 57-58)

An important aspect of subjectification is the experience of limitation, the experience of the resistances of the world and the resistance others to our desires. Biesta's vision of education has a strange and distant resonance with an older *bildung* ideal of education as a kind of ethical formation of the subject. Indeed, there are echoes of Schiller's *Letters on the Aesthetic Education of Man* (first published in 1794), not so much as Schiller's claim that aesthetic experience will re-integrate the fractured self of modernity, but rather that through subjection within the encounter with the external power of art, and more generally encounter with the world, the 'new comer' comes to their condition as subject, they become that kind of reflexive agent that assumes responsibility for itself. Elsewhere, Biesta has argued that subjectification "is itself a social, intersubjective and ultimately political process that can take place through engagement with knowledge and curricular content more generally." (2010, 109) He has also posited that "subjectification" is a more appropriate term to employ than "individuation" because it "articulates that being and becoming a subject are thoroughly relational and also…thoroughly ethical and thoroughly political." It is "not simply about expressing one's identity—not even one's unique identity, as uniqueness is not to be understood in terms of difference but in terms of irreplaceability in my ethical and political relationships with others who are not like me." (2010, 129) It is important to emphasize that this image of subjection—placing oneself in the encounter with the powers of the world—is not equated with the coerced subjection to an external authority, for we cannot "force" another human being "to exist as subject, as this would negate the very 'thing' we seek to bring about, namely that another human being can exist as subject of their own actions rather than as object of the ambitions of the educator." (Biesta, 2017a, 86)

Constructing education in fundamentally conflicted worlds

The subject and the horizon of the universal

Biesta signals in the opening phase of the argument in *Letting Art Teach,* that an art education cannot simply promote expression but must be focused on how the questions of what is *right and* what is *the good*, are produced: "What if the voice that expresses itself is racist? What if the creativity that emerges is destructive? What if the identity that poses itself is egocentric…?" (Biesta, 2017a, 56) He does not propose a normative answer in terms of the imposition of 'right thinking'. Instead he develops the more Kantian form of an answer that uses the terms of the self-legislating liberal subject. Kant famously responded to the question "what is enlightenment?" by talking of the human being maturing, becoming autonomous, becoming the operative of its own reasoning, and not simply submitting to the reasons of others. (Kant, 1784) Biesta echoes this Kantian formula of maturation with his contrast of 'infantile' and 'grown up' modes of being in the world. Biesta, however, also identifies his thesis on subjectification with Rancière's account of democracy, equality and the political which would seem to position his work in a post-Kantian frame.

It is interesting that one exemplary instance of problematic expression given by Biesta is that of a 'racist' voice. This is interesting because on the one hand, there are analyses of race/racism that point to the self-transparent liberal subject as precisely a figure that emerges within Western colonial-modernity already structured by the constitutive logic of racism. On the other hand, (liberal) institutions of higher education—as indeed the wider social and political landscapes of 'liberal democracies' – have seen a resurgence and a renewed intensity of explicitly racist politics (as opposed to the dog whistles of previous decades). This means that the presumptive liberal norm that "racist utterance is not right," cannot be assumed to have effect, even if we could assume—and we can't—that racist utterance will be unproblematically recognized as such by the speaker/group/teacher/institution. Indeed, this is precisely one of the sites of profound contestation in the contemporary university: that systemic, structural and institutional racism is operated, cognized and experienced differentially by always already (differentially) racialized subjects.

My purpose her is not *to prove* the claim that the subjectification that Biesta proposes is already inscribed within a racist logic. Rather my purpose is to establish that there exist such reasoned and well-wrought claims and that these warrant careful attention. The very possibility of such claims, signals a problem with the universalizing tendency of the analysis and the vision proposed for education. (And of course, I make no *ad hominem* accusation here either. This is a question of a possible limitation within the tradition of critique that both Biesta and I inhabit.) In order to construct this particular move in the discussion, I draw directly upon the work of Denise Ferreira da Silva, whose (2007) *Towards a Global Theory of Race* is crucial in re-describing the relationships between post-Kantian critique, the subject and the operation of race. In a recent interview with the German art journal *Text zur kunste*, Ferreira da Silva explains her analysis as follows:

> When commenting on racial critique, I have in mind the kind of engagement modeled after Immanuel Kant's formulation of critique, which he describes as systematic exposition and assessment of the conditions of possibility for X; that is, of its grounds and limits. … this specific analytical procedure has supported the claim that the rational mind … has access to the universal laws of nature because it shares their formal constitution.
>
> This presupposition is also shared by the kind of racial critique that stops at the diagnostic of the devaluation of human populations constructed as non-white/non-European. At its worst, it presents this devaluation as an effect of beliefs or ideology and, as such, a deviation from the universal (moral) principles said to rule modern existence; at its best, it presents devaluation as constitutive of modern thought, but then moves on to an argument based on the idea of incompletion (that universality is yet to be realized) or misapprehension (that a particular has mistakenly been taken for the universal). In both cases, universality is retained as the proper descriptor of the modern ethical program. (Ferreira da Silva, 2019)

Ferreira da Silva asserts the function of race within the construction of the universal, and in this way identifies the ways in which the paradigm of (post-Kantian) critique and the liberal subject *require* racial difference. Ferreira da Silva asserts that "foregrounding racial violence (and not racial discrimination or racial exclusion)" exposes that "the principles of universal equality and universal freedom are not the ultimate grounds for modern existence". The circulation of these principles of universal equality and universal freedom, is in fact "contingent upon the deployment of racial difference…to delineate the proper ethical domain of application of the universal principles under which colonial juridical forms of total violence prevail."

Even if one is not inclined to accept this argument, nor particularly interested in the philosophical intricacies of its construction, its mere existence should give one pause: There exists an argument that proposes that this construction of the transparent subject is part of a universalizing discourse that founds itself in the distribution of violence and an allocation of some bodies as always already to be violated. This points to the existence of a radical disagreement as to the nature of the subject vis-à-vis race, a disagreement that is about the ontology of the subject and the ontology of race. Different positions with respect to this disagreement will found different visions and programs of education. Furthermore, this disagreement may not be properly legible or decidable within the existing frame of critique within which it is now being cited and partially rehearsed.

Education and social ontology: Individuals, subjects, persons, humans, bodies…

> [T]he words "person", "human being", "child", and "individual", shall include every infant member of the species homo sapiens who is born alive at any stage of development. (Law Revision Counsel, 1934)

One of the challenges, that thinking about the processes and operations of education raises, is achieving clarity as to what kinds of entities we posit as making up the field of these operations. The entities proposed as constitutive of the world of education may include such foundational ones as individuals and subjects, but also a wider network of entities such as: 'adults', 'children', 'families', 'parents', 'schools', 'communities', 'teachers', 'learners', 'cohorts', 'literacies', 'skills', 'curricula', 'projects', 'disciplines' or 'subject areas' and so forth. A technical term for this type of construction, this positing of what entities make up the world, is an ontology, a set of categories of things that exist and constitute the world. Education appears to operate within some kind of social ontology, that typically has a range of operative categories that pertain to individuation: individual, person, body, subject, agent, human being, child, adult, learner, teacher, and so forth. In the confident prescription from the *U.S. Code of Laws* cited above, a common equivalence between some of these terms is suggested. However, in their considered usage these terms diverge significantly even though they are often casually employed as synonyms. Furthermore 'universal' alignment within these categories is selectively and partially withheld with respect to different bodies. For some commentators, this is a constitutive feature which cannot be overcome by including those currently excluded under a universal term.

Although these categories of individuation overlap and interact, they are not completely reducible to each other. For example, 'person' indicates a form of individuation before the law and it may apply in different circumstances to a corporation, a human being, and an environmental entity such as a river or a forest. An individual is that unit of human being that cannot be further sub-divided, an entity often understood as absolutely singular: the *unique* individual. While the term 'subject', hailing, from a particular strand of philosophical modernity, indicates individuation in terms of the supposedly generic grammar of being a 'self' that is reflexive: a self that takes cognizance of itself. We have seen that this generic grammar is for some upon built upon a distribution of violence. There are clearly many philosophical and social-theoretical knots that are activated by any talk of individuals, individualism and the formation of subjects. If we consider further such entities as 'community', 'group', or 'collective,' constructs that in turn interact with ways of constructing the terms of individuation, the challenges of framing with precision the social ontology that founds an account of education are quite formidable.

Do we really need all this talk of social ontology? Can't we just get on with the job at hand, and leave the philosophical knots to the philosophers and the social theorists, letting them tie themselves up with all that stuff, while we get on with the urgent practical work of learning and teaching in the 'real' world?

Of course, the constitution of the 'real' world is partly what is at stake here. There are two reasons why we might not be able to leave these matters to the philosophers and social theorists, though these colleagues are no doubt very able for the task. On the one hand, a change in the terms of our social ontology may radically re-orient how we construe the mission of education. On the other hand, the way education has been consistently framed as a key process and apparatus of formation, that is of forming subjects, of building communities, of shaping the social order–even among commentators who might otherwise be in fundamental disagreement–indicates that education cannot be unhooked so easily from a question of social order and thereby of social ontology. It might even be that education is a kind of generative engine for our operative models of social ontology. Education is a space of production and reproduction of various models of what the social world comprises whether that be a social ontology predicated on the priority of individuals, families, faith communities, and enterprises; or of classes, races, and genders; or of distributed socio-technical networks; or of natives, citizens, denizens and migrants; or of the mode of production.

The conceit that set this discussion in play, was that there is an analogy to be drawn between constructionism as a way of talking and activating educational practice, and the typical strategies of higher arts education. Through elaborating that analogy, it was proposed that something of the political stakes in constructionism and its contestation might become readable in a new way. The focus that emerged then was on the way in which higher art education individuated the artist-learner, requiring the artist-learner to give account of their decision-making with respect to both the art practice and the learning process. This individuation was then identified as a point of critique in terms of the figure of the hyper-individualized artist. At this point, Biesta's intervention to the debate on art education was drawn into the discussion. His thesis on subjectification was brought into view as an exemplary instance of the attempt to describe education in terms of a process of individuation that was oriented to a horizon of coming into the world conditioned by intersubjectivity and dialogue with the world, being in the world but not at the centre of the world.

The possibility that this account of subjectification might be caught within a limitation of the universalizing tendency of a particular type of critique was adduced by drawing upon the work Ferreira da Silva. The cursory treatment of Ferreira da Silva's analysis of the transparent subject and its prior constitution in a racial difference served as a means of evidencing the importance of different ontological commitments for founding very different accounts of education; and for signaling the unclear apprehension of the limits of the critique that both Biesta and I are practicing. The question that remains then is what has any of this last phase of the discussion got to do with constructionism?

Let us return to the analogy of fine art education and constructionism. The problematic that emerges within my analysis of higher arts education is the figure of the hyper-individualist. It is a presented as a problem about the core terms of the educational setting and mission. It is a problem not only of what is worth knowing (a problem of epistemology), not only of how to educate, (a problem of methodology), but also a problem of what constitutes the social world (a problem of ontology). My proposal is that the learner-centric disposition of constructionism requires attending to a similar problem. It is not that constructionism is necessarily producing the same hyper-individualists —but it is operating within the same broad neoliberal regime of individuation, and so faces similar challenges. The extreme case of higher arts education allows the problem to come into a more pronounced visibility, and that may be useful. However, the way in which the attempt to re-negotiate the terms of individuation through Biesta's model of subjectification, and the interruption of Biesta's work by the work of Ferreira da Silva, brings out the ontological problem of the subject and the ontological problem of race. Here I think constructionism also must engage. While constructionism may be well placed to navigate questions of epistemology, and questions of methodology, it is perhaps less equipped to address these questions of social ontology. However, it seems also important not to delegate these questions to philosophers and to social theorists. So what is to be done?

Firstly, constructionism does not need to become a theory of everything, neither a way of speaking of *all educations* nor a way of speaking *all of education*. Constructionism can be a way of talking about some education, a way of talking that is really useful for augmenting and enhancing some parts of that work, in some instances. It does not need to seek, and I will risk saying it should not seek, to be a totalizing discourse that assumes complete coverage. Secondly, constructionist accounts as these are deployed in so many different situations, are already operating implicit or taken-for-granted ontological commitments. Perhaps these can be excavated. In that excavation perhaps the question of race, for example, could be taken not as a question of externality—"constructionism *and* race"—but rather as a question of internal constitutive process, e.g., the question of what ways race/racialization might already be operative within constructionism. Importantly this is not proposed as a question of ethical reflection nor of moral self-examination, but rather of practical and conceptual analysis. Such an analysis might start with considering how to take on board the proposition that: systemic, structural and institutional racism is operated, cognized and experienced differentially by always already (differentially) racialized subjects. Attending to this claim would already suggest some parameters within which to begin the analysis.

Finally, let me revert to the area of my limited competence which is higher arts education. The project method of higher arts education has enormous flexibility and allows certain issues to readily come into focus, but cannot provide complete coverage. It has certain intrinsic limitations and problematic tendencies that need to be further excavated. Within this domain we have considerable challenges, and all that I have proposed with respect to constructionism applies *a fortiori* to fine art education. Biesta's proposition that as a learner and as a teacher one might risk *Letting Art Teach* is something that retains great value. My hunch is that the problematic of fine art educaton may be helpfully reconfigured if Ferreira da Silva's analysis of the matrix of universality/subject/critique/race, that has been very provisionally and only cursorily introduced here, can be more thoroughly explored with respect to Biesta's propositions.

References

Biesta, G. (2006) *Beyond learning: Democratic education for a human future*. Boulder, Paradigm.

Biesta, G. (2010) *Good Education in an Age of Measurement. Ethics, Politics, Democracy.* Boulder, Paradigm.

Biesta, G. (2017a) *Letting Art Teach*. ArtEZ Press, Arnhem.

Biesta, G. (2017b) *The Rediscovery of Teaching.* Routledge, London.

Biesta, G. (2019) *Obstinate Education*. Brill | Sense, Leiden.

Biesta, G. (2020) Educational Research: An Unorthodox Introduction. Bloomsbury Academic, London, New Delhi, New York, Sydney.

Boltanski, L. and Chiapelo, E. (2005) *The New Spirit of Capitalism*. Verso, London

Ceaser, J. W. (2008) How to Think About the Foundations of American Conservatism. The Heritage Foundation, Washington DC. https://www.heritage.org/political-process/report/how-think-about-the-foundations-american-conservatism [22/05/2020].

Cockcroft, E. (1974) *Abstract Expressionism, weapon of the Cold War.* Artforum, vol. 15, no. 10, 39–41.

Ferreira da Silva, D. (2007) *Toward a Global Idea of Race*. University of Minnesota Press.

Fereirra da Silva, D. (2019) *An End to "this" World Denise Ferreira da Silva interviewed by Susanne Leeb and Kerstin Stakemeie*. Text zur Kuntse, Berlin. https://www.textezurkunst.de/articles/interview-ferreira-da-silva/ [22/05/2020].

Guilbaut, S. (1983) *How New York Stole the Idea of Modern Art: Abstract Expressionism, Freedom and the Cold War*. University of Chicago Press, Chicago.

Heimans, S. and Biesta, G. (2020) Rediscovering the beauty and risk of education research and teaching: an interview with Gert Biesta by Stephen Heimans. Asia-Pacific Journal of Teacher Education, 48:2, 101-111.

Iber, P (2015) *Neither Peace nor Freedom: The Cultural Cold War in Latin America.* Harvard University Press, Cambridge MA.

Kant, I. (1784) *Beantwortung der Frage: Was ist Aufklärung?* Berlinische Monatsschrift. See translation "An answer to the question: What is enlightenment?" in Gregor, M.J. (ed.). Practical Philosophy. The Cambridge Edition of the Works of Immanuel Kant. Cambridge University Press. 11-12.

Kelley, S. (2013) 'But that was my idea!' Problems of Authorship and Validation in Contemporary Practices of Creative Dissent. Parallax, 19:2, 53-69.

Law Revision Counsel of the U.S. House of Representatives (1947) U.S. Code Title 1. GENERAL PROVISIONS Chapter 1. RULES OF CONSTRUCTION Section 8. "Person", "human being", "child", and "individual" as including born-alive infant, Government Printing Office. https://www.govinfo.gov/content/pkg/USCODE-2018-title1/pdf/USCODE-2018-title1-chap1.pdf [22/05/2020].

Lindström, S. (2016) *(Un)bearable freedom. Exploring the becoming of the artist in education, work and family life.* Linköping University, Norrköping.

Papert, S. and Harel, I. (1991) Constructionism. Ablex Publishing Corporation, New York.

Scruton, R. (2001) *The Meaning of Conservatism*. Palgrave Macmillan, Basingstoke, Hampshire.

Singerman, H. (1999) *Art Subjects: Making Artists in the American University*. University of California Press, Oakland.

Steyerl, H. (2010) Politics of Art: Contemporary Art and the Transition to Post-Democracy. e-flux Journal, 21, http://www.e-flux.com/journal/politics-of-art-contemporary-art-and-thetransition-to post-democracy/. [22/05/2020].

Stonor Saunders, F. (1999) *Who Paid the Piper? The CIA and the Cultural Cold War*. Granta, London.

Thompson, Christine Marmé (2015) *Constructivism in the Art Classroom: Praxis and Policy*. Arts Education Policy Review, 116:3, 118-127.

The significance of Constructionism as a distinctive pedagogy

Diana Laurillard, *d.laurillard@ucl.ac.uk*
UCL Knowledge Lab, Institute of Education, University College London, UK

Abstract

The paper sets the pedagogy of Constructionism in the context of the Conversational Framework, which was derived from theory and research on how students learn as a guide for teachers and academics. The framework shows how six distinct types of learning can combine to represent all the different types of exchange between teacher and learner, and between the learner and their peer, in order to support an effective learning experience. Four of these types of learning are especially relevant to the ways in which Constructionism has been characterised, and the paper interprets the framework in relation to one of them: learning through practice with intrinsic feedback, which is critical to the idea of Constructionism. To illustrate this interpretation, it then summarises the use of a constructionist pedagogy to design an adaptive digital game for the challenging context of supporting dyscalculic learners in developing their 'number sense'. It concludes by recommending a wider application of Constructionism that embraces learning that involves activities other than the more typical ones of programming and coding.

Keywords

Constructionism, pedagogy, Conversational Framework, Prediction-error learning, collaborative learning, intrinsic feedback, adaptive digital game, dyscalculia

Introduction

Constructionism as a pedagogy is clearly distinctive, due to its key elements, articulated originally by Seymour Papert, and followed through either completely or partially by everyone who references it:

- a digital environment designed to afford the learning of some system or set of concepts and powerful ideas (Healy & Kynigos, 2010; Noss & Hoyles, 2006; Papert, 1980).
- a microworld that affords the learning of a concept, "a place where the student, through playing, may stumble over and then ponder important inspirations and concepts" (Hoyles, Noss, & Adamson, 2002; p. 29).
- manipulating the designer's model of mathematical object behaviour (Edwards, 1998),
- actively engaged in constructing a public entity (Papert and Harel 1991).

The central insight of constructionism is that if learning is situated in a meaningful context with a meaningful goal in view, the learner can use just the direct feedback from the environment to improve their actions, without needing further external advice or guidance. The feedback is intrinsic to the action, showing the result of the action in relation to the intended goal, enabling the learner can work out how to improve their action without extrinsic teacher intervention (Laurillard, 2012).

The theoretical underpinnings of constructionism are difficult to pin down in most of its literature, but it had a strong influence on the work I did to develop the Conversational Framework, a way of summarising for the practising teacher the key findings of theory and experiment on how students learn. Teachers as designers, whether in a conventional or digital context, would benefit from being able to draw on this work, without having to read too many books and research papers.

Constructionism was important because it studied the learner learning from engaging with a dynamic digital environment.

This was important for some work I was doing on finding ways to support learners who are dyscalculic, i.e. who lack the more typical 'number sense' that most of us use for understanding the number line and basic arithmetic. The paper therefore uses this to illustrate one of the ways in which Constructionism is distinctive in the way it guides the designer of adaptive digital games.

To begin with, it introduces the Conversational Framework in terms of its original aim, to be a research-based approach to guiding an effective pedagogy, especially in the context of optimising the combination of conventional and digital methods (Laurillard, 2002).

What does it take to learn in formal education?

The question is a challenging one, but important for every teacher or academic who has to guide and nurture their students through understanding and using progressively complex concepts and skills as they move through the education sectors. The Conversational Framework was derived from a wide range of educational and psychological theories and empirical research to provide teachers and academics with a simplified, consolidated account of how students learn, to guide the design of their teaching and learning, in any context, conventional or digital, and in any education sector .

Figure 1 shows a simple way to think about how a learner is learning. The central panel represents the individual learner, who has some concepts, or knowledge, and some practices, or skills. These are influenced in their development by the teacher, who communicates at the conceptual level, and through learning activities at the practice level. They are also influenced by the learner's peers, who communicate through discussion at the conceptual level, and by sharing practice at the practice level.

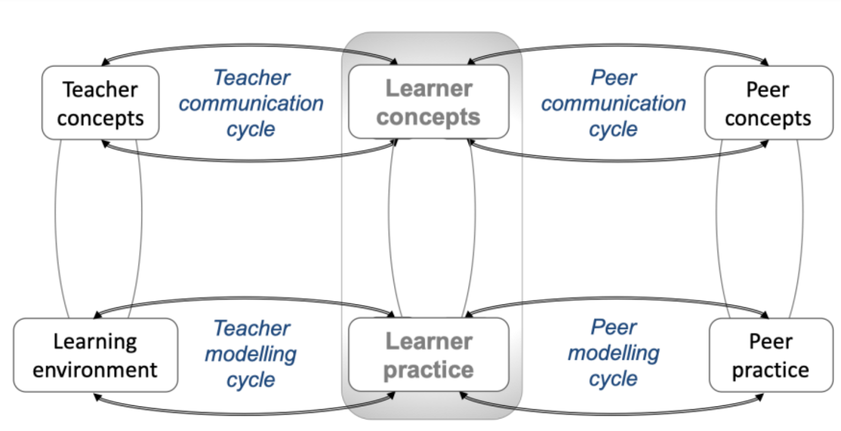


Figure 1: The Conversational Framework interactions between teacher and learner, and learner and peers

The Teacher concepts node stands for the human teacher, or their representation via a text, or video, or diagram, or podcast, etc. The Learning environment node stands for the context in which the learner is asked to put their concepts into practice: the materials or resources or tools for an exercise, or experiment.

At the concept level, there are continual iterations of ideas, as they gradually develop a concept, and join it up with other concepts. At the practice level as well, there are continual iterations of actions in relation to a goal, which gradually develop a skill, and then gradually more complex skills. It is also important that the concepts influence the learner's practice, and conversely, their practice influences the development of their concepts, in another continuous cycle. In formal education we have to make sure that all these continual cycles are working, and the learner is developing their concepts and practices together.

The teacher does that by engaging the learner in thinking about concepts, and by setting up the learning environment for them to put those ideas into practice: if you teach them about adding 2 and 2, you give them blocks to practice on; if you teach them about democracy, you run a mock election. Their peers are important too, because they discuss concepts, and they practice skills together.

The teacher-designer will aim to elicit all those iterative exchanges that help to develop and integrate concepts and practice, and this is made easier if the typical types of learning that are manifested in education are mapped onto the framework, for ease of interpretation. There could be many, but to reduce complexity the whole framework can be adequately represented through 6 contrasting learning types, utilising different parts of the framework, as shown in Figure 2.

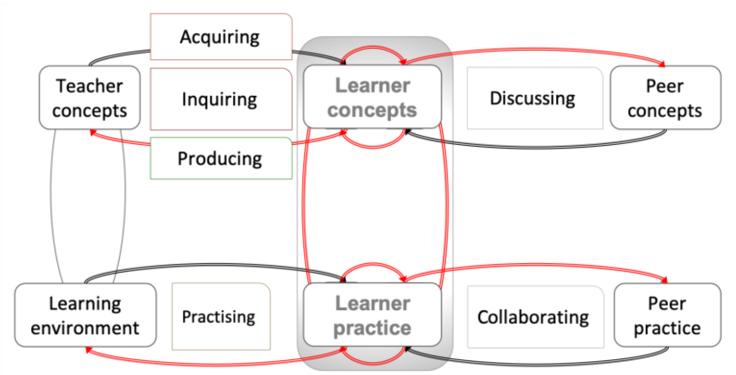


Figure 2: The range of common learning types mapped to the Conversational Framework

The first two learning types are briefly summarised as:

- 'learning through acquisition': the teacher (human, book, website, etc) communicates (one-way) concepts and ideas, and the learner reads, watches or listens
- 'learning through investigation' or 'inquiry': the teacher asks learners to explore or question (two-way) the Teacher concepts. In this case they generate their own ideas of what they want to know

The latter iteration produces more conceptual activity by the learner than 'acquisition' does, because they generate the questions, and go to the teacher concepts (represented in the teacher, or in books, or on the internet), directing their own learning. The more cycles there are, the more opportunities they have to change and develop their ideas.

The remainder are all more relevant to representing the pedagogy of Constructionism, e.g.

- 'learning through practice': the learner uses the learning environment set up by the teacher to create exercises for the learners; ideally it includes a goal, the means for learners to put their concepts into practice to achieve it, feedback on their action, and the opportunity to revise and improve it.

Before we come to the role of the learner's peers, we consider the framework exchanges in more detail, to look at an interesting property of Constructionism': that it presupposes 'intrinsic' rather than 'extrinsic' feedback on the learner's actions as they learn through practice. The two are contrasted in Figure 3, which maps the two types of learning through practice learning on to the framework. Figure 3a shows practice with extrinsic feedback, i.e. from some external judge, other than the learner, as the sequence: Goal - Action - Modulate Teacher Guidance - Generate learner

Action - Modulate teacher Guidance - Generate learner Action, and so on. Notice that this iteration does not need to change the Learner concepts, as the teacher guidance has told them what to do.

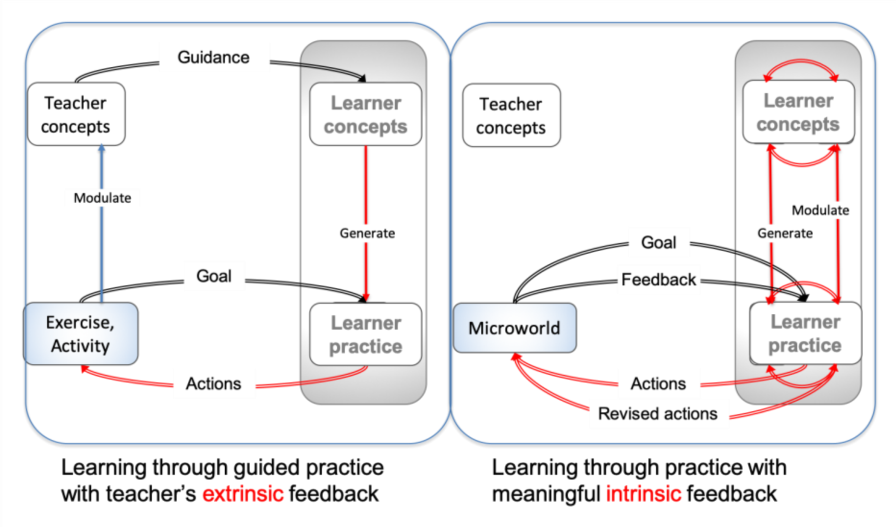


Figure 3: (a) Extrinsic feedback is given by the Teacher (person or program); (b) Intrinsic feedback is given by the Learning envronment (exercise or microworld)

Contrast that representation with the one for learning through practice with intrinsic feedback in Figure 3b, i.e. information from the learning environment on the results of the learner's action. Here the sequence is Goal - Action - Feedback - Revised Action - Feedback - Modulate Learner concepts - Generate Learner practice - Revised Action - Feedback, and so on. The microworld generates meaningful feeback on the results of the learner's actions, eliciting more careful engagement of their concepts to generate a better action. With luck. At least the opportunity is there.

This is why Papert could say that constructionist exercises enabled learning without a teacher. The teacher, in the form or a person, or a computer program running a multiple choice exercise, is not needed to comment or inform. The microworld, like the real world in the right context, can provide the 'informational feedback' the learner needs.

This is one feature of constructionism that makes it a distinctive pedagogy. It is recruiting the natural learning processes we use from day one to learn about how to act on the world around us.

Prediction-error learning

The importance of this kind of pedagogy is explained by appealing to the conclusions of neuroscientists who have studied how the brain learns, drawing on classical psychology, cognitive psychology, neural networks, and neuroimaging (Thomas & Laurillard, 2012). They emphasise the importance of feedback on actions in the world, and especially of feedback on errors. Errors are crucial for learning:

> ”learning occurs only if the brain selects the appropriate sensory inputs (attention), uses them to produce a prediction (active engagement), and evaluates the accuracy of the prediction (error feedback)” (Dehaene 2020, p202).

Setting a goal should elicit attention to appropriate sensory inputs by the learner; having the means to act to achieve the goal will actively engage them; receiving (meaningful) feedback should then enable them to evaluate how well they achieved the goal, and adapt their next action accordingly. “In AI, this type of learning, known as 'supervised', is the most effective, because it allows the machine to quickly identify the source of failure and to amend itself” (Ib, p209). It is the instant, meaningful, informational feedback that allows the learner to 'learn without a teacher'. And accordingly, Dehaene defines the optimal pedagogy for our classrooms in terms of the simple rules that Constructionism embraces, such as to "actively to explore, test and extend their understanding":

> "what is the best way to incorporate our scientific knowledge of error processing into our classrooms? The rules are simple. First, students must be encouraged to participate, to put forth responses, to actively generate hypotheses, however tentative; and second, they must quickly receive objective, non-punitive feedback that allows them to correct themselves" (Dehaene 2020, p214).

As the nature of the exchanges in Figure 3b show, if a microworld is constructed to enable appropriate exploration and testing, this learning environment can support meaning informational feedback on goal-oriented actions, and thereby support learning without a teacher (Dayan & Abbott, 2001).

Learning through collaboration and production

Now consider the role of peer learners in this pedagogy. In the Conversational Framework, similar iterations happen with 'learning through discussion', where the social construction of ideas helps learners develop their concepts; the generate questions, and respond to other learners with answers to their questions, again the iteration helps to develop their concepts.

However, 'learning through collaboration' is more demanding than simple 'discussion' in the top right-hand corner, as Figure 4 shows, because the learners are necessarily collaborating on constructing something together: that is the nature of collaboration.

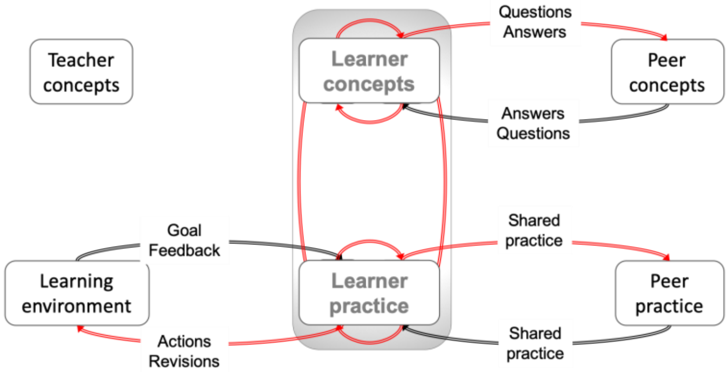


Figure 4: Learning through collaboration

Here, each learner is learning through practice by using the learning environment. And at the same time, they are discussing and sharing that practice. In order to do that, they are necessarily also linking the two, which helps them develop both concepts and practice with each other. The teacher need play no role at all, and yet there is a lot of active internal processing required of the learner during this process.

Finally, 'learning through production', happens when the teacher invites learners to reflect on and represent what they have learned, and communicate this to the teacher, shown in Figure 5.

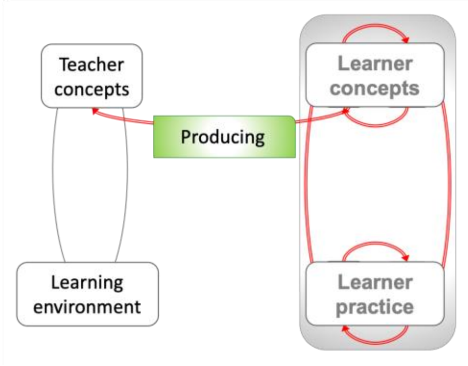


Figure 5: Learning through production

Here the learner must connect up concepts and practice, and then produce an essay, or performance, or design, or presentation to show what they have learned. Throughout this process the learner is actively processing both concepts and practice and the integration of the two. This is akin to what Papert referred to as constructing personally meaningful and shareable artefacts, where the sharing is part of the motivation to construct a successful artefact.

There is no specific ordering to the six learning types, as they can combine and be sequenced in many different ways. The main issue for a teacher is to be aware of the full ranges, and the extent to which their teaching embraces all these different types of conversation, between teacher and learner, learner and peers, and on the levels of both concepts and practice.

Constructionism is represented best through four of the 6 learning types. Learning through acquisition, and inquiry are not a particular focus. The role of the teacher is still critical, however, as it is a real design challenge to generate and modulate the learning environment that could achieve learning without a teacher. Very few achieve that as most rely greatly on the teacher to provide additional guidance and feedback. The teacher will also be the recipient of the artefacts produced by a constructionist pedagogy, able to use these for judging the value of it as a learning process.

A constructionist learning environment typically uses the kind of construction that involves programming or coding. By interpreting Constructionism as a pedagogy within the broad Conversational Framework it is easier to use it also to guide quite different kinds of digital learning, for a wide range of learning outcomes.

To illustrate that, I will use the example of a design challenge where we wanted to help learners with dyscalculia.

An illustration of a constructionist game: NumberBeads

A constructionist learning environment typically uses the kind of construction that involves programming or coding. By interpreting Constructionism as a pedagogy within the broad Conversational Framework it is easier to use it also to guide quite different kinds of digital learning, for a wide range of learning outcomes.

To illustrate that, I will use the example of a design challenge where we wanted to help learners with dyscalculia.

To create a digital game to help such learners it was important to help them see how to make numbers out of other numbers. No existing educational games helped with this, as they all tend to rehearse the use of a concept already understood, realying on multlple choice questions to encourage practice. A Constructionist pedagogy, on the other hand, would recruit their natural prediction-error learning capability in finding the action to achieve a goal. For understanding the internal structure of numbers, therefore, they had to make a target number out of other numbers.

The NumberBeads game is designed to elicit that goal-action-feedback- modulate concept-generate revised action - feedback cycle in the constructing of numbers. Figure 6 shows the game play area where the goal is a target set at the top, the action is to either split or join the sets on offer, and the feedback is the resulting set, which is either a set identical to the target, or something else.

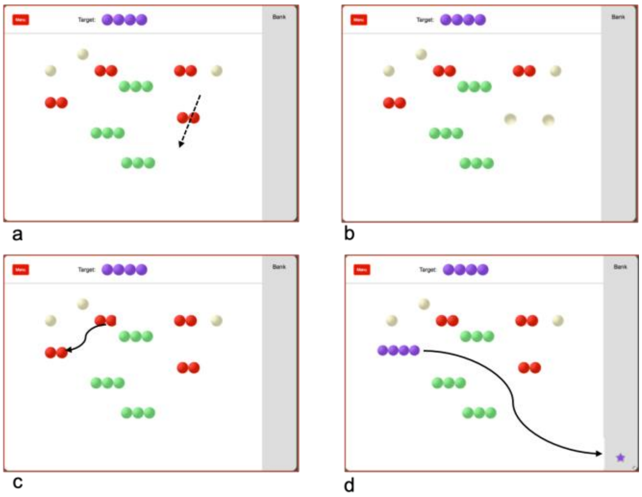


Figure 6: NumberBeads, showing (a) splitting a two set to (b) construct two one sets, then (c) joining a two set to a two set to (d) construct a 4 set, which matches the target at the top, and adds a star to the bank

The learner keeps exploring new combinations or splits to make more target sets. Each target set creates a star in the bank, until they reach 10 stars. There is no right or wrong, just the intrinsic feedback that splitting a 6 set by removing a 1 creates a 5 set. So it is informational feedback, that may help in seeing how to get closer to the target.

As a learner moves through the game, the beads and colours stage is followed by beads + colours + digits, then just beads + digits, then just digits. At a later stage these are repeated with addition, subtraction and = symbols, and the action of subtraction, rather than splitting, is to bring one set under another to remove that number of beads.

The NumberBeads game in this way acts as a microworld in which numbers behave as structures according to game transaction rules that obey the rules of arithmetic. The game thereby recruits the kind of prediction-error learning dyscalculic learners normally apply to the world around them, and gives them a constructionist experience of developing a number sense.

Testing the game against an MCQ version

An initial pilot of NumberBeads in UK schools, has shown that the game supports learners age 5-7 years for independent learning of the kind that low attaining learners will need in order to keep pace with mainstream learners (Laurillard, 2016).

To test the effectiveness of the constructionist pedagogy against that of games that employ an essentially multiple-choice transaction, we created a NumberChoice version of the game. This also shows a target, but for addition problems, for example, asked the learner to select which of 3 sets could be combined with a given set to create the target. It progresses through same sequence of formats.

The games have been trialled with two groups of students in Italy, 80 on NumberBeads (NB) and 60 on NumberChoice (NC) including a low-performing group of 13 on NB and a matched sample of 16 on NC. All were given a standard curriculum test before and after using the games, which they were asked to play at home for up to 15 minutes a day for 3 weeks, or less if they completed more quickly. The platform automatically records every learner action and its timing.

The data has been analysed and is ready for publication, so details cannot be given here. However, broadly, it shows that the constructionist pedagogy of NB does have an effect in comparison with NC, and is played more intensively, i.e. more completed stages in less time. It is also important that it showed that successful play to full completion was feasible at home, even for the low-attaining learners, without a teacher, and with very little guidance to parents.

The results are promising therefore, and will be reported to the Constructionism community at a later stage.

Concluding points

Constructionism is a distinctive pedagogy because it focuses the teacher-designer on what it takes to learn if the teacher is not present. The teacher is not available for very much of a learner's learning time at any stage of education, and even more rarely for 1-1 interaction and feedback at the point of needing it. Because of this simple fact, we have to inculcate independent and self-regulated learning at all stages, even early primary. The paper argues that Constructionism's fundamental idea of a digital environment that enables the learner to learn from the intrinsic feedback on their goal-oriented action is what makes it distinctive. The teacher still plays a vital role in designing that environment, whether using coding environments, modelling, simulations, or games. The teacher must then embed it in a social and collaborative engagement with other learners, and in the context of the concepts taught through learning through acquisition and inquiry.

Acknowledgement

The games, NumberBeads and NumberChoice were developed with funding from the Science of Learning Research Centre, based at the University of Queensland. The games were programmed by Cauldron.

References

Dayan, P., & Abbott, L. (2001). Classical conditioning and reinforcement learning. In P. Dayan & L. Abbott (Eds.), *Theoretical Neuroscience* (pp. 331-358). Cambridge, MA: MIT Press.

Dehaene, S. (2020). *How We Learn: Why Brains Learn Better Than Any Machine…for Now*: Penguin.

Edwards, L. D. (1998). Embodying mathematics and science: Microworlds as representations. *Journal of Mathematical Behaviour, 17*(1), 53-78. http://dx.doi.org/10.1016/S0732-3123(99)80061-3

Girvan, C. and Savage, T. (2019) Virtual Worlds: A New Environment for Constructionist Learning https://www.sciencedirect.com/science/article/pii/S0747563219301098?via%3Dihub

Healy, L., & Kynigos, C. (2010). Charting the microworld territory over time: Design and construction in mathematics education. *ZDM Mathematics Education, 42*(1), 63-76. http://dx.doi.org/10.1007/s11858-009-0193-5

Hoyles, C., Noss, R., & Adamson, R. (2002). Rethinking the microworld idea. *Journal of Educational Computing Research, 27*(1), 29-53. http://dx.doi.org/10.2190/U6X9-0M6H-MU1Q-V36X

Laurillard, D. (2002). *Rethinking University Teaching: A Conversational Framework for the Effective Use of Learning Technologies* (2nd ed.). London: RoutledgeFalmer.

Laurillard, D. (2012). *Teaching as a Design Science: Building Pedagogical Patterns for Learning and Technology*. New York and London: Routledge.

Laurillard, D. (2016). Learning *number sense* through digital games with intrinsic feedback. *Australasian Journal of Educational Technology, 32*(6), 13pp. doi:http://dx.doi.org/10.14742/ajet.3116

Noss, R., & Hoyles, C. (2006). Exploring mathematics through construction and collaboration. In R. K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (389–405). Cambridge, MA: Cambridge University Press.

Papert, S., & Harel, I. (1991). Situating Constructionism. In I. Harel & S. Papert (Eds.), *Constructionism: research reports and essays, 1985-1990* (1-11). Norwood, N.J.: Ablex Pub. Corp.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Brighton, Sussex: The Harvester Press.

Thomas, M. S. C., & Laurillard, D. (2012). Computational modelling of learning and teaching. In D. Mareschal, A. Tolmie, & B. Butterworth (Eds.), *Handbook of Educational Neuroscience* (46-76). Oxford: Wiley-Blackwell.

Panels, Demonstrations & Workshops

From design to implementation: adaptations and uses of ScratchMaths in different countries

Celia Hoyles, c.hoyles@ucl.ac.uk, *UCL Institute of Education, U.K.*
UCL Institute of Education, 20 Bedford Way, WC1H 0AL, U.K.

Ivan Kalaš, Ivan.Kalas@fmph.uniba.sk**,** *Comenius University,*
Department of Education, Bratislava, Slovakia

Richard Noss, *r.noss@ucl.ac.uk, UCL Institute of Education, U.K..*
UCL Institute of Education, 20 Bedford Way, WC1H 0AL, U.K.

Paul Goldenberg, *EPGoldenberg@edc.org, Education Development Center, USA.*

Elena Prieto-Rodriguez, elena.prieto@newcastle.edu.au, *School of Education, University of Newcastle*, Australia

Piers Saunders, *piers.saunders@ucl.ac.uk UCL Institute of Education, U.K.*
UCL Institute of Education, 20 Bedford Way, WC1H 0AL, U.K.

ScratchMaths(SM) is a two-year curriculum for students aged 9-11 years iteratively designed over several years with school partners to fit at the interface between Computing and Mathematics. SM develops Scratch programming skills and computational thinking with explicit links to areas of mathematics where programming can enhance mathematical reasoning. All materials freely available through UCL website http://www.ucl.ac.uk/scratchmaths.

The ScratchMaths project will be introduced, along with pedagogical approach adopted in England. The outcomes described along with some examples from the curriculum. ScratchMaths has proved rather successful (see references below) and is still in widespread use in England although there are challenges that will be explored. In addition, ScratchMaths has been taken up in many countries across the world, including Australia, Spain, New Zealand, China, Slovakia, Czech Republic. How and why these different countries have chosen to take up ScratchMaths varies enormously in terms of the curriculum and materials selected, the focus of attention, how far the content is changed, the research paradigm adopted, even the software used. Unsurprisingly, ScratchMaths use and spread is shaped in fundamental ways by the personal vision of the 'local' leads and the context of implementation: in England, for example there is a compulsory national computing curriculum and a National Centre for Computing Education all funded by the Government.

In the proposed panel discussion, each participant will describe the focus of their work in relation to ScratchMaths design and implementation, any findings or outcomes, some of the major adaptations made and the reasons for these, major obstacles and finally what the next steps might be.

Keywords

Programming, mathematics, primary, international, Scratchmaths

Chair's bio – Celia Hoyles

Celia Hoyles (Chair) is Professor of Mathematics Education at UCL Institute of Education, University College London. She taught mathematics in London schools and was inspired by the vision of using digital technology to open access to mathematics and has led many research and development projects to promote this aim. She has advised UK Government on policy in Mathematics. In 2016, Celia received the Suffrage Science award for Communications in acknowledgement of her scientific achievements and her ability to inspire others especially women into mathematics.

Celia will ask each panellist to talk to the topic of the panel from their perspective: what were the challenges in their implementation of ScratchMaths, what was their focus and why, and how they adapted SM if relevant and why.

Panellist bio - Ivan Kalaš

Ivan Kalaš is a professor of computing education at the Department of Education, Comenius University, Bratislava, Slovakia. His professional interests include development of constructionist educational interfaces for programming, and research in the field of educational programming for primary and secondary students. Currently he is head of the Department of Education at Comenius. Ivan is a co-author of several programming environments for children adopted by thousands of schools around the world. Between 2014 and 2016 he was a member of the UCL ScratchMaths project team, designing programming interventions for pupils aged 9 to 11. Since 2017 he leads a new design research project – Computing with Emil, focused on systematic constructivist development of computational thinking for primary pupils.

Panellist bio – Richard Noss

Richard Noss is Emeritus Professor of Mathematics Education at UCL Institute of Education. He has spent most of his academic career exploring the possibilities of programming as an alternative representational infrastructure for developing mathematical reasoning. Richard was director of the National Program, Technology Enhanced Learning, was the founding Director of the London Knowledge Lab and was co-director of the ScratchMaths project. Richard will compare vision and reality – trying to provide answers to some of the questions that arise from SM's findings.

Panellist bio – Paul Goldenberg

Paul Goldenberg is a distinguished scholar at Education Development Center in Boston, USA, designs mathematics and computer science (CS) instructional resources for learners and teachers, focusing on developing mathematical habits of mind including the disposition to puzzle things through, and drawing on natural curiosity and strengths-based approaches to support learning. He has taught elementary-school everything, high school CS, and graduate school mathematics and psychology for teachers. In his current work—Math+C and Think Math+C—7+-year-olds use programming as a language for expressing and exploring the mathematics they are learning.

Panellist bio - Elena Prieto-Rodriguez

Elena Prieto-Rodriguez, School of Education, University of Newcastle, Australia holds a Bachelor degree in Mathematics and a PhD in Theoretical Computer Science. From 2005, she has worked extensively in Mathematics education, including several Australia-wide research projects. She is currently engaged in projects focused on the use of technology for the learning of mathematics and teacher training and professional development. She conducted a small scale pilot of ScratchMaths in 2017 and found teachers deeply committed to the integration of programming into mathematics classes but needing explicit connections to coding within the Australian mathematics curriculum.

Panellist bio – Piers Saunders

Piers Saunders is the Head of Initial Teacher Education and lecturer in Mathematics education at the UCL Institute of Education, London. He is an experienced secondary school mathematics teacher, working in multiple schools in London. Piers has led the Secondary Teach First programme at UCL across eleven subjects, as well as leading the national mathematics Teach First provision and curriculum development working with seven local university providers. He has also had a career in the software industry spending significant time working in Canada as a software development manager for a large IT company. Piers is currently working on a Teacher Education project in partnership with the Queen Rania Teacher Academy (QRTA) in Jordan. Piers worked on the ScratchMaths project focusing on design of the curriculum materials and designing and delivering

References

Benton, L. Kalas, I; Saunders, P; Hoyles, C; Noss, R. (2018) Beyond Jam Sandwiches and Cups of Tea: An Exploration of Primary Pupils' Algorithm-Evaluation Strategies *J of Computer Assisted Learning* https://doi.org/10.1111/jcal.12266

ScratchMaths: Integrating Computing and Mathematics.(2019) *HelloWorld* May

Hands on with teaching ScratchMaths (2019) *HelloWorld Oct*

Noss, R., Hoyles, C., Saunders, P., Clark-Wilson, A., Benton, L., and Kalas, I., (2019 in press) *Constructionism can work: the story of ScratchMaths* In Holbert, N., Berland, M., Kafai, Y. Constructionism in Context: The Art, Theory, and Practice of Learning Designs. MIT Press.

A Defense of (s)crappy Robots

Ryan Jenkins *ryan@wonderfulidea.co*
Wonderful Idea Co, USA

Mairéad Hurley, *mairead.hurley@dublin.sciencegallery.com*
Science Gallery, Dublin, Ireland

Eva Durall, *eva.durall@aalto.fi*
Dept of Media, Aalto University, Espoo, Finland

Sebastian Martin, *sebastianm@exploratorium.edu*
Exploratorium, San Francisco, USA

A Defense of (s)crappy Robots

Constructionist education philosophy suggests that learners develop understanding about science, art and technology by creating physical artifacts that strengthen the connections. As makerspaces and STEAM learning environments have become more and more common, many commercial kits for robotics, electric circuits and digital technology have been created for schools and individual learners.

While these kits might lower the threshold for entry in a tinkering activity, we believe that the "polished" qualities of most commercial kits or products can work against constructionist educators. When the parts of a kit fit together perfectly, learners are less likely to develop skills around iteration, problem posing, collaboration and development of understanding. We're also concerned with the possibility that the approach and aesthetic qualities of these products tend to be directed to specific groups thus creating a less inclusive environment.

We propose a more scrappy, playful approach to thinking about engineering and robotics that relies on learners manipulating recycled containers, cardboard, vegetables or other everyday materials to create unique and whimsical designs.

We are inspired by the "shitty robots" made by Simone Giertz and the Hebocon robot sumo contest from Japan to create workshops where learners need to embrace frustration, celebrate moments where things don't work as planned and taking risks with designs.

In this hands-on workshop, participants will engage in making scrappy DIY robots out of everyday materials that are not designed to work perfectly. We'll use recycled materials, hobby motors, batteries and homemade switches in the construction process and show some possibilities for adding programming or digital tools to the mix. Participants will collaborate with others on the design of the machine and share their prototype with the rest of the group.

The experience of trying the activity as a learner will inform a discussion about the qualities of learning that we noticed in the workshop. We'll reflect on the value of creating robots from scratch and working with unexpected scraps. The workshop leaders will share practical tips and frameworks for running these workshops, discuss how the SySTEM 2020 project is developing principles for the design of inclusive non-formal learning activities and we'll think together about the ways that unexpected materials can replace or augment commercial, polished robotics kits to create more valuable learning experiences.

Advanced Yet Accessible CS Concepts in K12

Brian Broll, *brian.broll@vanderbilt.edu;* **Corey Brady,** *corey.brady@vanderbilt.edu;*
Akos Ledeczi, *akos.ledeczi@vanderbilt.edu*

Computational Thinking and Learning Initiative, Vanderbilt University, Nashville, TN, USA.

The 21st century is undoubtedly the century of computer science (CS). Distributed and cloud computing, artificial intelligence and machine learning, autonomous systems and cyber-security, big data and the internet of things are new frontiers of computing that are fundamentally transforming how people work, communicate and live. Computation is also transforming innovation in every discipline, becoming an integral tool that is spurring new ways of doing and thinking. Yet, the powerful ideas of these approaches are mainly exposed only to college CS majors. Based on Snap!, NetsBlox is specifically designed to offer an accessible introduction to these advanced CS concepts. Remote Procedure Calls (RPC) allow students' programs to access online data sources and services (e.g., Google Maps, weather and climate data, Twitter feeds, and much more). More than simply providing access to web APIs, a NetsBlox RPC can simplify both the query and the returned results, and NetsBlox RPCs be created to provide services where publicly available APIs do not exist. For example, NetsBlox supports cloud variables and charting through an interface to gnuplot running on the NetsBlox server.

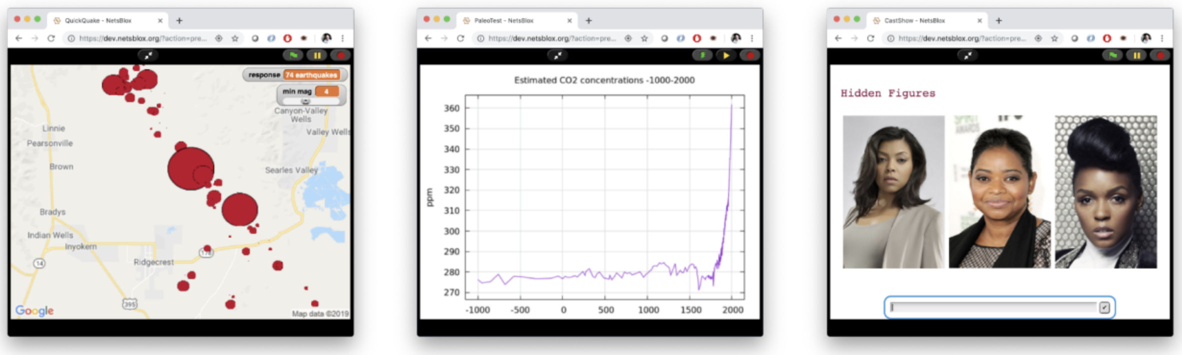


Figure 1. Representative projects: earthquakes near Ridgecrest, CA during the week of July 1, 2019 (left), estimated CO2 concentrations for the past 3000 years from the Dome C ice core in Antarctica (center), and the leading cast members of any movie based on its title (right).

Peer-to-peer communication is also supported by NetsBlox Messages. Messages are very similar to the Events already present in Snap! and Scratch. In NetsBlox, a Message is an Event that contains data payload and can be sent to other NetsBlox programs across the internet. This enables beginning students to create online multiplayer games and other truly distributed programs, often in their first NetsBlox projects. Completed projects can run on iOS and Android phones through the NetsBlox app. Finally, NetsBlox allows collaborative program development. Students can work on shared projects from their own computers simultaneously whether they sit next to each other or live in different countries. This allows true pair programming and team projects supporting the development of collaborative problem solving skills.

Our demonstration will highlight these and other features of the NetsBlox programming environment that make powerful ideas at the forefront of CS accessible. We hypothesize that a tool that enables teaching these advanced computing concepts and practices in interdisciplinary contexts will support learners' creativity and help them to see connections between computation and their own emerging identities and possible futures.

VotestratesML: Social Studies as a Vehicle for Teaching Machine Learning

Magnus Høholt Kaspersen, *magnushk@cs.au.dk*
Department of Computer Science, Aarhus University, Aarhus, Denmark

Karl-Emil Kjær Bilstrup, *keb@cs.au.dk*
Department of Computer Science, Aarhus University, Aarhus, Denmark

In this demonstration, we present VotestratesML; a collaborative learning tool for teaching machine learning (ML) in a high school Social Studies classroom using voter profile data. As ML becomes increasingly widespread in society (O'Neil, 2017), the importance of ML-literacy increases, and while traditional Computational Thinking (CT) as popularised by Wing (2006) covers technical aspects, a focus is needed on how ML changes society and our lives. We seek to Computationally Empower (Dindler, Smith & Iversen, 2020; Iversen, Smith, & Dindler, 2018) students to take part in the technological development and engage them in the larger questions about ML's role in democratic societies. Based on the goal of Scandinavian high schools to prepare and empower students to participate in democratic society, we investigate how the Social Studies classroom can be used as a vehicle to support students' learning and critical reflection about ML.

We have designed VotestratesML; a construction kit for ML, where Social Studies students use a web-application to explore the role of ML in political campaigns by constructing their own models. The design of VotestratesML is based on deconstructionism (J. M. Griffin, 2018) and specifically J. Griffin, Kaplan, and Burke (2012)'s *explore'ems*; interfaces designed to let students and teachers explore a technology, even if unfamiliar with its core concepts.

In VotestratesML, students collaborate in small groups to create the best possible model for predicting voter behaviour by tinkering with the data-set, features and model parameters to explore how different aspects of a ML model influence its predictions. Students are encouraged to draw on their Social Studies expertise, such as considering existing theories of voter-behaviour when selecting features in the real-world data-set. Following the group work, students compare their models by predicting the voter behaviour of a predetermined set of voter-personas and discuss the implications of using such models in political campaigns.

We have deployed VotestratesML in a study involving two Danish high school Social Studies classrooms and a total of 61 students, aged 17-20, in a three-lecture unit on the use of ML in political campaigns. We found that the contextualisation of VotestratesML as a Social Studies-specific tool was successful in motivating students in working with ML. We also found that students were able to use their Social Studies vocabulary to argue for choices made while working with VotestratesML, that they formed more nuanced views on the use of ML and were able to reflect on the pros and cons of using machine learning in political campaigns.

During the demonstration of VotestratesML, attendees will gain hands-on experience with VotestratesML, build their own ML-models for predicting voter behaviour, and be invited to participate in our on-going discussions on Computationally Empowering students in reflecting on the role of emerging technologies and in particular machine learning in society and their everyday lives.

References

Dindler, C., Smith R., & Iversen O. S. (2020). *Computational empowerment: participatory design in education*. CoDesign, Volume 16, Issue 1, pp. 66–80.

Griffin J. M., Kaplan E. and Burke Q. (2012), *Debug'ems and other Deconstruction Kits for STEM learning*, In proceedings of IEEE 2nd Integrated STEM Education Conference, Ewing, NJ, pp. 1-4.

Griffin, J. M. (2018). *Constructionism and De-Constructionism as Complementary Pedagogies*. In proceedings of Constructionism 2018, Vilnius, Lithuania, pp. 225–237.

Iversen, O. S., Smith, R. C., & Dindler, C. (2018). *From computational thinking to computational empowerment*. In proceedings of the 15th Participatory Design Conference (PDC) 2018, Hasseltand Genk, Belgium, ACM Press, pp. 1-11.

O'Neil, C. (2017). *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy.* Vol. 52, No. 2, Penguin Books.

Wing, J. M. (2006). *Computational thinking.* In Communications of the ACM. Volume 49, Issue 3, pp. 33-35

Target audience

The target audience of this demonstration is scholars in Computational Thinking whom are interested in learning tools for emerging technologies and particularly in machine learning. It is also intended for scholars interested in teaching (and in general discussing) ethical and moral issues surrounding the use of technology.

Creative, engaging and playful activities with Smartphones and Embroidery Machines

Bernadette Spieler, *Bernadette.spieler@uni-hildesheim.de*
Institute of Mathematics and Applied Informatics, University of Hildesheim, Hildesheim, Germany

Vesna Krnjic, *vesna.krnjic@ist.tugraz.at*
Institute for Software Technology, Graz University of Technology, Graz, Austria

Smartphones have become the predominant way for teenagers to interact online and mobile apps provide a unique opportunity to engage this age group. Teenagers use smartphones on a worldwide basis intensely every day around the clock. They use them both during leisure time, for schools, e.g., to research for homework, and to communicate with their friends. The Catrobat apps are used massively in exactly these areas, in the languages of the users, in particular for playing and creating games and in this way for learning about coding and other application areas they need to create their apps, e.g., mathematics or language skills. To engage teenagers and girls in particular in coding, a new project started in September 2018, with the name "Code'n'Stitch" (https://catrob.at/codeNstitch). During this project, our Pocket Code app (Android: https://catrob.at/pc, iOS: https://catrob.at/PCios) has been extended with the option to program embroidery machines (very similar to the existing TurtleStitch project https://www.turtlestitch.org/, a desktop-based environment). In this way, self-created patterns and designs can be stitched on t-shirts, pants, or even bags. Patterns and different forms can be created using our visual programming language Catrobat. As a result, teenagers have something they can be proud of, something they can wear, and they can show to others. A special emphasis is given to a gender-equitable conception to consider different requirements, needs, and interests of our target group. On the one hand, with this option, the team wants to show young women new ways of using technology, with a lot of fun in a sustainable way. On the other hand, young men can get inspired too through this digital design process and the possibility of new challenges in textile handicraft lessons. During this workshop, educators will learn about new concepts for game designs and handicraft lessons and how to apply programming classes only on mobile phones

Figure 1. Program creative designs and patterns on smartphones

Interactive Board Games as a Tool for Constructivist Pedagogy

Kate Delaney *hello@makecreateinnovate.ie*
Make Create Innovate, Ireland

Siobhán Clancy *hello@makecreateinnovate.ie*
Make Create Innovate, Ireland

Board game design lends itself to constructionism through the ‘gamification’ of learning which incentivises self-directed education and supports a variety of learning styles. It also facilitates the development of social skills through collaboration. In this hands-on workshop, the facilitator will role model a maker-based approach to the practice of constructivist pedagogy in a range of learning environments through the design and making of interactive board games. In line with the constructivist pedagogies promoted by The Digital Schools Strategy (2015-2020), we integrate Science, Technology, Engineering, Arts and Maths or S.T.E.A.M. subjects in applied learning opportunities for the benefit of all students, both academically and socially. STEAM-based approaches to learning have been identified in the Digital Learning Framework as one of the most effective ways to fulfil the strategy aims which include 21st Century Skills for all. In this workshop, we will demonstrate how by including the Arts in S.T.E.M, it is possible to facilitate the development of what Dr. Rueben Puentedura classifies as ‘high order’ learning (HECA, p94) game playing, development and design fulfils the Five Creative Habits of Mind espoused by Eric Booth which support the development of a 21st Century skill set of which creative problem-solving is key.

Figure 1. Kate Delaney facilitates an Interactive Board Game Workshop with educators (style: Figure caption)

This workshop will give participants an insight into the game development process and the critical thinking skills required to devise the rules and ethics of play in age appropriate ways. Participants can expect to develop skills in Design Thinking, conductivity, model making and creative technology usage by working in groups to produce an original interactive board game by the end of the workshop.

NetsBlox - Make Your Own Data Service Workshop

Brian Broll, *brian.broll@vanderbilt.edu;* **Corey Brady,** *corey.brady@vanderbilt.edu;* **Akos Ledeczi,** *akos.ledeczi@vanderbilt.edu* Computational Thinking and Learning Initiative, Vanderbilt University, Nashville, TN, USA.

NetsBlox is a web-based, open source, block-based educational programming environment based on Snap! and designed to offer an accessible introduction to advanced CS concepts, such as distributed computing and computer networking. Of particular interest to this workshop is NetsBlox's Remote Procedure Calls (RPC) that allow students' programs to access online data sources and services (e.g., Google Maps, weather and climate data, Twitter feeds, and more). An RPC is presented just like a custom block (or function) that happens to run remotely on the NetsBlox server. Related RPCs are grouped into services. For instance, the Weather service provides RPCs to query current temperature, humidity, and wind conditions, at locations by latitude/longitude. More than simply providing access to web APIs, a NetsBlox RPC can simplify both the query and the returned results, and can add useful helper functions. For example, the Google Maps service provides RPCs for coordinate transformation from image x and y coordinates to latitude and longitude and back. NetsBlox RPCs can also be created to provide services where publicly available APIs do not exist. For example, NetsBlox supports cloud variables and a charting service providing an interface to gnuplot as well.

Figure 1. The temperature RPC of the Weather service. Selecting a service from the first pull-down menu populates the second with the available RPCs. Selecting an RPC reconfigures the call block to present slots for the required input arguments and provide their names as a hint.

While our team keeps adding services on a regular basis and the service API is public and simple for developers, there has not been an easy way for *teachers* to add their own services accessing their personal, instructionally-relevant data sources. This workshop will introduce a new feature called *ServiceCreation* that lets users create their own data service within the NetsBlox environment. All the user has to do is import a data table (in csv format) and call the RPC *createServiceFromTable*. Upon calling this RPC, a new service is created with a set of automatically defined RPCs: one for returning each column of the table and one for returning a particular cell of each column indexed by the first column. ServiceCreation users can also create their own RPCs by providing corresponding scripts, implemented in the NetsBlox block language itself. This code is automatically compiled into JavaScript and deployed on the server. Finally, users can also provide documentation that will show up on the context menu help item.

This self-bootstrapping method of creating user-defined extensions to the NetsBlox environment using the same block-based language, is a powerful paradigm. It enables learners to make connections between personally-relevant data on the one hand and larger public datasets on the other. Researchers and teachers who want to create engaging STEAM activities in a local context in K12 classrooms should be interested in how to create their own data services.

After a brief hands-on introduction to NetsBlox and RPCs, workshop participants will create their own data services. They can use their own csv file with data they consider important or they can utilize one of several sample files we will provide, representing data locally relevant to the conference venue. Since user-created services show up immediately under the Services/ Community/ UserName menu, participants will be able to explore each other's services during the workshop (and afterwards).

Making Drawbots: A Junk Art and Raspberry Pi/Python workshop using the Bridge21 Learning Model

Helen O'Kelly, *hokelly@stratfordcollege.ie*
Stratford College, Rathgar, Dublin 6, Ireland.

Computer Science is now increasingly perceived as being relevant in primary and second level schools. As a subject, it has the potential to help develop programming, computational thinking and design thinking skills. Computer Science is being introduced or reinvented as a subject for upper second level education in countries throughout the developed world.

In this workshop participants will:

1. Design a drawbot from recycled objects.
2. Use jump leads to connect a propeller on the drawbot to the Explorer Hat Pro breadboard.
3. Use the IDLE editor to write Python code (supplied) to start the propeller motor.
4. Watch the drawbot "draw" once the propeller is activated by running the Python code.

Bridge21 is a constructivist model of 21st century teaching and learning and was designed by Trinity College Dublin. It is a candidate pedagogical model for the Computer Science classroom because it advocates a collaborative, project-based, hands-on approach to teaching and learning.

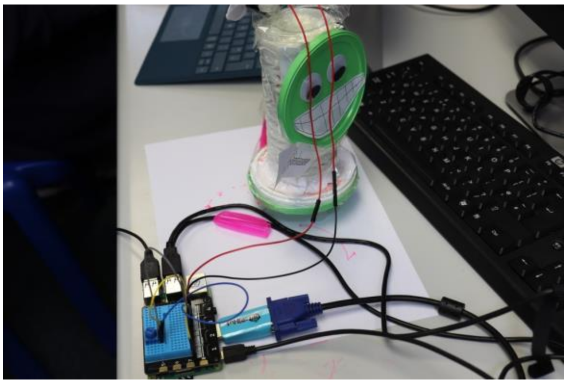


Figure 1. Drawbot connected to a Raspberry Pi using Explorer Hat Pro outputs a basic drawing to paper

The key ideas include a demonstration of the Bridge21 Learning and Activity model in framing this workshop, the use of Art with Technology in Computer Science and Art classrooms and the application of PRIMM as a structured pedagogical approach to teaching programming in a second level classroom. Participants will gain a first-hand experience of a constructivist Computer Science class from a student perspective underpinned by the Bridge21 Learning and Activity model, while also gaining technological pedagogical content knowledge.

Target audience: primary and second level teachers, particularly Art and Computer Science teachers; Computer Science Education researchers

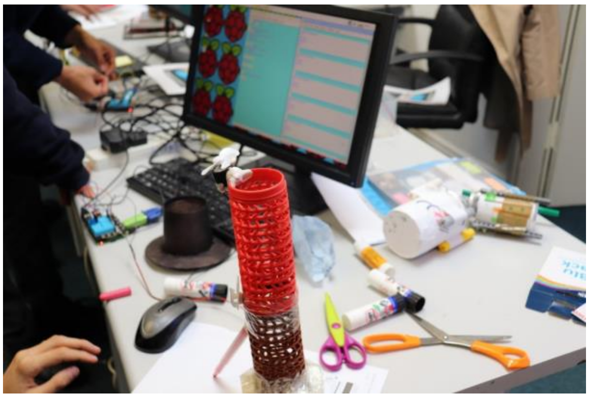


Figure 2. Drawbot with motor propeller connected to a Raspberry Pi outputs a basic drawing to paper

Objects-to-think-with in a climate emergency

Claire Garside, *edcga@leeds.ac.uk*
School of Education, University of Leeds, Leeds, England

Papert (1980) created Logo where the computer served as an ‘object-to-think-with’. The basis of this workshop is to explore constructionism in parallel with computational thinking, specifically when children interact with the internet of things. The physical computing device becomes a contemporary ‘object-to-think-with’ and a platform for children to further understand connected technologies with data transfer, linked to current global environmental issues and to control their own learning.

Putting tools, in this example low-cost sensors, into the hands of children to understand the world around them has seen a rise of problem-finding in project-based education programmes. The emphasis of this activity is on building and programming projects to link local problems to the UNs’ Sustainable Development Goals, thus giving purpose and real world relevance to children's learning.

Figure 1. Visualising air quality data using LED colour palettes

This is a hands-on workshop introducing the concept and maker education practices through the ‘Internet of Curious Things’. Educators will actively participate by building an air quality monitoring device and reflect on how digital making can leverage computational thinking as we explore multidisciplinary approaches including citizen science, geography and environmental literacies.

During the session participants will be challenged to program a microcontroller board with smart sensors, collect their own air quality data and visualise in a playful and imaginative way using papercraft. Discussion will focus on bringing together technology, art and science to empower the young generation and considering the value of constructionism to underpin social action in the curriculum.

Programming with Emil in Year 3

Andrej Blaho, *andrej.blaho@gmail.sk*
Department of Applied Informatics, Comenius University, Bratislava, Slovakia

Ivan Kalaš, *ivan.kalas@fmph.uniba.sk*
Department of Education, Comenius University, Bratislava, Slovakia

Drawing on our previous experience in developing programming environments for schools and our recent exceptional opportunity to work in the ScratchMaths team, we concluded that securing *sustainable computing education for every learner* requires paying much closer attention to lower primary years and providing teachers with systematic support and complex interventions. Thus, in 2017 we launched a long-term project Computing with Emil, with an ambition to provide all primary years with new software environments, teachers' and pupils' materials and corresponding PD sessions, based on design research strategy. We run all developments in close collaboration with our *design schools*. We strive to facilitate holistic learning process for computing education for all, borrowing from modern mathematics education, genetic epistemology and social constructivism.

Main design principles of our pedagogy include: pupils always work in pairs, discuss and learn together; they frequently meet in common whole group discussions carefully scaffolded by their teacher – a generalist primary teacher; software environments do not provide learners with any feedback – pupils themselves have to conclude (and justify in their discussions) whether their strategy or solution is correct or not; however, many problems and tasks have no solution or several solutions, or even 'unclear' solutions to be negotiated in discussions; software environment itself is useless without accompanying paper worksheets and vice versa, thus providing multiple representations of the problems and requiring pupils to transfer their thinking between those; 'wrong' solutions are considered important catalyst for learning; we strive to drive the learning process on intrinsic motivation of the learners. Intervention for Year 3 is currently being used in around 150 schools and requires 15 to 20 lessons.

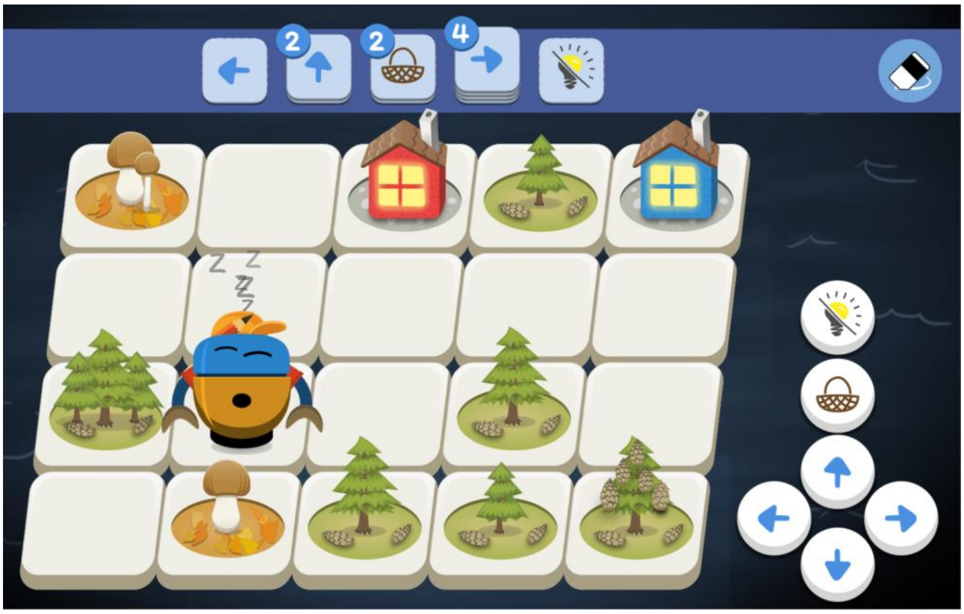


Figure 1. A learner has built a plan (see it in the panel above the stage) for Emil to pick some mushrooms and turn the light off in the blue house. When woken up, Emil will run the plan.

In this hands-on workshop participants will solve and discuss some of the tasks of the three worlds of Emil for Year 3 (i.e. for pupils around 8 to 9 years of age). They will experience our approach when we try to identify basic computational concepts and break them into natural progressions of what we call ‘pre-concepts’, so that no explicit lecturing is needed at all. Instead, each group of tasks brings in (usually one) new option or functionality or constraint – to be discovered by the pupils, adopted, used and validated in hands-on activities and continuous discussions.

In the workshop we will also discuss whether our method gives space to constructionist learning, in spite of the fact that the learning process is organized in gradations of (mostly predefined) tasks.

Our target group are educators, researchers or practitioners interested in supporting the development of early computational thinking (here with the focus on programming) at the lower level of primary schools.

Programming with Emil in Year 4

Andrej Blaho, *andrej.blaho@gmail.sk*
Department of Applied Informatics, Comenius University, Bratislava, Slovakia

Ivan Kalaš, *ivan.kalas@fmph.uniba.sk*
Department of Education, Comenius University, Bratislava, Slovakia

This will be a **sequel** to our *Programming with Emil in Year 3* workshop (see its Abstract for the background and motivation). With **Emil for Year 4** intervention we provide schools with another step of systematic support to develop computational thinking (here with the focus on programming) for every learner. The intervention for Year 4 requires 15 to 20 lessons.

Applying the same design principles, we decided to bridge Emil for Year 3 and Scratch, which we consider productive descendant of Logo philosophy and legacy, and very good follow-up of Emil.

In this world pupils progressively undertake key transition from ‘absolute frame’ navigation to relative one in the turtle geometry style – within a virtual programming environment (in contrast to relative navigation in the physical world of programmable toys like Blue-Bots etc. where body syntonicity makes such navigation rather natural).

In this new world, some basic commands have parameters (like filling a closed area with a colour, setting pen colour or setting pen width). We also take further steps in the thread of developing the concept of *procedures* (which has already been initiated in the third world of Emil for Year 3).

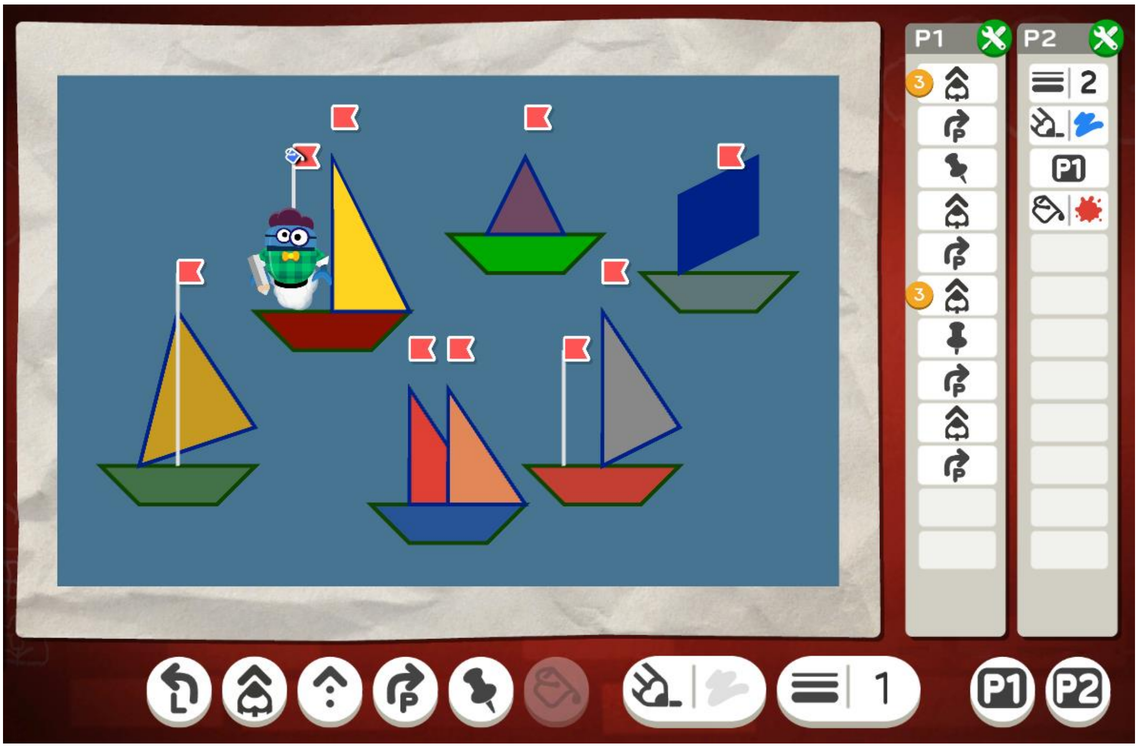


Figure 1. In Year 4 Emil gradually provides the learners with more and more simplified Logo-style commands for basic ‘turtle drawings’. In this task pupils work also with two ‘memories’ ***P1*** *and* ***P2*** *– a pre-concept to procedures. For our learners, transformation to Scratch will then be smooth and straightforward.*

In this hands-on workshop participants will solve and discuss some of the tasks of Emil for Year 4 (i.e. for pupils around 9 to 10 years of age). They will experience our approach when we try to identify basic computational concepts and break them into natural progressions of what we call ‘pre-concepts’, so that no explicit lecturing is needed in the class at all. Instead, each group of tasks brings in (usually one) new option or functionality or constraint – to be discovered by the pupils, adopted, used and validated in continuous collaboration and discussions.

In the workshop we will also discuss whether our method gives space to constructionist learning, in spite of the fact that the learning process is organized in gradations of (mostly predefined) tasks. In Emil for Year 4 however we take another step towards open programming language.

Our target group are educators, researchers or practitioners interested in our approach to supporting the development of early computational thinking (here with the focus on programming) at the lower level of primary schools.

Socio-syntonicity

Richard Millwood, richard.millwood@tcd.ie
School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland

Cynthia Solomon, cynthia@media.mit.edu
Cynthia Solomon Consulting, Boston, USA

Margaret Low, m.j.low@warwick.ac.uk
The University of Warwick, Warwick, UK

Artemis Papert, artemis@turtleart.org
Independent artist, Canada

Margaret Minsky, marg@media.mit.edu
New York University Shanghai, China

This workshop was intended to show the way in which some solutions to problems benefit from multiple sprites, which can be enacted by learners taking the role of each sprite and collaborating through their actions to debug their solution. The example to stimulate participants own ideas is derived from the challenge to 'fill in' a petal shape.

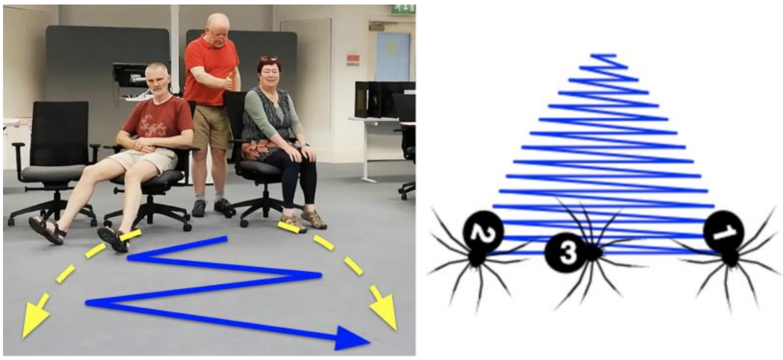


Figure 1. The three sprites filling in a petal shape in real life and in Snap!

Participants perform and program this example, and then are invited to invent their own problems which demand several individuals/sprites acting together.

In Mindstorms, Papert identified two key terms applied to the solutions made to problems through the creation and debugging of programs in Logo:

- **body-syntonic** "related to children's sense and knowledge about their own bodies " and
- **ego-syntonic** "coherent with children's sense of themselves as people with intentions, goals, desires, likes, and dislikes "

This workshop proposes a further term:

- **socio-syntonic** "connected to the expanded competence derived from several individuals acting together".

The workshop illustrates the potential for multiple sprites and concurrency to solve simple problems in ways which match learners' interest in acting together.

The HumaNature Project: Using Critical Constructionist Design to Re-think Humans' Relationship with Nature

Isabel Correa, *mic2130@tc.columbia.edu*
Dept of Math, Science, and Tech; Teachers College, Columbia University; New York, NY, USA

Nathan Holbert, *holbert@tc.columbia.edu*
Dept of Math, Science, and Tech; Teachers College, Columbia University; New York, NY, USA

New understandings of constructionism can help navigate these times of political, social, and environmental instability. Critical constructionist design (Holbert, Dando, & Correa, 2020) is a framework that draws from critical design and constructionism to tinker with humanity's pressing challenges by inviting learners to: 1) connect with the past, 2) reflect on the present, and 3) envision and design alternative futures.

In this workshop, participants will engage in critical constructionist design to critically reflect on the emerging climate crisis and its social, political, and ethical implications. To this end, they will construct future-thinking artifacts–as cautionary dystopian tales or utopian alternative futures–to represent new ways for humans to relate with the natural environment.

Following the three components of the framework (see figure 1), participants will engage in three consecutive activities. First, participants will connect with their past using natural clay to represent the different ways in which their ancestors, family, and cultural community relate to the natural environment. Second, participants will reflect on the present by questioning and re-shaping their clay work in light of their personal and local experiences with the environmental crisis. Third, participants will project alternative futures by designing a future-thinking artifact using colored modeling clay, conductive clay, and electronic pieces. The goal of these artifacts will be to challenge existing values and beliefs by presenting unexpected ways humans relate with the natural world to re-think: What roles will humans play in the ecosystem? How may we co-exist with other species? What and how humans will eat? Where may we live in the future? What rituals and activities we will celebrate? How might we dress and move around?

By engaging in critical constructionist design, participants will gain a renewed perspective of constructionism that aims to facilitate critical but playful reflection about the challenges of our time.

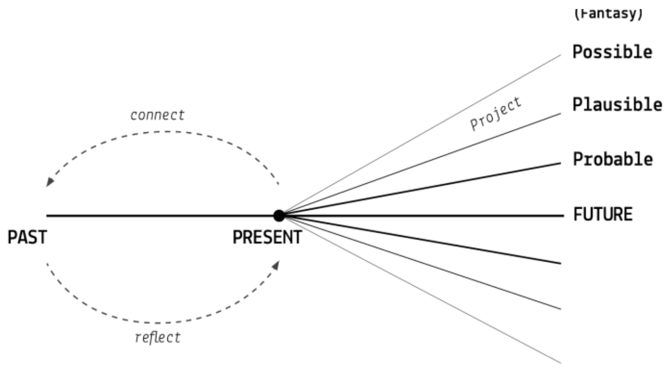


Figure 1. In a critical constructionist design practice learners engage in a cycle of connecting back to personal and communal histories and reflecting on present and local systems. They then use this connect and reflect cycle to project and create possible futures.

References

Holbert, N., Dando, M., & Correa, I. (2020). Afrofuturism as critical constructionist design: Building futures from the past and present. *Learning, Media and Technology.*

Turtlestitch

Richard Millwood, richard.millwood@tcd.ie
School of Computer Science and Statistics,Trinity College Dublin, Dublin, Ireland

Andrea Mayr-Stalder, mayr@sil.at
TurtleStitch, Austria

Mags Amond, mags.amond@gmail.com
Trinity College Dublin, Dublin, Ireland

John Hegarty, jhegarty@clongowes.net
Clongowes School, Ireland

Margaret Low, m.j.low@warwick.ac.uk
The University of Warwick, Warwick, UK

This workshop was intended to show the way in which the Turtlestitch environment for turtle geometry embodies a constructionist approach. Participants create their own design and stitch it on material using a consumer embroidery machine. The Turtlestitch programming environment is based on Snap!, but shows the output generated by movement of the turtle as stitches on screen. Feedback on the density of stitches and ‘jump stitches’ generated by the equivalent of ‘pen up’ commands, combined with an interface to zoom in on detail, allows the learner to anticipate problems that the embroidery machine may have in realising a design.

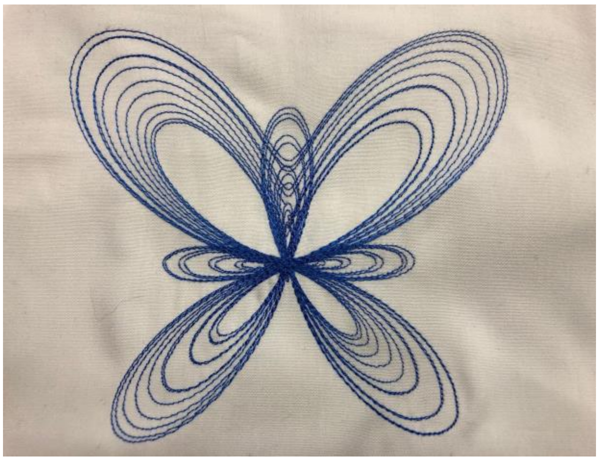


Figure 1. The outcome of programming and embroidering the Butterfly Curve (transcendental). Photo– Richard Millwood

In this way, embroidery in Turtlestitch exemplifies the constructionist ‘object to think with’ and introduces interesting challenges, extending turtle geometry to offer a tactile, fulfilling and practical outcome with new considerations for designing solutions compared with simple drawing.

The workshop provides new insights into the role of motivation and fulfilment for learners through the collision of craft thinking and programming, generated by the delightful anticipation of a product that can be worn or used to decorate.

Posters

A Foucauldian perspective on Computational Thinking and Initial Teacher Education

Jim King, *M.KING31@nuigalway.ie*
School of Education, NUI Galway, Ireland

Cornelia Connolly, *cornelia.connolly@nuigalway.ie*
School of Education, NUI Galway, Ireland

This presentation addresses a question that is particularly relevant in the current educational climate; *that is:* why is computational thinking (CT) understood as a constructionist orientation to formulating problems (as conversions of some input to an output and looking for algorithms to perform the conversion)—being regarded as an educational priority and, in particular, an essential element of any initial teacher education programme? (Lockwood and Mooney, 2017; Yadav *et al.*, 2017). The question—or problem—is being approached from a Foucauldian perspective (1984): that is, to distinguish CT—as a *field* of thought—from the ideas, and from the domain of attitudes, beliefs and customs, that underline and determine CT practices and behaviour. When looked at in this way, CT as a practice can, to use a Foucauldian term, be 'problematised' (1984, p. 117); *i.e.,* it can be considered as an *object* of thought, and reflected upon as 'a problem'. It is the act of intellectually stepping back in this way, and detaching ourselves from computational thinking as an idea—or set of ideas—that allows us then to establish CT as an object of thought, dispensing with prior theory, presuppositions and possibilities, or with any hints of solutions—*i.e.* a knowledge-oriented research approach. To question CT as an 'object of thought' in this way—along the parameters of meaning, conditions, and goals—is at the same time, according to Foucault, freedom in relation to what one does; and treating the object of thought as a problem, involving the development of a set of conditions within which possible responses—in this case, the constellation of challenges surrounding the introduction of CT as an integral part of any initial teacher education programme—could be proposed; but not as *a* solution; or as *a* response. In other words, an existential-ethical orientation toward the subject-matter in question (Foucault, 1984; Peters, 2007; Simons and Masschelein, 2014).

There is no explicit agreement on a definition of 'computational thinking', and it is being interpreted—and integrated (in learning environments)—differently (Csizmadia, Standl, and Waite, 2019); but there is consensus and agreement that it's located—cognitively and intellectually—at the intersection of computation, disciplinary knowledge, and algorithms (National Research Council [NRC], 2011, p. 5). Regardless of the uncertainties, it has been contended that CT represents a universally applicable attitude and skill set everyone, not just computer scientists, should be eager to learn and ready to use (Wing, 2006; Kong and Abelson, 2019). But a survey of literature concerning the practical and applicable aspects of CT reveals that many of the examples offered are directed at scientists, engineers, and professionals in nontechnical fields, such as archaeology and law (NRC, 2011; Yadav *et al.,* 2017, p. 56). Considered from this 'professional' perspective, it seems that CT would be relevant to individuals across the spectrum—technical, or otherwise—with graduate and postgraduate educations only. Yet, as has been noted above, there also have been ongoing efforts on an international scale focused on exposing learners (at all levels, and in all different fields) to CT and practices to aid them developing an understanding of how computing influences the world (Denning, 2009); of human behaviour (Wing, 2006); and even of how CT can improve people's lives (Wang, 2015; The Royal Society, 2017).

Therefore, it is our contention that a certain number of factors have made CT practice uncertain in the Foucauldian sense; and from some perspectives, unfamiliar; provoking some difficulties concerning CT and its relevance—especially outside of what are traditionally thought of as STEM subject areas (Figure 1). Moreover, it is the case that these factors result from wider social, economic or political processes that can, for the purposes of this presentation be referred to as *'The Politics'* (Foucault, 1984, p.117), and are located—from the point of view of an observer, or researcher—in an amalgamation of (pedagogical) contentions, ideas and memes about the relevance of computer science education (and constructionism in education) more generally (Denning, 2009). The amalgamation of contentions outlined above is referred to as *'The Forum'* (Foucault, 1983; Marshall, 2007, p. 23); contentions that are—and should be—open to interrogation along the parameters of meaning, conditions and goals through designing questions aimed at probing areas—or representatives—of the overlapping social, economic and political spheres of activity.

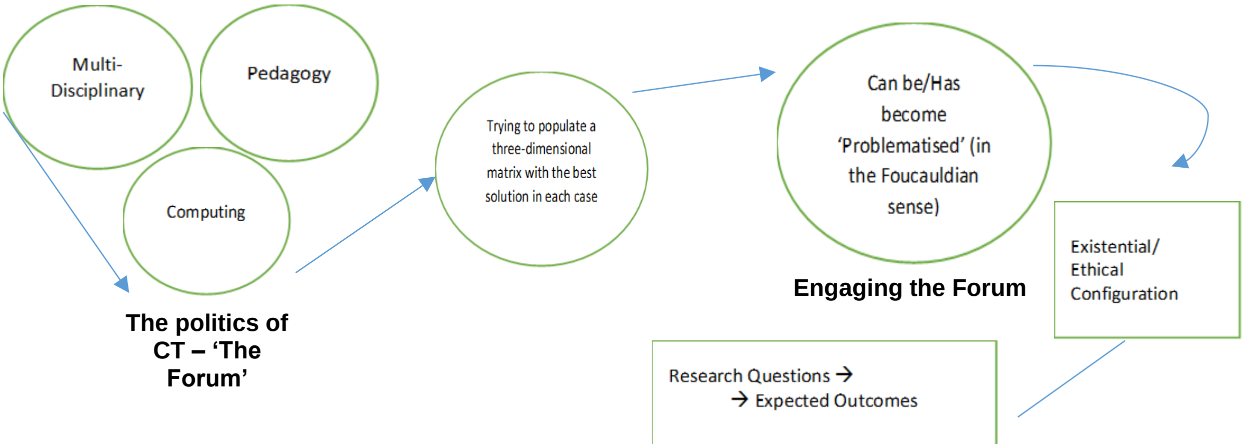


Figure 1. Overview of the CT Education Forum

References

Csizmadia, A., Standl, B., Waite, J. (2019). Integrating the Constructionist Learning Theory with Computational Thinking Classroom Activities. In *Informatics in Education,*18(1), 41–67

Denning, P.J. (2009). The Profession of IT Beyond Computational Thinking. In *Communications of the ACM*, 52(6), 28-30.

Foucault, M. (1984). Polemics, politics and problematisations: An Interview with Michel Foucault. In *Michel Foucault: Ethics, subjectivity, truth.* (pp. 113-119). Edited by P. Rabinow (1997). New York: The New Press.

Kong, S.C., Abelson, H. (2019). *Computational Thinking Education.* Singapore: Springer Open.

Lockwood, J., Mooney, A. (2017). *Computational Thinking in Education: Where does it fit? A systematic literary review*. Maynooth University, Rep. of Ireland: Dept. of Computer Science,

Marshall, J. D. (2007). Chapter 2: Michel Foucault: Educational Research as Problematisation. In *Counterpoints,* 292**,** 15-28.

National Research Council (2011). *Report of a Workshop on the Pedagogical Aspects of Computational Thinking.* Washington, D.C.: National Academies Press.

Peters, M.A. (2007). Chapter 12: Educational Research: 'Games of Truth' and the Ethics of Subjectivity. In *Counterpoints,* 292**,** 15-28.

Royal Society. (2017). *After the reboot: computing education in UK schools.* London: The Royal Society, (6–9 Carlton Terrace, SW1Y 5AG.)

Simons, M., Masschelein, J. (2014). "This Truth That I Say to You, Well, You See It in Myself": On Research As Education | As Philosophy | As a Way of Life. In *A Companion to Research in Education.* (Chapter 11). Edited by A. D. Reid, E. P. Hart, and M. A. Peters, Dordrecht, The Netherlands: Springer Science + Business Media.

Wang, P. (2015). *From Computation to Computational Thinking.* Boca Raton, FL: CRC Press.

Wing, J.M. (2006). Computational thinking. In *Communications of the ACM*, 49(3), 33–35.

Yadav, A., Stephenson, C., Hong, H. (2017). Computational Thinking for Teacher Education. In *Communications of the ACM,* 60(4), 55-62.

A Constructionistic Approach to Mathematical Concepts with Hand-Held Technology

Karl Josef FUCHS,
KarlJosef.FUCHS@sbg.ac.at
University Salzburg, Dept. of Mathematics and Computer Science Education, Salzburg, Austria

Simon PLANGG,
simon.plangg@phsalzburg.at
Salzburg University of Education Stefan Zweig, Salzburg, Austria

Aims and policy

The main goal of the research is the development and evaluation of materials for learning mathematical concepts and digital skills by students of the lower secondary level in Austrian schools. Collaborating with prospective math teachers, several materials have been designed at the University. The evaluation of those materials takes place in one class of 24 twelve-year-old students from a secondary school in Salzburg. The students are using the programmable wheeled robot TI-Innovator Rover and the Hand-Held TI-Nspire CX CAS (Fig. 1). Drawing on the method of design-based research (Cobb et al. 2003), the project aims to determine what and how those twelve-year-olds learn working with those materials in classes.

Materials, structure of class and methods

So far materials for the following topics have been developed: constructing triangles, triangle centres, quadrangles and regular polygons, proportions and similarity, operating with integers and functional dependencies. Figure 1 refers to one task within the topic of constructing triangles based on two side lengths and the included angle. According to Seymour Papert's turtle, the path of the Rover can be sketched by fixing a pen on the robot.

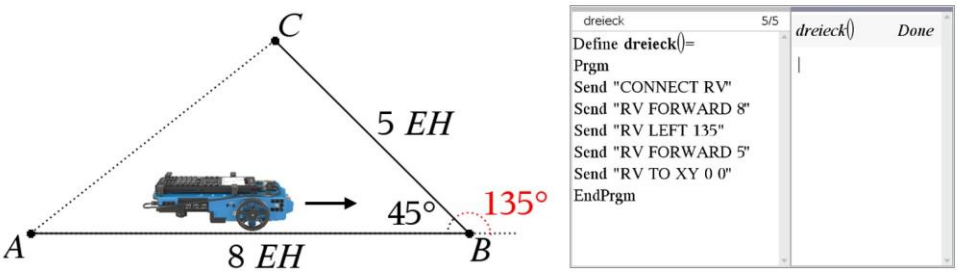


Figure 1. Possible outline (left) and code (right) for constructing a triangle with the TI-Innovator Rover

The evaluation of those materials started in spring 2020, namely constructing rectangles and regular polygons as a first introduction to the technology mentioned (Tab. 1), as well as the topic constructing triangles (Fig. 1) in another session. The structure of those sessions with the Rover are based on aspects of constructionistic theories, such as relations to the student's reality, personalization, the use of expressive digital media, modelling, abstraction, reflection and collaboration (Noss & Clayson 2015). Accordingly, the students collaborate in groups of two, sharing one robot working with problem- and activity-oriented materials at their own pace. Those sessions include one and a half to two hours working time. The introduction took place at the library of the school itself. Step by step instructions helped the students to get familiarised with the technology using tablets with a presentation. As the students proceeded, they solved the posed or chosen problems (Tab. 1) on their own, two teachers helping them when needed.

Table 1. *Posed problems within the introduction session using the TI-Innovator Rover*

Problems	
Rectangle	Let the Rover run a rectangle!
Triangle	Let the Rover run an equilateral triangle with a side length of 4 units!
Hexagon	Let the Rover run a hexagon!
Polygon	Let the Rover run another polygon of your choice!

In order to assess the students learning, several survey instruments are applied. The audio recordings for each group of students, combined with the created programs and the documented solutions, sketches and drawings as well as the students answers and essays to questions of contemplation, such as “How did you experience the today’s class?”, “What did you like the most / the least?”, “What did you learn today?”, “What difficulties did you experience?”, “What would you change for the next time?” are being analysed using qualitative methods.

Preliminary results and outlook

The first results from the introduction session, based on the students’ answers to the questions of contemplation, are provided. The handwritten answers and essays were analysed according to the method of thematic analysis (Brown & Clarke 2006) using MAXQDA (Fig. 2).

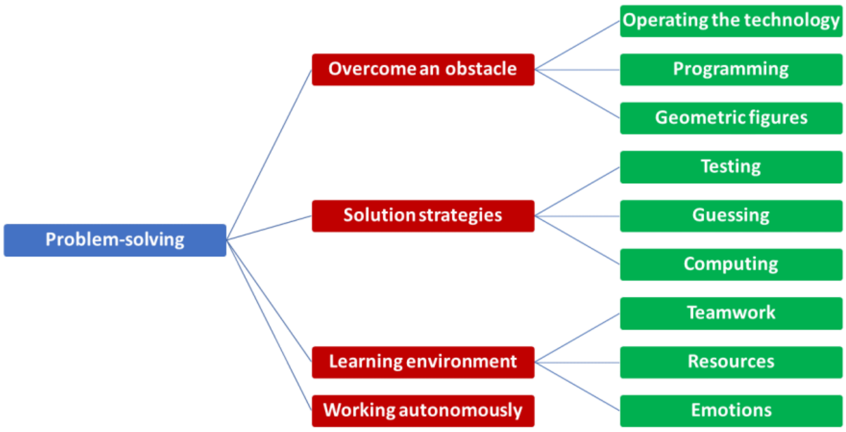


Figure 2. Developed thematic map based on the students’ essays after the first session

One of the main results is, that the students experienced the introduction, as intended, as a problem-oriented class while constructing geometric figures on their own. They used several methods to solve the posed and their own problems, especially while trying to determine the rotation angles of the Rover. Most of them appreciated the possibility to work autonomously at their own pace. The emotions were truly positive, stating that it was great fun, cool, interesting, exciting, funny and the desire to continue the classes with the Rover. Critical notes include that they would have needed more posters to sketch on and more working time.

Based on the results of the introduction session the challenge is to determine and to realize an appropriate scope of support coming from the materials and the teachers assuring that the students are able to solve the upcoming problems on their own in the given time.

References

Braun, V. & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101.

Cobb, P. et al. (2003). Design Experiments in Educational Research. *Educational Researcher*, 32(1), 9–13.

Noss, R. & Clayson, J. (2015). Reconstructing Constructionism. *Constructivist Foundations*, 10(3), 285–288.

A Thermometer for Kindergarten Data Inquiry

Ryan Cain, *ryancain@weber.edu*
Department of Teacher Education, Weber State University, Ogden, United States of America

Victor R. Lee, *vrlee@stanford.edu*
Graduate School of Education, Stanford University, Stanford, United States of America

As schools in the US work to engage K-12 students in more ambitious science instruction, children in kindergarten are expected to analyze and interpret weather data as part of NGSS performance expectation *K-ESS2-1* (NGSS Lead States, 2013). While air temperature is a salient weather variable for measuring, previous research has shown that young children have difficulty using thermometers (Havu-Nuutinen, 2007; Kampeza, Vellopoulou, Fragkiadaki, & Ravanis, 2016). Furthermore, conceptualizing heat and temperature can be fraught with misconceptions for children and even experts (Lewis & Linn, 2003).

This exploratory study examines the utility of a specialized thermometer while also eliciting participants' conceptions of air temperature. To pursue this line of inquiry, we pose the following research questions: RQ1 – how do the participating kindergarteners conceptualize air temperature, RQ2 – how they interpret a specialized thermometer designed and built for the study, and RQ 3 – how they make sense of graphic representations of temperature?

The specialized thermometer, the Early Childhood Thermometer (ECT), includes designed features to make it more readable by young children (e.g., discrete countable boxes, 10°F increment scale). The ECT is built with both open source DIY hardware and Circuit Python code (Cain, 2018). Using DIY electronics for classroom inquiry is consistent with previous work like the InSPECT Project (Hardy, Dixon, & Hsi, 2019), where high school students engaged in data science practices as part of a biology investigation. However, the current study investigates participants at the opposite end of the K-12 age range. The ECT color scale and box square increments are designed to be similar to typical kindergarten classroom manipulatives, such as building blocks.

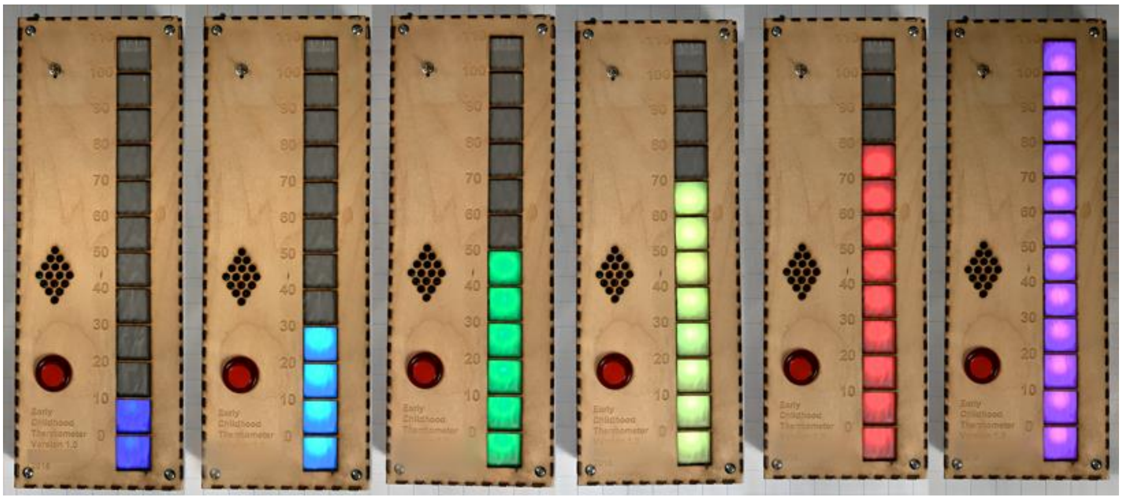


Figure 1. The Early Childhood Thermometer

Six participants were recruited from a public school in the intermountain west region of the US. Data for the study included 160 pages of transcripts and 23 participant made illustrations from a series of semi-structured interviews that engaged the youth in the practices of analyzing and interpreting temperature data. For example, children compared temperature measurements from shaded and full sun locations. Using an interpretive qualitative study design (Merriam & Grenier, 2019), we analyzed interview transcripts and children's illustrations using iterative cycles of process coding, thematic coding, and analytic memo writing (Saldana, 2009).

Analyses revealed a range of ways that the children: represented temperature, read thermometers, and interpreted temperature graphs. These included expressing temperature with colors, symbols, and effect the human body. Of particular interest in the children's drawings, 5 out of 6 included the ECT by depicting the discrete scale. While all of the children were able to make temperature measurements with the ECT, only four were able to demonstrate conceptual understandings of the temperature scale. In addition, all five of the children who were asked to pick a thermometer out to try chose the ECT.

The ECT could serve as a playful tool for young children to make sense of environmental data. The current study observed how some of the children could relate their existing conceptions of temperature to the scale of the ECT after only using it on two occasions. It is possible that more extended exposure could be more effective. Although the current study is early in its development, we aspire for the ECT to become an "object to think with (OTTW)" (Papert, 1980). Papert (1980) described OTTWs as existing at the "intersection of cultural presence, embedded knowledge, and personal identification" p11. The current study designed the ECT's block-like scale (kindergarten cultural presence) to engage the youth in sense-making of the complex topic of temperature change (embedded knowledge) by providing personal experiences with temperature differences to reference in the interviews (personal identification). This work demonstrates that the ECT is a promising tool worthy of further investigation. Future work will more thoroughly assess the tool's potential as an OTTW for engaging early elementary students in the practices of analyzing and interpreting temperature data.

References

Cain, R. (2018). Early Childhood Thermometer Code. Retrieved from https://github.com/ryespy/Early_Childhood_Thermometer/blob/master/main.py

Hardy, L., Dixon, C., & Hsi, S. (2019). From data collectors to data producers: shifting students' relationship to data. *Journal of the Learning Sciences*, *29*(1), 104–126. https://doi.org/10.1080/10508406.2019.1678164

Havu-Nuutinen, S. K. S. (2007). Young Children's Conceptions of Temperature and Thermometer. *The International Journal of Learning*, *14*(9), 93–101.

Kampeza, M., Vellopoulou, A., Fragkiadaki, G., & Ravanis, K. (2016). The expansion thermometer in preschoolers' thinking. *Journal of Baltic Science Education*, *15*(2), 185–193.

Lewis, E. L., & Linn, M. C. (2003). Heat energy and temperature concepts of adolescents, adults, and experts: Implications for curricular improvements. *Journal of Research in Science Teaching*, *40*(S1), S155–S175. https://doi.org/10.1002/tea.10091

Merriam, S. B., & Grenier, R. S. (2019). Introduction to Qualitative Research. In *Qualitative Research in Practice: Examples for Discussion and Analysis* (pp. 3–18). https://doi.org/10.1057/9780230505056_1

NGSS Lead States. (2013). *Next Generation Science Standards: For States, by States.* https://doi.org/10.17226/18290

Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. *New Ideas in Psychology* (Vol. 1). New York: Basic Books, Inc. https://doi.org/10.1016/0732-118X(83)90034-X

Saldana, J. (2009). *The Coding Manual for Qualitative Researchers*. *The coding manual for qualitative researchers*. Thousand Oaks, CA: Sage Publications, Inc.

Augmented map of digital literacy as a conversational tool for teachers, parents, and specialists in educational policies

Yevgeny Patarakin, *epatarakin@mgpu.ru*
Institute of System Projects, Moscow City University, Moscow, Russia

Vasiliy Burov, *vburov@mgpu.ru*
Institute of System Projects, Moscow City University, Moscow, Russia

Implementation of new educational policies needs the help of teachers as the change agents. Agency, locus-of-control or a "can-do" mindset is developed through the practice of making and modifying objects. Modern learning environments are putting hardware and software in the hands of children to conduct scientific explorations, create sophisticated worlds and games, participate in different types of digital collaborative fabrication and remixing. However, the experience of teachers in participating in collaborative making is extremely limited. This study investigated remixing of educational practices and ontologies in the formation of teacher's agency. We created a separate semantic wiki – http://smwiki.mgpu.ru/w whose articles fall into one of five categories: Competences, Concepts, Programming languages, Constructional toys, Learning practices. Semantic MediaWiki adds semantic annotations that allow a wiki to function as a collaborative database, records of which can be represented in various ways (Figure 1).

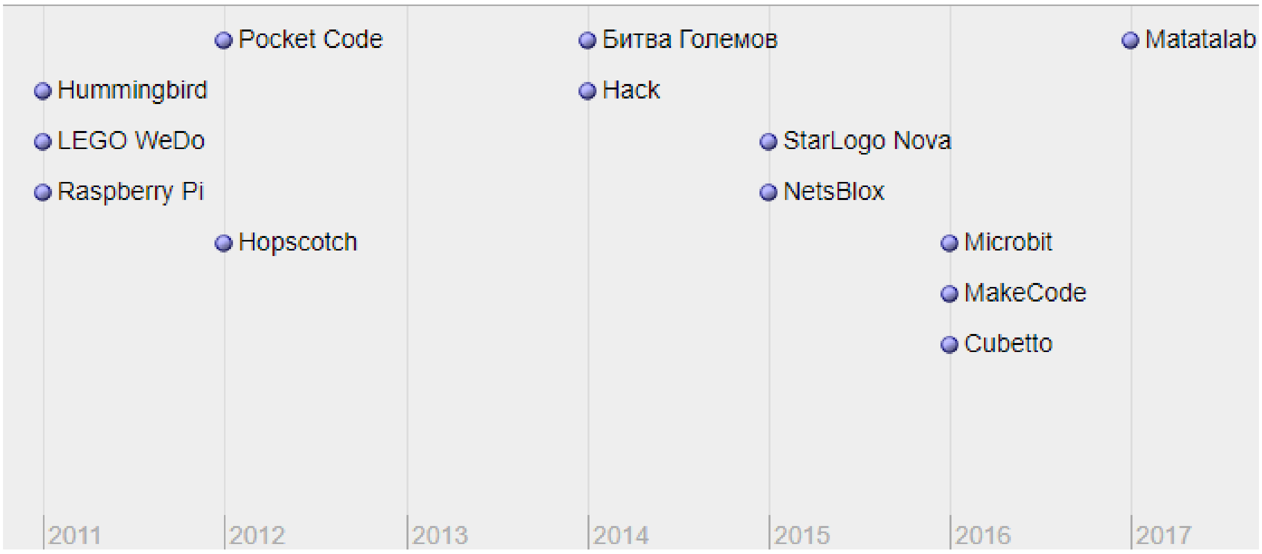


Figure 1. Time line smwiki.mgpu.ru/w

Modern learning environments are putting hardware and software in the hands of children to conduct scientific explorations, create sophisticated worlds and games, participate in different types of digital collaborative fabrication and remixing. Scratch, Pencil Code, Alice, and many others are low-threshold programming environments that make it easier for novices to develop games and interactive stories that can easily be shared with others. However, the experience of teachers in participating in collaborative making is extremely limited. In 2018, Moscow City University and the National Society for Technology in Education joined their efforts to develop a Russian network to deal with new learning activities shaping the 21 century skills - especially digital literacy, with the focus on collaboration skills). The organizers decided to find and form learning activities that would shape students' ability and willingness to share results of their activities, to work as a team, and to distribute tasks. They chose Scratch as an environment that already had tools to arrange and monitor such activities. The project brought together Scratch hackathons and schoolchildren's Collaborative Challenge competitions A learning activity script is not just an abstract description; it always describes a Scratch project. So, when a learning activity script changes through discussion, amendments and finalization, the Scratch project shaping this script changes too. We introduce a category of wiki pages that we call Taxonomical Sandboxes, which have served for remixing of taxonomy of learning environments.

The first sandbox is devoted to different families of novice programming environments. We created taxonomical sandbox as a place where everyone can produce their own remix of existing taxonomy. Teachers were given a taxonomy written in a DOT graph description language. They we asked to redesign the taxonomy of initial learning environments by adding new learning environments to existing taxonomy. First sandbox is available for editing and remixing as a wiki page: http://letopisi.org/index.php/Taxonomic_sandbox_1

The second sandbox contains a taxonomy of environments sorted according to the problems they were designed to solve. Each novice programming environment appears in the taxonomy only once. However, many of the systems in the taxonomy have been built on the ideas of earlier systems. Our second sandbox is available for editing and remixing as a wiki page: http://letopisi.org/index.php/Taxonomic_sandbox_2

There is wide production of standards and assessments of digital skills in the last years. Among these standards and assessments are: ACRL Information Literacy Competency Standards, Assessment and Teaching of 21st Century Skills, International Society for Technology in Education Standards, Technological and Engineering Literacy assessment, Russian Federal State Educational Standards. Each of the above documents contains a list of competencies. We extracted all separate competencies from standards and assessments and saved each competency as a separate page. This allowed us to present digital competencies as an augmented map

After experimenting with different sandboxes, we created a separate semantic wiki – http://smwiki.mgpu.ru/w whose articles fall into one of five categories: Competences, Concepts, Programming languages, Constructional toys, Learning practices. Semantic MediaWiki adds semantic annotations that allow a wiki to function as a collaborative database. For articles of each category, special forms and schemes have been created. Schemes require mandatory fields with special properties. For example, for an article about each individual competency, an indication of the source field is mandatory. This field indicates the name of the document in which the requirement for this competency is fixed. For articles describing educational practices, there is required a field indicating the target competencies, the formation of which this educational practice is aimed at. For example, articles about programming languages and constructional toys are automatically compiled on a common timeline (Figure 1) by using the following code:

```
{{#ask: [[Category:Construction toys]] OR [[Category:Programming language]] [[Year of creation::+]] |?Year of creation  |sort=Year of creation |order=descending |format=timeline }}
```

Conversation between people with the help of the augmented map of digital literacy may help in negotiating among the individual personal meaning systems brought to bear by work groups to solve common problems. Thanks to the mandatory framework, the results of the work of specialists in various fields are combined in a common field of collaborative activity. We hope that the system can be used by various categories of participants - teachers, parents, and specialists in the field of educational policies. In the proposed system, each of them can build her/his own version of the taxonomy, taking the existing text or graph as a basis, ask the system a question by modifying existing question templates, supplement the system by adding a new article or editing an existing one, offer her/his own version of educational practice, while relying on a given framework of target competencies, necessary concepts and existing learning environments.

Code-The-Mime: A 3D, Logo-based charades game for computational thinking

Marianthi Grizioti, *mgriziot@ppp.uoa.gr*
Educational Technology Lab, Department of Educational Studies, National and Kapodistrian University of Athens, Greece

Chronis Kynigos, *kynigos@ppp.uoa.gr*
Educational Technology Lab, Department of Educational Studies, National and Kapodistrian University of Athens, Greece

The idea of algorithmic thinking as the key to a new computational literacy described by Papert (1980) about 40 ago, is being re-discussed lately under the term ‘Computational Thinking’ (Wing, 2008). Despite the large number of studies on the field, there is still lack of designs that enable students to generate meanings on complex computational practices, such as abstraction, encapsulation and analysis, in an accessible and meaningful context. Many approaches for computational thinking, reject text-based programming as ‘obsolete’ and difficult for students and tend to focus on closed coding tasks with visual programming languages and simplified representations. Even though this approach can be beneficial for introducing students to specific programming concepts, it also prevents them from exploring computational practices and developing an integrated computational way of thinking. Therefore, the challenge remains to make complex computational concepts and practices accessible to young and inexperienced students, without lowering the threshold too much that would exclude or conceal these concepts.

In this poster, we present Code-The-Mime, a constructionist game that aims to provide students with multi-layered and multi-affordance access to powerful computational ideas. It is a program-to-play (Weintrop & Wilensky, 2014) word-guessing game, inspired by the game of charades, and it consists of a) a Logo microworld that creates a 3D, dynamically manipulated human model on the scene of MaLT2 online environment (Kynigos & Grizioti, 2018) and b) a deck of printed cards with names of sports (ski, basket, tennis, box etc). One group of players draws a card, and they have to describe the written sport to their teammates by modifying appropriately the 3D human model of the microworld (Figure 1). They first re-program, modify or extend the Logo code of the model and then they give the modified file to their teammates who should guess correctly the represented sport by using all MaLT2 affordances, namely Logo programming, dynamic manipulation of Logo parameters and 3D camera perusal.

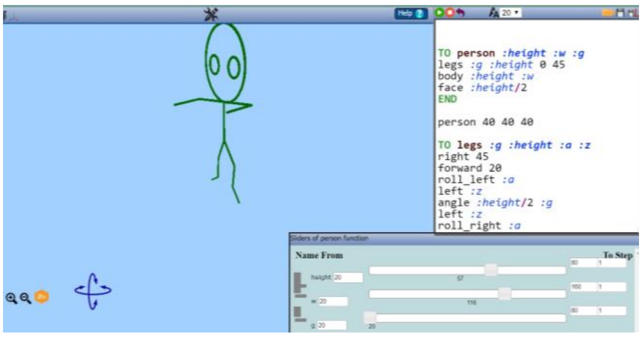


Figure 1. Students mod of the dynamic model while describing the card 'Box' (team 2)

The game Logo code is structured in three levels of parametric sub-procedures implementing the practices of decomposition and encapsulation. The main procedure (*person*) that draws the human model calls ‘level 2’ parametric sub-procedures that draw the body parts (e.g *head, body*) which they further call ‘level 3’ sub-procedures that draw basic geometric shapes (e.g *circle, angle*). The player, depending on the type of modification, can access, explore and modify the model in any of three levels, engaging in a process of programming abstraction. The parameters of an executed procedure can be dynamically manipulated with the sliders of the variation tool resulting in an animation of the constructed model (Figure 1). This affordance allows for programming parts of the model to be stable or dynamic (e.g. the angle between its legs, its position on scene, its size). Therefore, the programming of the appropriate parameters and their dynamic manipulation is central to the game.

We employed design-based research (Bakker, 2018) implementing an empirical study to investigate students meaning generation processes on computational thinking practices when they play the Code-The-Mime game. The study had a total duration of 9 hours and it was carried out in three three-hour parts. The participants, 12 students 14-15 years old, first experimented with the 3D model in MaLT2 environment and then they played the Code-The-Mime game in teams. During the study, we collected a dataset that included screen and audio recordings, interviews, questionnaires and worksheets.

The preliminary results of the qualitative analysis show that the continuous modifications of the 3D model with the integrated affordances enabled students to access powerful computational ideas, such as parameters passing, procedures, abstraction and decomposition, and to express personal meanings about them during the gameplay. The charades-like, guessing game fostered students to use computational concepts and practices to resemble real-life body movements with the 3D model. For instance, they approached the concept of the parameter as the dynamic movement of the model’s body and they created mathematical relations between parameters as a way to achieve co-variation of two body parts. Furthermore, to represent their card quick and efficient, they decomposed the 3D model into the respective Logo procedures and each time they chose which procedures and in which level they should extend or modify. Finally, some of the teams used the ‘level 3’ procedures as building blocks to add extra objects in the 3D scene, such as a ball by using the procedure ‘circle’. This study, instead of rejecting textual programming, it discusses a new approach, in which students use Logo programming in combination with other affordances, to express personal ideas in the context of a program-to-play, constructionist game.

Acknowledgements

The research of this poster has been financially supported by the General Secretariat for Research and Technology (GSRT) and the Hellenic Foundation for Research and Innovation (HFRI) (PhD Scholarship Code: 531)

References

Bakker A. (2018). *Design Research in Education*. Routledge, New York.

Kynigos, C. & Grizioti, M. (2018). Programming Approaches to Computational Thinking: Integrating Turtle Geometry, Dynamic Manipulation and 3D Space. *Informatics in Education*, 17(2)

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Weintrop, D., & Wilensky, U. (2014). Situating Programming Abstractions in a Constructionist Video Game. *Informatics in Education*, 13(*2)*, 307–321. https://doi.org/10.15388/infedu.2014.10

Wing J M. (2008). Computational thinking and thinking about computing, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366,1881: 3717–3725

Combining Learning Analytics and Qualitative Analysis for the Exploration of Open-Ended Learning Environments

Yipu Zheng, *yipu.zheng@tc.columbia.edu*
Teachers College, Columbia University, New York, USA

Paulo Blikstein, *paulob@tc.columbia.edu*
Teachers College, Columbia University, New York, USA

Nathan Holbert, *holbert@tc.columbia.edu*
Teachers College, Columbia University, New York, USA

Constructionist learning environments require learners to build projects, inhabit microworlds, create objects and code. Assessing and tracking the construction of these artifacts has always been a challenge for designers and educators. This work provides methodological input on how qualitative analysis and learning analytics based on log data can build upon each other to explore students' behaviors in an open-ended learning environment. Findings from both the data and the structures of the analysis process are reported. The work could suggest procedures and workflows for constructionist researchers to investigate new ways to examine learning in complex, open-ended environments.

Qualitative research methodologies such as interviews and field observations provide deep and thorough descriptions of complex learning phenomena. However, qualitative data collection and analysis methods are time consuming (Atieno, 2009). Recent research has suggested the potential of augmenting qualitative methods with learning analytics (Berland, Baker & Blikstein, 2014; Fields et al., 2016; Sherin et al., 2018; Worsley, 2018).

In this work we examine how qualitative data analysis and learning analytics can be used to make sense of complex process data in a complementary and iterative manner, where insights from qualitative analysis and learning analytics are put into conversation. In particular, we explore ways to gain deeper insights on students' gameplay behaviors in an online constructionist game (*Beats Empire*) through iteratively analyzing transcripts of interviews and think-aloud protocols, field notes of the social context, and log data of students' gameplay.

Beats Empire (Holbert et al., 2019) is an open-ended constructionist music management role-play game where players work to create a successful music studio by making decisions about what artists to sign, what songs to record, and where to promote their music by engaging with data about listener's musical preferences in a fictional US city. Thirty-five 7th graders in an urban middle school in the Northeastern US were engaged in one hour of gameplay, which was captured by a logging system. Concurrently with their classmates, seven students were interviewed by a researcher and played the game using a think-aloud protocol.

Through this exploratory analysis, we considered the complementarity of qualitative and log data analytics to be highly problem-driven. Depending on the research question, the analysis went back and forth spontaneously between qualitative analysis and log data analytics to understand the behavior of interest.

We started by taking an expansive view of the qualitative data, looking through field notes and interview transcripts, highlighting interesting gameplay behaviors or generating questions about potential game play patterns. For example, one anecdotal observation we generated was that some students seemed to spend more time than others cycling through auto-generated titles for soon-to-be-recorded songs, even though the choice of song title does not lead to any in-game rewards. We were interested in verifying the prevalence of this observation across subjects and exploring the reasons behind the actions. Thus, we processed and analyzed log data to help answer the question. To our surprise, out of 35 students, 24 students generated more than 50 song titles per song release. Two students even generated more than 175 song titles for some songs. (Noted that a few students initially thought that the song titles they had previously seen would cycle back around, so they kept going through the list to find the titles they liked.) This phenomenon led us to question why the song title generation feature was so prominent in the students' gameplay experience. We also wondered if the feature facilitated or hindered the students from exploring other meaningful features in the game that would better cultivate students' understanding and applications of data. Thus, we dived into the qualitative data again and compared interview transcripts of think-aloud players who generated about 15 titles per songs with those of who generated more than 50 titles per songs. We found students that generated few song titles engaged with song creation using mechanics that are explicitly defined by the game, while the others related their real-life music preference to the game. The qualitative analysis also suggested that some students who spent a lot of time scrolling through the song titles were actually looking for song titles that were suitable to the most popular moods indicated by the bar graphs in the data screen, so they were engaged with data in a different way in their decision making. In other words, an otherwise unnoticed aspect of gameplay (choosing song titles) ended up revealing a very meaningful pattern relating to students' experiences.

Furthermore, our work suggests five major roles log data analytics might complement qualitative data analysis: 1) capture transient actions, 2) identity possible trends in the sample population, 3) quantify a qualitative phenomenon using combinations of log data attributes and verify a qualitative phenomenon across a larger population, 4) identify particular chunks of qualitative data for further analysis, and 5) look for relationships among different phenomena. We believe that this work can also be applied to other types of microworlds, computational environments and constructionist games, enabling researchers to gain a deeper understanding into patterns of learning and interaction.

References

Atieno, O. P. (2009). An Analysis of the Strengths and Limitations of Qualitative and Quantitative Research Paradigms. *Problems of Education in the 21st Century, 13(1)*, 13-38.

Berland, M., Baker, R. S., & Blikstein, P. (2014). Educational Data Mining and Learning Analytics: Applications to Constructionist Research. *Technology, Knowledge and Learning*, *19(1-2)*, 205-220.

Fields, D. A., Quirke, L., Amely, J., Maughan, J. (2016). Combining Big Data and Thick Data Analyses for Understanding Youth Learning Trajectories in a Summer Coding Camp. *Proceedings of the 47th ACM technical symposium on computing science education*. ACM, 2016.

Holbert, N., Berland, M., Rutstein, D., Basu, S., Disalvo, E., Rochelle, J., Villeroy, M., Kumar, V. (2019). Designing Constructionist Formative Assessment Games. The Annual Meeting of the American Educational Research (AERA) 2019, Toronto, Canada.

Sherin, B., Kersting, N., & Berland, M. (2018). Learning Analytics in Support of Qualitative Analysis. In Kay, J. and Luckin, R. (Eds.) *Rethinking Learning in the Digital Age: Making the Learning Sciences Count, 13th International Conference of the Learning Sciences (ICLS) 2018, Volume 1*. London, UK: International Society of the Learning Sciences.

Worsley, M. (2018). Multimodal Learning Analytics for the Qualitative Researcher. In Kay, J. and Luckin, R. (Eds.) *Rethinking Learning in the Digital Age: Making the Learning Sciences Count, 13th International Conference of the Learning Sciences (ICLS) 2018, Volume 1.* London, UK: International Society of the Learning Sciences.

Constructing Creativity in Computer Science Pre-Service Teacher Education

Gemma O'Callaghan, *gemma.ocallaghan@gmit.ie.*
Dept of Computing & Applied Physics, Galway-Mayo Institute of Technology, Galway, Ireland

Cornelia Connolly, *cornelia.connolly@nuigalway.ie*
School of Education, NUI Galway, Galway, Ireland

With post-primary schools in Ireland increasing their capacity to teach computer science, it is imperative our pre-service Computer Science teachers have the capacity to develop creative thinking within their initial teacher education programmes.

Creativity is a core 21st century skill defined by business, government and education leaders (DES, 2016). In Ireland, the National Council for Curriculum and Assessment (NCCA) Senior Cycle Key Skills Framework has Critical and Creative Thinking identified in the Senior Cycle (NCCA, 2009).

The introduction of Computer Science at Senior Cycle will change the way Irish schools approach computing and information technology – replacing the idea of IT literacy and passive consumers of computing to innovators, creators and designers. According to the NCCA, Senior Cycle Computer Science "aims to develop and foster the learner's creativity and problem solving, along with their ability to work both independently and collaboratively." The new specification is based on three strands: Practices and principles, Core concepts and Computer science in practice, as shown in Figure 1.

Figure 1: Structure of Leaving Certificate Computer Science (NCCA, 2019)

Embedding Creativity in Computer Science initial teacher education is explored using the design thinking process. A framework to support pre-service teachers was developed to incorporate a student-centred problem-based learning approach while fostering characteristics such as creative confidence, risk taking and a strong sense of empathy in the pre-service teachers (Romeike, 2008; Resnick, 2014; Reilly et al., 2011). The literature shows that having pre-service Computer Science teachers solving a real-world design challenge using Design Thinking is an appropriate and pedagogical sound approach to use to develop creative thinking within their initial teacher education programme (DeSchryver & Yadav, 2015; Rauth et al., 2010).

Design Thinking aims to allow students to solve real-world complex problems using a human-centred approach. As shown in Figure 2, the process takes the students through the following 5 stages: Empathise, Define, Ideate, Prototype and Test. (Dam & Siang, 2019)

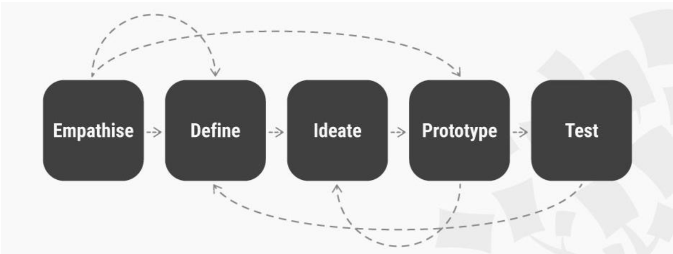


Figure 2: Design Thinking Process

The first three steps allow the student to empathise with the end user, by discovering what it is that they need, defining the problem to be solved and generating as many ideas as possible. The fourth stage in the process, prototyping, centres on the idea that making and refining the product several times results in product improvement as the student interacts with the product and reflects on its value. According to Kelley & Kelley, a bias for action is central to the design thinking process. (Kelley & Kelley, 2013) This concept has close links to Constructionism's central idea of "learning by making". (Papert & Harel, 1991) When students are constructing objects that others will see and critique, it reinforces their learning. The final solution is presented to the user in the last stage to allow for feedback and possible redefinition of the problem.

Throughout the process, students present and discuss their ideas with their peers which allows them to "boost self-directed learning, and ultimately facilitate the construction of new knowledge". (Ackermann, 2001). It has been argued that the design thinking process can be linked to Kolb's' Experiential Learning Theory (Luka & others, 2014) which consists of four phases: "experiencing, reflecting, thinking and acting". Each of these is contained within the Design Thinking process.

Closely linked to the design thinking process is the idea of creative confidence which when developed encourages students to generate new ideas and have the confidence to try them out. (Kelley & Kelley, 2012) It has been proven in the literature that using design thinking in education fosters creative confidence in the arts, business education and educational problems of practise (Henriksen et al., 2017). Due to the importance of developing creative thinking in Senior Cycle Computer Science, CS teachers would benefit from the use of a supportive framework which incorporates validated teaching strategies like the design thinking process.

References

Ackermann, E. (2001). Piaget's constructivism, Papert's constructionism: What's the difference. *Future of Learning Group Publication*, *5*(3), 438.

Dam, R., & Siang, T. (2019). *5 Stages in the Design Thinking Process*. The Interaction Design Foundation. https://www.interaction-design.org//literature/article/5-stages-in-the-design-thinking process

DeSchryver, M. D., & Yadav, A. (2015). Creative and computational thinking in the context of new literacies: Working with teachers to scaffold complex technology-mediated approaches to teaching and learning. *Journal of Technology and Teacher Education*, *23*(3), 411–431.

Henriksen et al. (2017). Design thinking: A creative approach to educational problems of practice. *Thinking Skills and Creativity*, *26*, 140–153.

Kelley, T., & Kelley, D. (2012). Reclaim your creative confidence. *Harvard Business Review*, *90*(12), 115–8.

Kelley, T., & Kelley, D. (2013). *Creative confidence: Unleashing the creative potential within us all*. Currency.

Luka et al. (2014). Design thinking in pedagogy. *The Journal of Education, Culture, and Society*, *2*, 63–74.

NCCA Specification for Leaving Certificate. (2019). Retrieved November 15, 2019, from https://curriculumonline.ie/getmedia/d73af6e3-b4e5-4edb-a514-6383e2306a4b/16626-NCCA Specification-for-Leaving-Certificate-CS-WEB-v4.pdf

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, *36*(2), 1–11.

Rauth et al. (2010). Design thinking: An educational model towards creative confidence. *DS 66-2: Proceedings of the 1st International Conference on Design Creativity (ICDC 2010)*.

Reilly at al. (2011). A synthesis of research concerning creative teachers in a Canadian context. *Teaching and Teacher Education*, *27*(3), 533–542.

Resnick, M. (2014). Give P'sa chance: Projects, peers, passion, play. *Constructionism and Creativity: Proceedings of the Third International Constructionism Conference. Austrian Computer Society, Vienna*, 13–20.

Romeike, R. (2008). Towards students' motivation and interest: Teaching tips for applying creativity. *Proceedings of the 8th International Conference on Computing Education Research*, 113–114.

Constructionism in Action in Nepal

Jonathan Lobel, *jlobel@gymnaziumbma.cz*
Gymnázium Beskydy Mountain Academy, Frýdlant nad Ostravicí, Czech Republic

Constructionist learning with technology can be a challenge to implement in developing countries, where teachers often face of lack of confidence, competence and access to resources (Bingimlas, 2009). This tablet project aimed to provide the tools, strategies, and essential professional development to overcome those barriers and offer constructionism as a confidence-building alternative to rote learning. It focused on four primary schools in former-slave communities in rural Far-West Nepal through an opportunity for cooperation with a US-based NGO ("Bridge to Nepal", 2019). Since inception in October 2014, the project has delivered 120 tablets, preloaded with a careful selection of apps, which now serve more than 1000 students across the four schools. Annual three-day workshops and mentoring for teachers have been essential to the ongoing success of the project, with teachers developing their own constructionist learning activities using the tablets. The following case study may prove useful for others aiming at constructionism in a developing world context.

Constructionism in Action in Nepal

For practical implementation, 8-inch Android ASUS tablets proved inexpensive and easy to load with a careful selection of educational apps before making the journey to rural schools in Far-West Nepal, where they would likely never again connect to the internet, due to lack of infrastructure. They were charged when electricity was available at night and used during the school day. The first 30 tablets arrived in 2014, many of which are still in service. Each following year, another wave of tablets was installed in conjunction with a teacher development workshop. The devices were used by students aged five to fifteen, in classes of up to thirty students. Devices were most often shared by a group of two to five students to facilitate cooperative learning.

Similar projects have had mixed results in developing countries. For example, the One Laptop Per Child initiative failed to have any positive impact on students' test scores in Nepal (Sharma, 2014), just as it infamously failed in Uruguay (De Melo, 2014). However, in neighboring India, a trial of Mindspark did positively impact scores (Muralidharan, 2019). This project generally follows the Integrated approach to Technology in Education (Charania, 2014), in which teachers design learning activities that integrate with existing curriculum and students build virtual learning artifacts. At annual workshops, teachers tested their designs with colleagues and gave each other feedback during group reflection sessions before trying them with students in the classroom.

Regular interviews during annual workshops with students and administrators revealed several trends and learning outcomes. Firstly, students often experimented and tinkered with the tablets in creative ways outside the formal learning space. For example, a student spontaneously used a digital model of a tea set that she had created to build a real-world version of the set from recycled materials, fitting Clayson's *situated computational thinking argument* that visual modeling allows for meaningful abstraction with various mediums (2018, p. 16). Secondly, teachers were initially slow to implement the project on their own in their classes. However, after each school assigned a “champion” staff-member to keep up motivation and communication, reported use of the devices in classes increased. Thirdly, teachers and administrators repeatedly mentioned their appreciation of the long-term human investment: much of the work at annual workshops turned out to be ongoing professional mentorship with the same people returning year after year. This kind of relational investment may be essential, but it is unfortunately not easily scalable. Fourthly, teachers reported feeling most engaged when they shared their designs with each other. They cited an increased sense of ownership of the project, deeper understanding after teaching colleagues, and fewer cultural and language barriers to overcome. This makes sense in light of Papert’s foundational belief that sharing ideas around our objects is key to learning (Ackermann, 2001). All of these outcomes underpin the project’s meta objective of confidence building in students and teachers in these former slave communities. Finally, the emails and Facebook messages that appeared with pictures of engaged students and triumphant teachers working together with the tablets gave an anecdotal and heart-warming indication that the project has indeed had a positive impact in this challenging part of the world.

References

Ackermann, E., 2001. Piaget's constructivism, Papert's constructionism: What's the difference. Future of Learning Group Publication 5(3): 438.

Bingimlas, K. A. (2009). Barriers to the successful integration of ICT in teaching and learning environments: A review of the literature. Eurasia journal of mathematics, science & technology education, 5(3).

Bridge to Nepal: True Sojourners. (n.d.). Retrieved September 1, 2019, from https://www.truesojourners.org/bridge-to-nepal.

Charania, A. (2014). ICT in education: An Integrated approach to Technology in Education. Mumbai: Sir Dorabji Tata Trust. Retrieved from http://www.dorabjitatatrust.org/pdf/ITE_Note.pdf

Clayson, J. (2018). Artifacts, Visual Modeling and Constructionism: To Look More Closely, to Watch What Happens. *Problemos*, 8-23. https://doi.org/10.15388/Problemos.2018.0.12345

De Melo, G., Machado, A., & Miranda, A. (2014). The Impact of a One Laptop per Child Program on Learning: Evidence from Uruguay. *Institute of Labor Economics*. Retrieved from http://ftp.iza.org/dp8489.pdf

Muralidharan, K., Singh, A., & Ganimian, A. J. (2019). Disrupting Education? Experimental Evidence on Technology-Aided Instruction in India. American Economic Review, 109(4), 1426–1460

Uttam Sharma, (2014), Can Computers Increase Human Capital in Developing Countries? An Evaluation of Nepal’s One Laptop per Child Program, No 169846, 2014 Annual Meeting, July 27-29, 2014, Minneapolis, Minnesota, Agricultural and Applied Economics Association

Constructionism: a dangerous adaptation

Micheál Ó Dúill, *logios.org@googlemail.com*
Dept of Design & Technology, British School of Sofia, Sofia, Bulgaria

The human capacity to construct, as described by Papert (1991 and earlier), is unique. It is most clearly exhibited in the capability to draw, be it language, representations, or mechanisms. No other animal can, or has been able, to do this. A hint as to the character of this capability, which must be a biological evolutionary adaptation, is found in Heidegger's (1947) obtuse "Question concerning technology." Here he asserted that technology "enframes" us, and that it is "challenged forth" rather than "at hand". Though generally prejudiced in its standpoint, these differences do typify the difference between us and all other animals. Whilst they, from cadis fly larvae through birds to Neanderthals and Denisovans, can construct complex artefacts from inbuilt evolved behavioural templates, no species other than the human originating in Africa had the capacity to design. And yet, as our genes show, we had regular conjugal relations with them. It follows that any genetic difference must have been socially and reproductively insignificant. Any yet, it is our species, not them, that split the atom. In so doing, we created an entropy differential far greater that any biological entity can survive. The question now is: how may a biological organism create such destructive physical entities?

The answer will lie not with the phenotype directly but with its genotype. The DNA that specifies the form and function of the phenotype embodies a vast amount of information about the physical world; its materials and forces. For instance, DNA creates haemoglobin or chlorophyll simply by substituting an iron atom for magnesium. Many forms, mechanisms, and chemical processes that DNA has evolved remain beyond our understanding yet must be its foundation. How did humans gain access to information that makes possible the creation of naturally impossible entities? Where has DNA expressed products of its creativity so that they can become accessible to a conscious organism?

One of the most amazing DNA creations is the set of sense organs. These produce information that goes well beyond the physical events that are their data. The most remarkable is vision. The only entities that impinge on the eye are reflected photons of varying wavelength. From the vast spectrum of these, vision relies only on those that trigger a reaction in rhodopsin and its variants. Data from three cone receptors in the eye activates a neural colour palette to give hues, blends, tones and shades based on red/green. Blue/yellow, and black/white. This palette is DNA created and bears no relationship to physical reality: it is an adaptive false-colouring of the world. Not only is the world colourised by DNA information but the picture itself is drawn using lines and angles, and direction of motion built into neural modules. Fortunately for us, these mechanisms are aeons old, so we share our worldview with many other organisms, conscious or reactive. It is a vast store of DNA created information that goes beyond that present in the physical world. It is inaccessible.

When we talk of a conscious organism, we restrict ourselves to mammalian species: possessors of a neocortex and in particular its prefrontal areas. Once the lobotomised silent area of the brain, we now know that prefrontal cortex preforms an executive function (Fuster 2015). Reciprocal connections from it to most parts of the brain, old and new, provide information: cognitive, affective, sensory and motor, which it modifies and returns, thereby planning futures based on historic data. As the species Homo evolved, the brain enlarged, most prominently in the prefrontal area whose neurones extended ever further. This facilitated complex social organisation and language. But it did not, except in our species, make (super-natural) construction possible.

At the first constructionist conference the author made the argument that constructionism required a scientific rather than philosophical foundation if it were to be influential. Today it remains a niche philosophy. Here, a decade later, in this obscure corner of the sixth edition is offered a proposition that offers a scientific basis. Though it requires some suspension of disbelief and an understanding of entropy and information, it is based on established science and is testable.

The proposition is simple in the extreme. It states that, in our species alone, prefrontal connection extended to the layer of raw sense data first found by Hubel (1995) and colleagues. This is the layer of DNA crafted information that is the foundation, the elementary particles, of perception. It is normally processed by a set of evolved mechanisms to provide the organism with a sensorially integrated perception of the world around it. Accessed directly by cognitive prefrontal cortex, these elements may be combined to create entirely novel, unnatural (super-natural) mental constructs. Returned to the effective part of the nervous system, they may motivate a physical construction; be it a sandcastle on the beach or a theory of the universe. The difference in the information flow between modern humans and prior species, including our conjugal cousins, is illustrated below.

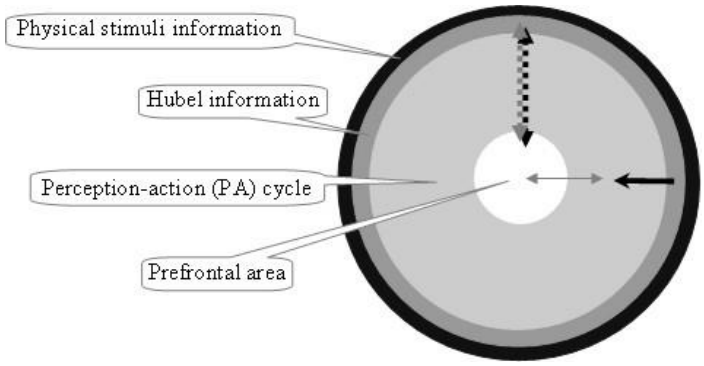


Figure 1. Schematic of the modern human brain showing the flow of information. Humans access raw Hubel information (grey/black dotted line) via prefrontal cortex, which composes it to create cognitively and affectively pleasing yet potentially dangerous low entropy constructions.

It is obvious, as mathematicians say, than constructs composed of DNA created information will be of lower entropy than any derived from sense data. Here is the entropy differential that provides our power over nature. As such it is dangerous. We may construct wisely (though this is difficult) or we can construct destructively. Rather than citing climate change, an example will be taken from mathematics education. The simplest way of enumerating is to put objects into bundles and mark them off on a tally. However, language, a proxy for the way the brain works, puts it differently. The Indian mathematicians developed our current decimal notation. Here, in contrast to a Roman X, the sequence 10 has no physical referent until you know what the permitted maximum is in any column. For decimal it is nine. This is reflected in language: nineteen - twenty. So, the brain works on a register system, not bundling like perception. We have two constructions to offer children as they learn to compute: the ten-bead abacus and objects to bundle into tens, or a four function calculator. The one is congruent with thought, the other with perception. Fascinatingly, committed constructionists prefer the concrete over the symbolic thus stunting cognitive growth. No comment.

References

Fuster, J. M. (2015) *The Prefrontal Cortex*, 5th edition. Academic Press, Amsterdam.

Heidegger, M. (1954, 1977) *The Question Concerning Technology*. Translated by W. Lovitt. Garland Publishing, New York.

Hubel, D. H. (1995) *Eye, Brain and Vision* (2nd edition). Henry Holt & Company, New York.

Papert, S. (1991) *Situating Constructionism*. Online at www.papert.org

O Duill, M. (2010) *Can there be a Science of Construction?* In: J. E. Clayson and I. Kalas, Proceedings of Constructionism 2010 (12th European Logo Conference), 16-20 August, Paris, France. Comenius University, Faculty of Mathematics, Physics and Informatics, Bratislava.

Creative Learning Workshops: an introduction to Constructionism

Juliana Ramalho Barros, *juliana@ufg.br*
Institute of Social and Environmental Studies, Federal University of Goiás, Goiânia-GO, Brazil

Paulo Roberto Ferreira de Aguiar Junior, *prf.geo@hotmail.com*
Institute of Social and Environmental Studies, Federal University of Goiás, Goiânia-GO, Brazil

We are all creative, however, as we grow, we come to believe that we have lost our skills to create and this belief is reflected in the way we perform our activities. With teachers, this is no different, since most have had an education based on little or no interactivity, focused more on the figure of those who teach than of apprentices. Therefore, if we want to prepare our children to develop their inventiveness, we must first foster the asleep creativity in teachers.

One believes that Constructionism, both as a learning theory as a strategy for education, can be a powerful tool, because one of its objectives is that students actively participate in the construction and reconstruction of their knowledge. Based on this approach, Lifelong Kindergarten research group at the MIT Media Lab has developed strategies to engage people in learning experiences that culminated in what is called Creative Learning, which is based on four pillars (RESNICK, 2017): *Projects*; *Peers*; *Passion*; *Play*.

Regarding an emancipating and empowering education, based on the ideas of Freire (2017) and Papert (1994), the bases of Constructionism to teachers and students of various undergraduate courses from the estate of Goiás - Brazil, we hold workshops that offer a hands-on experience, followed by reflection and theoretical subsidies to apply creative learning in the classroom. During the activity, we try to get people to engage in all phases of the creative process, which Resnick (2017) imagines as a great learning spiral (imagine, create, play, share, reflect, imagine…).

The workshops are part of a research project entitled "*Creative learning applied to the teaching of climatology and its school contents: looks, practices and awakening to science*", funded by the National Council for Scientific and Technological Development – CNPq, from Brazil, and are intended for teachers, pedagogical coordinators and students of undergraduate courses. To attract this public, the dissemination was done in schools, universities, on the website of the Brazilian Network of Creative Learning and in social networks. The applied methodology was developed based on Resnick (2017) and Clapp, Ross, Ryan & Tishman (2017).

The main objectives of the workshops are:

- To provide experience in a Creative Learning activity and present the basic principles of Creative Learning;
- To identify challenges and opportunities for the implementation of Creative Learning;
- To draft projects to be applied by the participants in schools and universities;
- To promote integration between teachers from various areas of knowledge and people interested in innovative learning methodologies.

The activity runs for 4 hours and, at first, participants (maximum 25 in each workshop), who work in groups, are invited to immerse themselves into microworlds. They are invited to think about how Brazil will be in the 22nd century and each group is placed at a workstation, with the most diverse types of materials (cardboard, fabrics, glue, batteries, led, scraps, etc.). Each workstation has a theme: fashion, housing, means of transportation and culture (Figure 1).

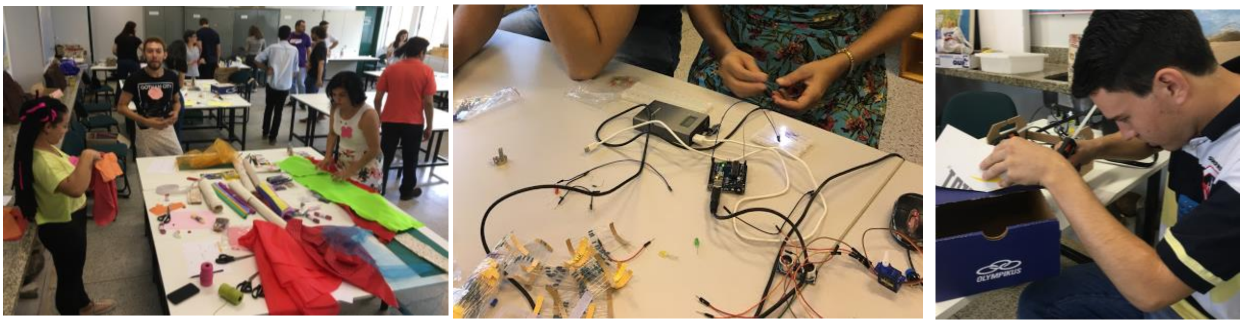


Figure 1. Workshop participants working on their projects

After 1 hour, the groups present their projects and share their ideas, mainly, how was the process of performing the activity and what they learned from it. Then, they are invited to answer some questions placed on panels on the walls, so that we can all reflect and dialogue about the results. In addition to dashboards, everyone receives a questionnaire so they can evaluate activity and insert their impressions without being identified. We also provide an auxiliary guide for them to think about implementing creative learning activities and projects at their schools and universities.

The last stage is a conversation wheel, in order to evaluate the positive and negative strains of the workshop, as well as what can be modified and also how each person felt when performing the activity. People are encouraged to expose their opinions freely and critics are very welcomed. In addition, we invite participants to integrate into our research network and work focused on Creative Learning.

It is hoped that the workshops can be replicated by participants in their schools and communities, as well as that the precepts of Creative Learning will be disseminated among teachers, students, school managers and people interested in education. Through evaluations and dialogues, it is possible to observe that people engage with their ideas and work with each other to overcome the challenges that appear throughout the activity.

The data are being organized, however, preliminary results already reveal that more than 80% of the people participating in the workshops seek to implement projects and activities based on active methodologies in their classes or engage in groups and networks that help promoting Creative Learning.

References

Clapp, E. P., Ross, J., Ryan, J. O. & Tishman S. (2017). *Maker-centered learning: empowering young people to shape their worlds.* San Francisco, CA: Jossey-Bass.

Freire, P. (2017). *Pedagogia da autonomia: saberes necessários à prática educativa* (55ª ed.). Rio de Janeiro/São Paulo: Paz e Terra.

Papert, S. (1994). *A máquina das crianças: repensando a escola na era da informática*. Porto Alegre: Editora Artes Médicas.

Resnik, M. (2017). *Lifelong Kindergarten*: *cultivating creativity through Projects, Passion, Peers and Play.* Cambridge, Massachusetts: The MIT Press.

Designing for Empathy in Engineering Exhibits

Kylie Peppler, kpeppler@uci.edu
Donald Bren School of Information and Computer Sciences, University of California, Irvine, USA

Anna Keune, akeune@uci.edu
Donald Bren School of Information and Computer Sciences, University of California, Irvine, USA

Maggie Dahn, dahnm@uci.edu
Donald Bren School of Information and Computer Sciences, University of California, Irvine, USA

Dorothy Bennett, dbennett@nysci.org
New York Hall of Science (NYSCI), New York, USA

Susan Letourneau, sletourneau@nysci.org
New York Hall of Science (NYSCI), New York, USA

Women are persistently underrepresented in science and engineering fields (Bix, 2014). However, when engineering problems are posed as personally meaningful and connected to people and communities, girls are more likely to express interest in solving them (Bennett, 2000). Personally meaningful problems can tap into learners' capacities for empathizing with those for whom they are designing, and can help learners develop positive conceptions of engineering (Cunningham & Lachapelle 2014). While some research has attended to this relationship between empathy and engineering, such empathic engineering engagement remains understudied in education. Constructionism helps to further explain that empathy and relationships between both people and material are central to the learning and design process. For example, Harel and Papert (1991) argued for engagement with carefully selected materials to support the development of personal relationships in which affect can blend with and inform formal domain learning. This can create moments of empathy in what we might call a human-centered design experience, a design process that takes a deep appreciation of the needs of others into consideration (Nelson & Stolterman, 2003).

While human-centered design elements are promising starting points for evoking empathy within engineering learning activities, it remains unclear how to design engineering activities to support empathy and how empathic design relates to engagement with engineering. Our research focuses on the design of engineering activities within museum contexts because informal STEM learning environments are inherently social, have the potential to reach diverse audiences, and can contribute to more inclusive and contextualized experiences with engineering. This research asked: *How do design elements of science museum-based engineering activities support empathy development and influence engagement with engineering for girls (ages 7-14)?* We conducted observational studies of three activities at an urban science museum on the East Coast of the United States. Exhibits included: 1) *Help Grandma*, in which visitors read activity cards with characters requesting help with everyday tasks, and designed inventions to help them; 2) *Chain Reactions*, in which visitors created contraptions to take care of pets; and 3) *Air Powered Vehicles*, in which visitors constructed vehicles to help someone travel over different landscapes (e.g., sandy deserts or forest floors).

We observed a total of 117 girls (ages 7-14; 38-40 observations per exhibit) using an observation protocol we co-developed to track engagement in engineering practices and evidence of demonstrating empathy within the design process. Our protocol included overall dwell time, empathy markers (i.e., desire to help, user-centered design, affective empathy toward designing, affective empathy toward user, perspective-taking, familiarity, societal issue), and engineering practices (i.e., imagining new possibilities, iteration, persistence, problem scoping, solution finding, testing, and tinkering). We photographed visitors' projects and conducted semi-structured interviews that asked visitors about their design process.

We conducted an analysis of variance (ANOVA) to compare dwell time, empathy markers, and engineering practices across activities. For all exhibits we found that the average dwell time within exhibits was well above average of science in museum exhibits (typically 1 minute) at an average of 17 minutes per observed visitor. This suggested that the designs presented opportunities for investigating engineering practices and their relationship to empathy. Of the engineering practices we observed, frequencies clustered around persistence, testing, and tinkering across exhibits perhaps also because observed visitors often started out tinkering with materials and went on to solve problems with the materials. Of the empathy practices we observed, frequencies clustered around perspective-taking and familiarity. Perspective-taking refers to acting out a use or explaining how someone would use the design, for example, a visitor who created an invention to help a grandma carry a heavy item stating "If it was real, the person would push it up" while acting out how the design would be used.

Of all exhibits, *Help Grandma* produced the highest frequencies across all empathy markers. Help Grandma included the design element of creating an innovation for a familiar person and we consider that it was this design element that supported empathy. Taking a deeper look at empathy, we recognized that empathy markers explained both engineering as well as dwell time. Those visitors who showed at least one empathy marker stayed with the exhibit over 30 minutes and performed at least five different engineering practices. Small design changes had an impact on empathy markers. For half of the observations, the animal characters of the Chain Reactions activity wore collars that read "I want to play" or "I am hungry". For the other half of the observations the collars were omitted. When the collars were there, visitors stayed with the activity significantly longer with more empathy practices compared to engagement without collars. The collars provided opportunities to attribute needs and desires to the characters, which, similarly to the Help Grandma activity, supported the possibility to design for characters. The findings that empathy markers are compelling predictors for engineering and dwell time and that empathy markers can be evoked through specific design elements has implications for the design of engaging engineering activities that appeal to girls and museum visitors more broadly.

References

Bennett, D. (2000). Inviting girls into technology: developing good educational practices.

Commissioned paper for the American Association of University Women. Excerpts in American Association of University Women (2000). *Tech-savvy: Educating girls in the new computer age.* Washington, DC: AAUW Educational Foundation.

Bix, A. S. (2014). *Girls coming to tech!: A history of American engineering education for women*. Cambridge, MA: MIT Press.

Cunningham, C. M., & Lachapelle, C. P. (2014). Designing engineering experiences to engage all students. *Engineering in pre-college settings: Synthesizing research, policy, and practices*, 117-142.

Harel & Papert, S. (Eds.) (1991). *Constructionism*. Westport, CT: Ablex Publishing.

Nelson, H. G., & Stolterman, E. (2003). *The design way: Intentional change in an unpredictable world: Foundations and fundamentals of design competence.* Educational Technology.

DESIGNING LOGOS USING LOGO: a constructionist approach to understand trigonometric functions through periodicity

Myrto Karavakou, *karavak@ppp.uoa.gr*
Educational Technology Lab, Dept. of Educational Studies, National and Kapodistrian University of Athens, Greece

This poster presents the main idea of a constructionist learning approach to understand trigonometric functions by engaging with a Logo-based application as well as some preliminary results gained by its first pilot implementation to 9th grade students. The presenting approach consists of a new idea which has only been implemented to a small scale of students (2 groups of 2 students each) and will be expanded and re-examined in the near future. However, by exploring the students working with the microworld in a first level, many interesting points connected to constructionist features on learning mathematics emerged.

The mathematical emphasis is on the trigonometric functions and specifically on their periodic property. The motive lies in three perspectives: 1. the surprisingly poor attention of research on the meanings generated by students around these concepts; 2. the non-constructionist way these functions are traditionally treated at school and 3. the affordances gained by exploiting the property of periodicity for creative and richly-constructionist ways for meaningful investigation and understanding through the appropriate medium. According to the existing literature, the initial stages of learning about trigonometric concepts, which are traditionally attached to the triangle model, are proven problematic as they consist of a restriction on the understanding of sine and cosine as functions and are fraught with difficulties in general (Weber, 2005). In fact, the triangle model is insufficient to represent the totality of trigonometric functional properties and also to host activities that awaken the creativity and imagination in students. Unfortunately this model still prevails as the predominant one because of its easily visualized nature. But what if there was a way to not only visualize the trigonometric functions as a snapshot of a certain angle-domain, but also include its unique dynamic nature as well as the ability to manipulate it? What if we could use these functions as an expressive tool for artistic ideas easily formed by Logo-based programming?

In the educational world where constructionist practices redefines the learning approaches where the emphasis is given on the meaning-making processes of students, the above idea can be supported (Kynigos 2015). There is a special tool which makes that possible: the medium of MaLT2 (Machine Lab Turtle-sphere); an online environment of our lab's design which integrates Logo textual programming with the affordances of dynamic manipulation and 3D graphics (Kynigos & Grizioti, 2018). In this case, students were encouraged to design an animated geometric logo using MaLT2 for a supposed company whose only requirement was to include the feature of periodic motion. At the first phase, students had to decode and reconstruct a given 2D animated logo specially designed by me -representing a periodically reshaping right angled triangle- as they only had access to the virtual outcome in motion (given as a repeated GIF; Picture 1, Phase 1). In the second phase, they were free to construct their own design with the added requirement for it to be 3D. The main hypothesis beneath this "artistic challenge" is that by designing the geometric logo and its motion by themselves through programming, they would physically grasp the essence of the trigonometric functions -in which the characteristic of periodicity resides- and experience their inner properties in a meaningful creative way. This assumption led to an open research question: What meanings do students construct on trigonometric functions when engaging with the logo-based environment of MaLT2 and its dynamic manipulation potentials?

The results gained by the two groups were quite enlightening on this issue. Even though 9th Graders had only engaged with the trigonometric concepts as ratios (in terms of a triangle or the

Cartesian coordinate system), they naturally adjusted to their functional aspect and its properties in order to handle them in their design. During the first task, they managed to uncover the -as mentioned by one of them- “bizarre functions” which cause the fluctuation of the triangle’s perpendicular sides; sine and cosine. This “uncovering” was reinforced by students’ intimacy of trigonometry in the triangle model, but they overextended its borders; they handled sine/cosine as functions, emphasizing to their dynamic features. They produced the code that constructs the requested shape and by dragging the slider which controls the values of the variable (:t), they realized the dynamic power of sine(:t) and cosine(:t) on their construction. The revealing periodicity impressed them and consisted of a strong motive for making more complicated constructions. The second task led to many experimentations with new codes with the trigonometric functions starring in them. One example, made by the first group of students, is presented in Figure 1 (Phase 2).

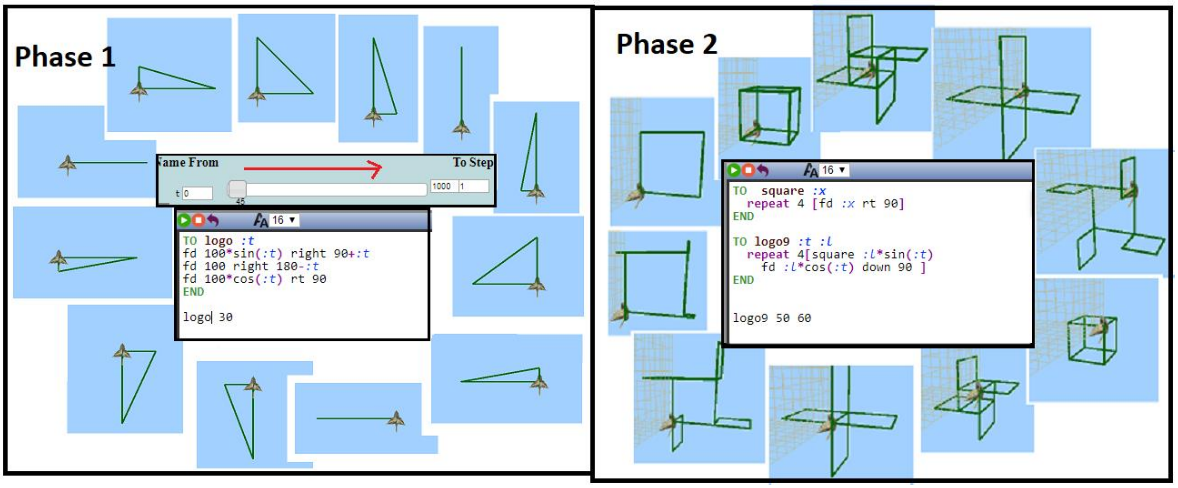


Figure 1. Examples of logos as constructed by students during the two phases

My closing remarks on the results are summed up in two points. Firstly, students’ collaboration and debate during the logo constructions reveal the reasons why they were impressed by the periodic feature of the two trigonometric functions; the fact that it provides feelings of harmony, uniformity and symmetry to the dynamically manipulated shape which add up to an artistic quality. Secondly, during their discussion -boosted by my questions- some properties of the two trigonometric functions were surfaced, such as the phase difference between them, their graph, their rate of change and the advantages of their periodic nature. Students naturally generated meanings on these properties based on their experience with the construction of the logos in a playful and healthily competitive mood. The hypothesis of the research was reinforced by these points, setting this “artistic challenge” even more challenging in order for my future research to gain richer results on the matter.

References

Kynigos, C. (2015). Constructionism: Theory of Learning or Theory of Design?. In *Selected regular lectures from the 12th International Congress on Mathematical Education* (pp. 417-438). Springer, Cham.

Kynigos, C., & Grizioti, M. (2018). Programming Approaches to Computational Thinking: Integrating Turtle Geometry, Dynamic Manipulation and 3D Space. *Informatics in Education*, 17(2), 321-340.

Weber, K. (2005). Students’ understanding of trigonometric functions. *Mathematics Education Research Journal*, 17(3), 91-112.

Developing children's sense of place by utilizing GPS-enabled Arduino toy in their community

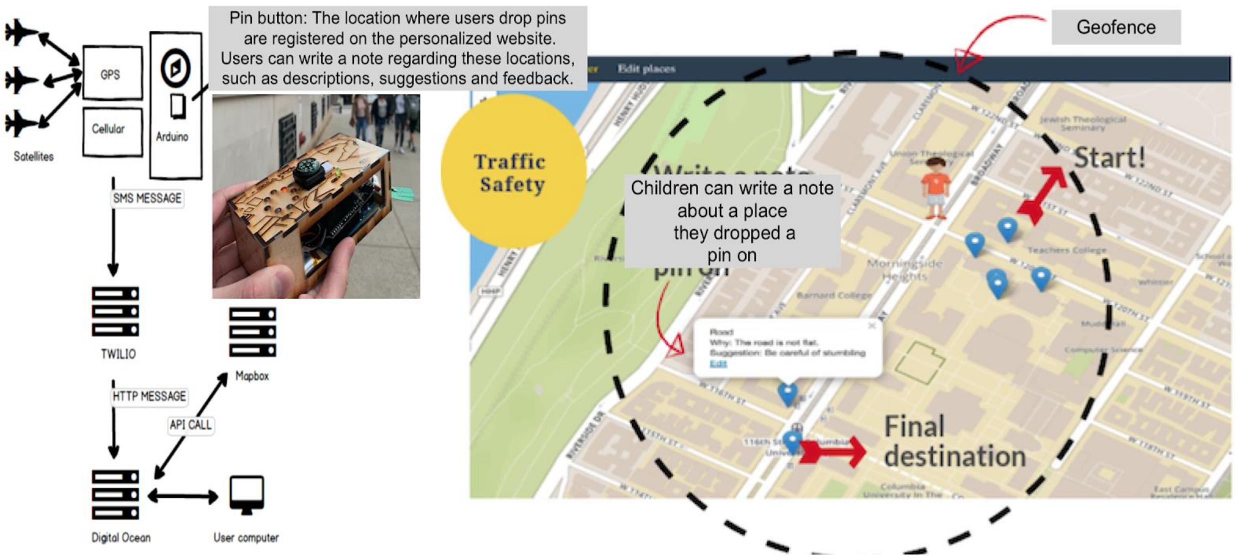
Ahyoung Kim, *ak4373@tc.columbia.edu*
Instructional Technology and Media, Teachers College, Columbia University

Children's sense of place, their feelings and perceptions with regard to the geographic area, is important as it has a powerful impact on children's cognitive and identity development (Qazimi, 2014). In a cognitive aspect, children can understand how places are shaped by different factors and develop spatial knowledge effectively when they interact with everyday environments (Pike, 2008). Having a sense of place through local geography, particularly during childhood experience, greatly influences the formation of individual identities and building the baselines for perceiving the new places as they grow up to be adults (Measham, 2007).

However, in social studies classrooms for grades 3 to 6 in South Korea, especially where excessive enthusiasm for education exists, children can hardly relate to the description of the local area in the textbook and show a lack of understanding in the geographical characteristics of surrounding spaces. Interestingly, according to their general daily routines, most are less exposed to exploring their neighborhoods or getting around places near them. Rather, they tend to dwell in limited places like their homes, schools, private institutes, or playgrounds. Therefore, it seems like there are not many chances to expand their understanding and use of spaces around them.

This situation in South Korea is often caused by the children's tendency to spend time on home entertainment or academic activities as well as parents' concerns regarding children's safety. Most children usually spend their daytime in private educational institutes after school and are exposed more to digital media rather than going around places. Moreover, parents usually worry about their children's safety as they assume their children might get lost going places or might be exposed to potential dangers. Consequently, children only visit a few places with their parents and therefore have a fear of new spaces, thereby missing opportunities to construct their local knowledge in the real-world context and to build knowledge about their communities.

Figure 1. Diagram of the device and its companion website overview



What if we could change this situation by minimizing parents' concerns and maximizing children's opportunities to interact with the places around their neighborhoods? In order for children from grades 3 to 6 to have more opportunities to experience the real-world and learn about their communities, I developed a toy by utilizing a GPS-enabled Arduino which allows them to visit new spaces with the initial help from their parents. The design of this toy is grounded by constructionism, in that it situates learners to explore the surrounding environment, provides a tool to build a cultural artifact on their personalized 'journey map', and offers an opportunity for them to share their ideas and feelings with others on this 'journey map' (Ackermann, 2001).

The digital compass on this toy sports 8 LED lights, each indicating a cardinal or intercardinal direction to the user based on the GPS information. Using the compass, users are able to figure out which way they need to go to arrive at a new place by following the LED lights. The toy also might help to increase children's interests and excitement of visiting unexpected spaces in the surrounding areas, ridding them of their fears of wandering around.

While they are navigating, children also have a chance to express their own perspectives regarding the places in their personalized 'journey map'. By pressing the white button on the device, children can drop a pin to save the places they have visited and written reports on the companion website which can be shared with other users. After they complete their trips, they are encouraged to write a reflective journal in their personalized 'journey maps' about what they like/dislike about the local place as well as what safety/traffic issues are around (Catling, 2005), which can be communicated and shared with other users.

To help parents make sure this experience is educational and safe enough, the predetermined location could be carefully designed by parents beforehand on the companion website. While children are navigating, parents are also able to receive an alert when children stray away from the specific location by setting up a geofence for the device. In this way, children can extend the boundaries of places gradually with the guidance of their parents.

Through their exploration of their communities with their devices and journaling, children can build special bonds within their communities, realize how the places in their communities are deeply related to their daily lives and connect their experience to school learning. This experience can also provide other learning opportunities such as understanding direction with the compass, studying the landscape formation of the surrounding environment, and starting to be interested in the area's history. Although this tool is initially designed for specific populations, it also can be applied to more comprehensive audiences such as children who want to research their communities' resources or train their independent mobilities in their local areas relative to their social and regional contexts.

References

Ackermann, E. (2001). Piaget's constructivism, Papert's constructionism: What's the difference. Future of Learning Group Publication, 5(3), 438.

Catling, S. (2005). Children's personal geographies and the English primary school geography curriculum. *Children's Geographies*, *3*(3), 325-344.

Measham, T. G. (2007). Primal Landscapes: Insights for Education From Empirical Research on Ways of Learning About Environments. *International Research in Geographical and Environmental Education* 16 (4): 339–350.

Pike, S. J. (2009). *Children and their environments in Ireland* (Doctoral dissertation, Queen's University Belfast).

Qazimi, S. (2014). Sense of place and place identity. *European Journal of Social Science Education and Research*, *1*(1), 306-310.

Digital Applied Learning and Innovation Lab: A Model of Constructionism in Higher Education

Erica Lobel, *erica@dali.dartmouth.edu*
DALI Lab, Dartmouth College, Hanover, USA

Tim Tregubov, *tim.tregubov@dartmouth.edu*
Computer Science, Dartmouth College, Hanover, USA

Previous work firmly establishes the value of the constructionist approach in K-12 settings, with few documented applications in higher education (Sacristan, 2018; Clinton & Rieber, 2010; Trust, Maloy, & Edwards, 2018). Universities are in need of academic innovation to prepare students for a changing world (Laurillard, 2002; Obama, 2009). Popular adaptations like makerspaces and capstone projects are limited by the transitory nature of the projects and teams and inert conclusions. The Digital Applied Learning and Innovation (DALI) Laboratory at Dartmouth College is an extra-curricular, interdisciplinary model of constructionism. Over one hundred students a year learn by making — gaining new knowledge and applying theories learned in the classroom by building tech solutions for early-stage entrepreneurial and research ventures. Qualitative reports indicate rich and multifaceted learnings. This poster highlights DALI's methodology, outcomes, and the value that constructionism adds to traditional higher education.

Figure 2. Peer mentorship plays a central role in student's growth.

The DALI Lab was founded in 2013 to empower students through the transformative process of building artifacts with real impact. Students join as early as their freshman year and 95% continue with DALI for the duration of their time at Dartmouth. Throughout their tenure, students work in teams (Figure 2) and develop technical, problem-solving, creativity, leadership, teamwork, and communication skills. The lab's project-based approach relies heavily on sourcing meaningful projects, a robust support system, and peer learning and mentorship (Figure 1) (Boud, Cohen & Sampson, 2001). The program is popular with students and partner organizations; acceptance rates hover at 30% for both. Throughout the project, teams reflect collectively and individually through surveys and structured conversations.

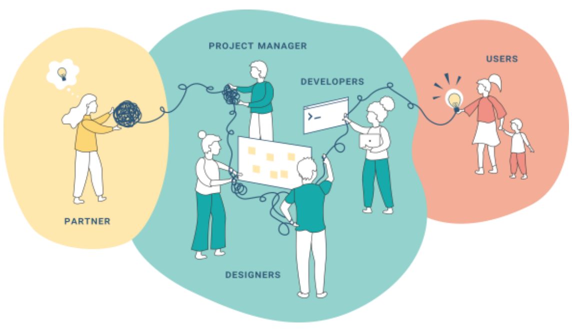


Figure 3. DALI students work in teams of project managers, developers, and designers to bring a partner's idea(s) to fruition.

Exploration of a case study will exemplify how constructionism is applied with compelling results. One team partnered with a physician and their Ph.D. candidate at the local hospital who study the role of mobile health modalities in physical therapy compliance and improvement over time in older adults. This project called for a mobile app easily accessible to older adults. Real projects entail real constraints, which can frustrate students, but also provide valuable teachable moments. In order to help the patient see their own progress and stay motivated, the team decided to connect an exercise resistance band with a force sensor to a smartphone app via Bluetooth. This also helps doctors keep track of patient's exercise habits for research purposes, an additional user persona for designers to keep in mind. In the next phase (as seen in Figure 3), design and development ran concurrently, requiring a high level of communication between the two sides of the team. Every other week the designers drove to the hospital to conduct user testing with the research participants. Their findings informed the next design iterations and how the developers implemented the app. At the end of the term, in just ten short weeks, the app was ready to be employed in the partner's field studies.

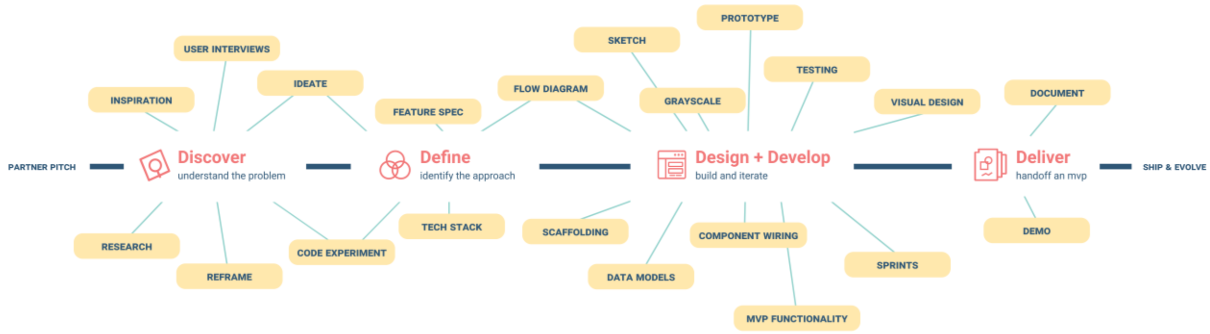


Figure 4. The DALI Process is a series of milestones guiding teams through the design and development

In the retrospective feedback, all team members reported substantial learnings. In addition to technical skills (working with Bluetooth, Android development, user testing), they reflected on gaining key problem-solving techniques. Every student was challenged by the fast pace and critical communication exchange between developers and designers, emerging at the end of the term as better teammates and communicators: "I developed as both a contributor and leader. By constantly communicating, asking questions, and understanding my teammates, I learned how to improve our team productivity as a unit instead of as just an individual." Finally, all team members strongly agreed (5/5 on a five-point scale) that the project was meaningful, fun to work on, and a great learning experience (5/5). They agreed (4/5 rating) that the project was conceptually interesting.

Papert's distinction between learning-by-doing versus learning-by-making rings true at DALI (Papert, 1991). As illustrated in the case study, those who begin with little technical, teamwork, or subject-matter expertise learn through quickly and collaboratively building an artifact. Over six years, the program has given 486 alumnae the opportunity to learn through building, and delivered 230 projects to partners around the world. The DALI Lab brings all the benefits of capstone projects to students early in their careers, building a larger knowledge base over a period of years, rather than months. It provides the context and motivation for learning in class, causing many students to perform better in coursework after a term or more of participating in DALI. Based on quarterly surveys, a recurring theme is that "At DALI I run into challenges that don't really occur in classroom settings." Students apply to the program expecting to gain skills that make them employable, (and indeed, upon graduation, they routinely earn competitive positions at industry-leading firms), but the ultimate value is in learning to problem-solve when there is no right answer and to have real-world impact rather than receive a grade.

References

Boud, D., Cohen, R., Sampson, J. (2001) Peer Learning in Higher Education. Routledge, London.

Clinton, G. & Rieber, L.P. (2010) The Studio experience at the University of Georgia: an example of constructionist learning for adults. Education Tech Research Dev 58, 755–780.

Obama, B. (2009, April 27). Remarks by the President at the National Academy of Sciences annual meeting. National Academy of Sciences, Washington D.C.

Laurillard, D. (2002) Rethinking University Teaching: A Conversational Framework for the Effective Use of Learning Technologies. Routledge, London.

Norman, D. (2013) The Design of Everyday Things. Perseus, New York.

Papert, S. (1999) "What is Logo? And Who Needs It?" In Logo Philosophy and Implementation (pp. VIII-IX) Montreal: LCSI.

Sacristán, A. (2018) The Constructionist Experiences for Mathematics Across Education Levels. Constructionism 2018, Vilnus.

Trust, T., Maloy, R.W. & Edwards, S. (2018) Learning through Making: Emerging and Expanding Designs for College Classes. TechTrends 62.

Embodied Participatory Simulations of Disease as an Entry Point for Network Analysis

Corey Brady, *corey.brady@vanderbilt.edu*
Department of Teaching and Learning, Vanderbilt University, Nashville, TN, USA

Lucas Yarnes, *lucas.yarnes@vanderbilt.edu*
School for Science and Math at Vanderbilt, Nashville, TN, USA

We have been working with a corporate partner to stabilize a flexible wearable-computing platform for embodied participatory simulations (Brady et al, 2015; 2016), involving hardware (electronic "badges"); software (a blocks-based authoring environment for creating programs that exploit the badges' native functionality); and network infrastructure (including facilities for creating publish-and-subscribe services and for storing information needed for or produced through badge activities). This poster describes the first pilot of a new WiFi-enabled version of this full platform in a course in Human Geography for pre-service social studies teachers, at a large private university in the southeastern United States. Our motivation was to explore how embodied participatory simulations (Brady et al, 2017; Collela et al, 1998) could provide an accessible entry point to ideas of network theory for non-technical pre-service teachers. We asked, "How can playing out an interaction network through embodied role-play offer resources for participants to reason about the role of network structure in the spread of disease?"

Network theory offers a powerful set of tools to make sense of, among other things, (a) emergent phenomena that are illuminated by structures of groups that go beyond their spatial distribution, or (b) the way that groups' structures are formed in interaction. These are powerful ideas for explaining complex systems, but they can be challenging to reason with. In connection with agent-based modeling, network analyses can offer exciting new insights on a variety of phenomena, including the spread of disease (Head et al, 2018; Vermeer et al, 2017).

Participatory simulations, or PartSims (Wilensky & Stroup, 1999a) have been an important tool for enabling groups to make sense of complexity through role-play, often supported by communications technology. In the context of agent-based modeling, NetLogo's (Wilensky, 1999) HubNet module (Wilensky & Stroup, 1999b), and a web-based implementation entitled GbCC (Brady et al, 2018) enable virtual PartSims and other group-centered activity designs that make emergence accessible (Wilensky & Stroup, 1999a; 2000). Embodied participatory simulations (Brady et al, 2016; Colella et al., 1998; Klopfer, Yoon, & Perry, 2005) have developed in parallel with the virtual versions described above and are useful where physical enactment is an asset.

In a unit on the spread of disease and other social phenomena of dissemination and diffusion, the classroom group engaged in several simulations of social interactions, using NetLogo and GbCC. To foreground group structure and its influence on interactions, we used the badge platform to enable the group to build interaction networks with different structures and characteristics, supported by real-time visualization. As a group, we then reasoned about how disease or other phenomena would spread on these networks, using replays of the interaction dynamics, where one or more nodes of the network were chosen to start the simulation as "infected."

In the first activity, students could interact with anyone else in the class, with the goal of finding out how many people shared a trait with them (a random number, 1-10). Each time they spoke with a classmate, their badges transmitted the interaction to a server, allowing a web-enabled NetLogo model to display the emerging network of interactions in real time (see Fig 1). The NetLogo model then allowed the interactions to be re-run virtually, with the option of supposing that one (or more) of the badges/participants began as infected.

Figure 1. Simulating social interactions with the badges. Right: a real time display of the network, here showing the final state, with the node-link representation cantered on the most-connected node.

Questions arose about whether and how disease might propagate differently on different networks. In response, the class then ran two other scenarios: one where they could connect only with classmates who had at least one other class with them (left) and another where they could connect only with individuals who were "friends" on a social media platform.

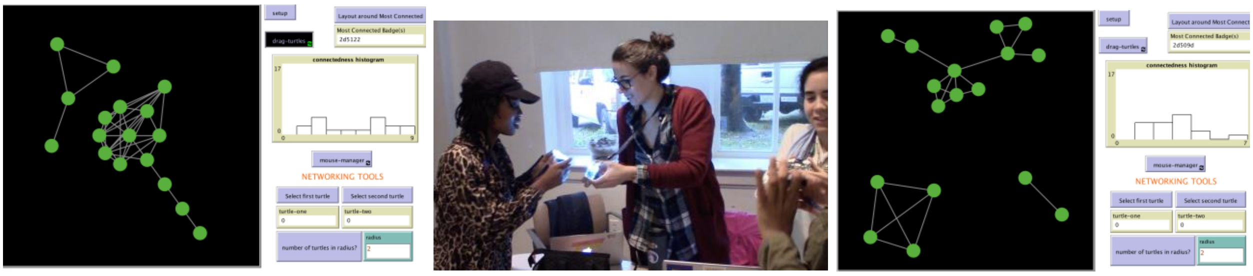


Figure 2. Creating different network structures. Left: two connected components; Right: three.

The first scenario produced a network with two connected components (corresponding to membership in distinct degree programs); the second, three (Fig 2). Running the spread of disease from a random node on the social-media defined network yielded the simulation below (Fig 3).

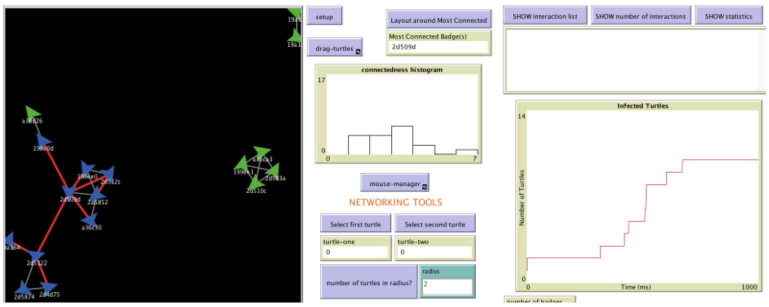


Figure 3. Disease spreading from a random node on the three-component network.

The badge-based activities provided a non-technical student group with a shared introductory experience with networking concepts and representations. Grounding the experience in familiar features of their lives (their programs of study and their social media use), they were able to make sense of concepts such as connectedness and network distance, as well as to appreciate the utility of these ideas in reasoning about disease spread. We argue that such activities can offer groups of learners shared experiences of connectedness and can create the need for key ideas in network theory. We further argue that the embodied nature of the simulations and the real-time visualization of the emerging network are valuable in making sense of node-link representations. Following on this successful pilot we are planning an implementation with high-school students of similar activities and activities with longer duration that unfold over entire days in the students' school environment.

References

Brady, C., Orton, K., Weintrop, D., Anton, G., Rodriguez, S., & Wilensky, U. (2016). All roads lead to computing: Making, participatory simulations, and social computing as pathways to computer science. *IEEE Transactions on Education*, *60*(1), 59-66.

Brady, C., Stroup, W., Petrosino, A., & Wilensky, U. (2018). Group-based Simulation and Modelling: Technology Supports for Social Constructionism. *Proc. Constructionism 2018 Conf*.

Brady, C., Weintrop, D., Anton, G., & Wilensky, U. (2016). Constructionist learning at the group level with programmable badges. In *Proc. Constructionism 2016 Conf* (pp. 61-68).

Brady, C., Weintrop, D., Gracey, K., Anton, G., & Wilensky, U. (2015, September). The CCL-Parallax Programmable Badge: Learning with Low-Cost, Communicative Wearable Computers. In *Proceedings of the 16th Annual Conference on Information Technology Education* (pp. 139-144). ACM.

Colella, V. et al. (1998). Participatory simulations: using computational objects to learn about dynamic systems. *Proceedings of SIGCHI 98,* New York: NY, 9–10

Head, B., Vermeer W., & Wilensky, U. (2018). The impact of structure of local networks on epidemic size in a multi-level system. *Complex Networks & Their Applications VI*

Klopfer, E., Yoon, S. & Perry, J. (2005). Using palm technology in participatory simulations of complex systems: A new take on ubiquitous and accessible mobile computing. *Journal of Science Education and Technology*. *14*(3), 285-297.

Vermeer, W., Head, B., & Wilensky, U. (2017). The effects of local network structure on disease spread in coupled networks. In Gaito S. Cherifi H., Quattrociocchi W. & Sala A (Ed.), *Complex Networks & Their Applications V: Proceedings of the 5th International Workshop on Complex Networks and their Applications* (pp. 487-498). Milan, Italy.

Wilensky, U. (1999). *NetLogo*. http://ccl.northwestern.edu/netlogo Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL

Wilensky, U. & Resnick, M. (1995). New thinking for new sciences: Constructionist approaches for exploring complexity. Paper presented at the AERA Annual Meeting, San Francisco.

Wilensky, U. & Stroup, W. (1999a). Learning through participatory simulations: network-based design for systems learning in classrooms. Proceedings of Computer Supported Collaborative Learning (CSCL'99). Stanford, CA.

Wilensky, U. & Stroup, W. (1999b). HubNet. http://ccl.northwestern.edu/netlogo/hubnet.html. Center for Connected Learning and Computer-Based Modeling, Northwestern University.

Wilensky, U. & Stroup, W. (2000). Networked gridlock: Students enacting complex dynamic phenomena with the HubNet architecture. In B. Fishman & S. O'Connor-Divelbiss (Eds.), Proceedings of the Fourth Annual International Conference for the Learning Sciences (pp. 282-289). Mahwah, NJ: Erlbaum.

Emerged Debugging Abilities in Early Childhood Education

Anastasia Misirli, *amisirli@upatras.gr*
Department of Educational Sciences and Early Childhood Education, University of Patras, Greece

Vassilis Komis, *komis@upatras.gr* Department of Educational Sciences and Early Childhood Education, University of Patras, Greece

Debugging process constitutes one of the key elements of Computational Thinking (CT) framework. A child/programmer within the CT will develop a cognitive process for algorithmic design, pattern recognition, decomposition and abstraction. In all these processes errors may occur and the child/programmer needs to detect and fix them, in other words needs to follow a debugging process. In the present research emerged debugging abilities has been investigated, for children in early childhood, when programming the Logo-like robot Bee-Bot. The main finding is emerged elements of syntax and semantic/logical knowledge since different categories of errors occurred (syntax and semantic/logical errors). A typology of debugging strategies emerged which may be useful to a further understanding of programming abilities related to debugging process.

Objectives - Learning activity design

The authors following that 'if we wanted to ensure a common and solid basis of understanding and applying computational thinking for all then the learning should be best done in the early years of childhood' (Wing, 2008, p.3720) set the following objectives of the study: a) to identify if debugging abilities are emerged using a Logo-like robot in early childhood education, b) to classify errors according to the main debugging categories and c) to organise a typology of debugging strategies as part of a broader conceptual programming model. The robotic environment chosen was the Logo-like robot Bee-Bot. A scenario-based teaching design (Komis, Romero & Misirli, 2017; Misirli & Komis 2014), was developed to teach key elements of computational thinking such as: algorithmic design, decomposition, and abstraction, in early childhood education (Wing, 2008). Debugging process was not included as such in learning activities rather were emerged through the programing knowledge each child developed. Learning took place in groups of fours, mixed in gender and age and always tried to balancing gender and age factors. In order to facilitate children's algorithmic planning we developed a visual representation ('pseudolanguage') of control and orientation-direction commands on cards. The 'pseudolanguage' worked as a medium for reflection on coding from the side of child/programmer. Syntax errors represented 'grammatical errors' when using a programming language. For the present programming structure that category of errors linked to control commands: i) "CLEAR" and (ii) "GO". Semantic and logical errors were the category of errors linked to location commands (orientation and direction).

Research context

Forty (40) educators, from mainstream public schools in Greece, being trained by the authors to implement programming activities (educational scenarios) into their classes (526 children aged 4-6 years old, boys and girls). Data collected from the algorithms (tasks) children/programmers set up in order to solve a problem as it was planned in each educational scenario. An example of the algorithm (task): each child/programmer was instructed to solve a problem how to move the robot in a specific place on a chi-squared mat. The task took place in privacy and separately of the others. The data analysis followed a qualitative model where the debugging process organised in six basic (06) variables ranged to twenty seven (27) descriptive categories. Through these variables we were able to capture and individualise the debugging route for each child and thus organise a typology of debugging strategies. Groups of sample, presenting similar debugging strategies, were matched and analysed applying qualitative factors analysis.

Results

Figure 1 shows the sequential programming structure (3 steps process: memory, movement(s), execution) of the Logo-like robotic environment linked to different categories of errors. The category of syntax errors led children/programmers to debug since control commands are involved for a program to run. For the logical errors category, most errors were described as lack of direction and orientation commands. A few cases characterized as semantic errors and related to the misconception of guidance commands. The range of qualitative categories brought to light that children/programmers are primarily developing strategies for debugging experienced programmers such as creating a new program (Sipitakiat & Nusen, 2012). Between the two ages, older children/programmers (5-6 year old) present an established programming behaviour where they recognise and identify syntax and semantic/logical errors and applying appropriate debugging strategy to fix them.

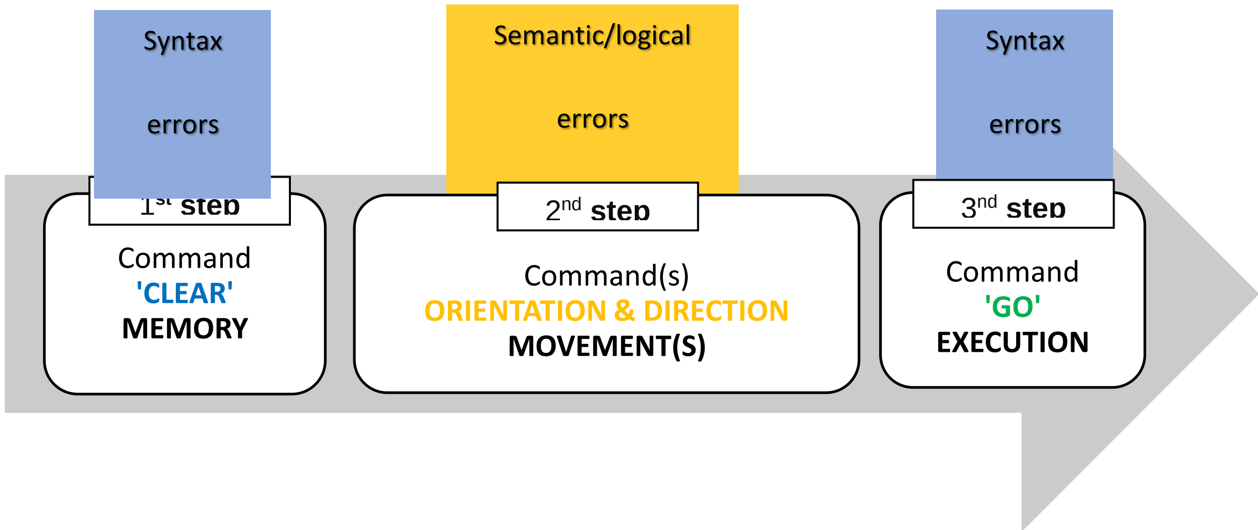


Figure 1: Types of errors in sequential programming structure with a Logo-like robot

Implications

The added value of the analysis was to capture a meta-cognitive and self-reflective process, which is mostly displayed in experienced programmers by organising emerged debugging abilities in typologies for each category of errors (syntax, semantic/logical) and for the overall debugging process. The number of sample provided a reliable context of analysis. Further work need to be done on a variety of robotic tools and the challenges and opportunities these provide (Wing, 2008), in order to evaluate the typologies of debugging process. Further study will help us to expand and improve our model of analysis and conceptual programming model (Komis, Romero & Misirli, 2017; Misirli & Komis 2014). The present knowledge may improve the teaching planning, regarding cognitive processes in programming, in future early childhood educators.

Acknowledgment

This work was funded by the European Commission through the FP7-SIS Framework (Grant agreement ID: 244684). Project: "The Fibonacci Project - Disseminating Inquiry-Based Science and Mathematics Education in Europe".

References

Komis, V., Romero, M., & Misirli, A. (2017). A scenario-based approach for designing educational robotics activities for co-creative problem solving. In D. Alimisis, M. Moro, E. Menegatti (Eds.) *Educational Robotics in the Makers Era* (pp 158-169). Springer. DOI: 10.1007/978-3-319-55553-9_12

Misirli, A. & Komis, V. (2014). Robotics and Programming Concepts in Early Childhood Education: A Conceptual Framework for Designing Educational Scenarios. In C. Karagiannidis, P. Politis, I. Karasavvidis (Eds), *Research on e-Learning and ICT in Education* (pp 99-11). Springer. ISBN: 978-1-4614-6501-0.

Sipitakiat, A. & Nusen, N. (2012). Robo-Blocks: Designing Debugging Abilities in a Tangible Programming System for Early Primary School Children. *Proceedings of the 11th International Conference on Interaction, Design and Children 2012, Bremen-Germany,* 98-105. https://doi.org/10.1145/2307096.2307108

Wing, J.(2008). Computational thinking and thinking about computing. *Philosophical Transactions of The Royal Society A Mathematical Physical and Engineering Sciences 366*(1881) 3717-25. DOI: 10.1098/rsta.2008.0118

Exploring Computational Thinking Practices through Collaborative Design Activities

Joey Huang, *joey.h@uci.edu*
Dept of Informatics, University of California, Irvine

Computational thinking (CT) entails a series of problem-solving processes, such as recognizing patterns, and systematically breaking down a problem, and then composing an algorithmic solution (Wing, 2006). This study focused on the two dimensions of CT learning: (1) Computational thinking concepts and (2) computational thinking practices (Brennan & Resnick, 2012). This work emphasizes the learning of CT through design activities in which a constructionist approach highlights the importance of students' interactions and engagement with design artifacts.

Collaborative design activities are defined to be a knowledge creation process which involves students actively communicating and working together to create a shared view of joint design ideas and decisions (Hakkarainen et al., 2013). Studies have shown that collaborative learning is beneficial for middle schoolers learning CT and programming knowledge, and these experiences relate to positive attitudes and confidence in learning computer science (CS) (Denner et al., 2014).

Prior studies have focused on individuals' CT practices and development in varied learning contexts. However, little attention has been paid to learning of CT through collaborative design activities, focusing on how CT is socially situated and practiced through collaboration (Chowdhury et al., 2018). This study aims to fill the gap of current research on CT and collaboration by investigating students' CT learning and social interactions through collaborative design from a constructionist approach. Particularly, this study is grounded in constructionist (Papert, 1991) perspectives on learning, which illuminate the impact of learning by creating, iterating, and interacting to investigate students' CT practices. In this study I ask: How do students learn CT practices through collaborative design activities?

The data were collected in a public middle school in a midwestern U.S. state. Students co-created Scratch projects and the processes were video recorded to examine their interaction and the process of CT practices. The data were collected as part of a five-week curriculum in an Introduction of CS class. The data included four triads groups, a total of 12 students. Groups were a mixture of novice to experienced students. The level of programming knowledge was determined based on pre-test results and the teacher's evaluations of students.

Preliminary coding results showed that patterns of CT practices emerged through the collaborative design processes. All four groups showed a higher number of experimenting and iterating in the planning stage while they were brainstorming the project. Particularly, the results showed that CT practices demonstrated different levels of complexity in different design phases (planning and coding). All four groups showed experimenting and iterating in both phases; however, compared to the experimenting and iterating practices through planning, students showed a deeper level of experimenting and iterating in coding. In the planning phase, students identified concepts of their project and developed a script to implement the design. In the coding phase, they were able to experiment and iterate their design by identifying the variables of the script and developing a plan to modify the variables. Excerpt one below showed how students negotiated and discussed the design and variables in the process of creating a storytelling project. They were trying to make the Scratch cat (the sprite named Mom in Figure 1) rotate consistently at certain speed. In the beginning of the discussion, they tried to figure out the coding variables to adjust the speed of rotation, and eventually they understood the function of variables and its relationship with rotation.

Excerpt I:

Chris: That's too fast! We shouldn't do 4.
Dan: I know what we should do. I know what that did. So what you want this random number to do. Before you pick (inaudible) bug what do you want this random number to do?

Chris: I need the score to be a random number.
Dan: Oh, so (.) I'm just going to copy this(.) note the random number will be changed every time.
Chris: Yeah I need it 15 through 18. And I know that it will be changed every time. That's what I want.
Dan: Well, no, it'll be constantly changed as the code(.)
Chris: Oh, crap.
Dan: It's fine. It'll only run it once eventually anyways...
Chris: Oh, yeah! And when I was doing that spinny thing -- as it got closer to 360 it slowed down(.)
Dan: Yeah, that's because it's just doing one complete rotation.

Figure 1. Screenshot of a Storytelling Project on Scratch

The results demonstrated students' CT practices through interactions. Constructionist learning approach allowed students to interact, negotiate, and reflect based on the Scratch projects, which facilitates CT practices through collaboration. This work contributes to the growing body of research on K-12 CS education with an emphasis on constructionist learning. I hope to extend the current scope of CT by providing an in-depth exploration of learning and collaboration for younger students. By bridging the framework of CT with collaborative design activities, our findings will enhance the understanding of CT in learning, collaborating, creating computing design projects.

References

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada* (Vol. 1, p. 25).

Chowdhury, B., Bart, A. C., & Kafura, D. (2018). Analysis of collaborative learning in a computational thinking class. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 143-148). ACM.

Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students?. *Journal of Research on Technology in Education*, *46*(3), 277-296.

Hakkarainen, K., Paavola, S., Kangas, K., Seitamaa-Hakkarainen, P. (2013). Sociocultural perspectives on collaborative learning: Toward collaborative knowledge creation. In C. E. Hmelo-Silver, C. A. Chinn, C. Chan, & A. M. O'Donnell (Eds.), *The international handbook of collaborative learning* (pp. 57-73). New York: Routledge.

Papert, S. (1991). Situating constructionism. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 1-14). Hillsdale, NJ: Lawrence Erlbaum Associates.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

Exploring sustainability issues of e-textiles with a co-creation game in ChoiCo

Chronis Kynigos, *kynigos@ppp.uoa.gr*
Educational Technology Lab, Department of Educational Studies, National and Kapodistrian University of Athens, Greece

Marianthi Grizioti, *mgriziot@ppp.uoa.gr*
Educational Technology Lab, Department of Educational Studies, National and Kapodistrian University of Athens, Greece

Christina Gkreka, *xristgreka@ppp.uoa.gr*
Educational Technology Lab, Department of Educational Studies, National and Kapodistrian University of Athens, Greece

Maria Daskolia, *mdaskol@eds.uoa.gr*
Environmental Education Lab, Department of Educational Studies, National and Kapodistrian University of Athens, Greece

Textiles is a multidisciplinary field with a broad variety of applications, that could motivate students from different specialities to get involved in participatory design and co-creation activities. Computational textiles (or e-textiles) have already been used in educational contexts to introduce students in computational and STEM concepts (Kafai et al, 2014, Buechley et al, 2013). Recently, the textile industry has been considered as one of the sectors that need to be synchronised with 2030 Agenda and the 17 Sustainable Development Goals of the EU (2019), considering sustainability issues of production and consumption of textile products (United Nations, 2015). A new challenge, therefore, is to integrate such issues in textile education. There is a need for new designs that would motivate students to ask questions and seek solutions about textile production in a realistic context, such as what material should be used, how sustainable would the product be, what is the usability of the product etc.

In this poster, we present a constructionist design that aims to foster students to investigate, discuss and express meanings on the needs and aspects of an e-textile production process, as they play and modify a choice-driven simulation game with ChoiCo game designer (Kynigos & Grizioti, 2020). The game, that is called “Sustainable Textiles”, simulates the production of a smart textile product and addresses socio-scientific issues that have no clear solution such as product sustainability. The student as a player makes choices related to the production of the smart textile such as fabric type (eg cotton, polyester etc), fabric structure and design (woven, knitted, etc), sensors (heartbeat sensor, light sensor, touch sensor etc), enhancements (flame resistance, UV protection, waterproof etc) and so on. Each choice has a positive or negative impact on several game attributes that represent product requirements and sustainability such as functionality, comfort, usability, cost, maintenance, resources, pollution etc. The aim is to keep the values of these attributes balanced without crossing specific limits (Figure 1). All choices have both positive and negative consequences, thus there is no ‘best’ or ‘worst’ solution, but a combination of selections that will keep the game, and therefore the textile production, balanced

Figure 1. Screenshot from the game area 'fabrics' of the demo version

The game mechanics and the game rules are designed so that they will trigger students to question, negotiate and modify the choices and their consequences. For the modification part, they will use the three integrated affordances of ChoiCo environment which are a) a map-based interface for setting the game areas and the available choices, b) a database for setting the game attributes and the consequences of each choice and c) block-based programming for creating the game flow and game rules. We aim to enable students to express their ideas on sustainability issues and generate meanings on the topic through construction, experimentation and collaboration. The game will be integrated as a learning resource into an online learning platform for supporting design thinking projects in schools and universities. We plan to implement a design-research with bachelor students of technical education with different expertise to investigate the meanings they would probably express and generate on e-textile and sustainability concepts when they play the game and modify its parameters.

Acknowledgements

The presented research has received funding from the European Commission in the context of the KA2 Erasmus+ project "T-CREPE (Textile Engineering for Co-Creation Paradigms in education)". Project No: 612641

References

Buechley, L., Peppler, K., Eisenberg, M., & Kafai, Y. (2013). *Textile Messages: Dispatches from the World of E-Textiles and Education. New Literacies and Digital Epistemologies. Volume 62*. Peter Lang Publishing Group. 29 Broadway 18th Floor, New York, NY 10006.

Eurostat., & Union européenne. Commission européenne. (2019). *Sustainable development in the European Union: Monitoring report on progress towards the SDGS in an EU context.* Publications office of the European Union. (https://ec.europa.eu/eurostat/web/products-statistical-books/-/KS-02-19-165)

Kynigos, C., & Grizioti, M. (2020). Modifying games with ChoiCo: Integrated affordances and engineered bugs for computational thinking. *British Journal of Educational Technology*.

Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., & Lui, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. ACM *Transactions on Computing Education* (TOCE), 14(*1*), 1.

United Nations. (2015). *Transforming our world: The 2030 agenda for sustainable development*. General Assembley 70 session.

Constructing Models of Local Segregation with Preservice Teachers

Max Sherard, *mksherard@utexas.edu*
Department of Curriculum and Instruction, The University of Texas at Austin, Austin, Texas, U.S.A.

Anthony Petrosino, *apetrosino@mail.smu.edu*
Department of Teaching and Learning, Southern Methodist University, Dallas, Texas, U.S.A.

Introduction

Segregation is can be considered a complex phenomenon which emerges from the interactions of many social actors (agents, or *de facto* causes of segregation) and actions of local, state, and federal governments (systems, or *de jure* causes of segregation). Agent-based models like NetLogo Web (Wilensky, 1999) have been used to scaffold attempts at uncovering the functions of complex systems (Wilensky & Jacobson, 2006). This study accounts for *how* preservice teachers make sense of complex and political phenomena when they engage together in constructing models of complex and political phenomena. Using constructionist perspective on learning, we hope to begin discussions on the following research question.

1. How do models constructed by preservice teachers vary in their representations of social actors and processes involved in segregation?

Learning Activity Design

The researchers planned a three-lesson series (Table 1) using the Model-based Learning (MbL) cycle (Louca & Zachariah, 2015). The MbL cycle follows four stages: (a) *observing and engaging* in some phenomena; (b) *using* a pre-created model; (c) *evaluating* the model; and (d) *revising* the model based on evidence or new experience.

Table 1
The three-lesson series, a short description, and the phases of the MbL cycle covered in each section (italicized).

Lesson	Description of Activities
1. Exploring the Initial Segregation Model	Participants *engaged* in discussing the phenomenon of segregation by sharing personal experiences and *observed* segregation by exploring maps which display racial settlement patterns. Participants *used* the NetLogo Segregation model (Wilensky, 1997) and *evaluated* and critiqued the model.
2. Walking Tour of Local Segregation	Participants read chapters from a book about the *de jure* aspects of segregation and *engaged* in a walking tour of campus guided by smart phones which discusses the history of segregation. This provided evidence to further *evaluate* and plan *revisions* to the model.
3. Revising the Segregation Model	Participants worked to *revise* the NetLogo segregation model by identifying sections of code to change, drawing a new hypothetical NetLogo model; and sharing their revisions with peers.

Participants, Data Collection, and Analysis

This research was performed with 15 preservice teachers enrolled in an elementary science methods course. The lesson series occurred over a total of nine hours (3 hours per lesson). Data was collected in the following forms: video recordings of the whole class, audio recordings of small group conversations, and photographs of the final revised segregation models. This poster focuses on describing the revisions participants made when constructing their final segregation model.

Example of the Results

The original NetLogo Segregation model deployed on the Group-based Cloud Computing (GbCC) platform operates by using a *de facto* explanation of segregation; meaning, segregation is a phenomenon that emerges strictly the actions of individual agents or turtles within a system. Preservice teachers critiqued the original model's emphasis on *de facto* mechanisms, specifically citing that the model did not account for systemic factors such as: historic red-lining, economic differences in families, property values, or school placement. Preservice teachers were tasked with modifying the segregation model to account for *de jure* factors in the segregation phenomenon. Below I offer one model revision (Figure 1) and describe how it differs from the original model.

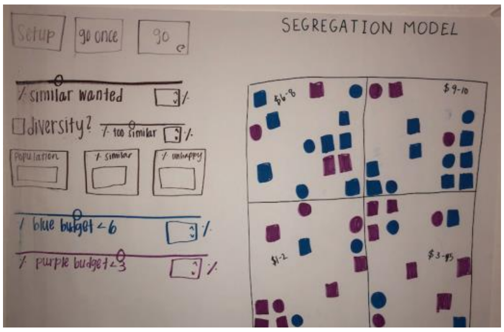


Figure 1. Example of a model revision

This group chose to preserve the original model's comparison between two types of turtles but selected the colours blue and purple to represent the two demographics of turtle. In the original model, both turtle types (red and green) are activated similarly; meaning, they possess the exact same ability to move patches if they are 'unhappy'. However, this group chose to activate each turtle type (purple, blue) differently depending on the turtle-types budget. The addition of a budget slider allowed the group to imagine a simulation where economic differences between the two demographics/turtles can be simulated. Furthermore, while the original model activated all patches similarly (meaning, each patch has the same probability of receiving a turtle); this group modified the model such that four types of patches exist with differing values, therefore restricting the motion of some turtles with lower budgets.

Implications

Examining the complex phenomenon of segregation using Agent-based Models provides a chance for preservice teachers to see multiple pedagogical tools at work, including: (a) integrated teaching and learning between the fields of computer science, social studies, and mathematics (Petrosino, Sherard, & Tharayil, in press); (b) model-based learning (Louca & Zachariah, 2015); and (c) developing critical consciousness (Ladson-Billings, 1995). While this only briefly summarizes one group's attempt to revise an agent-based model, further analysis of the data hopes to explain how preservice teachers collaborate to modify simulations of complex and political phenomenon.

References

Ladson-Billings, G. (1995). Toward a theory of culturally relevant pedagogy. *American Education Research Journal*. 32(3):465-491.

Louca, L. T., & Zacharia, Z. C. (2015). Examining learning through modelling in K-6 science education. *Journal of Science Education and Technology*. 24:192-215. doi:10.1007/s10956-014-9533-5.

Petrosino, A. J., Sherard, M. K. And Tharavil, S. A. (In press). The education philosophy, theories and models that enable STEM policy integration. In C. Zintgraff (Ed.), STEM in the Technopolis. Springer

Wilensky, U. & Jacobson, M. J. (2006). Complex systems and the learning sciences in R. K. Sawyer *The Cambridge Handbook of the Learning Sciences*. (pp. 319-333). Cambridge, U. K.: Cambridge University Press.

Wilensky, U. (1997). NetLogo Segregation model. http://ccl.northwestern.edu/netlogo/models/Segregation. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

MOON Board Game (CS Unplugged)

Witold Kranas, *witek.kranas@gmail.com*
OEIiZK (Computer Assisted Education and Information Technology Centre), Warsaw, Poland

Half century ago the Eagle lunar module was on its way to land on the moon, and it relied on the non-stop calculations of its on-board computer. MOON is an educational board game where players will simulate a simple computer, while learning how to: count in binary, perform logical operations, find out how a computer works. MOON is recommended for 10-year-old and older, for 1 to 4 players and has an estimated duration of 15-45 minutes. Game was developed by international team working in COMPUS project, sponsored by EU. All needed materials, cards and rulebook are free accessible on project site [compus.deusto.es/moon].

"*Designing the right story can be crucial for a project, a program or a promotion*" [Mönig Jens, 2018]. The story behind the game is Apollo 11 landing on the moon. The mission was supported by two twin computers designed at MIT (1966). The Lunar Guidance Computer (LGC) used a 16-bit word, was controlled by a clock with a basic frequency of 2048 kHz, had a mass of 32 kilograms, used a permanent ferrite memory with a capacity of 74 kB and a non-permanent memory 4 kB capacity. A few minutes before landing, astronauts were alerted by a computer message: error 1202, indicating a lack of free memory. Fortunately, the computer software designed by the MIT engineer team led by Margaret Hamilton was good enough to deal with this problem and focus all computing power on the landing manoeuvre. Steve Bales, a computer specialist at the flight centre, decided that astronauts could continue landing.

Figure 1. Game board at the beginning of a game.

We start the game by arranging the board. There are 4 registers: A, B, C, D. On the left are operations that can be performed on registers, on the right are energy resources and a stack of task cards. Processor registers have a number of bits that act as zero-one counters. Each item has a number assigned (1, 2, 4 and 8 in the 4-bit register). Available operations are: INC and DEC, consuming 2 units of energy, ROL, ROR, MOV, NOT, consuming 1 unit of energy, AND, OR and XOR, consuming half unit of energy. When we start the game, the A register is empty, and our goal is to place there the bits from the first task.

In easy version of the game we have 3 energy units. If you fail to complete the task with 3 energy units, you must take another task card and place the first face down under the current task. There are also task cards that do not have a bit combination, but contain ERROR. These special cards cannot be discarded and will block one of the positions for the rest of the game. If the task card is in the fifth position at the end of the round, our processor turned out to be too slow, the game is over and our lunar mission failed.

Figure 2. Task to be placed in register A.

You can also play in the competitive mode, in which each player solves the tasks, and the winner is the one who finally has the most solved problems after finishing the pile of tasks. It's also possible to play with 6-bit processor.

COMPUS team prepared game materials (Rulebook) in English, Spanish, Romanian, Polish and Basque. To gain additional founds the main developer Pablo Garaizar started a Kickstarter campaign successful in quick gathering demanded founds but also in translation of the materials to French, Portuguese, German, Catalan and Dutch [www.kickstarter.com/projects/garaizar/moon-0].

We organized several workshops for CS teachers to familiarize them with the MOON game. The main goal was to learn how to play the game. 24 participants completed a survey about their feelings during the game. Brief summary of the results:

The statements that were strongly (more than 70%, more than 5 on the 7 grade scale) supported

- I felt just the right amount of challenge during the activity.
- I did not notice time passing.
- I had no difficulty concentrating.
- I was totally absorbed in what I was doing.

The less supported (less than 30%, less than 3 on the 7 grade scale) statements are:

- I was completely lost in thought.
- I was worried about failing during the activities I had to perform..

Now we are developing supporting educational materials for the students. An example is Scratch project developed to support understanding of binary representation in MOON game [scratch.mit.edu/projects/316072601]. The next step will be dissemination of the game and educational materials at schools.

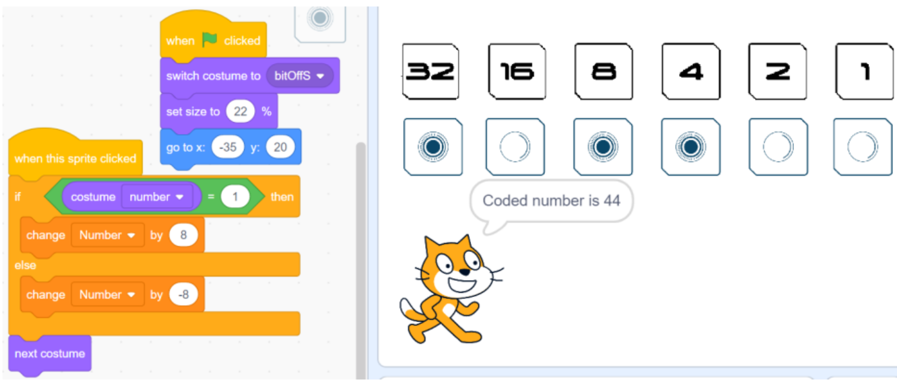


Figure 3. Scratch project helping students to understand binary representation.

References

Mönig, J. (2018) *Bones, Gears and Witchcraft.* In: Constructionism 2018, Vilnius, Conference Proceedings. Edited by Valentina Dagienė and Eglė Jasutė.

Natural Language Processing 4 All

Arthur Hjorth, *arthur@cs.au.dk,*
Department of Computer Science, Aarhus University

In this poster I present Natural Language Processing 4 All (NLP4All), a web-based text analysis system designed for use in classrooms, specifically high schools. NLP4All is designed to let students use Machine Learning approaches to explore language and words in corpora of texts, and to facilitate an explorative process of investigating similarities and differences between texts. To do so, NLP4All draws on concepts from the emerging field of Explainable AI – AI that is grey- or white-boxed so that users of ML understand how AI is making the decisions or classifications that it does (Gunning 2017; Holzinger et al. 2018). I present a a use case of NLP4All in which students use the system to explore how different political parties, specifically in a Danish context, use different language on Twitter.

Giving learners without programming skill the ability to use Machine Learning is gaining increasing attention (e.g. Kahn and Winters, 2018.; Hitron et al. 2019). The idea behind NLP4All is slightly different: to let leaners use Machine Learning to explore an existing school practice, namely text analysis, but using Machine Learning as an exploratory process.

A use-case to illustrate NLP4All

In collaboration with a high school civics teacher, we built an 6-hour classroom unit in which students used NLP4All to understand the different ways in which political parties in Denmark use language on Twitter. Text analysis, and specifically the “analysis of ideology and power in language” is part of the educational national standards for high school civics. However, teachers reports that text analysis is seen as easy, or less valuable by students because of its qualitative nature. Using NLP4All will help students quantify their analysis of political texts, through machine learning-assisted analyses of Tweets.

For this unit, 13 sets of approximately 10,000 tweets each have been downloaded from the official Twitter accounts of each of the 13 political parties in Denmark. The purpose of the learning activities is for students to train a “robot” – a Bayesian classifier – to be able to tell the difference between the various political parties by identifying the most strongly predictive words. The corpus of Tweets can be uploaded to NLP4All which stores it in its database. Each Twitter account is created as a “Category” of texts, and a teacher can now select some number of categories that students should work with, and create a “Project” for this (see Figure 5).

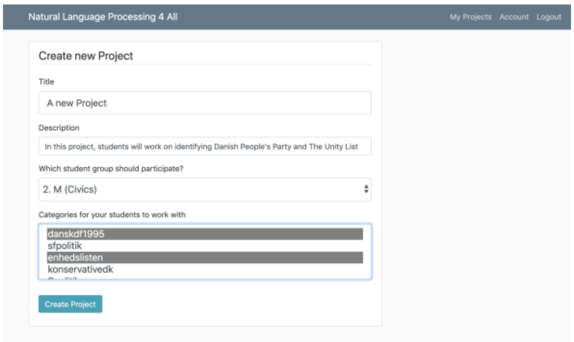


Figure 5: Creating a new student Project

This allows students to enter the Analysis page (Figure 6). The analysis page randomly samples a tweet from each of the political parties specified by the teacher in the Project, and students can now collaborate on “tagging” these tweets as having been authored by one of the parties. In the example below, tweets are selected either from the Unity List, a left-wing party, or from the Danish People's Party, a right-wing party.

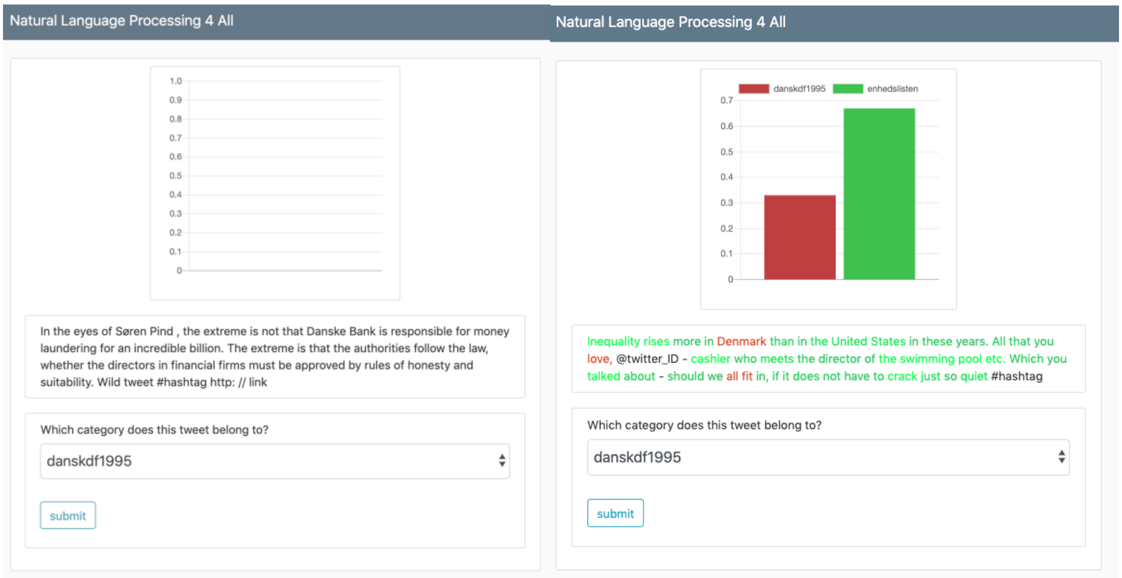


Figure 6: The Analysis Tool: On the left, a new analysis. On the right, an analysis that has already learned from students' tagging tweets. (Tweet is in Danish, but translated by Google.)

As students tag more and more tweets, their Bayesian Classifier learns how each word is predictive of one particular party. In Figure 6 above, we see how NLP4All helps students make predictions about the true underlying category of tweets by giving an overall category prediction (the bar chart). In this tweet, talking about increasing inequality in Denmark, NLP4All predicts with a .65 probability that the Tweet comes from the Unity List, the left-wing party. The colour of each of the different words in the tweet correspond to the party that is predicted in the bar chart. And even just with a few tweets tagged, we see that the word "Denmark" better predicts that the tweet comes from the Danish People's Party, whereas the word "Inequality" better predicts that the tweet comes from the Unity List.

By collaboratively tagging tweets and reflecting on why certain words or phrases predict particular parties, students reported that they had gained both a qualitative and a more quantitative understanding of how ideology and party politics are embedded in language. Further, students reported being more likely to see themselves working with Machine Learning either in their jobs, or as a hobby, a week after the unit had been completed.

References

Gunning, David. (2017). "Explainable Artificial Intelligence (Xai)." *Defense Advanced Research Projects Agency (DARPA), Nd Web* 2.

Hitron, Tom, Yoav Orlev, Iddo Wald, Ariel Shamir, Hadas Erel, and Oren Zuckerman. (2019). "Can Children Understand Machine Learning Concepts?: The Effect of Uncovering Black Boxes." In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 415. ACM.

Holzinger, Andreas, Peter Kieseberg, Edgar Weippl, and A Min Tjoa. (2018). "Current Advances, Trends and Challenges of Machine Learning and Knowledge Extraction: From Machine Learning to Explainable Ai." In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, 1–8. Springer.

Kahn, Ken, and Niall Winters. (2018). "AI Programming by Children." In proceedings of the Constructionism 2018 Conference. Vilnius, Latvia.

Pomelo, a Collaborative Education Technology Interaction Robot

Ece Tabag, *ece.tabag@hisarschool.k12.tr*
C.S. IdeaLab, Hisar School, Istanbul, Turkey

Yoel Nasi, *yoel.nasi@hisarschool.k12.tr*
C.S. IdeaLab, Hisar School, Istanbul, Turkey

Can Aydın, *can.aydin@hisarschool.k12.tr*
C.S. IdeaLab, Hisar School, Istanbul, Turkey

Andy Emre Kocak, *emre.kocak@hisarschool.k12.tr*
C.S. IdeaLab, Hisar School, Istanbul, Turkey

Rana Taki, *rana.taki@hisarschool.k12.tr*
C.S. IdeaLab, Hisar School, Istanbul, Turkey

Batuhan Bayraktar, *batuhan.bayraktar@hisarschool.k12.tr*
C.S. IdeaLab, Hisar School, Istanbul, Turkey

Sedat Yalcin, *sedat.yalcin@hisarschool.k12.tr*
C.S. IdeaLab, Hisar School, Istanbul, Turkey

Pomelo is an educational robot that can be programmed through psychical code blocks and can move in desired patterns. Pomelo is suitable for elementary students between the ages of 4 to 7. Its aim is to teach coding and algorithmic thinking to younger kids in a simple and entertaining way.

There are many different coding platforms for kids to learn basic programming but most of them are in a computer environment that may create an isolated setting when they are getting started with programming which might discourage some students from learning. This approach is not suitable for younger ages since they are more interested in playing, rather than coding on a abstract platform. Pomelo, however, counteracts this as it is a physical robot that can be coded with simple blocks.

Pomelo consists of two main parts which are the code blocks and the robot. Our main goal when designing Pomelo was to create an environment for people to learn the basics of programming at a young age. To accomplish this, we implemented physical code blocks to program Pomelo so that even elementary school students can give instructions without difficulties. The icons on the code blocks make it easy to program Pomelo for even students that can't quite read. The code blocks are similar to the“ABC” blocks that children play within elementary schools. These blocks contain several commands such as move forward, turn left and turn right. When kids stack these code blocks, Pomelo reads the instructions from top to bottom and moves accordingly. This allows kids to learn how programming works with fun activities like games or puzzles. We imagined Pomelo being used in a classroom environment where the teacher creates puzzles for students to solve together using Pomelo. These kinds of assignments will encourage collaboration between students as well as teach them essential skills like problem-solving and critical thinking.

Fig.1 - A sample side of a code block

The shell of the prototype has a friendly dog shape. The natural and amiable appearance improves the interaction with Pomelo and the kids. Before designing and 3D printing the shell, we showed different designs to elementary school students and a majority of students said that they would prefer to play with the dog-like design. To make the interaction less mechanical and more organic, Pomelo has eyes on the screen at the front of it that changes emotions accordingly to the interaction. An LCD screen displays Pomelo's eyes that were drawn and rendered using the Maya 2018 software.

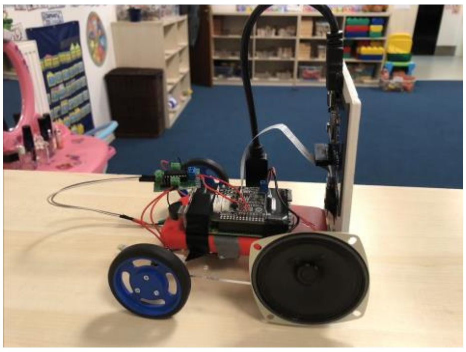


Fig.2 - 1st Prototype of Pomelo

Fig.3 - The Most Recent Prototype of Pomelo

By making a physical toy-robot to teach algorithmic thinking, collaborative thinking, and problem solving skills, we wanted to enable students to collaborate together to solve problems like mazes and puzzles rather than working on the problem on their own. The main goal for this project was to create a robot that elementary school students can play and learn to code in a way that would make them interested in the STEM fields. We believe that learning to think like a programmer is not only beneficial to the future programmer but it is beneficial for everyone since skills like problem-solving and analytical thinking are essential for everybody.

References

1. Burke, J., Murphy, R., Rogers, E., Lumelsky, V., & Scholtz, J. (2004). Final Report for the DARPA/NSF Interdisciplinary Study on Human–Robot Interaction. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), 34(2), 103–112. doi: 10.1109/tsmcc.2004.826287

2. Hiraki, K. (1995). Machine learning in human-robot interaction. Advances in Human Factors/Ergonomics Symbiosis of Human and Artifact - Future Computing and Design for Human-Computer Interaction, Proceedings of the Sixth International Conference on Human-Computer Interaction, (HCI International 95), 1127–1132. doi: 10.1016/s0921-2647(06)80178-4

Primary students' meanings around spatial and geometry concepts formed with the joint use of a programmable simulator and a robotics CNC Drawing Machine

Chronis Kynigos, *kynigos@ppp.uoa.gr*
Dept of Educational Studies, National and Kapodistrian University of Athens, Greece

Christina Gkreka, *xristgreka@ppp.uoa.gr*
Dept of Educational Studies, National and Kapodistrian University of Athens, Greece

Marianthi Grizioti, *mgriziot@ppp.uoa.gr*
Dept of Educational Studies, National and Kapodistrian University of Athens, Greece

Given that spatial ability consists an important part of geometry learning, students' difficulties have been attributed to the emphasis on the axiomatic, proof-oriented teaching and learning approaches over the exploitation of the spatial aspects of the subject (Clements & Battista, 1992). Meanwhile, conceptual difficulties are related to the kind and number of representations used in classrooms (Mesquita, 1998). While, static images of two dimensions fail to help students come to the right conclusions about 3d objects and their properties (McClintock, Jiang & July, 2002) and the use of multiple representations supports them in fostering their understanding (Ainsworth, 2006).

This design based research aims to study the kind of spatial skills that learners develop, using different tools, thus interacting with different representations of the same mathematical object in a way that supports their conceptualization about geometry concepts.

During the implementation of the activity plan which was designed, students create physical models of temporary houses simulating the practices of professionals who design and build container houses. Firstly they design virtual models, using MaLT2 (Machine Lab Turtle-sphere2) (http://etl.ppp.uoa.gr/malt2). MaLT2 is a Logo-based programming environment which integrates dynamic manipulation of students' digital constructions and periscopic view of these constructions provoking mathematical inquiry, thus supporting mathematical meaning-making (Kynigos & Grizioti, 2018). During the second phase students create a robotic CNC Drawer Machine similar to the real machine which cuts solids' nets. Finally they program their robot to draw cube and other solids' nets on the paper (Figure 1). These drawings are cut and tiled in order to create physical models of container houses. Educational design was theoretically supported by constructionist principles as during the activities students' meaning generation is grounded on their participation in the construction of their own artefact.

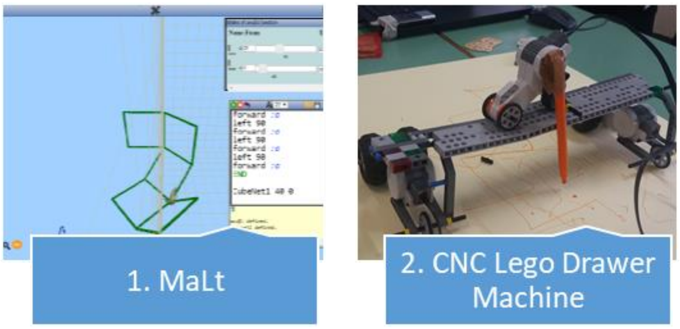


Figure 7 Students create virtual models by orienting an entity in 3D space, using MaLT2 (Machine Lab Turtle-Shere2) and a CNC Lego Drawer Machine that simulates the operation of a real CNC Cutter.

The workshop was implemented in a public primary school of Athens, Greece as a part of ER4STEM research project (Lammer et al, 2017). The participants were 18 11year old students of the 6th grade, who worked in groups of 4 during two 2-hour sessions and one 3-hour session. Students had prior knowledge of dimensional geometrical shapes and their properties, two of them had educational robotics workshops in the past and four of them had programmed with block programing software. Data was collected through recordings of students' discussions, screen recording files, students' digital, robotic artifacts and final solid models. Analyses was based on the identification of "critical incidents" (Cope & Watts, 2000), which took place during the implementation.

The results of the research show that students implemented spatial orientation and spatial visualization skills during their interaction with both tools. These skills supported processes of problem solving and meaning generation. The use of different tools triggered representational interpretation procedures. The orientation of the MaLT2 entity and the physical characteristics of the artefact helped students develop dynamic, kinaesthetic, concrete and pattern imagery (Presmeg, 1986). The development of spatial skills favored meaning making concerning cube nets and solids properties.

Future research could study the development of spatial skills through the combination of digital and physical representations at other ages and the design of activities that encourage the use of different tools by students in order to investigate the type of representations the combination of which could help students conceptualize mathematical concepts.

References

Ainsworth, S. (2006). *DeFT: A conceptual framework for considering learning with multiple representations.* Learning and instruction, 16(3), 183-198.

Clements, D. H., & Battista, M. T. (1992). Geometry and spatial reasoning. *Handbook of research on mathematics teaching and learning*, 420-464.

Cope, J., & Watts, G. (2000). *Learning by doing–an exploration of experience, critical incidents and reflection in entrepreneurial learning.* International Journal of Entrepreneurial Behaviour & Research.

Kynigos, C., & Grizioti, M. (2018). *Programming Approaches to Computational Thinking: Integrating Turtle Geometry, Dynamic Manipulation and 3D Space.* Informatics in Education, 17(2), 321-340.

Lammer, L., Lepuschitz, W., Kynigos, C., Giuliano, A., & Girvan, C. (2017). ER4STEM Educational Robotics for Science, Technology, Engineering and Mathematics. In *Robotics in Education* (pp. 95-101). Springer, Cham.

McClintock, E., Jiang, Z., & July, R. (2002). *Students' Development of Three-Dimensional Visualization in the Geometer's Sketchpad Environment*.

Mesquita, A. L. (1998*). On conceptual obstacles linked with external representation in geometry.* The Journal of Mathematical Behaviour, 17(2), 183-195.

Pitta-Pantazi, D., & Christou, C. (2010). *Spatial versus object visualisation: The case of mathematical understanding in three-dimensional arrays of cubes and nets.* International Journal of Educational Research, 49(2-3), 102-114.

Presmeg, N. C. (1986). *Visualisation in high school mathematics for the learning of mathematics*, 6(3), 42-46.

Digital Learning Club

Mayte Gonzalez *gonzalmt@tcd.ie*
Director Digital Learning Club, Mediaskool, Gorey, Co Wexford, Ireland

The problem we are trying to address is the digital divide among the 45+ learner cohort in a rural location. The underpinning theories are Constructionism and Sociocultural Constructionism. The digital learning club (DLC) is a free service that delivers open, friendly and casual mentorship on the use of technology to adults. It usually appeals to people who do not feel confident in the use of IT and who need help in a relaxed and friendly environment. It provides support for people who want to learn at their own pace. The DLC has been piloted for 26 weeks. It has attracted over 140 members to date between learners and mentors with a regular cohort of 40 cycling through weekly sessions.

A recent study in the Netherlands found that there are three levels of digital divide. The first one is access to hardware, the second one is attitude, (still relevant after twenty years) and most significantly, the impact of social setting and infrastructure (for example a relative with a printer who can help). Importantly the absence of this support network has a particularly negative impact (Van Deursen 2019). Pinkett (2000) defines Sociocultural Constructionism as a theory ‘about individual and community development that can inform efforts to engage populations traditionally underserved by technology’. It states that ‘achieving a certain level of social and cultural resonance is critically important to any effort that seeks to engage populations with computers and the Internet that have not traditionally enjoyed the benefits of these technologies’. Every individual has a particular cognitive potential with Vygotsky’s studies (Vygotsky,1980) highlighting the critical role played by cultural artefacts - tools, language, people - to achieve this (Ackerman, 2001).

Constructionism informs the learning activity; ‘Constructionism - the N word as opposed to the V word - shares constructivism's connotation of learning as "building knowledge structures" irrespective of the circumstances of the learning. It then adds the idea that “this happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe”’ (Papert, 1991). Papert’s research (1980) suggests than ‘when people “dive into” situations they experience a connectedness that aids in gaining a deeper understanding of such a situation. Their world view can change once they experience alternative points of view. This helps the individual to expand his or her knowledge. This knowledge is under construction (Ackerman, 2001).

The objective of DLC is to extend digital learning to educational cohorts who frequently find themselves left behind and left out of traditional education presentations. Volunteer mentors are an integral part of our process. Their work is to help people to articulate their query, answer it successfully and encourage them to learn more about their device and hence the technology behind it. Learners are encouraged to help each other depending on learner’s previous knowledge and current skills. Another objective is for learners to become mentors thereby demonstrating embedded learning.

For several years, a one to one teaching method was used to help people to learn and to express their concerns regarding digital knowledge. Initially, it was considered that a confidential approach was better for the person to open up and ask ‘stupid questions’. Three strong themes emerged: ‘the family is not patient enough’, ‘there is no-one to help’, ‘I must be the only one that doesn’t know how to do this’. It was decided to try a different approach, thus the social environment in which people can discuss their experiences making them feel less isolated. They can relate to what the other person has experienced, and their questions, in many instances, are relevant to all. Participants get consciously engaged building artefacts e.g.

text messages, emails with attachments or taking and sending photos. When they are ready and confident they share the artifacts socially so that others can see. These are the principles of constructivism according to our understanding.

DLC has a hands-on approach and is learner-centred. Participants learn to complete measurable tasks. There is a focus on self-efficacy and self-direction as learners define their objectives. In collaboration with mentors they decide what is important to them. We have formulated our methods over several years of observations, although it is a work-in-progress. The DLC is an informal way of getting instruction on the use of technology (figure 2). Preliminary results are positive based on attendance and immediate feedback, e.g. “aha” moments. The programme can be seen as a proof of concept that this is a model that has the potential to be replicated across multiple target communities.

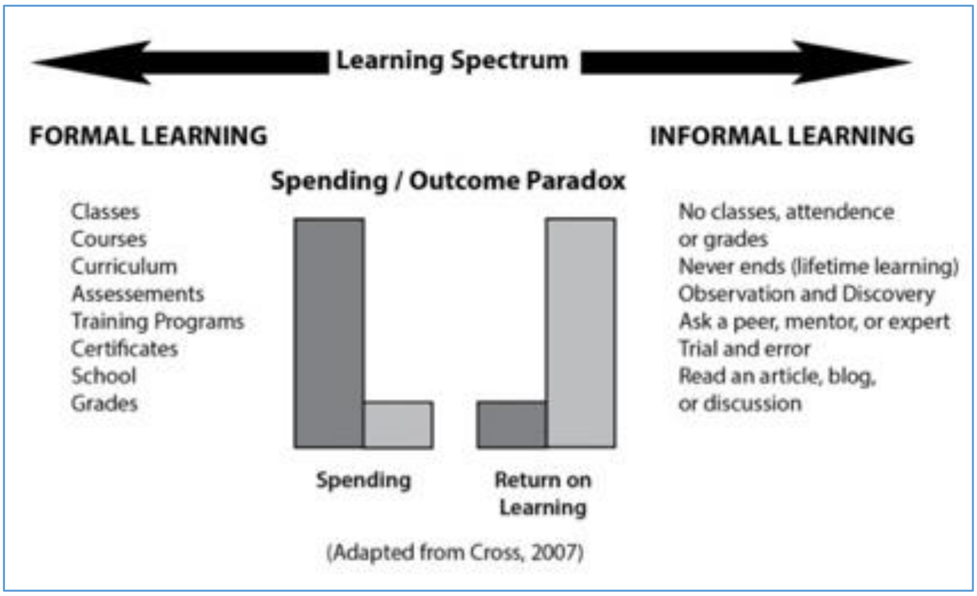


Figure 1. Informal learning: Rediscovering the natural pathways that inspire innovation and performance from Cross, J. (2007).

In response to current thinking, the learning space is laid out in a way to encourage interaction and shared learning. is beneficial for adult learners, especially if any have bad associations with previous education in their personal experience (figure 1).

Figure2. Room Layout Digital Learning club 2019

Positive results, voiced by participants, suggest that an informal educational approach is advantageous in particular with non-traditional education cohorts who might otherwise self-select out of the educational opportunity. This further indicates the need to accommodate a broad range of educational structures and in particular to privilege ‘learning by doing’.

Based on verbal feedback by library staff, recorded verbal feedback and feedback paper forms by participants we conclude that the learning experience, with the design informed by the constructionism theory of learning, works well in an informal environment with certain cohorts (albeit without a longitudinal and rigorous research methodology). In our experience, we see that a similar approach could give satisfactory results in a number of situations where adult learners 45+ are concerned since they share similar cultural and life experiences regarding the learning and use of modern digital technologies.

References

Ackermann, E. (2001). *Piaget’s constructivism, Papert’s constructionism: What’s the difference.* Future of learning group publication, 5(3), 438.

Cross, J. (2007). *Informal learning: Rediscovering the natural pathways that inspire innovation and performance*. San Francisco, CA, US: Pfeiffer/John Wiley & Sons. [accessed 14 Oct 2019] https://lidtfoundations.pressbooks.com/chapter/informal-learning-by-boileau/

Papert, S., & Harel, I. (1991). *Situating constructionism*. Constructionism, 36(2), 1-11.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc..

Pinkett, R. D. (2000, April). *Bridging the digital divide: Sociocultural constructionism and an asset-based approach to community technology and community building*. In *81st Annual Meeting of the American Educational Research Association (AERA), New Orleans, LA* (pp. 24-28).

Van Deursen, A. J., & van Dijk, J. A. (2019). *The first-level digital divide shifts from inequalities in physical access to inequalities in material access*. New media & society, 21(2), 354-375.

Vygotsky, L. S. (1980). Mind in society: *The development of higher psychological processes*. Harvard university press.

The use of GeoGebra in Brazil from a constructionist perspective

Fernando Carnaúba, *fa2529@tc.columbia.edu*
Teachers College, Columbia University, New York, USA

Paulo Blikstein, *paulob@tc.columbia.edu*
Teachers College and (by affiliation) Computer Science, Columbia University, New York, USA

Introduction

GeoGebra is one of the most widespread software packages for geometry, with more than 100 million users (GeoGebra, 2019). There is growing evidence that the software brings positive learning outcomes (Arbain and Shukor, 2015, Sinclair *et al*., 2016). Recent quantitative meta-analytic studies on the effects of GeoGebra have also shown promising results (Chan and Leung, 2014). Much less work has been devoted, however, to examine *how* this software is being used. GeoGebra is a flexible tool that allows teachers to design learning experiences that are very different in nature. On one extreme, teachers can design closed-ended mathematics tasks that resemble paper worksheets. At the other end of the spectrum, teachers can use GeoGebra as an open-ended environment for students to create mathematical artifacts of their own. We refer to this as a *constructionist* use of the software, in the sense that it intends to help students "understand something by making an artifact for and with other people that, to be built, requires the builders to use that understanding" (Holbert, Berland, and Kafai, 2019). The goal of this study is to understand to which degree GeoGebra is being used by Brazilian educators in a constructionist manner. More specifically, we ask the following question: *to what extent is GeoGebra being used by mathematics teachers as a constructionist learning environment?*

Methodology

Given the scope of this poster, this is more of a methodological exploration. In order to investigate the nature of GeoGebra activities from this constructionist perspective, we created a simple task taxonomy intended to map the students' degree of "control over the computer" in the proposed tasks. One end of the spectrum (Level 1) starts with no student interaction with the software, while the opposite end (Level 4) captures constructionist activities in which students are able to create their own mathematical representations.

- *Level 1 - Teacher-led use of software*. Students do not interact with the software, as Geogebra is used only by the teacher as a visual resource that complements a lecture (ex.: teacher plots graphs of functions using the overhead projector.)
- *Level 2 - Student step-by-step implementation of a software routine*. Students implement a routine in GeoGebra following step-by-step guides. There is a "right answer" for what students do. (ex.: students follow a list of procedures to create visualizations that illustrate a problem.)
- *Level 3 - Student open-ended use of software to solve a mathematical task*. Students use GeoGebra as a tool to solve a mathematical task or illustrate a theorem. There is no "right answer" for what students do (ex.: students use the graphing tool as a resource to solve a problem on linear equations.)
- *Level 4 - Student mathematical creation*. Students create a mathematical object of their choice using GeoGebra (ex. students create a visualization for a theorem/mathematical property of their choice.)

Data

Our data source are masters' theses in mathematics education that used GeoGebra as their tool of choice. We analyzed thesis defended under the program Mestrado Profissional em Matemática em Rede Nacional (Profmat) between 2013 and 2019 in Brazil. Profmat is a network master's program in which all affiliated universities offer the same curriculum and undergo a centralized evaluation process. Profmat was developed specifically for secondary mathematics teachers, most of whom undertake their coursework and research alongside teaching jobs. As a professional master's program, Profmat allows graduate students to write an action research piece as their final graduation project. This context thus provides us with a singular database that lies on the interface between research and practice in mathematics education in Brazil. It also comprises a fairly large sample, with 4,865 theses submitted and accepted between 2013 and October 2019, spread across more than 100 different universities from all Brazilian states (Profmat, 2019). Using the search mechanism in Profmat's thesis repository, we found 291 works (6%) that included the word "GeoGebra" in their titles.

Preliminary results

Out of these 291 works, we selected the 15 most recent for our analysis. We noticed a wide range in the nature of Geogebra use, with at least one thesis in each of the proposed categories. Six proposals provided students with a step-by-step implementation of a routine designed by the teacher and were classified at Level 2. Slightly in the direction of more open-endedness, four activities were classified at Level 3, in which students explore a given applet in an unstructured manner, with the goal of solving a mathematical problem. We also identified activities in each of the extremes of our taxonomy - one was designed to be used only by the teacher (Level 1), and one allowed students to develop their own mathematical objects (Level 4). Three theses did not provide clear suggestions and were not coded. Our goal here was to provide a proof of existence on the viability of this type of analysis into a very promising dataset - one that could reveal, ultimately, how educational software ends up being used in real classrooms. We were able to fit a small sample of 15 theses into a set of discrete categories of "open-endedness," showing that these works indeed greatly differ in their usage of the software. This study could be useful to understand how a tool that has both constructionist and instructionist affordances is used, and guide further analysis into the data. Ultimately, we hope to shed light into the usage patterns of other tools as well and reveal the types of implementations in existence.

References

Arbain, N., &. Shukor, N. A. (2015). The effects of GeoGebra on students' achievement. *Procedia - Social and Behavioral Sciences*, 172, 208-214.

Chan, K. K., & Leung, S. W. (2014). Dynamic geometry software improves mathematical achievement: systematic review and meta-analysis. *Journal of Educational Computing Research*, *51*(3), 311–325.

GeoGebra. *What is GeoGebra?* Retrieved November 17th, 2019, from *www.geogebra.org/about.*

Holbert, N., Berland, M., & Kafai, Y. 50 Years of Constructionism (2019). In N. Holbert, M. Berland, & Y. Kafai (Eds.), *Constructionism in Context: The Art, Theory, and Practice of Learning Designs*. Cambridge, MA: MIT Press.

PROFMAT (2019). *Apresentação sobre o programa*. Retrieved November 17th, 2019, from http://www.profmat-sbm.org.br/organizacao/apresentacao/.

Sinclair, N., Bartolini Bussi, M.G., de Villiers, M. et al. (2016). Recent research on geometry education: an ICME-13 survey team report. *ZDM Mathematics Education*, 48(50), 691-719.

Using Epistemological Pluralism to Explore Math in Weaving

Naomi Thompson, *naothomp@indiana.edu*
Learning Sciences, Indiana University, Bloomington, Indiana, USA

The constructionist and feminist lens of epistemological pluralism (Turkle & Papert, 1990), or validating diverse ways of thinking and knowing, can help educators and researchers consider more carefully how diverse mathematical practices can be recognized and valued. This involves, in part, giving learners space to solve problems in playful and unconventional ways, use playful and unconventional tools, or even to question the nature of math. This inclusive approach has potential to reaffirm youth practices as valid, allowing diverse experiences to be valued in STEM spaces. Weaving, with deep roots in innovations by women (Plant, 1995) has been shown to be highly mathematical in certain cultural contexts (e.g., Saxe & Gearhart, 1990), yet work has not explored the mathematical nature of weaving in schools. There is also a rich material and historical link between mathematics and weaving, including the invention the Jacquard loom (Essinger, 2004). Providing space for learners to participate in math in playful and hands-on ways may also work toward affirming student dignity in learning spaces (Espinoza & Vossoughi, 2014) as it positions multiple student practices as valuable and valid. The specific link between play, weaving, and emergent diverse mathematical practices is not yet known. I ask, how does playing through weaving support middle school youth in engaging with mathematics?

Methods I implemented a 6-session weaving unit with middle school youth in a project-focused school in the Midwestern United States. Thirteen youth signed up to do weaving activities with me during their "design studio" time. I prompted the youth to consider the mathematics in their weaving, but math content was not explicitly taught. In these sessions, youth used frame looms that were laser-cut from an open-source pattern to create their woven artifacts, and grid paper to plan their designs. Due to the accessible nature of the looms, youth were able to keep their looms and continue work on their projects after the end of the workshop. Following a constructionist ethic, the unit was open-ended and allowed youth to design and solve personal problems. The six sessions were designed to provide time for the youth to steadily work on their projects while also prompting them to engage with mathematical ideas through explaining their work and showing their processes on paper. I captured video/audio using four cameras. I also periodically checked in with individual youth asking prompting questions such as "What are you working on? Did you have to use any math to solve this issue?" Additionally, I took photos of woven artifacts and drawn designs produced. I used techniques inspired by mediated discourse analysis (Wohlwend 2014) to pay attention to movements, actions, and talk, in addition to looking for evidence of math in the artifacts themselves.

Findings In general, it was clear that youth became more adventurous and playful with their designs and artifacts throughout. As youth progressed, 30% of participants shifted toward either planning or implementing new "over, under" sequences. This suggests increased reasoning around seeking, understanding, and building underlying mathematical patterns. This is also evidence of mathematical engagement as Common Core Math Practice Standard 8 (2010) calls for learners to "Look for and express regularity in repeated reasoning." An example can be seen in Figure 1, A where the youth adopted an "over two, under three" sequence that resulted in a horizontal chevron pattern. Additionally, 76% of participants shifted their practice to experiment with embedding shapes into their designs. This suggests increased reasoning around considering multiple dimensions of measurement, shape, and proportion in concert as well as increased invention of unknown measurements such as slope and area, particularly by building curved lines and shapes into the inherently gridded structure. This is evidence of mathematical engagement as Common Core Math Practice Standard 7 (2010) calls for learners to "Look for and make use of structure." An example of this can be seen in Figure 1, B where the youth tried different methods to create a feeling of moving water. In both examples, the youth move above and beyond the standards by inventing and playing with the concepts and discovering new results.

Case: Marg For Marg's final project, she decided to recreate the Human Rights Campaign flag: a royal blue background with an embedded yellow equal sign in the center. She planned her design on grid paper, taking care to ensure the equal sign was centered and symmetrical. She also colored in every other square for the first two rows to indicate she planned to use an over one, under one pattern (Figure 1, C). Yellow yarn was not available on Day 4. Marg worked on the blue rows to the point where she wanted to add the yellow. The yarn was still not available on Day 5, and I encouraged her to think about another way she could continue her project. Through discussion, Marg decided to continue weaving with the blue, skipping the area where the yellow would be and bringing the blue thread behind the strings in this area. This strategy required Marg to hold the imagined shape of the equal sign in her mind while she built the blue structure around the empty space, a complex spatial visualization task. On Day 6, the yellow yarn was brought back, and Marg got to work, moving back and forth between the two colors (Figure 1, D).

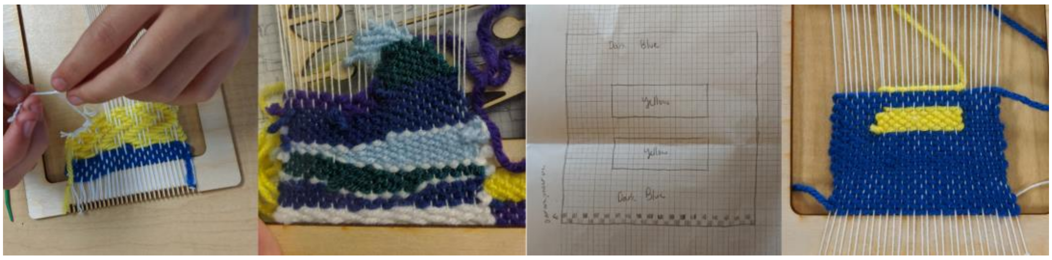


Figure 1. A: Chevron pattern; B: Moving water; C: HRC design; D: HRC artifact

Implications The playful and experimental approach to mathematics that was made possible through weaving allowed Marg and her classmates to follow their passions, experiment with mathematics principles and practices in meaningful and interest-driven ways, and experience a new way of participating in mathematics. Through a lens of epistemological pluralism, we can recognize knowing and doing in mathematics in broader and more equitable ways. When we understand that math happens in design and play, and may look different for learners from different backgrounds, experiences, and epistemologies, we can begin to adjust our frameworks to better value youth's intellectual work.

References

Espinoza, M. L., & Vossoughi, S. (2014). Perceiving learning anew: Social interaction, dignity, and educational rights. *Harvard Educational Review*, *84*(3), 285-313.

Essinger, J. (2004). *Jacquard's web: how a hand-loom led to the birth of the information age*. Oxford: Oxford University Press.

National Governors Association Center for Best Practices, Council of Chief State School Officers (2010). *Common Core State Math Standards*. National Governors Association Center for Best Practices, Council of Chief State School Officers, Washington D.C.

Plant, S. (1995). The future looms: weaving women and cybernetics. *Body & Society*, *1*(3–4), 45–64.

Saxe, G. B., & Gearhart, M. (1990). A developmental analysis of everyday topology in unschooled straw weavers. *British Journal of Developmental Psychology*, *8*, 251-258

Turkle, S., & Papert, S. (1990). Epistemological pluralism: Styles and voices within the computer culture. *Signs: Journal of women in culture and society*, *16*(1), 128-157.

Wohlwend, K. E. (2014). Mediated discourse analysis: Tracking discourse in action. In P. Albers, T. Holbrook, & A. S. Flint, (Eds.), *New methods in literacy research*. New York: Routledge.

πππgraf: A device to read charts for the visually impaired

Pavel Petrovič, *pavel.petrovic@fmph.uniba.sk*
Dept of Applied Informatics, Comenius University, Bratislava, Slovakia

Pupils in the joint school for visually impaired in Bratislava learn physics in practical exercises. In particular, they are using the Pro'sKit MT1820 multimeter with a temperature probe to make hands-on experiments with heat exchange. As part of these experiments they record data and visualize them in tables and charts. Following up on a previous cooperation [1], students at the Faculty of Mathematics, Physics and Informatics cooperate with the physics teacher from the joint school through their project on Information Systems Development course [2]. Their task is to develop a program that communicates with the multimeter, collects the data and shows them on PC display so that they are easily visible for those that can see and accessible for a screen reader for the others. However, the screen reader is not capable of reading line charts.

We have developed a device that detects the movements of the slider potentiometer and sends its position to the PC. The software on the PC allows drawing charts. It reads the position of the potentiometer as the x-coordinate in the chart. It uses the sound output to signalize the position of the line chart on the y-axis. In this way, the visually impaired or blind student can "listen" to the chart curve.

Apart from using in the heat exchange experiments with recording a line chart of temperature, this device has a broader potential. Pupils in usual schools are using wonderful tools, such as [5] to learn mathematics. The purpose of πππgraf is to improve the experience of visually impaired pupils while studying mathematics and physics. The πππgraf device was designed and built by participant of a robot club at age 13 at a different elementary school. It contains the four potentiometers: two linear ones for each axis, and two circular for additional applications, see Figure 1.

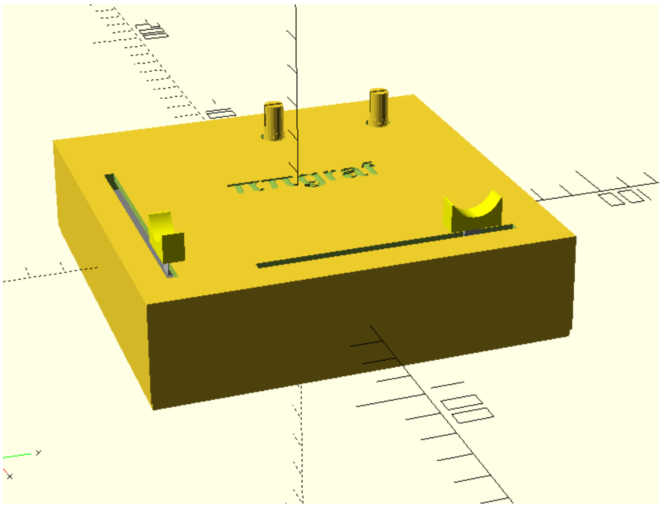


Figure 1. πππgraf as designed in OpenSCAD and produced on a 3D printer.

The prototype software to test the capabilities has been written in Imagine Logo [3], currently it allows reading (or playing loud) only line charts. In the case of line chart, after starting the operation, the PC is producing a musical note with the pitch that is proportional to the y-coordinate at the currently selected x-coordinate. We plan to add additional chart types: 1) bar charts (histograms) will produce the same tone in the same bar, producing a clicking sound when moving to the next bar; 2) xy-scatter plots, where the second linear potentiometer comes to play, for the selected x-y position in the chart, the PC will produce noise that will be proportional to the density in the position neighbourhood; 3) pie-charts, where the circular potentiometers will find their usage, while the behaviour will be alike the bar charts.

In the first three modes, the circular potentiometers can be used to regulate volume, and possible zoom in or out for a specific region of the chart.

Currently, we have built several prototypes of ππgraf, the students have finalized the software and performed successful tests with the pupils in the early 2020. Ππgraf is open-source and available at its Github page [4]. Ππgraf is very easy to build and very low cost. It has been presented to the public for the first time at Mini Maker Faire in November 2019 in Bratislava.

Recently, we have added an example VBA project for MS Excel that can use ππgraf to play chart shown in Excel using musical notes, also available at [4].

This project is a manifestation of how constructionist approach in after school robot club can be targeted at producing useful devices that help the peers of the robot club participants.

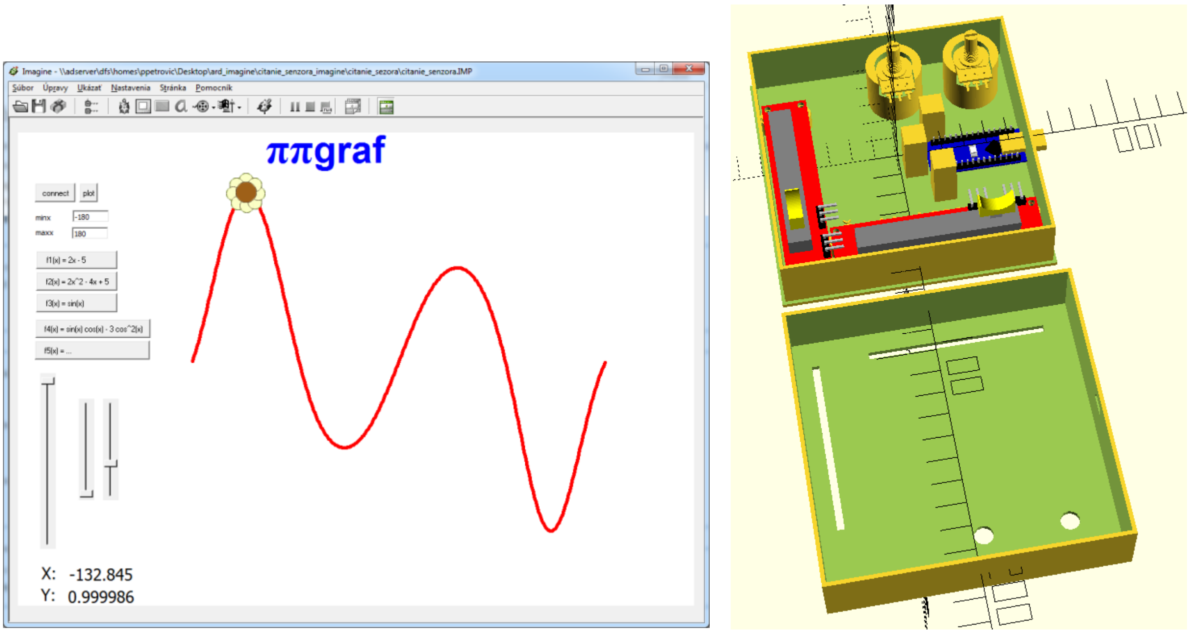


Figure 2. A protype software for ππgraf implemented in Imagine Logo (left) and the design of ππgraf in OpenSCAD as created by Oliver who is 13 years old.

References

[1] Jašková, Ľ. and Stankovičová M. (2017) *Teaching Materials in Informatics for Lower Secondary School Blind Students*. Informatics in Schools: Focus on Learning Programming: 10th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2017, Helsinki, Finland, November 13-15, 2017, Proceedings, pp.37-48.

[2] Gál, M., Keszeghová D., Sojka R., Švorc J. (2019) *Software for visually impaired – a measuring device*, Github repository, online: github.com/TIS2019-FMFI/meranie-velicin

[3] Kalaš I. and Hrušecká A. (2004) *The Great Big Imagine Logo Project book*, Logotron.

[4] Kurňava O. and Petrovič P. (2019) *ππgraf* Github repository, online: github.com/Oliver81594/pipigraf

[5] King, A. (2017) *Using Desmos to Draw in Mathematics*, Australian Mathematics Teacher, v73 n2 pp. 33-37.

Papers

A Collaborative Cross-Device Microworld for Physics Simulation Experiments in High-School

Linda Hekman Nielsen, *hekman@cs.au.dk*
Department of Computer Science, Aarhus University, Åbogade 34, Aarhus, Denmark

Mikael Hartoft Andersen, *mail@mhartoft.dk*
Department of Computer Science, Aarhus University, Åbogade 34, Aarhus, Denmark

Marianne Graves Petersen, *mgraves@cs.au.dk*
Department of Computer Science, Aarhus University, Åbogade 34, Aarhus, Denmark

Niels Olof Bouvin, *bouvin@cs.au.dk*
Department of Computer Science, Aarhus University, Åbogade 34, Aarhus, Denmark

Hermes Arthur Hjorth, *arthur@cs.au.dk*
Department of Computer Science, Aarhus University, Åbogade 34, Aarhus, Denmark

Abstract

This paper concerns itself with identifying and addressing challenges in high school-level teaching with real-time cross-device collaborative interactions. Specifically, it identifies a challenge of inappropriate workload distributions within student groups within these interactions and presents a concrete learning activity with elements specifically designed to combat this challenge. These elements are based on a notion of limiting the innate freedom of a *model* shared real-time by introducing carefully chosen constraints in the various ways of *viewing* the shared model, with the goal of making collaboration the path of least resistance in achieving a goal. The paper discusses and evaluates the approach and choice of constraint by carrying out the learning activity in real classroom settings.

Keywords

Education, Cross-device interaction, Evaluation

Introduction

Constructionism has a long-standing tradition for designing Microworlds – virtual places for learners to construct, modify, and explore an external representation of some domain (Edwards 1998; Hoyles and Noss 1992). Microworlds provide a range of benefits to all stakeholders in education and educational research; they can cheaply provide all learners with access to equipment that would be prohibitively expensive or otherwise impractical for a classroom; they are often easy to scale up because they run on hardware that students already have access to (e.g., cheap laptops or tablets), and they provide both teachers and educational researchers with a "window" through which to see students' thinking and learning as it happens, providing a fruitful approach into what learning is (Masson and Legendre 2008; Noss and Hoyles 1996).

However, the open-ended nature of Microworlds can sometimes feel intimidating to teachers, who experience uneven workload distributions when students work in groups, and who therefore worry that the more dominant students are the only ones who learn from the activities. In this paper, we present a design-based classroom study in which we deliberately and temporarily constrained the open-endedness of a web- and browser-based Microworld of our own design using the interaction model of collective interaction (Petersen and Krogh 2008). The Microworld was designed to help Danish high school physics students learn about diffraction grating. Collective interaction (ibid) is an interaction model which enforce collaboration and negotiation between people when providing input for a system. This type of interaction is less efficient than individual control, but invites for shared engagement, negotiation and potentially playful interaction (Petersen et al 2010). It has previously been applied in library and domestic settings (Petersen and Krogh 2008) and for children's' making of music (Grønbæk et al 2016). We embedded this model in a cross-device system (Houben et al 2017) where students use each their personal device to access and control parts of the shared experiment.

In this paper, we focus on those parts of our testing and data analysis that explored the potential benefits of temporarily constraining the open-endedness of the activity.

Our study

In this section, we introduce the study, describe the particular educational context it was carried out in, what preliminary inquiries were performed, and how the study was conducted. We interviewed teachers from Danish high schools about their existing use of classroom technologies and, in particular, challenges associated with that. Identifying a common challenge and a suitable topic, we then designed a Microworld with potential solutions. Finally, we implemented it in two different classrooms, and collected data in order to evaluate it.

Identifying a topic and potential challenges

We reached out to natural science teachers from Danish high schools nearby and conducted semi-structured interviews in order to identify challenges they faced in their teaching pertaining to digital tools for real-time collaboration, and to identify a topic for our Microworld.

We interviewed three different teachers at two different schools. Both schools reside in large Danish cities, and they have between 500 and 700 students. Both have self-described strong natural science presence, as evident in various good placements in inter-school science competitions. They are both also part of a nation-wide network of Science High Schools, with the goal of furthering interest in math and science in general education.

Topic: Diffraction grating

A staple in physics teaching for first year students in Danish high schools is an experiment with diffraction grating, which consists of shining a laser through a grating and measuring the resulting diffracted light. Though simple in its setup, the experimentation aspect of the experiment proves ineffective, according to the teachers, as the equipment available to students does not enable gradual variable adjustments. Instead, students switch out lasers and/or gratings with discrete properties, and the knowledge product of these properties' effect on the resulting diffraction is potentially lost. Additionally, the experiment is fragile as small imperfections in the setup such as the laser's angle can invalidate the results.

These difficulties are easily mitigated in a virtual Microworld, as the gradual adjustments and a controlled environment are trivial to implement.

Existing challenge: Uneven workload distribution

We found a recurring theme in our semi-structured interviews, which would become the main focus and design challenge of our study, namely an *inappropriate workload distribution* within groups. Particularly, high-performing students have a hard time engaging in cooperative activities as they tend to assume control of the group and do all the work themselves. All three reported finding it problematic that students have a hard time cooperating.

Teachers' experiences were particularly based on using Google Docs for shared group activities. They told us that groups often will split up the task at hand when faced with such a live collaboration tool. This results in each student engaging with, and therefore potentially only understanding, a fraction of the particular task. Consequently, live collaboration often means not actually collaborating, but individually accomplishing a smaller, simpler task, and as a group provide the requested conclusion as the sum of individuals' contributions, when the intention was for students to see that it is more than the sum. Preventing students from doing this is, according to the teachers, almost impossible to avoid in tools like Google Docs.

All teachers further noted that the division of labor is not always even, either. It is commonplace that one student will perform the entire task, as they often already possess the necessary prerequisites for accomplishing the task, because they find it cumbersome to wait for the rest of the group. This is in practice often not even dissatisfactory to the others – they are happy with just delivering a conclusion, having taken part or not. This is, of course, very dissatisfactory to the teachers, as it undermines the educational motive for doing the activity and fails to achieve the learning goals. For two of the three teachers, it means they actively avoid live collaboration tools.

We therefore set as a design goal to build a learning activity that allows for real-time collaboration, and that ensures that all students participate.

Guiding question

To guide the design and evaluation of our prototype, in this paper we address the following question:

How can we design and evaluate a multi-device Microworld that

- prevents or discourages uneven workload distributions
- aligns with existing learning standards

Our prototype and design rationale

We give here an overview of how our Microworld looks and works so that the reader can better understand what happened in the classrooms.

The prototype is web-based, and the experiment is a virtual equivalent to the physical experiment living in the student's browser. We chose the web as A) every student is expected to have a browser, B) distribution is trivial, and C) a technology stack called Webstrates (Klokmose 2015) allows for real-time sharing of state (namely, the DOM) across multiple devices with a low development overhead, allowing for rapid prototyping of collective interaction.

The first core element to the design is the virtual experiment. An overview screenshot of our design can be seen in Figure 1 below.

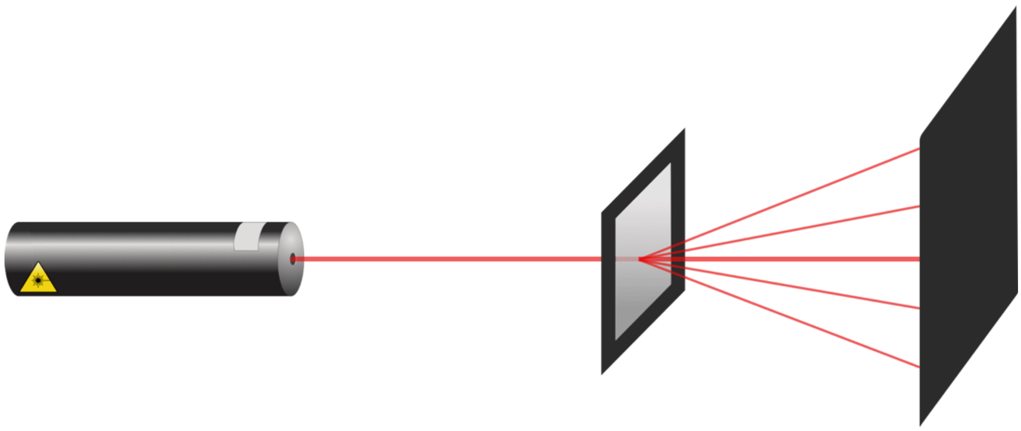


Figure 1. Visual Experiment Setup Representation

It is collaborative across devices, meaning students can look at different parts of the same experiment, and thereby work together to solve the problems presented in the learning activity. For instance, one student can work with the grate, while another student works with the laser – just like in the real-world experiment. This visual representation is a continuation of existing practices for showing the experiment and students are familiar with it from their textbooks.

There are three elements in the experiment; a laser, a diffraction grating, and a screen showing the resulting diffractions of the laser light. Each element is inspectable by clicking on it. This results in a popup specific to the element, as shown in the screenshots in Figure 2.

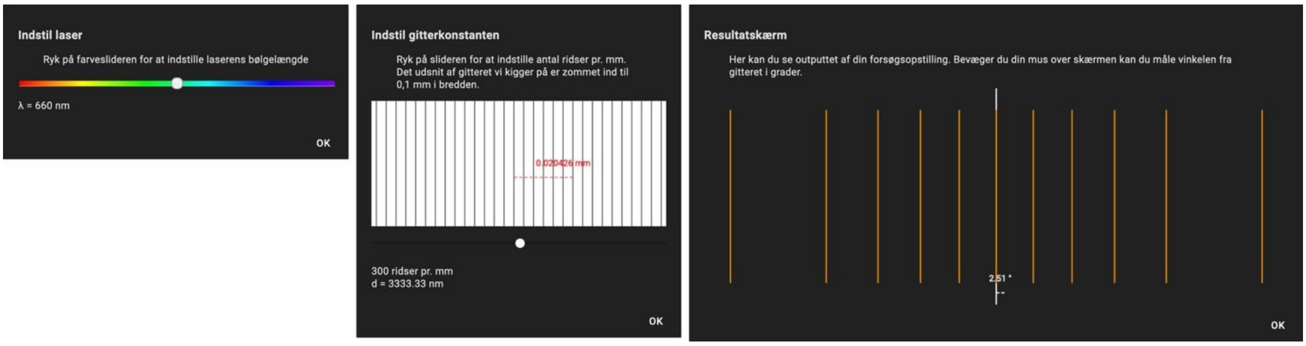


Figure 2. Per-element inspection modals.
From left to right: The laser, the grating, and the result screen

The laser can be manipulated with a color slider, and the accompanying wavelength is shown.

The diffraction grating's internal distance between slits can be manipulated as well, also by a slider. The inspection will show a scaled-up segment of the grating, in order to visually aid the notion of distance between slits. The students actually give input in the form of "slits per millimeter", and the matching "distance between slits" will be shown.

Inspecting the screen will give a full-screen pop-up modal where the resulting diffraction is easily detectable and measurable. Mousing over the diffractions, an angle measurement from the grating is shown.

The important aspect of these inspection pop-ups is that *only one can be visible and manipulable at any given time on a single device*. This practical constraint means that whenever the students need to solve a problem requiring synchronous actions and reactions, the students need to work together to accomplish their task, as they need simultaneous information from multiple devices.

Figure 3. Students during the learning activity, each with their own modal. Collaboration is necessary to obtain necessary information

Structuring of the Learning Activity

The experiment is embedded within a structured learning activity, which is a step-by-step question and answer flow, as seen in Figure 4. The answers are *individual*, i.e., not collaboratively shared within a group, and each student has to provide an answer.

The students are instructed to use the tool to obtain the answers to the questions. The *work-flow* is that students, when met with a question, will perform the necessary manipulations in one place in order to read out the answer in another. The challenge is then to first interpret the question correctly, then find and perform the correct manipulation, and lastly correctly identifying where to look for, and how to interpret, the answer.

There are three main parts to the structure of the learning activity. An introductory part, an exploratory part, and a more guided part. This is done to ease the students in – both in using the tool and understanding the underlying theory. The complexity gradually increases, and ultimately ends up with students having to demonstrate use of acquired knowledge in compliance with the learning goals of their curriculum.

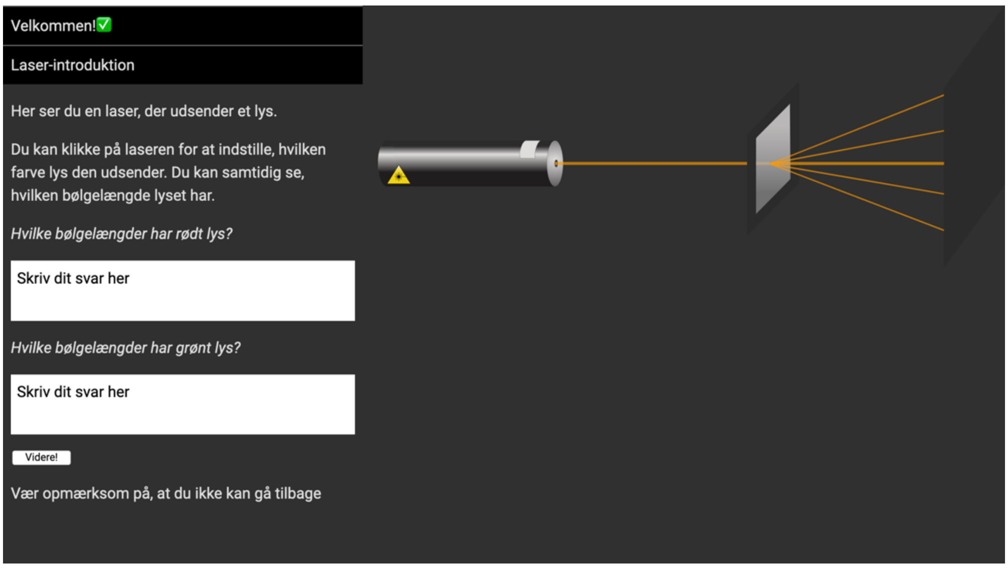


Figure 4. Guidance

In the **introductory part**, the experiment is *not* shared within the group, as each student has their own experiment. These questions introduce the students to the tool, what each element does, and how to control it. This will ensure an understanding of what role each element plays, and by extension how each element affects the experiment.

In the **exploratory part**, the transcluded experiment is shared within a study group of three people. This means that if one student changes the wavelength of the laser, the other students will also see the effects of that change – the laser color will change, and, if open, the result screen will update. The associated questions are very exploratory in their motivations. They ask students to perform manipulations – not number-specific, but direction-specific – and ask them to reflect and report on what effects they cause. The outcome hereof should be an observation-driven intuition of the positive and negative relations between wavelengths, lattice constants, and order counts and angles. Then, the students are tasked with putting this relation intuition to use to accomplish simple tasks like producing the highest number of orders they can.

In the **guided part**, the questions are almost exact copies of questions from teachers' existing teaching materials, in order to align with the learning goals of the experiment. These are more fact- and precision-oriented questions and should report on students' abilities to apply their knowledge in a less exploratory, but more fact-seeking sense.

Design rationale

Now that we have gone through *how* the design looks and works, we want to briefly explain *why* the design looks and works like it does, and how we intended for it to meet our design goals of accommodating inappropriate workloads, and meeting with existing learning goals.

Accommodating Inappropriate Workload Distributions Within Groups

In our domain inquiry, this was a prevalent concern from teachers when discussing real-time collaborative tools. Students should be unable to split up tasks between them. Multiple approaches were considered, including forcing various interactions to be performed by multiple people synchronously in order to register, forcing all group members to answer before proceeding, or actively withholding information so no student alone would ever have enough information to answer the questions. Many of these approaches were deemed too restrictive and forceful, and at high risk of being perceived as annoying and obnoxious.

We settled on a more subtle approach of withholding information. The practical constraint of only showing one inspection pop-up at a time, does not lock a student completely out of all required information, but instead heavily *favors* collaboration as the information is much more easily obtained by including multiple people and devices.

The second important aspect of achieving this are lines of questions in the learning activity that specifically asks for the information more easily obtainable through collaboration. This could be questions about how changes in one variable relates to changes a second variable, not their numeric relations, but their directional and magnitudinous relations. We accomplished this by asking students questions like, "*What happens to the result screen when dragging the wavelength slider?*", or, *"What happens to the result screen when dragging the lattice constant slider?*

Meeting with existing learning goals

In addition to the design challenges relevant for this project, it was important to make sure we met the existing learning goals with our learning activity, as our activity was included in real classes' real schedules. This is evident in the last part of the knowledge-applying questions as they are heavily inspired by materials from existing learning activities.

Evaluation and Data collection

Procedure

The prototype was evaluated in two different schools with two different teachers and two different classes. It was assessed that the activity would take roughly an hour for a class to complete, and plans with the teachers were made accordingly.

The students would have already read the textbook chapter pertaining to the experiment and would therefore have the same basis for carrying out the activity as they would have with the traditional physical experiment.

Beforehand, 30 different instances of the web-based prototype were prepared and distributed as links to the students by the teacher.

The teacher would briefly introduce the researchers, and we would go on to describe the place it has in their curriculum – it being a replacement for the physical experiment. We would not disclose any design goals or any additional information, only that their answers were for our eyes only, not their teachers', in an attempt to get them to answer as freely as possible. The teacher then organized the groups, and students started the activity.

Prototype evaluation

We wanted to evaluate the prototype in the context that it was intended for, for which reason the evaluation took place in a natural setting (Preage 2002). In contrast to controlled environments, natural settings tend to be messier in the sense that the students can get interrupted by the person next to them, by the teacher, or other sources. The more uncontrolled setting can make it harder to test a very specific hypothesis about the interface or use of interface. This makes it more difficult to identify what caused a specific type of behavior.

Co-discovery, where users collaboratively explore the activity, which incurs natural out-loud thinking through helping each other, was chosen as an evaluation method. Depending on the data to be collected, the evaluator has the opportunity to take an active part in the evaluation by asking questions or monitor the "think aloud" between the groups. By asking specific questions we manage to direct the output towards our interests.

We brought our prototype into two classrooms with a total of 60 students, each for a period of 1.5 hours. During the implementation, the role of the teacher was introducing the activity in relation to its place in their curriculum, and the researchers participated by helping with technical issues throughout the activity. The researchers were observing and collecting data throughout as well.

In order to answer evaluate the success in preventing inappropriate workload distributions, we collected two different kinds of data: Audio recordings, and written work by students.

Audio Recordings

During the lesson, students were divided into 10 groups that each worked at a separate table. We put audio recorders at 3 tables. As the researchers moved around the classroom during the lesson, we achieved co-discovery evaluation through these audio recordings, where we asked the specific groups to think aloud as much as possible. We would come by now and again to ask questions or answer any questions they might have. The degree to which the students thought aloud was varying, a circumstance we knew beforehand with our approach.

Written responses

In addition to in-person observation and audio recordings, we also gather the answers the students provide in the activity for evaluation purposes.

Lastly, we conduct an open discussion with the class after the activity in order to hear their thoughts on the activity.

Evaluation questions

For each data set, we set up questions for evaluating our design challenge of inappropriate workload distributions. We will outline these here.

Answer inputs

To analyze the answer inputs, we look at the following questions for evaluating each our design challenge:

- **Does everybody answer?** If all students provide answers, it is an indication of them participating in the activity, and not simply letting others do the work.
- **Do they agree on answers?** If a group internally agrees on answers, it is a strong indication of collaboration and thereby all-round participation. It *can* be the case, however, that one group member merely dictates an answer to the others. Examples of slightly varying, but substantially equal answers are a great success indicator.

Audio recordings

The questions used for analyzing the audio recordings for evaluating the design challenge are as follows:

- **Does everybody participate?** If there is evidence of all group members actively participating in the audio recordings, it is a strong indication of successful mitigation of this challenge.
- **Do they use each other, i.e. collaborate?** Evidence of collaboration and discussion in audio recordings would likewise indicate mitigating this challenge

Findings

Here follows a presentation of the data we collected. Since we collected in two ways, one was the answer inputs from the students, and the other was audio recordings of random groups, they are presented separately. We will here use the data to answer our evaluation questions pertaining to the answer inputs.

Answer inputs

Does everybody answer? We have found less than 50 instances of different forms of non-answers in a pool of roughly 1500 student responses, which is just above 3%. These range from simply not answering, to inputting "??", to more creative ways of avoiding answering.

Do they agree on answers? The outcome here surprised us. Students tend to strongly agree on the explanation answers. But they tend to *not* provide the exact same wordings of their answers. This is a strong indication that they arrive at the same understanding, but that they are not dictating answers for each other, both of which are good in relation to the *inappropriate workload distributions* challenge. As an example, when asked "What happens to the result screen when dragging the wavelength slider?", group 10 of the 2nd school provide the following answers:

- "Larger wavelength means greater angle"
- "Larger wavelength means greater angle to 1st order"
- "Larger wavelength means greater angle between 0th and 1st order"

They have clearly arrived at a common understanding by discussing it with each other but are not completely identical in their wording.

Audio recordings

Due to noise and sound issues, we only have two full transcripts, as they had good conditions, and half a transcript as a recording device just stopped 10 minutes in.

These transcripts have been analyzed in relation to the relevant evaluation questions. Examples will be provided of particularly interesting points.

Does everybody participate? This was difficult to accurately answer, as voices were difficult to tell from each other. It is our general interpretation that all group members participated, though at times, some more than others.

Do they use each other, i.e. collaborate? The students discussed the questions and the activity *a lot*, and silences were few and far between. To save space, we provide simply one vignette to illustrate the ways in which students collaborate and coordinate during the learning activity.

Student	**Quote**
Girl 1	Which variable are you controlling now?
Girl 2	I am controlling the lattice constant right now.
Boy	If you change one thing, then I am able to see the results on the screen.
Girl 2	Ahh!
Girl 1	Can you guys see me changing the wavelength?
Girl 2	Yes. Is it possible for you to go to the lattice constant's setting and change the lattice constant to 800 nm?
Girl 1	Hmm, no that's not possible.

After discovering the collaboration possibilities, they use each other to ascertain whether an action is possible or not instead of trying individually. They coordinate and provide information verbally and actively show interest in what the others are doing.

Furthermore, the students actively engage in discussions in order to arrive at the right answers. Her is one example:

Student	Quote
Girl 1	What is the maximum amount of orders you can achieve?
Girl 2	The lattice constant should be as high as possible, right?
Girl 1	Yes…
Boy	No, I think it was as low as possible if we want many (orders).
Girl 2	No, it is high.
Boy	When you put it as high as possible, then these two appears (orders).
Girl 1	No, it is high.
Girl 2	Do you mean the lattice constant or the wavelength?
Boy	I mean the lattice constant. When you make the lattice constant high, then two orders appear.
Girl 1	No, then more is shown. I think that you are thinking of the fact that it says 100 slits.
Boy	Of course… I have swapped them around (lattice constant and wavelength)
Girl 1	And the wavelength should also be… high.
Girl 2	Yes, the wavelength should be high… or wait… is it the other way around?
Girl 1	Okay, let me try and change it (the wavelength). Does the amount of orders increase?
Girl 2	Yes!

As you can see, the students are constantly consulting each other on how to interpret what they are seeing and how to proceed from here is evidence of students actively working together in order to achieve a specific goal, i.e. collaborate. Through the practical constraint of only having access to one modal at the time, the students are forced to consult each other on the effects on what they are doing individually.

This is in contrast to the typical behaviors described to us by the teachers when working with real-time collaboration – that students divide the workload in order to do them individually.

Discussion and Conclusion

Identifying the recurring theme in our domain inquiry of *inappropriate workload distributions*, our guiding question became how we could mitigate this behavioral pattern from students, while still meeting with existing learning goals through the use of our Microworld.

This led to designing a practical constraint after translating the existing physical experiment to a digital cross-device real-time collaboration Microworld, and the evaluation emphasis was placed on the effectiveness of this constraint. From the design process, the introduced constraint became that of having access to only one of the three *views* of the experiment state on a single device at any given time. Coupled with questioning imploring interaction with several of these views simultaneously, we were able to actively encourage collaboration, and we saw a mitigation of the problem of inappropriate workload distributions within groups.

The take-away from this project for designers of tomorrow's learning activities involving real-time collaboration Microworlds should be awareness of potential problems in constraint-free designs, namely students' behavior regarding workload distributions, and to consider possibilities of introducing *constraints* in the interaction to counteract this.

References

Edwards, L. D. (1998). Embodying mathematics and science: Microworlds as representations. *The Journal of Mathematical Behavior*, *17*(1), 53-78.

Grønbæk, J. E., Jakobsen, K. B., Petersen, M. G., Rasmussen, M. K., Winge, J., & Stougaard, J. (2016, October). Designing for children's collective music making: How spatial orientation and configuration matter. In *Proceedings of the 9th Nordic conference on human-computer interaction* (p. 60). ACM.

Houben, S., Marquardt, N., Vermeulen, J., Klokmose, C., Schöning, J., Reiterer, H., & Holz, C. (2017). Opportunities and challenges for cross-device interactions in the wild. *interactions*, *24*(5), 58-63.

Hoyles, C., & Noss, R. (1992). A pedagogy for mathematical microworlds. *Educational studies in Mathematics*, *23*(1), 31-57.

Klokmose, C. N., Eagan, J. R., Baader, S., Mackay, W., & Beaudouin-Lafon, M. (2015, November). Webstrates: shareable dynamic media. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (pp. 280-290). ACM.

Krogh, P. G., & Petersen, M. G. (2008). Collective Interaction–Let's join forces. *From CSCW to Web 2.0: European Developments in Collaborative Design Selected Papers from COOP08*.

Masson, S., & Legendre, M. F. (2008). Effects of Using Historical Microworlds on Conceptual Change: A P-Prim Analysis. *International Journal of Environmental and Science Education*, *3*(3), 115-130.

Noss, R., & Hoyles, C. (1996). *Windows on mathematical meanings: Learning cultures and computers* (Vol. 17). Springer Science & Business Media.

Petersen, M. G., Krogh, P. G., Mortensen, M. B., Møller-Lassen, T., & Mortensen, D. H. (2010, October). Collective interaction by design collective controllers for social navigation on digital photos. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries* (pp. 353-362). ACM.

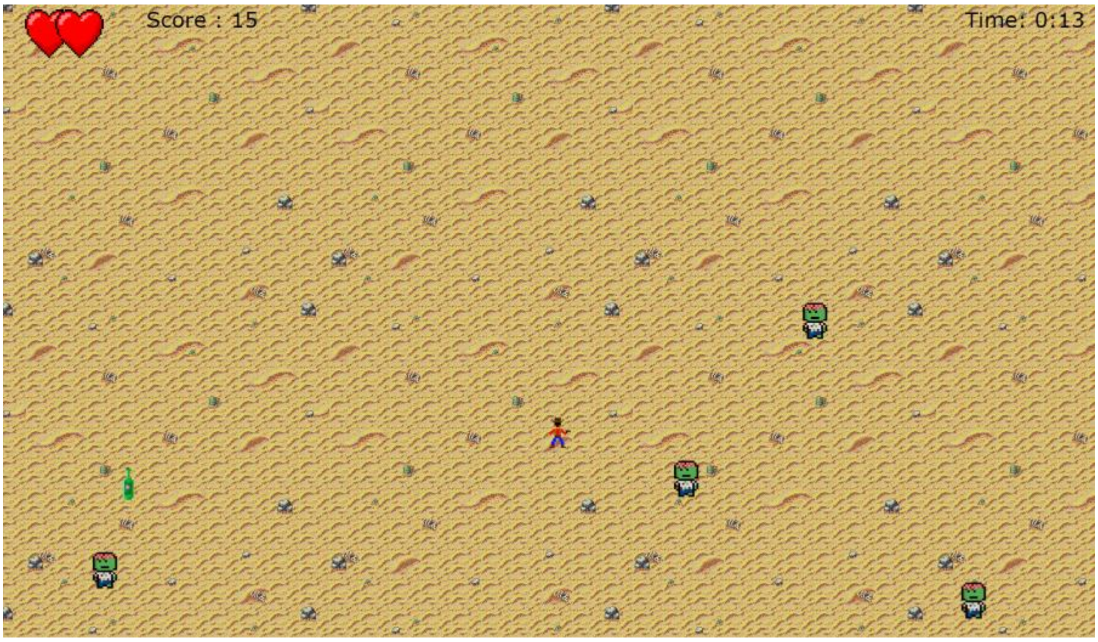
Preage, J., Rogers, Y. & Sharp, H. (2002). *Interaction design: beyond human-computer interaction*. NY: Wiley.

A Constructionist Approach to Learn Coding with Programming Canvases in the Web Browser

Márton Visnovitz, *visnovitz.marton@inf.elte.hu*
Győző Horváth, *horvath.gyozo@inf.elte.hu*
Department of Media and Educational Informatics, Eötvös Loránd University, Budapest, Hungary

Abstract

In the era of web applications running on all platforms, from the browser to smartphones and the desktop, it is important to look at the possibility of the educational use of the Web as a programming platform. In this paper we investigate the possibility of using web programming to support constructionist learning activities. We present an overview of web programming, with a special focus on HTML canvases and their programming API in the JavaScript programming language. We describe how canvas programming in the browser can be used in programming education, incorporating constructionist principles to create games. We also present our experiences of using the proposed method in a summer programming camp with children of ages 13-18. We also pinpoint some possible improvements to the framework/platform that we use in our practice and investigate possible future directions that the proposed method could take.



A game created during the summer programming camp nicknamed: Zombieland

Keywords

constructionism, web programming, JavaScript, canvas, game development

Introduction

The Web is evolving at a tremendous pace these days; browsers are improving, and new technologies are coming with them. This results in an unprecedented number of applications to appear on the web. The widespread use of this platform is due to not only the continuous growth of internet penetration, but also by the fact that applications written with web technologies run on a growing number of devices.

In addition to the traditional forms of programming education, the features and capabilities of the Web have opened new ways for teaching and learning. Innovative programming platforms (such as Khan Academy[1], Udacity[2], Hour of Code[3]) primarily take advantage of the ubiquitous nature of the web and consider it as a delivery medium and a rendering layer. We believe that the web cannot only serve as a medium for programming education, but also as its toolkit. We consider it as a platform that can motivate students to learn programming with its many creative capabilities. In this article, we explore the potential of using this platform as a means of supporting constructionist learning.

The Web as a programming platform

The usage of web technologies is on the rise for the past decade, the popularity of JavaScript among developers is undeniable[4]. In the browser three programming languages are used to create web documents and applications. HTML is used to describe the content, CSS is used to style it, and JavaScript is the de-facto language to describe behaviour and add user interaction.

While some older investigations found that JavaScript is not suitable for education (Mannila & de Raadt, 2006), later sources say that JavaScript and the web as a programming platform provides many benefits for introductory programming (Horváth & Menyhárt, 2014). This is due to the rapid development of the language and a platform in the recent years, especially since the introduction of the 6th edition of the ECMAScript standard[5] which serves as the basis of JavaScript implementations.

While a lot of educational platforms look at the web as a runtime for educational software, it is also possible to exploit the potential of the web as the *target* of our educational process (Visnovitz & Horváth 2018). Web technologies provide an exciting, interactive and familiar environment for students to program (Mahmoud et al., 2004). JavaScript requires zero boilerplate code to work, thus it is easy to get started with, but it still provides a large set of features. Furthermore, there are a lot of resources available online for JavaScript allowing students to look for the information necessary to solve their problems.

Creating programs in the web comes down to three main parts (“pillars”): *Application State Representation*, *User Interface Generation*, *Event Handling* (Visnovitz & Horváth, 2018). These three “pillars” represent the core of modern web programming, but as concepts are easy to grasp. All three of the above-mentioned components can be designed and planned without extensive programming knowledge (Visnovitz & Horváth, 2018). This means that unplugged activities can be used to support the design process of the students’ programs.

Programming with the Canvas API

Learning all core technologies (HTML, CSS, JS) of the web requires a significant effort, however it is possible to eliminate the need for CSS and minimize the usage of HTML to a few lines of code

[1] https://www.khanacademy.org/
[2] https://www.udacity.com
[3] https://hourofcode.com
[4] https://medium.com/javascript-scene/how-popular-is-javascript-in-2019-823712f7c4b1
[5] https://www.ecma-international.org/ecma-262/6.0/

by using canvases to create our programs. The `<canvas>` HTML element[6] and the Canvas API[7] of the JavaScript programming language provide an easy-to-use interface for programmatically creating graphics in the browser not unlike the Logo programming language. Using this technology, it is possible to introduce low level programming concepts such as variables, loops, conditions and functions to create repeating patterns in our graphic at an early stage.

Combining the usage of the Canvas API with the “three pillars” concept is also possible. Creating animations requires storing the current state of the animation in some form (Pillar 1). This is analogous with the state of applications with complex user interfaces. This internal state can be rendered onto the canvas element (Pillar 2) using the Canvas API.

Using the built in `requestAnimationFrame` function of the JavaScript language, it is easy to make these animations come to life, or even create games. This method of the browser allows the creation of animation loops. In the animation loop a function modifies the state of our application (Pillar 1) and renders the changes onto the canvas (Pillar 2). These animations can be used to create simple graphics or even a simulation or a model of a real system.

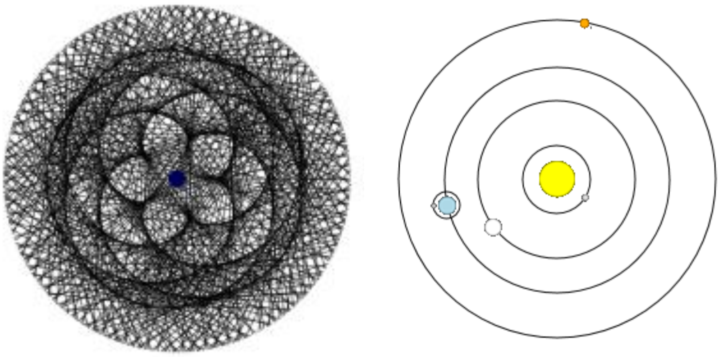


Figure 1. Examples of a drawing (left) and an image from an animation of the solar system (right) created with canvases

The `addEventListener` method of JavaScript allows us to respond to events happening in the browser (e.g. clicking or pressing a button on the keyboard). Programming the responses to these events (Pillar 3) can make our animations interactive, opening the possibility of creating games. When using the canvas, we usually listen to the events of the browser globally, eliminating the need to bind our event listeners to specific HTML elements. This makes event handling easier than in conventional HTML element-based applications.

[6] https://developer.mozilla.org/en-US/docs/Web/HTML/Element/canvas
[7] https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API

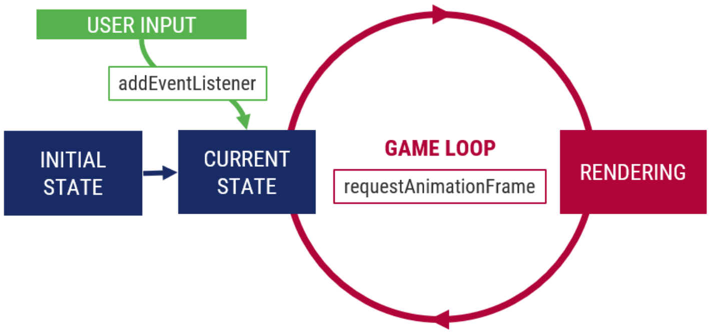


Figure 2. Schematic of the game loop

The versatility of the browser and the Canvas API allows us to use a variety of teaching strategies as described by Bernát and Zsakó (Bernát & Zsakó, 2017). These strategies include teaching by (turtle) graphics, teaching with animation or teaching with game-development. Organising our code around objects and the interaction between these objects has a great potential for students to learn about object-oriented patterns and design, especially when building games (Corral et al., 2014). This form of encapsulation shows similarities to how modern front-end frameworks (like React[8] or Angular[9]) work.

Constructionist principles in web programming

Constructionism as a learning theory can be applied in many areas. One such area is software programming. Learning programming the constructionist way, started with the Logo programming language (Harel & Papert, 1991). The idea of creating visual artefacts with programming was a steppingstone to many programming environments such as Scratch (Resnick et al., 2009) or NetLogo (Tisue & Wilensky, 2004).

Resnick's theory of using the iterative process of imagining and creating to support constructionist learning (Resnick, 2007) can be applied to the process of learning programming by creating software artefacts. This theory shows similarities to modern Agile software development methodologies (Monga et al, 2018).

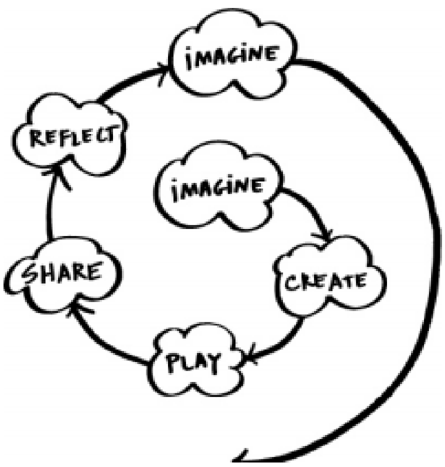


Figure 3. Scheme of Resnick's theory of the iterative process of learning (Resnick, 2007)

[8] https://reactjs.org/
[9] https://angular.io/

The Canvas API makes it possible to create anything from a basic drawing to a complex video game with the basic set of tools it provides. With its gradual learning curve, it is optimal for self-exploring the options it provides, and to create software artefacts of different types and complexity. It is also possible to create a curriculum based on Bernát and Zsakó's proposed teaching strategies (Bernát & Zsakó, 2017) – with maybe one exception: robotics – using canvas programming, as it supports both turtle-like graphics, animations and game development. All these strategies align well with the constructionist principle of creating personally meaningful artefacts.

The popularity of video games inspired the integration of games and game creation into the constructionist learning theory (Caperton, 2010). While Weintrop and his colleagues write about games that support creation within themselves (Weintrop et al., 2012), it is also possible to look at the creation of a game itself as a constructionist process. The Canvas API along with the three pillars concept provides a framework in which games can be created following the constructionist methodology (Horváth et al., 2016). Creating the various parts of a game allows our students to go about exploring technology as they improve their program. After creating an initial, basic application – analogous to the Minimal Viable Product (MVP) of several Agile methodologies – students can start extending their game with new elements. Adding these features to the game happens in accordance with Resnick's learning spiral (Figure 2). These new elements (like introducing a new type of opponent or a new game mechanic) can be viewed as the software artefacts that students create to improve their design. With proper guiding questions these features can be designed and implemented along with the three pillars of web applications (Visnovitz & Horváth, 2018).

Field experiment

Experiment environment

During the summer of 2019 Eötvös Loránd University organised a five-day programming camp dubbed "Programming in the browser" along with seven other courses with different topics. The participants of the camps – aged 13 to 18 – came from various underprivileged areas of the country based on teacher recommendations. For the activities of the "Programming in the browser" course we chose the Canvas API of the JavaScript programming language as the technological framework, and we applied the proposed constructionist method of learning programming. For tools, we used the Google Chrome[10] browser and Microsoft Visual Studio Code[11] for code editing.

We designed the activities of the summer camp to cover the "4 P's" of Seymour Papert (Resnick, 2014; Resnick, 2017). The five days of work was all about students working on their own *projects*. The topic we chose was creating games, because we found that children are very *passionate* about games at any age. The games that they created (also the process of making them) are a means to *play*. They *played* while learning and *played* with the games they created. Finally, they were working in groups of three, learning about programming along with their *peers*.

Student engagement

Students in the camp only got a short technology introduction to begin with. After a small demo, they started to work on their own game projects. All groups started to build radically different games, each based on their own interests. Not only the themes of the games were different, but also there were examples of games from multiple genres, like platformer, side-scroller or reaction games. Each group worked on their own projects with high motivation and enthusiasm. Effectively splitting the task between themselves they successfully cooperated to create their games. Some groups even spent their time outside the lab planning the next phase of their project and designing assets.

The "Eight Big Ideas" of a Constructionist Learning Lab (Stager, 2006) could be found in the pupils' activities: They worked on their projects, *learning along the way,* creating artefacts in the *digital*

[10] https://www.google.com/chrome/
[11] https://code.visualstudio.com

world, using *modern technologies*. While at the initial steps their projects looked basic but *taking the proper time* to *investigate and learn about technology* and to *fix their mistakes,* they created diverse, interesting applications that even we couldn't think possible within the framework of this camp. *We*, the instructors of the course *also learned a lot* by facilitating the work of the children. We even incorporated some of their ideas into university web programming classes. While at times children had a *hard time* finding solutions to their problems, they eventually all found a way to make their ideas work and enjoyed the fruit of their struggle.

Student questionnaire

After the five-day course, all 19 participants were asked to fill out a questionnaire about their prior knowledge in programming, and their experiences in the "Programming in the browser" camp. The questions were designed to try to address the various aspects and principles of constructionism. The responses are shown in Table 1 (for each question they had to give an answer on a scale from 1 to 5).

Question	**Answer average**
1. How would you rate your programming skills?	3.26
2. I think the programs we created are visually impressive.	4.79
3. I feel I created something new.	4.63
4. I enjoyed this kind of programming.	4.84
5. I believe I learned a lot about programming.	4.68
6. I always understood what exactly the code I wrote did.	3.89
7. I felt that I could express my creative ideas through this kind of programming.	4.74
8. I would like to create similar programs in the future.	4.89
9. I feel that I created my program on my own without help.	3.95
10. During breaks I felt like continuing the work instead.	4.00

Table 1. Responses to the student questionnaire

The responses show that students came with some prior knowledge in programming, but they didn't feel confident about their programming skills (Q1). It is important to note, that while all of them but one had learned some form of programming before, only 5 of them had encountered the JavaScript language before the camp. The answers also show that student motivation was very high (Q2, Q4, Q7, Q8, Q10) and that they felt that they were part of a creative process of creating something new (Q7, Q3). The learning outcome of the experiment is also highly positive (Q5), however it would be important to validate the actual acquired knowledge. Feedback from the participants suggest that they will continue to work on similar projects in the future (Q4, Q8), further expanding and solidifying the knowledge they gained. An example of this are the two pupils who participated in the same summer camp a year before, and they continued to work on their own projects during the next year. Also, many students from this summer showed interest in returning to the camp in the coming years.

Conclusions and future directions

Our experiences from Eötvös Loránd University's summer camp showed that it is possible to create learning activities that are centred around the principles of constructionism using web programming – more specifically canvas programming. Based on a questionnaire that students filled out it was clear that this activity resulted in high student engagement and motivation by allowing them to create their own video games with low level tools.

Based on our experiences in the summer camp we would like to make some minor improvements to the environment we provided for students. In addition to the questionnaire we also asked the participants about their experiences with the environment they had to work with, what was hard, where could we improve on the APIs provided by JavaScript. One glaring point was the handling of the canvases as it is the only point where they had to interact with HTML code. While some of the pupils had some experience with HTML it was confusing for most of them to work with two languages at the same time. To address this problem a prototype abstraction layer was created on top of the Canvas API to provide a single JavaScript-only method of handling canvases (layers) in the programs.

Another aspect that can be improved is the module loading of JavaScript files. In the camp students had to load their various scripts manually in the HTML documents. The module syntax of the ECMAScript 6 (a.k.a. ECMAScript 2015) standard provides an easier, JavaScript only way to handle the load order and dependencies between pieces of code. This allows us to eliminate the need of writing HTML code altogether and focus solely on one programming language.

The next step in developing a truly comprehensive web programming-based learning environment is to support multiple programming paradigms and strategies in the same platform. A lightweight library could be developed to further extend the capabilities of the Canvas API to support more “turtle-like” commands like “turn” or “forward”. This could make getting started easier and make the environment suitable for younger children as well. Many online learning environments, like Khan Academy follow this direction, however their abstraction layer on top of the Canvas API is usually higher level. Introducing this Logo-like abstraction could also open the possibility of creating an agent-based environment like NetLogo (Tisue & Wilensky, 2004). The benefit of creating such a system would be having a unified environment in which a wide range of different activities could be performed with low-level tools and using a single programming language (JavaScript) and a single environment (browser).

References

Bernát, P., Zsakó, L. (2017). *Methods of Teaching Programming – Strategy.* In Proceedings of DidMatTech 2017, Travana.

Caperton, I. H. (2010). *Toward a theory of game-media literacy: Playing and building as reading and writing*. International Journal of Gaming and Computer-Mediated Simulations.

Corral, J. M. R., Balcells, A. C., Estévez A. M., Moreno G. J., Ramos M. J. F. (2014). *A game-based approach to the teaching of object-oriented programming languages*. Computers and Education, Volume 73. pp. 83-92.

Harel, I. E., Papert, S. E. (1991). *Constructionism*. Norwood. Ablex Publishing Corp.

Horváth, Gy., Menyhárt, L. (2014). *Teaching introductory programming with JavaScript in higher education*. In Proceedings of 9th International Conference on Applied Informatics. Eger. pp. 339-350.

Horváth, Gy., Menyhárt, L., Zsakó, L. (2016). *Viewpoints of programming didactics at a web game implementation*. New methods and technologies in education and practice. pp. 79-88

Mahmoud, Q. H., Dobosiewicz, W., Swayne D. (2004). *Redesigning Introductory Computer Programming with HTML, JavaScript, and Java*. In Proceedings of SIGCSE '04 Proceedings of the 35th SIGCSE technical symposium on Computer science education, Norfolk. pp. 120-124.

Mannila L., de Raadt M. (2006). *An Objective Comparison of Languages for Teaching Introductory Programming*. In Proceedings of 6th Baltic Sea conference on Computing education research. pp. 32-37.

Monga M., Lodi M., Malchiodi D., Morpurgo A., Spieler B.: *WG6: Learning to Program in a Constructionist Way*. In Proceedings of Constructionism 2018, Vilnius

Stager, G. (2006). *Eight Big Ideas Behind the Constructionist Learning Lab.* In: *An Investigation of Constructionism in the Maine Youth Center*. Melbourne.

Resnick, M. (2007). *All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten*. ACM Creativity & Cognition conference, Washington DC.

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). *Scratch: Programming for All*. Communications of the ACM, Vol. 52, no. 11. pp. 60-67.

Resnick, M. (2014). *Give P's a Chance: Projects, Peers, Passion, Play*. Constructionism and Creativity conference, opening keynote. Vienna

Resnick, M. (2017). *The Seeds That Seymour Sowed*. International Journal of Child-Computer Interaction.

Tisue S., Wilensky U. (2004). *NetLogo: A Simple Environment for Modeling Complexity*. In Proceedings of International conference on complex systems. Boston. pp. 16-21.

Visnovitz M., Horváth Gy. (2018). *The Web – A Platform for Creation*. In Proceedings of Constructionism 2018, Vilnius. pp. 736-740.

Weintrop D., Holbert N., Wilensky U., Horn M. (2012). *Redefining Constructionist Video Games: Marrying Constructionism and Video Game Design*. In Proceedings of Constructionism 2012, Athens. pp. 645-649.

A Robot-based Activity for Kindergarten Children: An Embodied Exercise

José Armando Valente, jvalente@unicamp.br, NIED, UNICAMP, Brazil

Ricardo Caceffo, caceffo@ic.unicamp.br, Institute of Computing, UNICAMP, Brazil

Eliana Alves Moreira, eliana.moreira@ifsp.edu.br, Federal Institute of São Paulo, IFSP, Brazil

Rodrigo Bonacin, rodrigo.bonacin@cti.gov.br, CTI Renato Archer, Brazil

Julio Cesar dos Reis, jreis@ic.unicamp.br, Institute of Computing, UNICAMP, Brazil

Marleny Luque, marlenyluque@gmail.com, Institute of Computing, UNICAMP, Brazil

João Vilhete D Abreu, jvilhete@unicamp.br, NIED, UNICAMP, Brazil

Fabrício Matheus Gonçalves, fmatheus@ic.unicamp.br, Institute of Computing, Unicamp, Brazil

Camila Brennand, camillatenorio123@gmail.com, Institute of Computing, UNICAMP, Brazil

M. Cecilia C. Baranauskas, cecilia@ic.unicamp.br, Institute of Computing, UNICAMP, Brazil

Abstract

The objective of this article is to explore the concept of programmable environments to create a robot-based activity for kindergarten children, who have not yet mastered the use of written language, so they can, as a group, interact with a robot and reach a particular task. This work is part of a wider research project "Socioenactive Systems". In this work, we focus on the creation and use of the robot-based activity. The designed activity is based on the *Little Red Riding Hood* narrative adapted to the children's daily lives. Children played the role of rangers (instead of hunters) who had to interact with themselves and coordinate their actions to help a robot (a mBot characterized as a Robot-Wolf) to find the Grandma's laboratory so she could fix its GPS. They wore boots that were used to interact with the Robot-Wolf. The design of the robot-based activity was based on the co-design methodology. The study with the children was conducted using the action-research method. It was implemented in a school setting for kindergarten students. Participated in this study 26 children (11F, 15M) between 4 and 5 years old. The children's activities were videotaped and analysed using the Grounded Theory methodology. Figure 1a shows the Robot-Wolf, Figure 1b a child interacting with the Robot-Wolf through his boot, and Figure 1c shows a group of four children acting as rangers, interacting with the Robot-Wolf guiding it to Grandma's lab.

(a) (b) (c)

Figure 1a, the Robot-Wolf is at the Grandma's lab; Figure 1b, a child interacting with the Robot-Wolf through his ranger's boot; Figure 1c, a group of children are guiding the Robot-Wolf to Grandma's lab

The children's behaviour was categorized in terms of 4 categories: Awareness, Predictability, Cooperation, and Type of Interaction. Our results show that this activity was able to engage kindergarten children and demonstrate how they behave as a group to solve a particular task.

Keywords: Embodied-Interaction; Robotics; Preschool; Educational-Technology; ICT; Children;

Introduction

Several programmable environments adopt multiple agents such as NetLogo (Wilensky, 1997) and StarLogo (Resnick, 1996) which are used to simulate natural and social phenomena. In these environments, agents can be instructed independently regarding a particular behaviour and as they interact among themselves, a particular goal is reached or it is possible to observe connection between the micro-level behaviour of individual agent and/or macro-level patterns that emerge from their interaction.

This article addresses how to design a robot-based activity that implements a physical version of the multi-turtles modelling environments as StarLogo and NetLogo, so kindergarten children can interact among themselves and with the robot to guide it to a particular target. We used the embodied action concept, proposed by Varela, Thompson, and Rosch (2016), as the theoretical basis for the development of this activity.

This work is part of a research project (Baranauskas, 2015) developed at the University of Campinas, Brazil. The objective of this project is to build a conceptual framework from exploring different scenarios (Caceffo *et al.*, 2019; Caceffo *et al.*, 2020), to instrumentalize those involved with the design and development of socioenactive systems to produce technological solutions that effectively take into account the cultural context of those who participate, including their differences, needs, preferences, abilities, and values.

In this article, we concentrate on the creation of the robot-based activity and use it in an educational context. The objective is to show how to develop a robot-based experience for kindergarten children, age 4 to 5 years old, so they can interact with a computational technology that is enriched with sensors and actuators in an environment in which they have to cooperate to solve problems and reach a particular task. The narrative for constructing this activity was based on the Little Red Riding Hood story, adapted to a context closer to the children's daily lives and to address playful aspects of interaction and cooperation. Children played the role of rangers (instead of hunters) who had to interact with themselves and coordinate their actions to help the Robot-Wolf to find Grandma's laboratory so she could fix its GPS. They wore boots that were used to interact with the Robot-Wolf. The sessions with these children were observed, video recorded and analysed in terms of the children's behaviour (Caceffo *et al., 2018).*

Catlin and Blamires (2010) explain that working with physical robots is an active learning process and exploring the idea of embodiment could lead to new understandings about educational robots. According to these authors, "Embodiment is about how we engage with the world, extract and share meaning through our interaction with it and the objects it contains… It is self-evident that this applies to robots." (Catlin and Blamires, 2010, p. 1)

The remaining of this article is organized as follows: the next session presents the background and theoretical referential, followed by the methodology conducted in the study. Then we present the computational activity and how it was used with the children, the results and discussion; finally we conclude and suggest further work.

Theoretical Background

Several themes are related to this study. First, the programmable modelling environment that provides the metaphor for the development of the computer activity. Second, how children interact with already programmed robots. Finally, the embodiment aspects of the activity since children are using their bodies to interact with the robot.

Seymour Papert and collaborators developed, in the late 1960s, the programming language Logo, which changed how children were using the computer. Instead of being instructed by the computer, they could "teach" a turtle how to do things. This was an active way of constructing knowledge by developing computational products through programming, which Papert called constructionist (Papert, 1986).

Mitchel Resnick expanded Papert's approach and designed StarLogo, a massively-parallel programming language, a screen-based "multi-turtles" program to simulate complex systems' behaviours (Resnick, 1996). With StarLogo people can write rules for thousands creatures on the computer screen, then observe the group-level behaviours that emerge from their interactions. In these decentralized systems, patterns are not determined by some central authority, but by local interactions among many parallel components.

NetLogo, developed by Uri Wilensky is derived from StarLogoT (Wilensky, 1997). Earlier versions of StarLogo were developed for supercomputers and became very popular among researchers, although they were originally intended to be used in schools. NetLogo was developed to satisfy both audiences. It includes almost all the StarLogo features and many new ones (Tisue and Wilensky, 2004).

One of these new features is the "participatory simulations" (Wilensky and Stroup, 1999a), "in which a group of students acts out the behaviour of a system, each student playing the role of an individual element of the system" (Tisue and Wilensky, 2004, p. 4). Each student can play the role of a single agent or a set of agents of a particular phenomenon. For example, students may each "be" an atom in a molecule or a bird in a flock. This feature was accomplished by including a technology called HubNet (Wilensky and Stroup, 1999b). A student operates a NetLogo computer "client" connected through wireless hubs to the NetLogo "server", which "scoops up" student input, processes and displays this collective input, and sends messages back to the students' computers (Abrahamson and Wilensky, 2004).

Even though in the HubNet students can play a role as an agent in the phenomenon being simulated, the interfaces in StarLogo and NetLogo are generally screen-based. In our activity, the children can be seen as physical "turtles" in these systems. They have to interact and coordinate their actions as "multi-turtles" to guide the Robot-Wolf to Grandmother's lab. Our role as researchers was to observe what they were doing and to understand their behaviour as problem solvers.

Charoenying, Gaysinsky and Ryokai (2012) presented a three-parameter model of interaction that takes place in computer-supported instructional environments. This describes how the interplay between a learner's prior knowledge, the enactive experience as the learner is attempting to complete some task objective, and goals embedded into the instructional situation contributes to the emergence of new conceptual schemes. Regarding educational robotics applications, there are two ways children can interact with robots: one, when children construct and program a robot; another, when they interact with an already programmed robot. In this analysis, we focus on the latter. The interaction between children and robots is possible through a variety of ways, such as remote navigation (Druga *et al.*, 2018) in which children remotely control the robot through computer software or a smartphone app, touch (Yadollahi *et al.*, 2018) and haptic interfaces supported by tactile feedback (Yadollahi *et al.*, 2018; Asselborn *et al.*, 2018), and through conversational interfaces (Westlund *et al.*, 2018; Sun *et al.*, 2017; Chandra, Dillenbourg and Paiva, 2017; Leite, Pereira and Lehman, 2017; Leite and Lehman, 2016). In our work, children-robot interaction takes place through their (both children and robot) body actions.

Our literature analysis indicates that interaction among children and robots can be classified as one child interacting with the robot (Westlund *et al* 2018; Sun *et al*, 2017; Leite, Pereira and Lehman, 2017; Leite and Lehman, 2016), a group of children interacting individually with the robot (Asselborn *et al.*, 2018), and a group of children interacting with each other and with the robot (Chandra, Dillenbourg and Paiva, 2017). Chandra, Dillenbourg and Paiva's work is the only one which is related to children's interaction with each other and with a robot, proposing to explore an educational scenario in which a robot acts as a facilitator to a pair of children performing a collaborative writing activity.

In our work, children have autonomy to interact with robots and, simultaneously, with other children in a task-based narrative scenario. As the immersion of children into the physical environment is a fundamental point, we decided to go beyond the traditional storytelling approach — in which stories are read and told — to a role-play approach (Ortiz and Harrell, 2018), where stories and characters are to some extent lived and interpreted. We are particularly interested in investigating and exploring the social components and their role in the interaction with a robot, so to create socioenactive educational scenarios.

The fact that our children are interacting with a physical robot using their body, we understand, as Catlin and Blamires (2010), that this is an embodied activity. The embodiment concept was initially proposed by Maturana and Varela (1987), further elaborated by Varela, Thompson, and Rosch (2016) as *embodied action*. These authors stated that

> By using the term *embodied* we mean to highlight two points: first, that cognition depends upon the kinds of experience that come from having a body with various sensorimotor capacities, and second, that these individual sensorimotor capacities are themselves embedded in a more encompassing biological, psychological, and cultural context. By using the term *action* we mean to emphasize once again that sensory and motor processes, perception and action, are fundamentally inseparable in lived cognition. Indeed, the two are not merely contingently linked in individuals; they have also evolved together (p. 173).

They also proposed the concept of *enaction* or enactive approach to cognition which is based on two principles: first, perception consists of perceptually guided action, and cognitive structures emerge from the recurrent sensorimotor patterns that enable action to be perceptually guided. Kaipainen *et al.* (2011) applied this concept to what they called enactive technology, consisting of a two-way feedback system cycle. In this sense, the behaviour of such systems would not be entirely predictable, since it is the consequence of the impact of the technology on the human, and considers the effect of the human experience on the technology.

In our Socioenactive Systems project, the objective is to expand the enactive technology proposed by Kaipainen *et al.* (2011) to consider the social aspects of the interaction among children with the technology. Our research in this article emphasizes the design of the activity, aiming at promoting a socioenactive experience among children in the technological scenario.

The research questions that motivated this study were: how to design robot-based activity for kindergarten children who have not yet mastered the written language? How to describe their behaviour as they interact with themselves and with the robot and solve the task of commanding it to reach a target? Thus our goal is to describe the process of designing the robot-based activity, to implement this activity in an educational setting and to analyse the data collected to understand the children's behaviour.

Methodology and Constructed Scenario

The designing of the robot activity was based on the co-design methodology (Baranauskas, 2014). The robot-based activity was implemented using the mBot (Block, 2019) artefact, an educational robot toolkit for children, based on Scratch. The mBot was dressed as a wolf in the fairy tale narrative. The system prototype was discussed among the researchers and its first version was used in a workshop with the children's teachers. The teachers' feedback was implemented and the final version was used in the workshop with children in two different scenarios.

The study was conducted using the action-research method (Thiollent, 2000). It was implemented in a school setting for kindergarten students within the University of Campinas campus. Participated in this study twenty-six children (11F, 15M) between 4 and 5 years old (mean 5.31, SD = 0.32). They participated in two workshop sessions. The children were from two different classrooms, led by two different teachers. Classroom 1 had twelve children (4F, 8M) and Classroom2 had fourteen children (7F, 7M). All children had previous experience with computational artefacts, specifically with tangible interfaces (Carbajal and Baranauskas, 2019; Baranauskas and Carbajal, 2017).

The children activities in these scenarios were videotaped. The videos were selected, the activities categorized and analysed in terms of these categories. The data analysis was based on the Grounded Theory (Strauss, 1987; Lazar, Feng and Hochheiser, 2017). The excerpts from the videos were selected, the children's actions were categorized and coded according to the categories defined.

The computational system and its use

The first version of the system prototype was designed driven by elements of the narrative to promote social interaction and cooperation among the children in the adapted Little Red Riding Hood theme. The children's embodied interaction with the robot is a relevant aspect in our study context. Figure 1 shows the different phases for the system development.

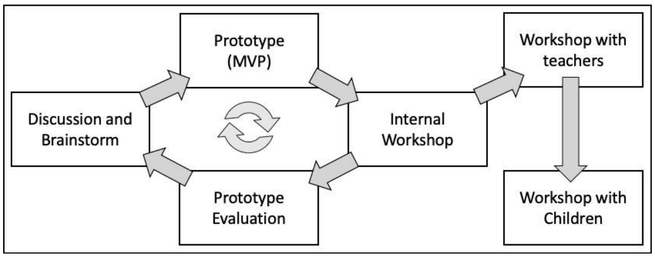


Figure 1: System prototyping process and the system use in two workshops

After a discussion and brainstorming session among the researchers, an initial prototype version was developed and evaluated in an internal workshop in which participated the researchers group, simulating the real environment with children. This led to another discussion and brainstorming session. After 3 iteration cycles, we conducted a preliminary workshop with two teachers, who were invited to use the system playing the role of the children. After this workshop, the teachers proposed that the children should be working in a cooperative way, which changed the children's role in the story, from hunters to rangers who would help the Robot-Wolf complete his task. In addition, the teachers indicated that, as children would be eager to interact with a robot, it would be interesting to introduce a brief warm-up activity, in which each child would have the opportunity to individually interact with the robot as a first step in the workshop.

The mBot (Block, 2019) robot supports a great variety of sensors which can be controlled through a Scratch program. The algorithm states that the mBot must walk forward continuously, until a sensor detects an obstacle. In this situation, the mBot would walk back some steps, changing randomly its direction (to the left or right) in 45 degrees, and then resuming its forward walking.

For the preparation for the workshop with children, the researchers customized the mBot as the Robot-Wolf character, wearing it with fabric ears and a wolf tail. Figure 2a shows the mBot characterized as a friendly Robot-Wolf at the Grandmother's lab. The teachers helped each child to build a customized boot (using ethylene-vinyl acetate sheets and glue) to interact with the Robot-Wolf (Figure 2b). The teachers explained to the children that the only way to interact with the Robot-Wolf would be through their boots, which should be positioned in the path of the robot.

(a) (b)

Figure 2: (a)The mBot is characterized as a friendly Robot-Wolf ; (b) a child interacting with the Robot-Wolf through his ranger's boot

The two workshops sessions with children were performed independently in each class, within an average time period of 90 minutes each. The workshops were organized in two parts, in which two distinct scenarios were explored. The rational of evaluating two scenarios was to understand the way the cooperation and social interactions among the children could change in function of the designed scenarios. Both groups of children (Classroom 1:12 children; Classroom 2:14 children) took part in both scenarios. For the purpose of this article only scenario 1 is discussed.

In scenario 1 (Figure 3), a selected group of 4 children, acting as rangers, guided the Robot-Wolf to reach Grandma's lab so its GPS could be fixed. As the children succeed, a strong applause sound would be heard (played in a hidden notebook). The size of the group was guided by the dynamics of the narrative to provide opportunity for all children of the group interact with the robot. In the Classroom 2, some children participated twice and always the groups had 4 participants. The allocation of the children in the groups was done by the researchers with the help of the teachers. Children sitting side by side were naturally selected to the same group in order to preserve the already existing affinity among the children. No gender criteria were used for the allocation.

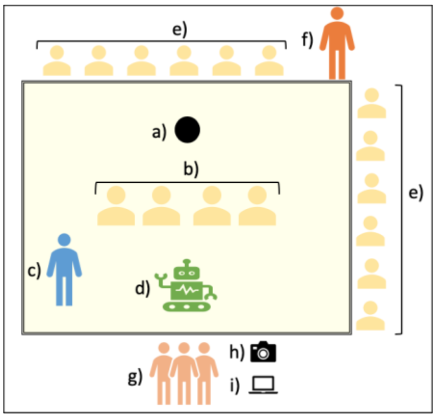


Figure 3: Workshop Scenario 1 scheme refers to: a) Grandma's lab, where children should guide the robot; b) group of children in the current round, acting as rangers; c) researcher, acting as facilitator; d) Robot-Wolf; e) children around the scene, acting as trees; f) teacher; g) researchers; h) recording camera and; i) notebook in which the robot behaviour was implemented. The scene, representing the enchanted forest, is characterized by the rectangle (in the floor) at the centre, which contains the elements a), b) c) and d).

The rangers had the mission of preventing the Robot-Wolf to leave the scene, the enchanted forest in the story, indicated in Figure 3 by the yellow rectangle in the floor. On the situation of crossing the line frontier, a "car crash" sound was played on the notebook, which means the rangers had failed as a group. The children sitting around the scene were oriented to play the role of "enchanted trees", staying seated in their positions and moving only the required to place their boots in the robot path, thus helping the rangers in preventing the robot to get out the forest. Several different children groups acted as rangers (each in its turn).

A researcher acted as facilitator and was responsible for bringing the robot to the starting point between sessions. The other researchers were responsible for other activities, like taking notes, operating the recording camera and the notebook, in which the robot was connected, restarting the game whenever necessary. The children's teacher also helped in the workshop organization.

Figure 4 illustrates a moment of the scenario 1 activity in which a child places his boot in the Robot-Wolf path, forcing it to change its direction, thus reorienting its heading. Meanwhile, the other children in the rangers group shout instructions to each other. The children outside the scene cheer for the rangers, also trying to help by positioning their boots and shouting tips.

Figure 4: Scenario 1 activity: group of children, acting as rangers, guide the robot

towards the Grandma's lab

Results and Discussion

We created a robot-based activity that implemented an analogous physical version of the multi-turtles modelling environments as StarLogo and NetLogo. The major difference is that kindergarten children in our study were instructed about the task and had to autonomously interact among themselves and with the robot, acting as parallel "multi-turtles", so the task could be accomplished.

However, it is as though we are reversing the activity that is done in these programmable modelling environments – we are observing the children's behaviour as they interact with the robot to reach a particular task and trying to understand their behaviour in terms of their engagement in the activity, the way they interact among themselves and with the robot. Differently from being a 'participatory simulation', their participatory embodied interaction (among themselves and with the robot) constitutes the 'program' which solves the task.

For analysing children's behaviour the workshop sessions were fully recorded by video cameras. Grounded Theory (Strauss, 1987; Lazar, Feng and Hochheiser, 2017) methodology was used to analyse the videos since it provides an empirical process to explain the children's interaction which is not foreseen by existing child-child and child-robot interaction evaluation models. First, video excerpts were selected, transcribed into natural language sentences and categorized. The criteria adopted considered the actions related to children, for example, whether an individual or a group was interacting with the robot, and the actions related to the robot, whether they were expected or not.

The robot's actions were categorized as *expected*, according to the programmers' expectations, or *unexpected*, related to bugs or actions not planned when the scenario was designed. Based on preliminary analysis of the video, the children's behaviours were classified into 4 dimensions. For each dimension, relevant properties were identified, totalling 9 properties. Synthetically, the identified dimensions and their properties were:

Awareness. Whether the child' action is: (1) performed intentionally/deliberately by him/her (aware property) or; (2) as an instinctive reaction/involuntary action (unaware property).

Predictability. Whether the child' behaviour is: (1) according to the instructions given before the workshop (expected property) or; (2) not according to the instructions (unexpected property).

Cooperation. Whether the child' action is: (1) a result of an individual initiative (individual property) or; (2) a decision coordinated between two or more children (collective property).

Type of Interaction. Whether the child' action is: (1) an interaction with the robot (robot property); (2) an interaction with another human being (social property) or; (3) it was not an interaction (introspective property).

These categories are relevant to enable further comprehension of social interaction elements in the target scenario. The results are summarized as follows: the children's actions in the proposed scenarios were mostly made deliberately (89.4%); *i.e.*, their actions were performed intentionally towards the objective, according to the underlying narrative of directing the robot to the Grandma' lab. This means they accepted the social roles proposed by the narrative and the scenario. Children were totally immersed in the activity and cooperated via embodied interaction with the robot and among themselves (also via verbal communication and gesture) to attain the goal of the group activity. This suggests a complete engagement with the scenario elements: the robot and its behaviour, their peers, their actions as groups and as individuals capable of acting and changing the situation towards the expected result (*e.g.* indirectly leading the Robot-Wolf to reach the lab). The children mostly acted as expected (62.8%), *i.e.* according to the narrative and scenario rules. Although their actions were mostly initiated individually, illustrating their autonomy in the system (63.7%), coordination with others for achieving the common goal was present (36.3%) signalling the social aspects of the interaction.

Results show a clear focus (57.5%) on the main actor in the scenario — the robot – with actions directed to achieving the robot's desired behaviour, by socially interacting with other children (31%), and by expressing themselves individually (11.5%). The natural engagement, shown individually as well as collectively, added to the joint attention to the robot's behaviour. Joint actions to reach the goal suggest a kind of participatory sense-making for the scenarios experimented.

The frame to design and understand the activity was that of multi-agent modelling, but in this case children are a part of what is going on by intentionally interacting with the agent-robot, the narrative and the others. While it is indeed very complex to understand what learning is going on and what kind of meanings these children make, the results of the activity could reveal some components of a socioenactive experience being constructed: the children's *autonomy* to perceptually driven actions, expressed by their actions coordinated to the robot's actions as co-dependent processes; their *coupling* with the entire environment (themselves, the robot, the others) promoted by the ubiquity of the computational technology; their *embodiment* and the inseparability of mind-body-world, illustrated by their joint focus on the task at hand (at bodies in fact) and; their *emotional* and *social* engagement with the activity.

Conclusion

Our conducted research is part of a much larger project related to the study of Socioenactive Systems. This article focused on the development and use of the robot-based activity as part of this major project. We described how the robot-based activity was developed through the co-design methodology; how this activity was implemented in a school setting for kindergarten children using the action-research methodology; and how the children's actions were analysed through the grounded theory methodology in terms of four categories.

We found that the conducted activity was successfully used with these kindergarten children. Our results demonstrated to which extent they were engaged as well as they behave as a group to solve a particular task. This indicates that the designed activity and system can be adequate to kindergarten children as a way to promote shared meaning construction among them via embodied technology. Our findings can be further explored as a way to investigate how children create knowledge in technology-based environments.

Our next step is to study how the robot-based activity can help children to construct knowledge related to, for example, computational thinking. This is one of the on-going goals proposed in the Socioenactive Systems project.

Acknowledgements

This research is supported by São Paulo Research Foundation (FAPESP), grants #2015/24300-9, #2015/16528-0, #2014/07502-4 and #2018/06416-8. Additional support was provided by the Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES), the National Council of Technological and Scientific Development (CNPq #306272/2017-2) and the University of Campinas (UNICAMP).

References

Abrahamson, D. and Wilensky, U. (2004). *SAMPLER: Collaborative Interactive Computer-Based Statistics Learning Environment.* In: Proceedings of the 10th International Congress on Mathematical Education, Copenhagen, Retrieved from: https://www.researchgate.net/publication/235247149_SAMPLER_Collaborative_interactive_computer-based_statistics_learning_environment.

Asselborn, T., Guneysu, A., Mrini, K., Yadollahi, E., Ozgur, A., Johal, W. and Dillenbourg, P. (2018). *Bringing letters to life: Handwriting with haptic-enabled tangible robots*. In: Proceedings of the 17th ACM Conference on Interaction Design and Children, IDC '18, ACM, New York, NY, USA, 2018, pp. 219–230.

Baranauskas, M. C. C. (2015). *Socio-enactive systems: Investigating New Dimensions in the Design of Interaction Mediated by Information and Communication Technologies*, FAPESP Thematic Research Project #2015/16528-0.

Baranauskas, M. C. C. (2014). *Social awareness in HCI*. Interactions, vol. 21, no. 4, pp. 66–69, Jul. 2014. Retrieved from: http: //doi.acm.org/10.1145/2621933.

Baranauskas, M. Cecilia C. and Carbajal, M. L. (2017). *The Social Nature of Programming: Children and Fluency.* Lecture Notes in Computer Science. 1ed.: Springer International Publishing, v. 10272, p. 291-308.

Block, M. (2019). *Make block inc*. Retrieved from:https://www.makeblock.com/steam-kits/mbot-2, 2019.

Caceffo, R., Abreu, J., Gonçalves, F., Takinami, O., Romani, R., Baranauskas, M.C.C. (2018). *Socio-enactive Systems: The Educational Scenario*. In Technical Report 18-01, Institute of Computing, University of Campinas, SP, Brasil. 40 pages. March, 2018

Caceffo, R., Moreira, E., Bonacin, R., Reis, J., Carbajal, M., D'Abreu, J., Brennand, C., Lombello, L., Valente, J. A. and Baranauskas, M. C. C. (2019). *Collaborative Meaning Construction in Socioenactive Systems: Study with the mBot*. In: Zaphiris P. and Ioannou A. (eds) Learning and Collaboration Technologies. Designing Learning Experiences. HCII 2019. Lecture Notes in Computer Science, vol 11590. Springer, Cham. Retrieved from: https://doi.org/10.1007/978-3-030-21814-0_18.

Caceffo, R., Gonçalves, D., Reis, J., Baranauskas, M. C. C. (2020). *Relatório sobre as Atividades do GT-OntoEscola*. Em Relatório Técnico 20-02, Instituto de Computação, Unicamp, SP, Brasil. 18 páginas. Março, 2020.

Carbajal, M. L. and Baranauskas, M. C. C. (2019). *Exploring and Evaluating "TaPrEC+mBot" Environment with Preschool Children*. In: XXV Workshop de Informática na Escola, 2019, Brasília. Anais do XXV Workshop de Informática na Escola (WIE 2019), 2019. p. 521

Catlin, D. and Blamires, M. (2010). *The Principles of Educational Robotics Applications (ERA): a framework for understanding and developing educational robots and their activities*. In: Proceedings of Constructionism Paris, 2010.

Chandra, S., Dillenbourg, P. and Paiva, A. (2017). *Classification of children's handwriting errors for the design of an educational co-writer robotic peer*. In: Proceedings of the 2017 Conference on Interaction Design and Children,IDCC '17, ACM, New York, NY, USA, 2017, pp. 215–225.

Charoenying, T., Gaysinsky, A. and Ryokai, K. (2012). *The Choreography of Conceptual Development in Computer Supported Instructional Environments*. In: Proceedings of the 11th International Conference on Interaction Design and Children. ACM.1, January 2012, pp.162-167.

Druga, S., R. Williams, R., Park, H. W., and Breazeal, C. (2018). *How smart are the smart toys?: Children and parents' agent interaction and intelligence attribution*. In: Proceedings of the 17th ACM Conference on Interaction Design and Children, IDC '18, ACM, New York, NY, USA, 2018, pp. 231–240.

Kaipainen, M., Ravaja, N., Tikka, P., Vuori, R, Pugliese, R., Rapino, M. and Takala, T. (2011). *Enactive systems and enactive media: Embodied human-machine coupling beyond interfaces*. Leonardo, vol. 44, no. 5, pp. 433–438, September 2011. Retrieved from: http://dx.doi.org/10.1162/LEON a 00244.

Lazar, J., Feng, J. H. and Hochheiser, H. (2017). *Research Methods in Human Computer Interaction* (Second Edition), Morgan Kaufmann, Boston, pp. v – viii.

Leite, I. and Lehman, J. F. (2016). *The robot who knew too much: Toward understanding the privacy/personalization trade-off in child-robot conversation*. In: Proceedings of the 15th International Conference on Interaction Design and Children, IDC '16, ACM, New York, NY, USA, 2016, pp. 379–387.

Leite, I., Pereira, A. and Lehman, J. F. (2017). *Persistent memory in repeated child robot conversations*. In: Proceedings of the 2017 Conference on Interaction Design and Children, IDC ’17, ACM, New York, NY, USA, 2017, pp. 238–247.

Maturana, H., and Varela, F. J. (1987). *The Tree of Knowledge: The Biological Roots of Human Understanding.* New Science Library, Boston.

Ortiz, P. and Harrell, D. F. (2018). *Enabling critical self-reflection through roleplay with chimeria: Grayscale*. In: Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play, CHI PLAY ’18, ACM, New York, NY, USA, 2018, pp. 353–364.

Papert, S. (1986). *Constructionism: A new opportunity for elementary science education.* A proposal to the National Science Foundation, Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group, Cambridge, Massachusetts.

Resnick, M. (1996). *StarLogo: An Environment for Decentralized Modeling and Decentralized Thinking*. In Conference Companion on Human Factors in Computing Systems. CHI ’96, pp.11–12.Strauss, A. L. (1987). *Qualitative Analysis for Social Scientists*. Cambridge University Press, Cambridge.

Sun, M., Leite, I., Lehman, J. F., and Li, B. (2017). *Collaborative storytelling between robot and child: A feasibility study*. In: Proceedings of the 2017 Conference on Interaction Design and Children, IDC ’17, ACM, New 619 York, NY, USA, 2017, pp. 205–214.

Thiollent, M. (2000). *Metodologia da pesquisa-ação*. 10th edition. Cortez, São Paulo.

Tisue, S., and Wilensky, U. (2004). *NetLogo: A simple environment for modeling complexity*. Paper presented at the International Conference on Complex Systems, Boston, May 16–21.

Varela, F. J., Thompson, E. and Rosch, E. (2016). *The Embodied Mind: Cognitive Science and Human Experience*. Second edition. MIT Press, Cambridge, Mass.

Westlund, J. M. K, Park, H. W., Williams, R. and Breazeal, C. (2018). *Measuring young children’s long-term relationships with social robots*. In: Proceedings of the 17th ACM Conference on Interaction Design and Children, IDC ’18, ACM, New York, NY, USA, 2018, pp. 207–218.

Wilensky, U. (1997). *NetLogo Ants model*. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. Retieved from: http://ccl.northwestern.edu/netlogo/models/Ants.

Wilensky, U., and Stroup, W. (1999a). *NetLogo HubNet* . Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University.

Wilensky, U., and Stroup, W. (1999b). *Learning through participatory simulations: Network-based design for systems learning in classrooms*. Paper presented at the International Conference of the Learning Sciences (ICLS), Atlanta, GA.

Yadollahi, E., Johal, W., Paiva, A. and P. Dillenbourg, P. (2018). *When deictic gestures in a robot can harm child-robot collaboration*. In: Proceedings of the17th ACM Conference on Interaction Design and Children, IDC ’18, ACM, New York, NY, USA, 2018, pp. 195–206.

An initial experience with Logo in a multigrade rural school in Mexico

Ana Isabel Sacristán, *asacrist@cinvestav.mx*
Department of Mathematics Education, Cinvestav, Mexico City, Mexico

Homero Enríquez-Ramírez, *homero.enriquez@cinvestav.mx*
Department of Mathematics Education, Cinvestav, Mexico City, Mexico

Abstract

In this practice paper, we focus on the experience of two primary-school teachers who participated in a PD programme, in introducing Logo, for the first time, in a rural, multigrade, primary-school in the state of Oaxaca, Mexico, for teaching mathematics. The school (Figure 8) consisted of two classrooms where the two teachers taught children in Grade 1-3 and 4-6, respectively. The school only had a 12-year old desktop computer as hardware.

The two teachers collaborated in designing a few Logo tasks for addressing two topics from the compulsory primary-school curriculum: spatial awareness and the representation of geometric figures (polygons). For that, they borrowed some laptops so that the children (who, for the most part had never touched a computer) could carry out the Logo tasks in teams (Figure 9).

The inexperience of the teachers led to some difficulties in understanding the rotation of the turtle as an angle; and also in providing, from the beginning, a more constructionist experience and allowing children more control.

Nevertheless, the experience was very positive, with children eventually being empowered and able to carry out their own projects while learning from their mistakes. In general, the experience illustrates how a small intervention, working within the conditions and teacher knowledge available, can create motivating learning experiences and introduce students to a meaningful use of digital technologies, to which they had no access before.

Figure 8. The multigrade primary-school in rural Mexico (in a region of the state of Oaxaca)

Figure 9. Students working with Logo at the multigrade primary-school in rural Mexico

Keywords

Digital Technology, Logo, Rural primary-school, Teaching, Mathematics, Geometrical figures and angles.

Introduction

The situation regarding digital technologies in Mexican (rural) primary-schools

In rural Mexico, the use of digital technologies (DT) in schools, particularly in state primary-schools, is still scarce, as observed in a recent study in the lower Mixtec region of the state of Oaxaca (Enriquez & Sacristán, 2019a). One problem is the lack of hardware and software (ibid). Another problem is that teachers have not had sufficient professional development (PD) for the integration of DT to their teaching practice, and particularly in their didactical use (Enríquez & Sacristán, 2017). In fact, as has also been noted by others (Santiago, et al., 2013; Trigueros, Lozano & Sandoval, 2014) as well as ourselves (Enriquez & Sacristán, 2019b), the ways in which teachers use, if at all, DT, is mainly as a *replacement* and *amplifier* –using the categories proposed by Hughes (2005) for describing pedagogies supported by technology– rather than as a *transformative* use, as would be expected in a constructionist approach.

This situation of limited digital resources and teacher training for their integration is not exclusive to this region of Mexico. In the past fifteen years, there hardly have been initiatives to integrate DT in Mexican state schools. The last significant programmes by the Ministry of Education, were in the early to mid 2000s –the Enciclomedia programme for primary education, where schools were equipped with computers and smartboards, and interactive apps accompanied the official textbooks; and the EMAT (Teaching Mathematics with Technology) programme that introduced Spreadsheets, Dynamic Geometry and Logo programming for mathematics in middle-school education (see Sacristán & Rojano, 2009; and Trouche et al., 2012). But after 2006, DT in state schools, and the training of teachers on their pedagogical use (such as for the teaching of mathematics), became very limited or non-existent.

Thus, in the year 2018-2019, we implemented a PD programme to train 15 participants from our previous study (Enriquez & Sacristán, 2019a), seeking to address some of the challenges identified in that study and help them integrate DT (including Logo) to their practice.

Aim of the paper

In this practice paper, we describe the experience of two teachers who participated in our PD programme, when they introduced Logo for the first time in a rural, multigrade, primary-school in the state of Oaxaca, Mexico, for teaching mathematics.

The professional development programme

For the design of our PD programme, we took into account, as identified in our previous study (Enriquez & Sacristán, 2019a), the schools' and teachers' reality in this rural area of Mexico. We designed a face-to-face course (with official validity) in a venue close to the teachers' schools and/or homes. We chose a few offline (i.e., not requiring an Internet connection) digital resources, that could run on the old hardware available in the region's schools. These had to be easy for teachers to use, allowing for a significant and constructionist implementation in class, for "doing mathematics" rather than learning about mathematics, as promoted by Papert (1971).

The PD course was implemented over a period of five months, organized in three modules to cover three types of digital resources, respectively: diverse interactive apps for specific topics (including the Enciclomedia apps); dynamic geometry (GeoGebra); and Logo. The interactive applets in the first module deal with specific curricular contents and are easy to manipulate. GeoGebra helps address many geometry activities of primary-school. Finally, Logo also helps address the compulsory curricular contents of primary-school mathematics, but through programming –allowing children to express themselves (construct), be creative, and develop computational thinking (an important skill in today's world). Furthermore, Logo's Turtle Geometry allows for "body syntonicity" opening "young programmers to affordances in tune with their body languages" inducing "their own reasoning, knowledge and skills to which the mind in the head may or may not have access" (Clayson, 2018, p. 33).

We chose Logo over the nowadays popular Scratch, because: Logo has less technical requirements and can be run offline on the available computers; Turtle Geometry is more straightforward in Logo, which also has less "distractions" than Scratch, allowing students to engage more directly in math-related tasks. Also, by having to type Logo-commands, rather than choosing Scratch blocks, children may need to pay attention and reflect more on what they are doing (as well, perhaps, on the implicit mathematics involved).

We didn't just introduce teachers to digital resources; rather, we aimed to promote a use that was transformative (Hughes, 2005), changing teachers' pedagogical practices and students' learning-processes and roles in the classroom. Thus, the work with teachers during the PD programme attempted to follow constructionism principles (Papert, 1991; Kynigos, 2015; Sacristán, 2018): e.g., by having tasks that would promote active, collaborative and autonomous exploration and expressivity of the learners. Teachers and trainers also engaged in discussions of strategies for using the DT resources in constructionist ways, taking into account the material limitations (e.g. of hardware), in terms of group organization, didactical interventions, etc.

The activities of each PD module were carried out in four stages: (i) Introduction to each resource through worksheets and in-class experimentation and discussions on how it could contribute to the teaching and learning of mathematics, accompanied by readings on pedagogical recommendations, including some on constructionism and Logo (e.g., Papert's, 1980, *Mindstorms*). (ii) Designing (with teams of teachers working together) math-tasks and experiences for their students, with each resource; teams developed their own plan, taking into account their particular school characteristics, syllabus and students. (iii) Implementing the designed tasks in their schools. (iv) Discussing the classroom-experience with the other PD participants.

The experience described here, is the implementation that two teachers, Isabella and Thelma, did in their school of the Logo tasks designed during the PD programme.

The two teachers and their school

At the time of our observations, Thelma had 16 years of service and Isabella, 11, as primary-school educators. Both are enthusiastic and dedicated teachers who constantly engage in school-projects (such as participating in our PD programme) for improving their practice and educational results. However, although they had a positive attitude towards DT, neither had training in their didactic use, scarcely using any DT in their practice (only Thelma had used some Enciclomedia resources), and unaware of what could be used for mathematics (other than the Enciclomedia apps).

The school in which they worked, is a multigrade (Grades 1-6) rural primary-school in Oaxaca, consisting of two classrooms, an administrative room, a library and a courtyard (see Figure 8). Isabella and Thelma were the only teachers at the school. At the time of our study, 30 children (ages 6 to 12) were enrolled. For most activities, Isabella was in charge of Grades 1-3 (13 children); and Thelma of Grades 4-6 (17 children).

The school had only one computer, a projector and a non-working smartboard used as a screen –Enciclomedia hardware dating from 2006 (shown in Figure 9)—, and no Internet, nor mobile, connectivity. This hardware, located in the administrative room, was in poor condition due to age and lack of maintenance; its use had been mainly for administrative tasks; rarely, if at all, for didactic purposes. But it was what was available for the digital resources' task implementations. The students, for the most part, had never had any contact with computer equipment before.

Introducing digital technologies and Logo to the students

In order to integrate each module's digital resources to their practice, Isabella and Thelma had to hold class in the administrative room where the only computer available was located: they brought chairs and tables for the students, and projected the DT work with the single computer.

During the first two modules of the PD programme, they used the room separately. But for the Logo module (after 5 Logo PD training-sessions), Isabella and Thelma designed a joint class-plan (see previous section) where they wanted to cover the compulsory curricular topic, for all school grades, on geometric figures (polygons) and spatial awareness; so they joined the two groups of children. They used the MSWLogo[12] version dating from EMAT program (i.e. from 2004) that ran in the old hardware available in their school. Next we describe how they put into practice their class plan.

First day: introducing Logo as a group activity

First, children drew polygons (squares, rectangles and triangles) on paper and on the courtyard floor; then the teachers had them "play turtle" (Figure 3) through verbal commands, such as "walk forward 5 steps" or "turn right" (with no measurement), although the children had not yet been introduced to Logo, nor its turtle. Back in the classroom, the teachers introduced the turtle and the basic primitives (Forward, Back, Right, Left, ClearScreen), which they wrote in a poster (Figure 11). They also made a poster with different types of angles and their types of measure (Figure 12); but this led to difficulties tracing figures with Turtle Geometry (see further below).

Figure 10. Children "playing turtle" by following the teachers' commands

Figure 11. The teacher-created poster with the basic Logo primitives

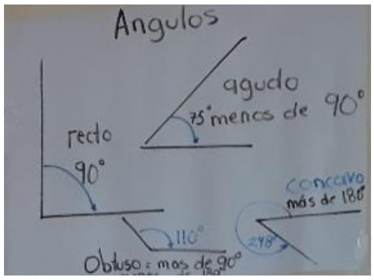


Figure 12. Poster showing the different types of angles (right angle, acute, obtuse, reflex)

Figure 13. A girl shows the turtle's turn-direction

Figure 14. Girl shows her team the turtle's turn-direction

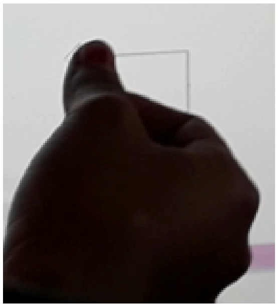


Figure 15. Another student shows the turtle's turn-direction

The teachers asked a student (one of the oldest girls) to come forward and point the direction of the turn (Figure 13). Isabella also stood in front of the projected turtle to show its direction, telling the children: "[the turtle] has now stayed like this, how should we now place it?" This was a strategy that the teachers used over the entire session to help children identify the direction of the turtle's turn for the construction of the polygons. In later sessions, children would use this strategy (Figure 14 and Figure 15). After identifying the turtle's turn-direction, Isabella typed the Right primitive and

[12] See http://www.softronix.com/logo.html and http://www.matedu.cinvestav.mx/~asacristan/LogoEMAT.exe for the EMAT version.

asked the older children (of grades 4-6), who were familiar with angles, for the turtle's turn-measurement, needed to produce a square. Determining the measure of the turn was a challenge for the children: Some first said "Right 150"–repeating the same input value as the one for the side–, which of course was not the right angle; the children tried to correct it, by proposing various turns left and right, but were not able to produce the expected figure (Figure 16). Professor Thelma intervened, pointing to the poster with the angles; the older children then said that the angle had to be 90. After drawing the second side of the square, Thelma asked "I'm going to turn left, how many degrees?"; one student again said "150" but the teacher emphasized it had to be angles, so children quickly corrected their answers to "90". However, in a square the turning angle and the square's internal angle are both 90, so neither the teachers nor the students realized there were dealing with different angles.
The teachers then proposed drawing a triangle: Isabella started with RT 90, so the first side would be horizontal, then typed the commands for the first two sides of a right triangle (FD 150, LT 90, FD 150), similar to tracing a square. She then asked for the turtle's turn-measurement needed for the third side of the triangle, but none of the children's proposed values worked. Thelma intervened, reminding and showing the students the 90º and 45º angles. She placed her finger to where the turtle had to turn, but pointing to the internal angle of the triangle (not the turning angle), and asked if the triangle's angle would be greater or less than 90º and for its value. Some children said less, and proposed "70", so she typed "LT 70". When the teachers saw that it was not what they expected, they abandoned the activity deleting what had been done. Since, in a square, the turning and internal angles are both 90, neither teachers, nor students, realized that the turtle's turn and the internal angle are different. Isabella finished the class drawing an equilateral triangle, by copying the commands, with 120º turns, that she had seen in the PD course, but with no discussion, nor reflection.

Figure 16. First attempt to draw a square

Figure 17. The younger children in a Logo session with Isabella

Second day: Children working directly with Logo in teams

The following day, Thelma brought her own laptop and two other borrowed ones (for a total of four computers: 3 laptops and the school's desktop), so that the students could work with Logo. In a morning session, Isabella worked in her classroom with the younger children (Grades 1-3) and the laptops, while Thelma used the administrative room and desktop to work with the older children (Grades 4-6). They both planned working with squares and rectangles; no longer with triangles, to avoid the previous day's problems.

In the morning session, Isabella divided the group in four teams of 3 students (she didn't feel comfortable having larger teams): while three teams took turns with the laptops, the remaining team's students drew in notebooks (Figure 17). She asked them to draw a square and a rectangle with the Logo words (primitives) they had learned the previous day. Many children had never used a computer before, so Isabella helped them with the keyboard, mouse, cursor, etc. (Figure 17). Because it was the first time working directly with Logo, children also needed to learn to act like the turtle, to figure out turtle's turn-direction. They either simulated the movement of the turtle with their body (Figure 14) or placed their hands in front of the computer screen in order to orient themselves (Figure 15). Isabella also helped them with the turtle's orientation (particularly when the turtle faced sideward). In writing the commands, children had trouble initially because they omitted the space between a command and its input (later, the older children developed a strategy to write, in their notebooks, hyphens to represent those spaces –Figure 21 and Figure 22, further below). For drawing a square, they remembered that the angle was 90, even though, at this stage, the concept of angle was not clear to them. But they quickly got the hang of working with Logo. For the rectangle, after Isabella reminded them that some sides were longer than the others, the students drew it quite easily. The children were delighted with these activities and when this first session was over, they didn't want to go out to recess.

In the administrative room, Thelma divided the 17 older students in five teams of 3-4 students. Teams took turns drawing polygons with Logo on the desktop computer. These older children had worked with apps and GeoGebra in the previous months, so they were more familiar with the computers than younger children. Before their turns, each team drew in their notebooks the figures they wanted to trace with Logo (squares and rectangles), writing the instructions they would have to type. In general, children didn't have many problems and other teams cheered when each figure was completed. When they all finished (which was quite quickly), Thelma improvised an activity in the schoolyard where the children traced different polygons (squares, rectangles and triangles) on the floor using a protractor, writing the instructions for drawing them in Logo (Figure 18). However, she again did not realise that she was measuring the internal angles and not the turtle's rotation angles. After the recess, Thelma exchanged rooms with Isabella, so that the older children could use the computers; the teams tried to reproduce in Logo the triangles traced in the courtyard, but obviously their written instructions didn't work. So they began experimenting in Logo; when they remembered the instructions for an equilateral triangle given the previous day, they drew such triangles. Thelma didn't intervene –we suspect she was confused with the angles, despite five PD sessions where the difference between the rotation angle and the internal angles was discussed: Teachers seems to have an instilled way of thinking of the angles of figures as always internal. Isabella referred to this difficulty with the angles in a later interview, saying that the measure of the turtle's turn was still confusing because it wasn't the internal angle of the figures.

Figure 18. Thelma draws figures in the courtyard with a protractor to help determine the angles

Figure 19. The younger children try to trace the boat with Logo

Figure 20. Failed attempt at drawing the boat

After the recess, Thelma's students planned personal projects of complex figures to be traced in Logo (and would another day), writing the instructions on paper. In the meantime, Isabella's group of younger children used the computers in the administrative room: The students followed a worksheet from the Ministry of Education (SEP)'s EMAT Logo activity book for middle school (Sacristán & Esparza, 2005), that focused on drawing a boat with the aim of learning about different ways of inputting commands. She assumed it would be a straightforward activity because children would be copying commands. But this activity uses negative numbers (e.g., RT -90, BK - 29), operations (e.g., FD 220 / 2, FD 20 * 7) and parenthesis –symbols that the younger children had never seen. Since they didn't understand the symbols, they tended to skip them and they couldn't trace the boat (Figure 19 and Figure 20). When faced with this situation, Isabella translated the commands to simple and positive inputs, and the children still had great fun in being able to trace the boat.

Isabella and Thelma then exchanged rooms and Thelma's students were given the "Boat Worksheet" to work with Logo. But in view of what happened with Isabella, Thelma translated immediately all the commands to simple inputs and the activity was very fast, so the older children did not explore the operations, nor the meaning of the symbols.

The teachers explained that they hadn't thought of analysing the worksheet before giving it to the children, because they had already done it in the PD sessions. Nevertheless, children from both groups were so motivated and immersed in the Logo activities that the entire schoolday was devoted to Logo (either with the computers, or in writing with chalk or pencil).

Third day: Drawing more complex figures

Six days later, only Thelma and the older children were present and had access to the 3 laptops and the desktop. The children had the opportunity to try out, on their own, their personal projects, from the second Logo workday, of more complex figures –more expressive and constructionist activities. We observe that, in their plans, the confusion between the turtle's turn and the internal angles remains. For example, in the written instructions for drawing a flag (Figure 21) the children marked the internal angles (the Logo commands would still work because they were tracing rectangles and the internal angle coincides with the turning angle), but it shows the problem transmitted by the teachers.

Figure 21. Students' plan for a Flag

Figure 22. Students' plan for an Arrow

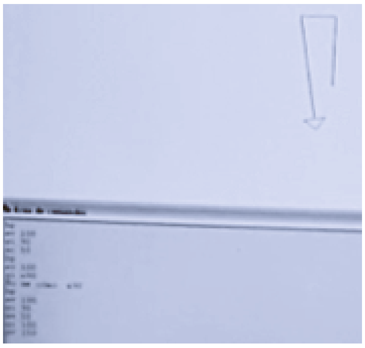


Figure 23. Failed attempt for the arrow

Figure 24. New attempt for the arrow

Another team of students attempted to draw an arrow (Figure 22). They copied their planned instructions from their notebook, but after running the first commands (FD 100, LT 90, FD 50, LT 100), a member warned: "those made me go wrong" (see Figure 23). They tried to continue but realized the expected figure was not achieved and started afresh by trying turns according to what they saw on the screen. After experimenting with Logo, it was clear to them that for drawing right angles, the input had to be 90; but with other turns the children tried different options until they reached one that seemed right. For the arrow, they realized that a turn of 120 or greater than 90, was more effective than smaller turns (Figure 24). Under this trial-and-error approach, they were able to carry out their projects.

In later days, children kept asking when they would get to work with the computers and Logo again, because they loved the experience. Unfortunately, they wouldn't get a chance again in that school year: the teachers couldn't borrow laptops again; Isabella didn't dare work with all the younger students with a single computer, while Thelma was distracted by another project.

Results of the experience and discussion

On the development of spatial awareness and the concept of angle as rotation

The Logo tasks did seem to fulfil the teacher's aims and compulsory curriculum of promoting the development of spatial awareness and learning about polygons, but also going beyond curricular standards. In relation to spatial awareness, children used laterality and spatial imagination to place themselves in, or mimic, in a physical or imaginary way, the turtle's position in order to define the turn-direction. Further to that bodywork, children (and teachers) had to reflect, not only on the figures' internal angles, but also the turtle-turn angles, thus expanding their conceptions of angle.

Nevertheless, for both students and teachers, measuring turtle-turn angles –seeing angles as rotations, rather than as a measure between two intersecting segments– was new and challenging, conflicted with their previous knowledge and made teachers question their knowledge of that concept. Research has shown that it is not easy for children to incorporate the turn into the concept of angle –and as we have seen in this paper, it isn't either for teachers– because the experiences with physical rotation (e.g., a door, a windshield wiper of a car, etc.) do not result in reflective abstractions that lead to capturing elements of the mathematical concept of angle (Mitchelmore & White, 2000). Thus the difficulty in conceiving the turtle's turn as an angle. In fact, Clements and Burns (2000) explain that turns in Turtle Geometry integrate two schemes: turn as body-movement and turn as number (a measure). Moreover, Mitchelmore and White (2000) warn that work with Logo needs to be complemented by pedagogical mediation for children to develop angle abstractions and transfer them to other contexts. But in our case, the teachers themselves hadn't integrated this view of angles, which made that mediation impossible. We believe that the teachers' difficulties come from a deeply instilled way of thinking of angles of figures as always internal (i.e., as they have, previously, taught angles in polygons and figures); this became an obstacle to thinking about angle otherwise. The sessions in the PD course were insufficient to change that, despite some pedagogical mediation. It is clear that, in the PD course, we took for granted the understanding of the turtle's turn angle: more time and attention needs to be given to this and the mathematical elements involved (e.g., emphasizing more that the inner angle and the rotation angle are supplementary). In the children's case, the obstacle of dominant inner angles thinking was more easily overcome, because it wasn't as deeply ingrained and they "got the hang" of Logo through trial and error. But of course, in their case, deeper mathematical knowledge is yet to be developed.

Appreciation of the Logo experience: motivation, collaborative work and DT integration

The Logo implementation promoted collaborative work among teachers and among children: both teachers agreed on a topic for children of all grades; and children worked together in teams of mixed ages. The Logo activities also combined courtyard, paper-and-pencil and computer work. Both teachers appreciated that the children became very creative, interested and motivated in the novel activity, and worked collaboratively.

Thelma valued how children's imagination developed, how they created figures using their own instructions, and that teamwork enhanced students' learning "through the exchange of ideas, clarification of doubts, correction of mistakes and [strategizing] to find correct results". In a report, she wrote that Logo helped children "develop their cognitive and logical abilities, and find independent processes that helped them improve their learning, get quick and easily verifiable results, and be able to identify their mistakes".

She referred, however, to the challenge of using DT resources: in her 15 years of teaching practice, she had never used DT for mathematics. Also, some children had never touched a computer before, and encouraging them to use it was not easy because she lacked confidence herself. But the outcome was very positive: using the DT resources (GeoGebra, Logo), gave her confidence and a new outlook on how to teach mathematics, where children explore, make mistakes, do and undo, etc. It also "turned her teaching upside-down": instead of explaining concepts and procedures followed by examples, Logo tasks started with experiences and problem-solving that then led to the concepts and procedures. She felt that this was very enriching (not only in class with the children, but also when she shared it with the other teachers in the PD course). She also expressed her surprise in seeing underachieving children (some of which drew a cross as their project) rise above expectations and even above what some "better" students did.

Being a Grade 1-3 teacher, Isabella (and her students) had also never used DT, and she lacked confidence in implementing it. She was very surprised by students' working abilities and of thinking on their own during the Logo tasks. She also appreciated that children learned about laterality, polygons and angles, in spite of the confusions between internal angles and turtle's turns; she said that she became aware of the need to be more attentive in analysing the angles.

Both students and teachers were very motivated by the Logo experience (particularly when doing their own projects). One student said: "For me, this was like a party". Another said: "With the chalk, I simply traced [the figure] directly, but with Logo I had to think".

The experiences mirror many that were documented during the EMAT program (Sacristán & Rojano, 2009) and reflect Papert's (1980) descriptions on the value of working with Logo.

On the implementation of the Logo tasks and their constructionist (or not) nature

The experience, however, highlights the difficulties in training teachers not only in math, but in the pedagogical and constructionist use of digital resources. In the first sessions, it was difficult for the teachers to grant children control during the tasks. For instance, when "playing turtle" in the courtyard, it was the teachers who dictated the commands. And in the first experiences drawing with Logo, the teachers also "dictated" what to do. These point to the limitations that these inexperienced teachers had in harnessing the potential of Logo with their students. And when using the "Boat Worksheet", we also saw a loss of fidelity (Noss & Hoyles, 2019) of the activity's design. On the other hand, Thelma, in particular, did empower her students in the end (providing a real constructionist experience), letting them engage in their own personal projects; this became the richest and most motivating experience for the children.

Final remarks

Forty years ago, Papert (1980) envisioned a future where children would have their own computers and their (constructionist) use (such as with Logo) would transform education and learning. The above experience shows that in many places there is divide that places this vision far from reality: it highlights the challenge of introducing constructionist implementations of digital resources (e.g., Logo) given limited hardware and contextual conditions, even when those are taken into account in a PD course design. Conditions such as those of the school described here, where: (i) the only two teachers/staff are insufficiently trained (as identified in our diagnostic study –Enríquez & Sacristán, 2019a), both in mathematics (which explains some of their difficulties in understanding the relationship between the internal and rotation angles); as well as in the use of DT and how to harness these; and (ii) students also have no previous experience using DT. The DP course aimed to overcome some of the teachers' limitations, but it is clear that more time and practice are needed for teachers to appropriate themselves of both the tools and (constructionist) pedagogical model. This experience points to the need of more efforts between researchers, authorities and teachers for promoting a meaningful integration of DT for the teaching and learning of mathematics in schools.

On the other hand, the experience also shows how a small intervention, working within the available conditions and teacher knowledge, can create motivating learning experiences and a meaningful use of technologies, to which teachers and students had no access before. It was also a constructionist experience in itself, that helped both teachers' PD by having them face how to use DT tools and tasks for mathematical learning, organise the children, break with their usual ways of teaching and also of understanding mathematical concepts (e.g. of angles), and accept that they can empower children.

Acknowledgements

We thank the two teachers and the children whose experiences we described in this paper.

References

Clayson, J. (2018) Look Closely, Watch What Happens: Visual Modelling and Constructionism. In V. Dagienė & E. Jasutė (Eds.), *Constructionism, Computational Thinking and Educational Innovation: Constructionism 2018 Conference Proceedings.* (pp. 30-37). http://www.constructionism2018.fsf.vu.lt/proceedings

Clements, D. H., & Burns, B. A. (2000). Students' Development of Strategies for Turn and Angle Measure. *Educational Studies in Mathematics*, 41(1), 31-45. https://doi.org/10.1023/A:1003938415559

Enríquez, H. & Sacristán, A.I. (2017). Acceso y Uso de Tecnologías Digitales para la Enseñanza de las Matemáticas. *Seminario Nacional de Tecnología Computacional en la Enseñanza y el Aprendizaje de las Matemáticas 2017*. Pátzcuaro, Michoacán, México

Enríquez, H. & Sacristán, A.I. (2019a). La Orquestación y Uso de la Tecnología Digital en la Enseñanza de las Matemáticas en la Escuela Primaria: el caso de algunos profesores de la Región Mixteca de Oaxaca. *Seminario Nacional de Tecnología Computacional en la Enseñanza y el Aprendizaje de las Matemáticas* 2019. Chetumal, Quintana Roo, México.

Enríquez, H. & Sacristán, A.I. (2019b). Cómo Profesores de Primaria Enseñan Matemáticas con Tecnología Digital: Dos Estudios de Caso de una Zona de Oaxaca. *5º Coloquio de Doctorado en Matemática Educativa.* Mexico City, Mexico: DME-Cinvestav.

Hughes, J. (2005). The Role of Teacher Knowledge and Learning Experiences in Forming Technology-Integrated Pedagogy. *Journal of Technology and Teacher Education,* 13 (2), 277-302. https://www.learntechlib.org/primary/p/26105/

Kynigos, C. (2015). Constructionism: Theory of Learning or Theory of Design? In S. J. Cho (Ed.), *Selected Regular Lectures from the 12th International Congress on Mathematical Education* (pp. 417–438). Cham: Springer. https://doi.org/10.1007/978-3-319-17187-6_24

Mitchelmore, M. C., & White, P. (2000). Development of angle concepts by progressive abstraction and generalisation. *Educational Studies in Mathematics*, 41(3), 209-238. https://doi.org/10.1023/A:1003927811079

Noss, R., & Hoyles, C. (2019). Micromundos, Construccionismo y Matemáticas. *Educación Matemática*, *31*, 7–21. https://doi.org/10.24844/EM3102.01

Papert, S. (1971). *Teaching Children to be Mathematicians vs. Teaching About Mathematics* (Artificial Intelligence Memo 249 / Logo Memo 4). http://dspace.mit.edu/handle/1721.1/5837

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books.

Papert, S. (1991). Situating constructionism. In Harel, I. & Papert, S. (Eds.), *Constructionism: Research reports and essays* (pp. 1-18). Westport, CT: Ablex Publishing Corp. http://www.papert.org/articles/SituatingConstructionism.html

Sacristán, A. I., & Rojano, T. (2009). The Mexican National Programs on Teaching Mathematics and Science with Technology: The Legacy of a Decade of Experiences of Transformation of School Practices and Interactions. In A. Tatnall & A. Jones (Eds.), *Education and Technology for a Better World* (Vol. 302, pp. 207-215). https://doi.org/10.1007/978-3-642-03115-1_22

Sacristán, A. I. (2018). Constructionist Experiences for Mathematics Across Educational Levels. In V. Dagienė & E. Jasutė (Eds.), *Constructionism, Computational Thinking and Educational Innovation: Constructionism 2018 Conference Proceedings.* (pp. 83–93). http://www.constructionism2018.fsf.vu.lt/proceedings

Santiago, G., Caballero, R., Gómez, D. & Domínguez, A. (2013). El uso didáctico de las TIC en escuelas de educación básica en México. *Revista Latinoamericana de Estudios Educativos* 43(3). http://www.redalyc.org/html/270/27028898004/

Sacristán, A.I. & Esparza, E. (2005). Programación Computacional para Matemáticas de Nivel Secundaria. Programa EMAT. México: SEP. http://www.matedu.cinvestav.mx/~asacristan/LOGO_EMAT_ActividadesPUBc.pdf

Trigueros, M., Lozano, M.D. & Sandoval, I. (2014). Integrating Technology in the Primary-school Mathematics Classroom: The Role of the Teacher. In A. Clark-Wilson, O. Robutti, & N. Sinclair (Eds.), *The Mathematics Teacher in the Digital Era* (Vol. 2, pp. 111-138). https://doi.org/10.1007/978-94-007-4638-1_6

Trouche, L., Drijvers, P., Gueudet, G., & Sacristán, A. I. (2012). Technology-Driven Developments and Policy Implications for Mathematics Education. In M. A. (Ken) Clements, A. J. Bishop, C. Keitel, J. Kilpatrick, & F. K. S. Leung (Eds.), *Third International Handbook of Mathematics Education* (pp. 753–789). https://doi.org/10.1007/978-1-4614-4684-2_24

Build and Animate Your World Construction & Computing from Six to Ten

Michael Doyle, *logios.org@googlemail.com*
Dept of Design & Technology, British School of Sofia, Sofia, Bulgaria

Vesselka Ilieva, *vessela.ilieva@abv.bg*
Primary Dept, NSOU Sofia, Sofia, Bulgaria

Abstract

The time and place relocation of Constructionism 2020 was a minor side-effect of Covid-19. More telling was the closure of schools, particularly the primary phase, when it became starkly obvious that the ubiquitous smartphone, so superb for gossiping, was an inhibitor of online teaching. The long abandoned home laptop might sometimes be resurrected but in many cases the poor teacher had to squint at out of focus snaps of exercise book pages. This highlights an unregarded truth about constructionism: it can have negative as well as positive consequences. This means that a constructionist teaching method may actually inhibit learning.

The introduction of microcomputers in the early 1980s led, by their end, to a few teachers having a good understanding of what the new medium offered. Very few had the chance to influence policy, which was driven by academics and the computer industry. Serendipity led one jurisdiction in Eastern Europe to delegate the primary school curriculum to a couple of teachers. Their design was to introduce children to all the possibilities the computer offered. An option within it was "Build and Animate Your World" based on LEGO and DOS Logo. This latter is our anchor subject.

Certain possibilities that the computer, aka Turing machine, offered have been resisted by those who control education. Notably, primary education is still paper based. This situation is considered in the light of the aim to raise standards, with issues of assessment. Also considered is the damage caused by certain leading constructionists. Our quarter century of prototyping offers a positive and very relevant pathway as Ireland hesitates on the brink of reform to its primary school curriculum.

Figure 1. One of the spring projects built and animated by children at the British School of Sofia.

Keywords (style: Keywords)

Primary school, construction and computing, curriculum, teaching method, standards.

A Quarter Century of Prototyping

At the inaugural Constructionism gathering, Ilieva (2010) presented a summary of fifteen years primary school teaching using LEGO construction materials animated with Dacta Control Lab. Her project-oriented method was based on small teams of children within a class modelling familiar situations. These models were then incorporated into a whole-school scene on a seasonal theme.

Her method was adopted by Doyle (2013, Ó Dúill 2020) who extended its scientific, mathematical and language aspects. This was motivated by the requirements of the English national curriculum requirements in design and technology for KS 1 and 2. These developments were consolidated in 2019 in a British School curriculum document.

Both teachers evaluated later LEGO robotics kits. WeDo, targeted at primary education, compared badly with Control Lab as adapted for and integrated into primary school teaching. It was retained.

Control Lab: fit for purpose

Technology companies are driven by innovation as toymakers are by novelty. The obsolescence cycle is shorter than a child's primary school career. Mirrored in the programmable offerings from the LEGO Company, this contrasts starkly with the longevity of their plastic parts. Its electrical and electronic parts, and current pictographic software, are increasingly inflexible and closed to users.

Control Lab was marketed at secondary education, therefore adaptation was needed for primary. The original Apple-like DOS implementation will not run after Windows XP, but the 1997 Windows version works on all. On Windows 10, a reliable serial connection using a USB adaptor is possible. Below, figure 2, is a screenshot of the user interface. Its facilities are described below.

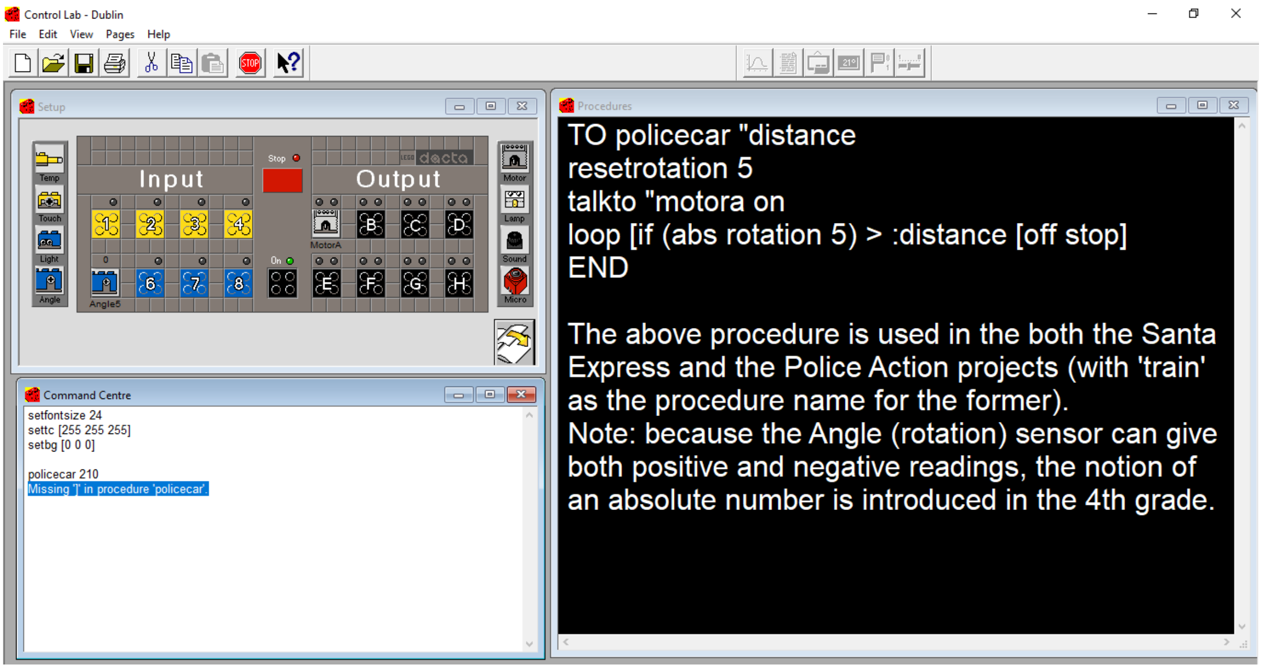


Figure 2. Control Lab user interface (MS-Windows) showing the three windows used.

The Setup and Procedures pages, and Command Centre are used. The Project pages, on-screen gadgets, and dialogue boxes are ignored. A small subset of the vocabulary (primitives) provided is used in both direct Command Centre (command line) and procedure-based modes.

The Setup page mimics the LEGO Interface B. Where the latter has lamps, the image has monitors for inputs and radio buttons for outputs. Logo is told what is connected where by moving a picture onto an input or output, when its default name appears. With a mouse, clicking on the left or right radio buttons sets that side positive and sends full power to whatever is attached. (On the interface the output in the centre is permanently on and is used as a test contact.) For the sensors either a numerical or a Boolean result is shown. Temperature can be set to read Celsius or Fahrenheit.

In the Command Centre, Logo commands for immediate execution are entered (and animated by pressing Enter) and blue errors messages are displayed. On start-up, output A is active and commands will animate whatever is connected. The Procedures page is also active, making it possible to alter its appearance and edit text by command, e.g. change font size or colour.

On the Procedures page new words are defined using existing vocabulary and mathematical and Boolean operators. Procedures enable complex primitive-level actions to be reduced to a single meaningful term. Note: TO and END are capitalised to avoid confusion with the English infinitive. Plain text may (carefully) be combined with procedures, after an END and before the next TO. Procedure names are single words with German-style concatenation. Punctuation includes the double quote prefix for Logo words (called rabbit ears to avoid confusion with speech marks), square brackets to delineate a list. The Logo 'thing' is used instead of the colon prefix for a variable because it makes more sense to the children. An issue in teaching this age range is the space character, added or omitted, which also proves problematical in their pencil and paper writing.

LEGO elements

There is a range of motors that may be attached but, for pedagogic reasons, we use the original ungeared one. Lamps have been converted from filament to LED, some of which are bi-colour. The cables, with their ingenious brick connector, are repairable (figure 3).

Figure 3. Polarity switch, motor, cables, electric plates, lamp, and special parts used in projects.

The original 9 volt electrical system is extremely versatile with battery boxes, electric plates, and polarity change switches. The plates are useful for making special parts; two minutes with a hack-saw will divide them and they may be soldered to.

For technical constructions, like the police car, the original Technic bricks, plates, and connectors work best. From the vast array of gears only a few are needed: spur (8, 16, 24, 40 teeth), worm, crown, and rack. With chains, pulleys and belts these suffice constructionally and pedagogically. In addition the original serrated train track is used as it may be built onto baseplates using small bricks. Turntables, connectors, wheels and axles, complete the catalogue (figure 4).

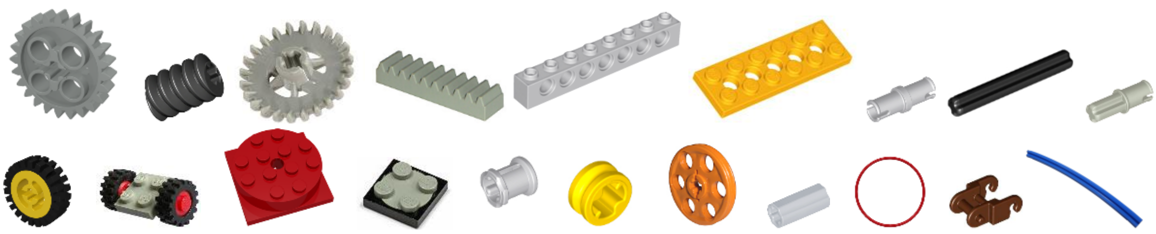


Figure 4. Technical elements used in animated constructions.

The humble LEGO brick is the foundation of all constructions. It comes in two basic thicknesses (fat 2 wide and thin 1 wide) and a variety of lengths. There are only two odd lengths: 1 and 3. Thereafter bricks are even in length. In addition to the basic bricks there are round and modified. Plates, one third brick thickness and slopes of varying degree complement the bricks (figure 5).

Figure 5. Lego basic brick types. The original colours were: red; yellow; blue; green; black; white.

For those who build LEGO following the instructions in sets, success comes easily. Bereft of them, the situation is very different. Even well-educated adults will stack bricks like infants rather than bond them. Doyle designed a test for his fourth grade that required children to construct a simple house given two views and exactly the bricks required (figure 6). The record was nine minutes. When university students and academics tried, many completely failed, stacking not bonding.

Hints

1. Look carefully at the pictures to see where the bricks go.
2. Use the roof plate to help you with the size of the house.
3. Apply logic and principles of sound construction.
4. NOTE: the colours of windows, door, and roof and building plate may vary.

Figure 6. The house test pictures used to guide construction and the hints given.

A couple of issues

As we worked with the children two programming (coding) issues stood out. One was the reverse direction (rd) and setleft and setright commands. The other was zero.

Control Lab Logo is descended from LEGO TC Logo. There, ‘rd’ was defined in terms of direction of motor rotation, with the remark that you can’t reverse the direction of a lamp. This is scientifically misleading. What really is reversed is the direction of the current, this is what produces a different result. For a DC motor it is direction of rotation, for a sound element it is a different tone, for a bi-colour LED it is a different colour. The ‘setleft’ and ‘setright’ commands switch the polarity of the output just like moving a polarity change switch. To help young children understand this two fairies, Ellie electron and Freddy photon, were introduced (Ó Dúill 2020).

Zero is little considered in educational software. The classic problem is division by zero. Scratch states that the result is “infinity” and deals with it chaotically. In fact, the operation is mathematically undefined. Control Lab signals an error but also deals more subtly with zero. The commands ‘wait’ and ‘onfor’ refuse to accept zero as an input. This is sensible in terms of reality – zero time has no meaning. ‘Random’ gives a value from 0 to 1 less than the input, so can deliver a zero to wait. Add one and the problem is solved: a difficult idea for children taught number with by current methods.

Course outline

The construction and computing curriculum is based around four termly whole-school projects: autumn (Halloween), winter (Christmas), spring (Easter) and summer (Fun Fair). A small Ski Week model is also constructed. The course, over the first four years of primary school from six to ten, encompasses the neurologically critical years of affective and cognitive prefrontal maturation.

Grade I. Autumn: Enclosures for Duplo animals using 4-button fat bricks introduce bonding. Simple fat-brick houses with door and window follow. Battery box and bi-colour lamp light the house and introduce electricity, helped by Ellie electron and Freddy photon. Winter: Thin bricks are introduced beginning with 1, 2 and 3 length to make stars and tiny flat houses. The mouse is used to operate the interface, noting polarity effects. Spring: Now their writing skills are adequate, coding is introduced. The first robot is a replacement for men with red and green flags at roadworks. They act the operation of the system to work out the algorithm. Battery boxes and polarity switches are used manually to implement it. The Command Centre and procedures are used for the first time. Road works, small houses and cars complete the scene. Summer: Simple roundabouts, seesaws, stalls and cafés are made.

Grade II. Autumn: Buildings are now constructed with thin bricks, introducing brick thickness as a factor. Electricity, light, and coding is developed using a three colour traffic light. Winter: Electric plates are used to distribute electricity around a village of houses of varying size and fenestration. A day/night clock using a light sensor turns on and off house and street lights. Spring: Windmills are constructed by building a house with a tower for a motor. Gears and pulleys reduce speed of rotation. Ratios are related to rote-learned multiplication tables. Speed of sail rotation is further reduced by setting the power to the motor. Coding to make windmills ‘dance’ for different periods at different speeds and in both directions, introduces random numbers and the issue of zero in computing. A bicolour lamp attached below the motor adds to the effect. Summer: Free rein is given to imagination in building fairground attractions.

Grade III. Autumn: Houses with touch-sensor doorbells programmed as independent processes are built. Winter: The idea of coding is introduced with Morse code. List processing is used to flash letters by laser. ASCII coding is introduced for ‘space’. Their message is read by a light sensor on a second computer where Logo procedures decode and print it. Alpine houses with no backs, a porch, hinged plate roofs and interiors follow. Overhead cables deliver electricity to their lamps. Ski Week: A rack and pinion lift uses grade IV chassis; hotel, café, ski shop. Spring: Design and construction with reference to their own reality. A) Model the environs of the school: its building, houses, apartments, café, hotel, etc. Rack and pinion is repurposed as a building-site elevator. B) Build and program local pedestrian controlled crossing lights. Recall grade II work then model the, more complex, new Bulgarian system, first acting out the algorithm. Summer: Design and build fairground rides, and a disco with temperature sensor air conditioning. Consider body temperature.

Grade IV. Autumn: Mobile robot work begins by building a chassis with motor and gears. It will be a scary train. There is work around friction and keeping the train on a straight track, without using flanged wheels. Electric plates are used to mount lamps front and rear. The train is moved using the ‘onfor’ command. Always, some child enters a long numeral string. A teacher-written program turns their number into years, weeks, etc. to demonstrate the result of adding an extra digit. Winter: The train is turned into a Santa Express. Flanged wheels and their history is introduced. An angle sensor is added to measure distance. This introduces positive and negative outputs and the idea of absolute number. Programming introduces a loop with a stop condition. Ski Week: A ski lift is built using a worm and spur gear drive, as in the vehicle chassis. Spring: Police Action is a burglar alarm simulation involving a robotic police car. Buildings of the children’s own design are equipped with either a touch or light sensor behind the door. A police station and car based on the train chassis with siren and light bar are built. Programming reuses code developed earlier. The algorithm detects a door opening; sounds an alarm in the police station, indicating the burgled house; and despatches the police car to that house. An added feature is a delay. The children suggest reasons for a delay and these are made into a list with an associated time. List processing and random selection are used, the latter revising the issue of zero in computing. Summer: Rides are designed and made, powered by variations on the worm drive of the ski lift and vehicle chassis. Test: The challenge is to construct the house of figure 6 and write a simple sensor procedure.

Each project entails the selection of LEGO bricks and other elements from store, their appropriate use in construction, relevant programming, exploration of other grades' work, and finally careful dismantling and storing. Teamwork, the capability for which is developed over the four years, is the fundamental working method. Whilst templates, demonstration and verbal instruction are used in teaching, LEGO step-by-step style instructions are not. Children select bricks of the size and colour they need. Recall that LEGO bricks above 3-length are all even numbered and the basic colours are the 6 neurological opposed-pair primaries: red/green, blue/yellow, and black/white. The aim of this curriculum is not to make architects, engineers and programmers of primary school children. It is to introduce the fundamentals of robotics in an educationally constructive context.

Animation programming is carried out interactively with the class. The teacher's computer displays on a large screen where the ability to set font size in the Procedures page is valuable. Children work individually using a variety of recycled Windows laptops. Even without an interface attached, they can check their work by running procedures from the Command Centre. The Logo vocabulary subset is shown in the following paragraph, with that used in projects in italic. To give a flavour of how the language is used, the pelican crossing procedures follow. Procedure names are chosen by children. The algorithm is worked out and the procedures written step by step with the class.

abs, and, asci, bg, *butfirst*, *butlast, cc, char, count,* counter, *empty?, END, equal?, false, first, if,* ifelse**,** *last, launch, light#, loop, make, not, off, on, onfor, or, output, power, random, rd, repeat,* resetcounter, *resetrotation,* resett#, *rotation,* run, sentence, setbg, setfontsize*, setpower, setleft, setright,* settc, *show,* showcc, *startup, stop,* stopall, *stopme, talkto,* tc, *temp#, thing, timer#, TO, tone, touch#, true, type, wait, waituntil,* word.

TO pelican "mode
make "pedestrian "false
make "pressed "true
if equal? (thing "mode) "button
[launch [button]]
loop [trafficlight (thing "mode)]
END

TO trafficlight
carred on peoplered on
if equal? (thing "pressed) "true
  [if equal? (thing "mode) "button
  [make "pedestrian "true]
  launch [pedestrianlight]]
wait seconds 30
if equal? (thing "mode) "button [make
  "pedestrian "false launch [button]]
caramber on wait seconds 3
carred off caramber off
cargreen on wait seconds 30
repeat 3 [off wait 5 on wait 5] off
caramber on wait seconds 3
caramber off
END

TO pedestrianlight
buttonlight off
wait seconds 1 peoplegreen on
wait seconds 1 beeper "slow on
wait seconds 15 setright
beeper "fast peoplegreen
repeat 3 [off wait 5 on wait 5] off
beeper "fast off
peoplered on
stopme
END

TO button
make "pressed "false
waituntil [not (thing "pedestrian)]
loop [if touch1 [make "pressed "true
  buttonlight on stopme]]
END

TO carred
talkto "lampa
setleft
END

TO caramber
talkto "lampb
setleft
END

TO cargreen
talkto "lampc
setleft
END

TO peoplered
talkto "lampd
setright
END

TO peoplegreen
talkto "lampd
setleft
END

TO beeper "speed
talkto "sounde
if equal? (thing "speed) "slow [setleft]
if equal? (thing "speed) "fast [setright]
END

TO buttonlight
talkto "lampf
setleft
END

TO seconds "number
output thing "number * 10
END

Aspects of the crossing that may be perceived are shown in normal text, hidden workings in italic. Some lights have an automatic pedestrian phase. The commands that must be deleted or disabled (with a preceding semi-colon) are indented. This program text may pasted into the Control Lab Procedures page.

Note 1: the coloured text used to highlight lamp colour is, sadly, not possible on the Procedures page, settc affects all text.

Note 2: "mode may be "button or "automatic,

Assessment

In a project oriented team-work based setting, the completed project is the outcome. However, there is a need to assess how the children perceive not only the world around them but also what they have produced. In addition to formative interventions, drawing is used (figure 8).

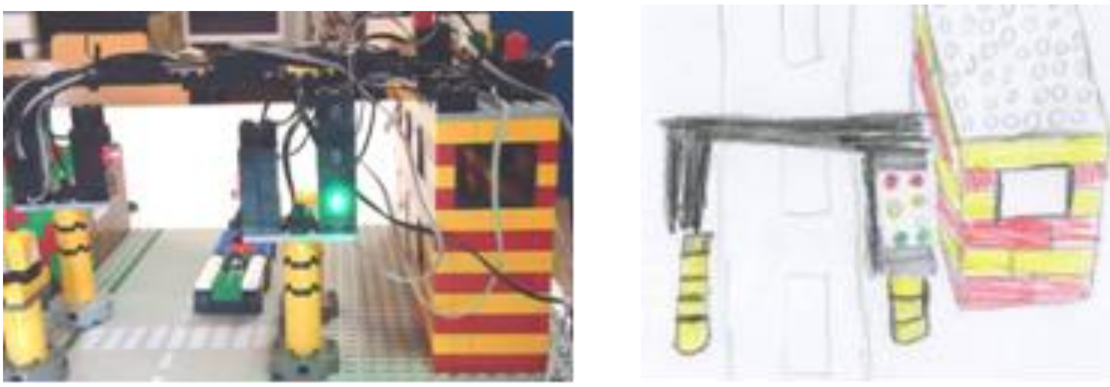


Figure 8. LEGO construction of pelican crossing with nine a year old's drawing of it.

The light sequence and operational algorithm had been discussed and programmed in Logo. The lights had been modelled using bicolour LED and special two colour lamp assemblies. The touch sensor and lamp was built in for the pedestrian button. All watched it go through its sequence. The child's drawing shows what they have perceived: All pedestrian phase elements are left out. The two sets of lights are drawn together: more like a verbal description than a visual representation. Another child drew three-colour lights for red/green road-works lights. As Cox (1992) put it, children draw what they know, not what they see. Drawing, though qualitative, is highly informative.

Turing Medium Possibilities and Human Misconceptions

From our combined experience of working with the computer as the teaching medium for over a quarter century we can assert that working in the Turing medium (Ó Dúill 2013) is a prophylactic against failure and anxiety throughout the primary phase.

- Letters require recognition not recall.
- Keyboarding requires less motor control than letter formation.
- Every child's product is equally neat.
- Spelling and grammar may be checked on the fly.
- Mistakes and rethought are edited, eliminating crossings-out and the need for fair copies.

Though never implemented, it is feasible with speech synthesis technology to produce a Standard English 'pronunciation' that maps directly to spelling. Such an acoustic model would help children to avoid the error of writing phonetically in their own dialect, as we all did before dictionaries. Note: Oxford English Dictionary (OED) IPA phonetic pronunciation is incompatible with correct spelling.

Literacy will be enhanced by Turing teaching. Mathematics and number may radically be reformed.

Constructionism's fatal flaw

The literacy points were suggestive of a flaw at the heart of constructionist philosophy: constructs may be cognitively incompatible. The classic example, on Papert's theory-of-the-universe plane, is the terracentric/heliocentric controversy that got Galileo into such trouble. Its relevance here is because his new construct was roundly rejected by the (religious) establishment. In our case the first construct to consider is the Turing machine and its physical manifestation. To paraphrase Turing: A pupil provided with paper, pencil and rubber, and subject to strict discipline is in effect a computer. I.e. the concept of a digital computer stems directly from our mark-making capability. The difference between a school-child with pencil and paper and one with a stored program digital computer lies in the cognitive demands made. The computer supplies storage and processing, the human the algorithm and data. This means that, by systematic experiment, the student, can learn the relationship between the symbols input and the algorithm. One of the authors discovered this when using a calculator to teach a perceptually disabled child who could not count. If a child can learn to understand number by experimenting with numerals and algorithms, then a question mark must be placed over current teaching method.

Children are taught to count to ten with bricks and other countables, counting frames and abaci, well researched materiel like Dienes blocks, and fingers even. Like the Romans we teach them to write down the result. For ten they used X whilst we use the numeral sequence 10.

- The Roman X lives in a tally-based numeral system and the referent of X is a bundle of ten.
- Our present-day sequence of 1 followed by 0 has no intrinsic meaning. Therein lies the rub.

We treat 10 as if it were the equivalent of the Roman X, but how do we know that it means ten? Its closest physical referent is an abacus, not collections of bundles. Its true source is language. At the first constructionist conference, Clayson (2010) dismissed the notion of a language origin citing the French ‘quatre-vingt-douze’ (92). Though an interesting Celtic/Viking historical artefact, viz. English score and dozen, his assertion supports rather than demolishes the language idea. Consider an abacus. Its base is set by the maximum number of beads allowed in the columns. In the majority of languages, including French, there is a transition after nine in all decades, thus:

- Dix-neuf transitions to vingt. I.e. our number system is defined by nine, not ten.

We have developed two mutually incompatible number constructs. First came fingers, perceptually derived from the vertebrate bauplan. Then came the abacus, with columns of five or ten beads, i.e. one or two hands. But an abacus can have empty columns. Here lies cognitive conflict. In India sometime, somewhere, somebody noticed a connection with language and developed the decimal notation. This notation is incompatible with Piagetian inspired counting-to-ten structural apparatus.

- The morphology of fingers and words is incompatible. Their conflation confuses children

Our LEGOLogo work avoids this. When working with bricks children use number for dimensions. Length is not used as an aid to understanding arithmetic. This is left to the Turing environment.

A note on standards and research

It should be immediately obvious that a change from Paper to Turing media poses real problems for setters of standards and consequently researchers. Given that the new medium is interactive and informative, teaching method and hence learned competencies need, of necessity, to change radically. It follows, given that the short timescales of academic research and the comparisons used, that it is not feasible to carry out the studies required for an evidence based transition.

The Two Faces of Constructionism

Faulty philosophy

Constructionism, because it is a philosophy, lacks the tools critically to analyse constructed entities and test them for contradictions, incompatibilities, and inconsistencies. The example of number, discussed above, is a glaring cultural example. The constructionist community has similarly failed. Papert (1980) mistakenly promoted turtle geometry (Abelson & diSessa 1980) as an easy entrée to “LOGO,” an educational philosophy. It was the nemesis of the Logo programming language, so well integrated with literacy and numeracy, which we still use in primary school with young children. More recent pictorial offerings such as ToonTalk, NI LabView and Scratch, break the essential link to natural language. They are a bolt-on to education blind to its deep interrelationships. Children deserve to receive help in learning from Turing technology, not to be subject to academic conceits. Many conceptions may be constructed, not all constructions prove constructive.

Application to Future Curricula and Teaching Method

It is immediately obvious that Control Lab integrates with the core curriculum of literacy, numeracy and science in primary school. It expands the children's concept of language. It enhances their understanding of number and logic. It provides a platform for introducing the fundamental scientific behind technology. It is quintessentially Turing, the medium that educationalists consider cheating.

Covid-19

The pandemic that has closed schools worldwide has catapulted digital teaching and learning. Despite decades of research into distance learning, little thought had been given to primary school. Ilieva, teaching a third grade class, started off squinting at phone photos of children's exercise-books. A week later she knew that all also had laptops at home, so an editable pdf took over. We have a medium with entirely different properties emulating its paper predecessor. This is because teaching method remains unchanged despite half a century of 'integrating' technology in schools.

There is no reason why the transition to Turing teaching cannot be initiated now. Costs are no longer an issue. Four function calculators can be had for a Euro. The world is awash with unused but perfectly serviceable desk and laptop computers. Schools could use these to drive home the need to re-cycle and re-consider fitness for purpose, as our work shows. Calculator use and the maths functions of e.g. Word tables are the type of life skill that children need to develop for their futures; as is the effective use of spelling and grammar checkers, layout, and keyboard, and the capability to mix text with illustration. Inhibition comes from an academic and administrative milieu unwilling to countenance the consequences of the capabilities of the no longer new medium.

Our quarter-century old software and hardware has been deemed obsolete by most constituencies and yet it is educationally more fit for purpose than its replacements. It's not perfect and would benefit from updating. It is considerably less obsolete than the curriculum and teaching method in primary education: a numeracy method that not only conflates incompatibilities but which turns children into what Turing called "paper machines" rather than educating creators of algorithms for that Turing machine. At the time of writing, Ireland is consulting on its primary school curriculum. With the Covid19 experience of e-teaching in mind, might this be an opportunity to catalyse the transition from the current oppressively obsolete paper curriculum and method to Turing teaching?

References

Abelson, H. and diSessa, A. (1980) *Turtle Geometry*. The MIT Press, Boston MA.

Clayson (2010) Personal communication.

Cox, M. (1992) *Children's Drawings*. Penguin, London.

Doyle, M. and Ilieva, V. (2013) *Primary Robotics: Laying Sound Foundations*. In Proceedings of WCCE 2013, Volume 2, Edited by N. Reynolds and M. Webb, Torun, July 1-7, pp. 28 – 35. Nicolas Copernicus University Press, Torun.

Ilieva, V. (2010) *LEGO and LOGO in the primary school – a simple way for learning through creation.* In Proceedings of Constructionism 2010, Edited by J. E. Clayson and I. Kalaš, Paris, August, p. 58 and CD. Faculty of Mathematics and Informatics, University of Bratislava, Bratislava.

LEGO Dacta, (1993) *Book 822207 Control Lab Reference Guide*. The LEGO Company, Billund.

Ó Dúill, M. (2013) *Teaching: Traditional versus Turing*. In Proceedings of WCCE 2013, Volume 2, Edited by N. Reynolds and M. Webb, Torun, July 1-7, pp. 9 – 16. Nicolas Copernicus University Press,Torun.

Ó Dúill, M. (2020) *Trans-constructionist Prototyping with a Rare Vintage.* In Educational Robotics in the Context of the Maker Movement. Edited by M. Moro, D. Alimisis and L. Iocchi. pp. 45 − 56. Springer, Cham Switzerland.

Papert (1980) *Mindstorms*. Harvester, Brighton.

Building & transferring constructionism from higher education to public education

Márta Turcsányi-Szabó, *tszmarta@inf.elte.hu*
Dept of Media and Educational Technology, ELTE University, Faculty of Informatics, Budapest, Hungary

Abstract

The advent of emerging technologies and the ways it changes our everyday lives as well as handling and processing information must also have an effect on the way learning could be improved to suit the needs of the 21st century. The notion of changing needs in higher education can well lead to developments that might produce a whole paradigm shift in education effecting public education as well.

The paper describes the challenges that higher education faces and how these were carefully addressed in a university setting to administer progressive changes in pedagogies from personalised project-based learning, through problem-based learning, till challenge-based learning. Considering teacher education situated problem-based learning to constructionism even managed to contribute to public education, injecting innovation and paradigm shift in learning. A model of Cyberspace learning environment was created, and different methods of dissemination were attained. The paper described the learning theories behind the developed methodologies and illustrates them with practical projects.

Figure 1. Characteristic picture of our activities

Keywords

Project-based learning, problem-based learning, challenge-based learning, constructionism, university education, public education

Digital Constructionism

Advanced learning theories have been emerging for centuries now, *Constructionism* being the „mysterious duckling" that never seemed to suit proper educational systems. This paper tries to draw a process how theory and practice was built and enhanced in a university setting and expanded further to be injected into formal public education and informal museum education settings.

First the underlying theories and processes are defined and explained. Then, the different university level learning setups are explained from personalised project-based learning, through problem-based learning, till challenge-based learning was formed. Then it explains how teacher education setting was formed from situated problem-based learning that contributed to public education through constructionism. Then, it ends by forming and disseminating Cyberspace learning environment models for both formal and informal education.

Digital tools and the necessity for paradigm shift in education

Our approach in education in based on *experiential learning* occurring in a natural setting yet underlined with learning theories and teaching strategies in order to achieve effective learning outcomes.

Experiential learning is a term (in informal setting) occurring through direct participation in a situation of live or (in formal education) by students who acquire and apply knowledge, skills and feelings in a relevant setting learnt. The former could be viewed (Wolf, 2010) as *situated learning* that takes place through social learning and interaction with the individual, while the later (Kolb, Fry 1975) as *experiential learning cycle*, which involves in sequence (1) concrete experience (2) observation and experience (3) abstract concept formation (4) testing in a new situation.

However, the speed and trac of technology development in the 21st century requires a paradigm shift in education, that takes into consideration not just the affordances of technology, but also the pedagogical considerations for learning to take place.

Characteristics of Learning 1.0, 2.0 and 3.0 is explained through the affordances of Web 1.0, 2.0 and 3.0 technologies towards a paradigm shift in education, whereby learning is transformed towards learner-centred model in accordance with technology development. (Figure 3. a. and b., Turcsányi-Szabó, 2012.)

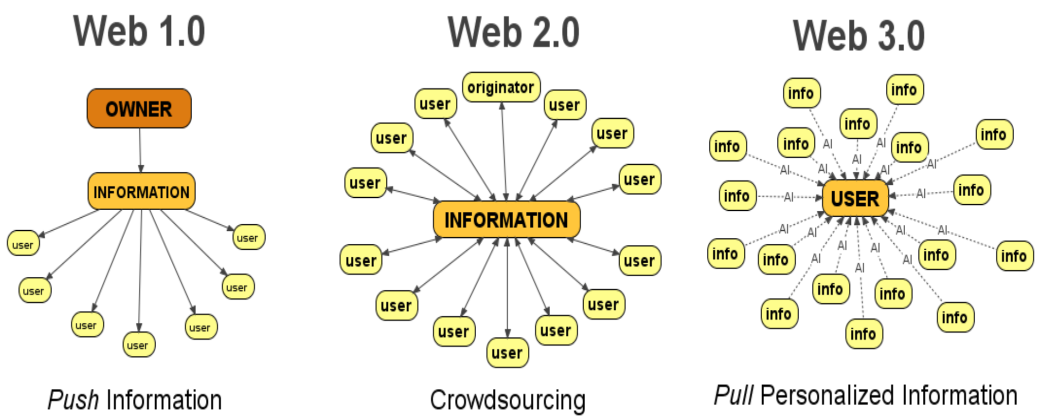


Figure 3.a. Web 1.0, 2.0, 3.0

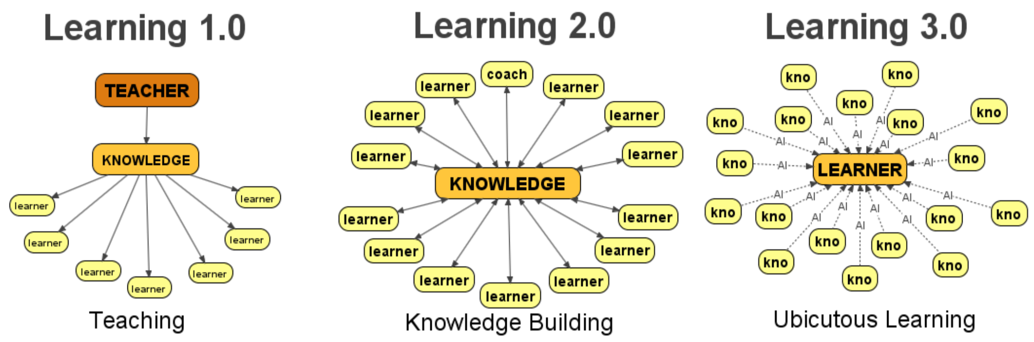


Figure 3.b. Learning 1.0, 2.0, 3.0

21st century learning cannot take place just by knowledge transfer, but – apart from utilization of relevant technology – needs the active participation of the learner in individual information retrieval and critical thinking is a key issue for relevant knowledge building in social context. Thus, teaching has to provide all the levels of knowledge building and deepening with the help of modern technology embedded in proper learning taxonomy and also provide motivation for self-directed learning to take place.

Extended Digital Bloom's Taxonomy is based on Blooms Taxonomy and is extended by one level higher stating that sharing with the broad public escalates knowledge to one level higher (Turcsányi-Szabó, 2012).

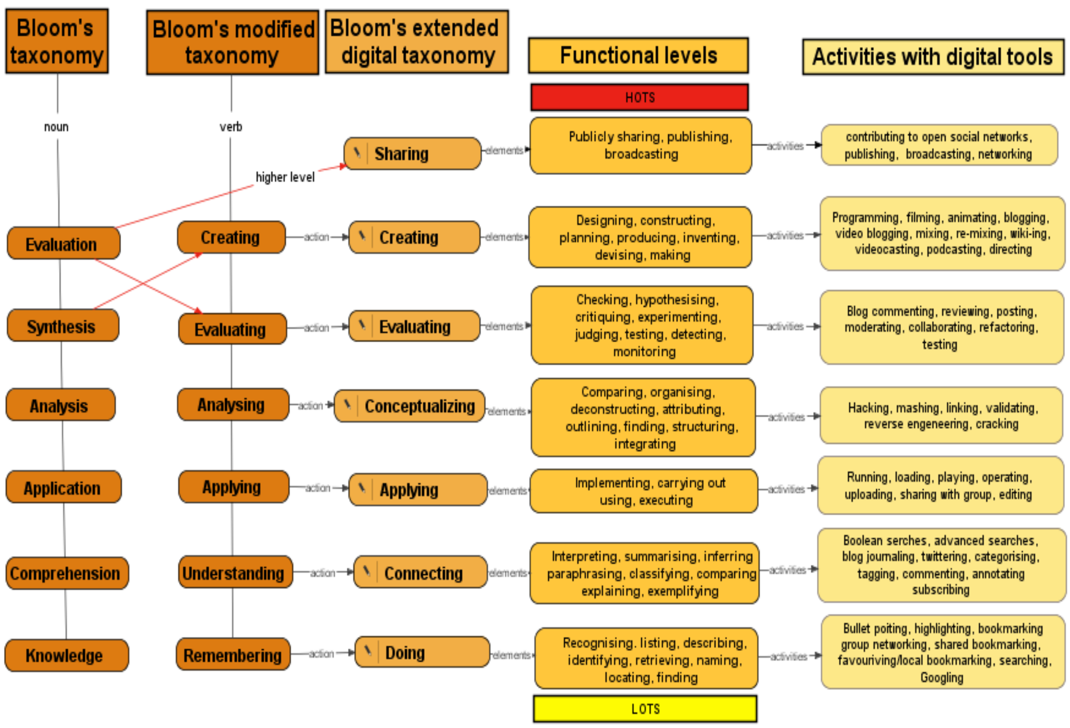


Figure 2. Extended Digital Bloom's taxonomy

Applying the different levels of Bloom's taxonomy deepens knowledge and allows the learner to develop higher order thinking skills. Experience shows, that the extension of sharing developments with the public provides a challenging step that needs confidence, responsibility and thus a higher level of involvement in quality assurance to allow the individual or group to give ownership to results attained.

However, the structure and context of individual tasks does not always end up being relevant to students and thus might not provide enough motivation for learning, which could jeopardise the main aims of education. So, a meaningful context for understanding a full picture, paving the road towards the main aims of a learning goal has to be established as a cover story.

Problem-based learning is both a curriculum and a process. The curriculum consists of carefully selected and designed problems that demand from the learner acquisition of critical knowledge, problem solving proficiency, self-directed learning strategies, and team participation skills. The process replicates the commonly used systemic approach to resolving problems or meeting challenges that are encountered in life and career (Barrows, 1980).

Problems often involve interdisciplinary areas and thus has to involve participants with diverse backgrounds and an openness towards communication on common backgrounds. This could provide soft skills too, preparing students to become workforce of the 21st century. Moreover, some might be more ambitious in understanding situations and coming up with own challenges and willing to take the risk in coming up with a possible initiate for an innovation.

Challenge-based learning is an effective learning framework initiated at Apple, Inc. and used in universities, schools, and institutions around the world. The framework empowers Learners (students, teachers, administrators and community members) to address local and global Challenges while acquiring content knowledge in math, science, social studies, language arts, medicine, technology, engineering, computer science and the arts. Through Challenge Based Learning, students and teachers are making a difference and proving that learning can be deep, engaging, meaningful, and purposeful. (Nichols, et. al. 2016)

Our present time needs this sort of creative attitudes and entrepreneurial thinking very much, thus opportunities have to be given to students to find their own challenges and suitable partners enabling fruitful collaboration for success. At the same time, considering the scope of a single learner, constructionism is the individual driven construction process through informal mentoring and community feedback.

Constructionism (Papert, 1991) takes a view of learning as a reconstruction of knowledge and extend the idea of manipulative materials to the idea that learning is most effective when part of an activity the learner experiences as constructing a meaningful product (Sabelli, 2008).

Constructionism is thus student-centred discovery learning, where feedback and other interactions within a community has key importance. Mentors scaffold individuals through participation in project-based learning to make connections between information they already know with background knowledge and different ideas arising, as well as self-driven knowledge acquisition. The process is undertaken through emerging digital tools and thus arises the term *digital constructionism*. It must be emphasised that the process is less teacher driven and thus it is more likely to occur in informal settings and thus teachers feel uncomfortable in such environments, not being able to take full control of the process itself.

Informatics Education

Personalised project-based learning

In between 2000-2009 a one semester course „*Designing multimedia materials*” at ELTE University was attended yearly by about one hundred program designer and informatics teacher training students, in their fourth and fifth year of the university studies divided into five groups evenly. The goal of the course was to make students become aware of the efficiency factors of e-learning content and master authoring tools with which they can realise required features. The development of multimedia content demands various expertise. Students got acquainted with the importance of visual, auditive elements and with the significance of cognitive style that can be effective in e-learning materials.

A single framework was used by all the students that allowed the identification of added value of collaborative work by precise registration of the individual’s activities. Students worked in groups of four, taking responsibility for developing different media elements (a. video and audio, b. text and mindmap, c. graphic and animation and d. interactive simulation). Each person in the group had to choose one of these tasks and work together to create a coherent result. Their project goal was development and integration of multimedia elements fitting to selected chapters of the course digital learning material.

CECIP (Collaboration – Evaluation - Critical thinking - Individual assessment – learner Profile) was based on the (C) collaborative construction of student's knowledge, (E) developing evaluation and assessment skills, (C) developing critical thinking skills, (I) integrating individual evaluation and (P) generating learner profile. Personalised tasks and assessment were based on Bloom's Taxonomy. (Figure 2., Turcsányi-Szabó et. al. 2012 a.)

Knowledge: The course material necessary for the learning of theoretical and practical part of the knowledge was available on-line and each group received one chapter to develop as e-learning material. The students created an online knowledge base collaboratively by setting up the „golden rules”. The application of golden rules was mandatory for the development of their own learning material.

Understanding: Students got acquainted with creating mindmaps, and using authoring tools through sequential e-learning materials and practical hands on lab activities.

Application: They identified and critically analysed the above listed principles in the works submitted by previous-year-students.

Analysis: They actively analysed and evaluated their own impressions on different presentations they were exploited to during an interactive lecture.

Synthesis: They had to apply all what they learned in theory and practice within their own tasks to make a coherent final group work in collaboration.

Evaluation: They evaluated their own and the others' works.

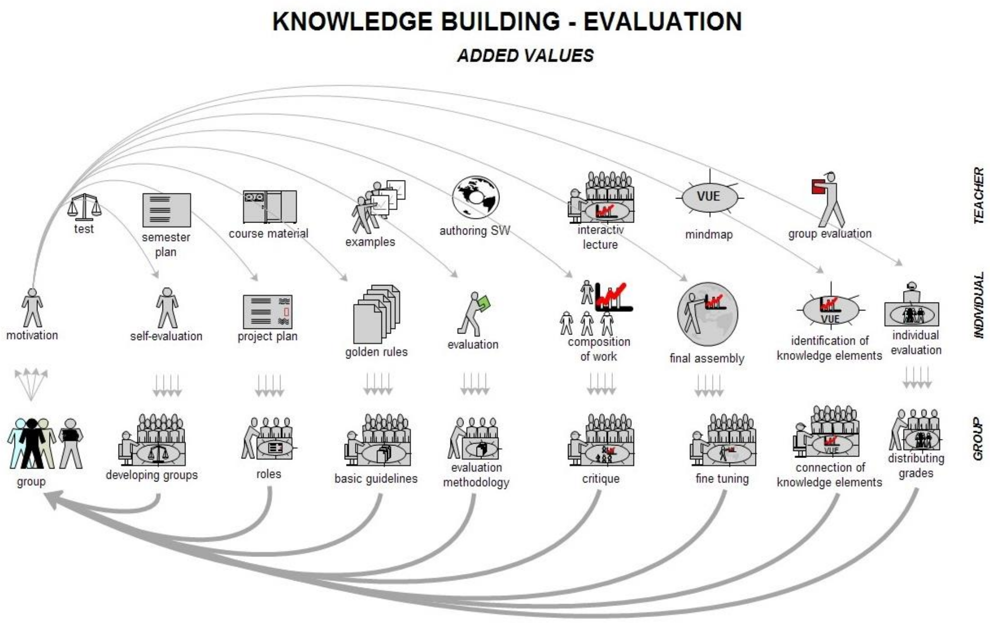


Figure 4. Knowledge building and evaluation model used

One of the most important part of the model is the evaluation strategy (Fig. 4). In a collaborative e-learning environment value added by individuals, the performance and the acquired knowledge in a project work is very important to be traced with proper evaluation strategy, because in this educational model teacher cannot monitor students' individual performances. In an electronic educational environment self-evaluation and peer-evaluation has more importance. The used evaluation strategy has an essential role in knowledge-building, thus the evaluation of the project- and individual work emerged from the following elements:

- Students evaluate the multimedia materials made in the previous semesters by others using pre-determined evaluation criteria.
- The students and the teacher together assess all their own development of multimedia materials on the basis of predetermined evaluation criteria and students then improved their work according to the assessment.
- Students also evaluated their team-members' attitudes, collaboration and work as part of the online evaluation strategy.
- The teams receive collective grade points for their whole project work, so they need to divide it according to individual added value.
- Individual grades are issued by the teacher in accordance with the previous agreement.

Most of the students experienced online collaborative environment for the first time within our course. Their most important experiences in our blended learning context were team responsibility and peer dependency. Students felt that working in project teams, keeping deadlines; dividing grades need more attention and responsibility than traditional teaching methods. Collaborative knowledge building seems to be more effective in identifying critical rules within the overall theory to be implemented even if not followed in practice.

Being aware of the significance of learning styles may enhance students' self-awareness for their own and others learning strengths and weaknesses which is an important skill for future multimedia content developers.

Extension to problem-based learning

The previously described course was later (between 2010-2015) switched for the *"Interactive media development"* set of courses that embeds interdisciplinary projects originating form industry, education or museums, and is also offered for B.Sc., M.Sc. and Ph.D. students in different faculties as well as in MOME university of applied art. Students can register for any course within the set offered under different course titles and are able to fulfil requirements dedicated to the specifics of each course. E.g. *"Interactive media application"* B.Sc. in C.S. (Computer Science) does not require coding only design through applications, *"Interactive media programming"* M.Sc. in C.S. requires coding, *"Designing media for museums"* M.Sc. in Museum Communication requires media design with professional tools, *"Interactive media research"* Ph.D. in C.S. requires the setting up of a research involving designed artifacts, *"Creative media design"* M.A. in Arts requires creative design of artifacts, *"Museum management"* M.A. requires project management.

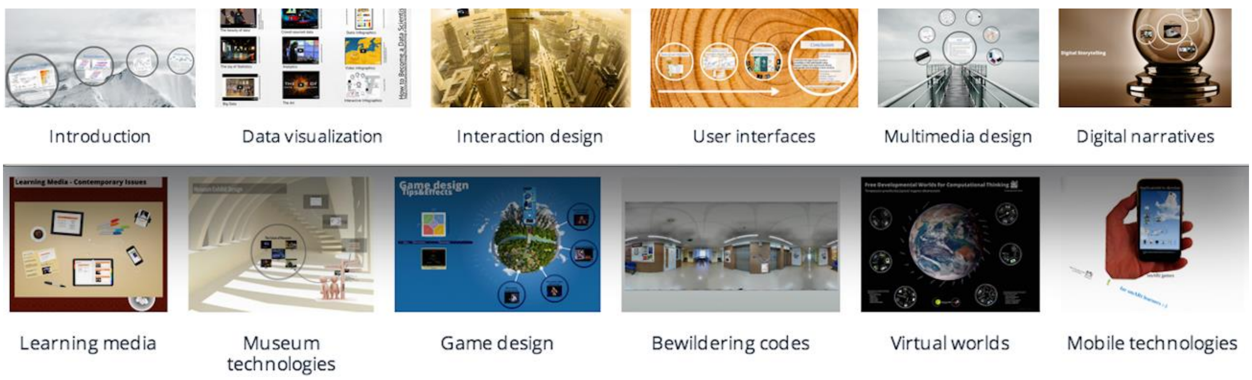


Figure 5. Chapters of Interactive media course (ref link)

Example course content for C.S. B.Sc. and M.Sc. students contains several themes to choose from: Introduction, Data visualization, Interaction design, User interfaces, Multimedia design, Digital narratives, Learning media, Museum technologies, Game design, Bewildering codes, Virtual worlds, Mobile technologies (Figure 5.). Requirements include:

- *basic understanding* of three chosen Chapters (which should be mastered self-paced - using flipped classroom model - depending on interest and specifics of course requirements the student registered for) and contribution is required to the online knowledge building on the theme.
- *user level awareness* of all projects produced during the semester and providing online feedback to each project.
- *personalized developer role* within one of the actual interdisciplinary projects in collaboration to produce interactive media for edutainment.

This project demonstrated how different technologies serve the developer learning community of practice and how products for industry, education and museum are piped back into the immersive learning process. It also created a model for bridging faculties and institutions (Turcsányi-Szabó, et. al. 2011) producing products in co-operation or for institutional use (Turcsányi-Szabó, et.al. 2012 b). The course setup won a Tempus prize for showing good practice in STEM education on higher education level.

Opening up to challenge-based learning

The later described course was altered (between 2016-present) in profile to tackle challenge-based learning and is also offered for international students in C.S.. This evolved from the fact that the growing number of students enrolling for international studies at our university confronted us with new challenges, diversity in backgrounds and cultural differences, guidance and orientation, working and learning styles, language and communication difficulties, … etc. We realized that these issues can be solved best if we include the students and the whole community in raising awareness about their own emerging problems, their relevance in students' lives, and best ways of dealing in order to emerge with a solution. Example projects included: housing issues – developing portals to match supply and demand, how to manage cooking according to own cultural background – developing a portal to search for basic supply and recipes for cooking dishes, dressing issues – developing an augmented reality application to help the search for suitable items, sentiment analysis – developing an app to analyse text within chat windows to identify opinions of products, virtual exhibitions – exhibiting diploma works of art students … all under the umbrella of sustainable development building environment friendly solutions.

Teacher education

Situated problem-based approach

The 21st century is dedicated in bringing up a knowledge-based society in which required competencies strive to follow the extremely fast development of tools that are needed for enhanced work and Life Long Learning. But, the structure of teacher education is not suitable to handle the extent of changes progressing in our daily lives influencing the next generation of learners:

- Required competencies strive to follow the extremely fast development of tools,
- Structure of teacher education is not suitable to handle dynamic changes,
- Continuous flow of sustainable innovation needs to shape public education.

Besides, Informatics teacher educations struggles to keep students on teaching tracks due to the fact that industry offers them much higher salaries. Thus, teacher education had to be reformed by switching test and exams for project work directly addressing public education, in order to raise the motivation of practicing teachers for professional development and continuously channel a sustainable flow of innovation to shape public education towards innovation. We actually took advantage of students' created projects while they are still pursuing their studies (Turcsányi-Szabó, 2006).

The nature of the learning design of the "*Educational Technology*" is *transmissive*, *dialogic*, *constructionist* and *co-constructive*, illustrating how each element contributes to the adaptation of theory into practice (Bower, Hedberg, & Kuswara, 2010). The course attempts not just to teach about learning technologies and their adaptation within the learning processes, but also to understand learning situations and be able to design a suitable environment for enjoyable learning, using innovative tools and methodologies situated in a given context that molds relevantly into the lives of children. In doing so, they must be able to choose from activities that are engaging, have adequate motivation to learn themselves, understand well the process and depth of learning and how suitable environments can be built for different learners, and what role the teacher has in this environment. Then, they have to design the artifact in collaboration with other team members and share their final product with public education or professional community from where they get their feedback on quality. This lies within the highest level as "Sharing" within Bloom's Extended Digital Taxonomy (Turcsányi-Szabó, 2012) and has to lead into the everyday professional practice within an active CoP (Community of Practice) (Lave, Wenger, 1991). Example projects co-ordinated by T@T lab (http://tet.inf.elte.hu/) are plentiful: https://tinyurl.com/TeTlabprojects, where a series of projects launched were designed on challenge based learning (http://kihivas.ini.hu/).

Model for Cyberspace

Opening model of Cyberspace at university

In 2016 January T@T lab opened our IT Hut (http://tet.inf.elte.hu/tetkucko/) as a small model for cyberspace used as maker space, that could be adapted from small scale (school setting) to enlarged scale (museum spaces). The Hut gives place for classes mentioned above as well as other similar courses developing technology and methodology for enjoyable learning activities moving towards constructionism. Demo classes for activities produced are also done here for which we invite school classes to take part. We also provide orientation activities for schools when they can get acquainted with different IT related jobs too. Open days and Researchers' Nights are also very frequented activities, besides Family Days when we try to build a bridge between children and parents to enjoy and understand their digital lives with common understanding. We also launch challenges on Facebook https://www.facebook.com/tethalo/.

Figure 6. Picture of IT Hut (http://tet.inf.elte.hu/tetkucko/)

Disseminating model of Cyberspaces

We also developed a larger scale model for Cyberspaces within a governmental project and the first such institute opened in January 2019 with great success. Further such institutions (using the same model) are about to open next January in three other towns. Their success is our proof of the methodological model building we went through in the past many years. In 2020 we launched an elective course within all faculties of ELTE university called "Experiential informatics" (http://tet.inf.elte.hu/tetkucko/experiential-informatics-course/) that atempts to transmit our philosophy of learning that shall be opened to public education this September.

Figure 7. Picture of IT Hut (http://mobilis-gyor.hu/mobility-gyor)

Conclusion

In a situation where multiple issues are urging changes in higher education one should analyse deeper to find solutions that are not just solving the most alarming issues, but look for ways that might have further benefits within the overall settings. After all university education (and especially teacher education) is not an entity for itself, it not just serves to produce well educated professionals with plentiful skills for the 21st century1s knowledge building society, but also awaits their contribution towards public education and the boarder society.

Applying and transferring Constructionism from Higher Education towards Public Education can well be attained through the challenge-based approaches applied in courses, where the emerging innovations can be well injected into societal use.

Acknowledgements

The results shown could not have been done without the hard integrated work of students and colleagues taking part in the classes and projects of T@T lab (http://tet.inf.elte.hu/).

References

Barrows H.S. & Tamblyn R.M. (1980) Problem-Based Learning: An Approach to Medical Education. New York: Springer Publishing Company, p.1.

Bower, M., Hedberg, J. G. and Kuswara, A., (2010), A framework for Web 2.0 learning design, Educational Media International, 47: 3, 177-198.

Kolb. D. A. and Fry, R. (1975) 'Toward an applied theory of experiential learning;, in C. Cooper (ed.) *Theories of Group Process*, London: John Wiley.

Lave, Jean; Wenger, Etienne (1991). Situated Learning: Legitimate Peripheral Participation. Cambridge: Cambridge University Press.

Nichols, M., Cator, K., and Torres, M. (2016) Challenge Based Learner User Guide. Redwood City, CA: Digital Promise. http://cbl.digitalpromise.org/wp-content/uploads/sites/7/2016/10/CBL_Guide2016.pdf

Papert, S. (1980). Mindstorms. Children, Computers and Powerful Ideas. New York: Basic books.

Sabelli, N. (2008). *Constructionism: A New Opportunity for Elementary Science Education*. DRL Division of Research on Learning in Formal and Informal Settings. pp. 193–206.

Turcsányi-Szabó, M. (2006). Blending projects serving public education into teacher training. In Kumar, Deepak; Turner, Joe (Eds.) *Education for the 21st Century - Impact of ICT and Digital Resources*, IFIP 19th World Computer Congress, TC-3 Education, IFIP series Vol. 210, pp. 235-244, Springer.

Turcsányi-Szabó, Márta (2012) *Aiming at Sustainable Innovation in Teacher Education – from Theory to Practice*, INFORMATICS IN EDUCATION: AN INTERNATIONAL JOURNAL 11 : 1 pp. 115-130. , 16 p. https://infedu.vu.lt/journal/INFEDU/article/163/info

Turcsányi-Szabó, Márta; Simon, Péter; Abonyi-Tóth, Andor; Ekker, Nikoletta; Ruttkay, Zsófia (2011) *Augmenting Experiences – a Bridge Between Two Universities* In: IEEE (eds.) Mixed and Augmented Reality - Arts, Media, and Humanities (ISMAR-AMH), 2011 IEEE International Symposium On Bâle, Bázel, Switzerland : IEEE, pp. 7-13. , 7 p.

Turcsányi-Szabó, Márta; Magyar, Tímea; Béres, Ilona (2012 a.) *Towards a Personalised, Learning Style Based Collaborative Blended Learning Model with Individual Assessment*, in INFORMATICS IN EDUCATION: AN INTERNATIONAL JOURNAL 11 : 1 pp. 1-28. , 28 p. https://infedu.vu.lt/journal/INFEDU/article/158/info

Turcsányi-Szabó, Márta; Ekker, Nikoletta; Abonyi-Tóth; Abdor, Pluhar; Zsuzsa (2012b) *Augmented Edutainment on campus.* In: Gardner, M.; Garnier, F.; Delgado, Kloos C. (eds.) Proceedings of 2nd Immersive Education Summit Párizs, France : Presses de l'Université Paris-Sorbonne, pp. 204-209. , 6 p.

Woolf, B. P. (2010). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning.* Morgan Kaufmann.

Building For Robots: An Alternative Approach of Combining Construction and Robotics

Janet Bih, *bihjane@umd.edu*
College of Education, University of Maryland, College Park, USA

Daniel Pauw, *dpauw@umd.edu*
College of Information Studies, University of Maryland, College Park, USA

Tamara Clegg, *tclegg@umd.edu*
College of Information Studies & College of Education, University of Maryland, College Park, USA

David Weintrop, *weintrop@umd.edu*
College of Education & College of Information Studies, University of Maryland, College Park, USA

Abstract

Robots and robotics toolkits are becoming a growing part of the introductory computing landscape. One very common approach for the use of robotics in learning contexts is to have kids construct their own robots. This paper explores the affordances of focusing the activity of construction in robot-related activities on the setting rather than on the robots themselves. Using conventional arts-and-crafts materials (e.g. pipe cleaners, paper cups, cardboard), this paper describes a study in which groups of children (ages 7 to 13) engaged in an experiential learning activity focused on collaborating, designing, building landscapes for robots to explore followed by programming Sphero robots to navigate them. Shifting construction to the setting in which the robot runs rather than on building the robot itself enabled new and productive forms of engagement, participation, and learning. The benefits of the building-for-robots approach includes relying on inexpensive materials, supporting new forms of collaboration and sharing, and opportunities for learners to draw on prior knowledge and experience. Further, the shift towards constructing for the robots introduces productive interplay between the construction and programming components of the activity where constructing for the robot can influence what is programmed, while authoring the programs shapes the way the world is constructed around the robot. This paper provides examples of these new affordances and explores the potential of this shift towards designing for robots. In doing so, it advances our understanding of ways to productively use robotics as a means to support meaningful engagement with powerful ideas of computing.

Keywords

Constructionism, Robotics, Design, Youth

Introduction

Robots and robotics toolkits have an increasing presence in both formal and informal learning contexts. Be it in homes, classrooms, or on television and in the movies, robots are becoming ubiquitous. One common argument made for robots and robotics toolkits either as toys at home or as devices to be integrated into classroom activities is the potential they have to serve as meaningful learning devices (Chambers, Carbonaro, & Rex, 2007). A quick internet search reveals dozens of books, toolkits, videos, and articles all espousing the virtue of robots for kids. This enthusiasm is not unwarranted as much research has been done showing the potential and effectiveness of robots and robotics toolkits to support learning and serve as a means to engage learners in productive learning activities (Blikstein, 2013; Papert, 1980; Resnick & Rosenbaum, 2013; Yu & Roque, 2018). Looking specifically at work that has grown out of the Constructionist tradition, an emphasis has been placed not just on using robots but on constructing them as well (Blikstein, 2013; Resnick et al., 1998; Resnick & Silverman, 2005). Having learners construct robots provides opportunities for deep engagement with foundational concepts of engineering, mathematics, science, and computing. Further, providing learners with the building blocks can impart a powerful sense of belonging and ownership over both the device and the concepts that it demonstrates. While there is much promise in the use of robots and robotics toolkits for authentic learning, challenges remain related to robotics and education, including costs associated with tools and toolkits, opportunities for collaboration and drawing on prior knowledge and experience during construction, and issues around reuse of construction kits and their components.

In this paper, we explore an alternative way to bring Constructionist principles and robotics together that seeks to draw on the strengths of both while addressing some of the aforementioned challenges. To do so, we asked learners to construct a world around the robot, rather than construct the robot itself. We call this approach *Building for Robots* and contrast it with the more conventional approach of building the robots themselves. These setting-based constructions were created using inexpensive art-and-crafts materials most children are familiar with such as construction paper, paper plates, pipe cleaners, and tape. We argue combining low-tech materials with high-tech robots can provide an alternative approach for construction and robotics that retains many of the benefits present in conventional robot construction activities while also introducing new forms of engagement and opportunities for learning. This paper presents the motivation for this strategy and then outlines three affordances of the Building for Robots approach that make it distinct from more conventional robot construction activities. In doing so, we seek to advance our understanding of ways to blend construction and robotics as a means to provide engaging learning experiences for youth. Our approach introduces new design opportunities and opens new pathways into the world of robotics, engineering, and computing.

The paper continues with a description of how constructionist ideas undergird this work, and a discussion of prior research on robot toolkits for constructionist learning. We then introduce the Building for Robots approach relative to existing approaches to learning with robots. The methods for the study conducted follows before we present our findings in the form of three distinct affordances of the Building for Robots approach. The paper concludes with some challenges and limitations of the approach.

Theoretical Orientation and Prior Work

Constructionism & Robot Construction Kits

Papert’s constructionism builds upon Piagetian Constructivism and identifies the act of constructing public and shareable artifacts as being an especially productive and generative learning activity (Harel & Papert, 1991). Adding to this idea, distributed constructionism focuses on situations in which more than one person is involved in design and construction activities (Resnick, 1996). Distributed constructionism asserts that collaborative activities that involve information exchange, design and construction of meaningful artifacts lead to effective knowledge-building communities (Resnick, 1996). Both building and collaboration are central to the Building for Robots approach discussed in this paper. The Building for Robots approach also draws from Friere's perception of learning as a form of empowerment (Blikstein, 2013). Friere argued learners should go from the "consciousness of the real" to the "consciousness of the possible" as they perceive the viable new alternatives" beyond "limiting-situations" (Freire, 1974). Therefore, learners' projects should reflect their personal and community problems as they design solutions to problems that would become both educational and empowering (Blikstein, 2008). Recent versions of educational robotics activities involve programmable bricks, that enable learners to control the behavior of tangible objects using virtual and physical environments. Such tools make possible new types of experimentation, in which children can investigate everyday phenomena in their lives both in and out of the classroom (Resnick, Bruckman, & Martin, 1996).

Physical computing devices and robot construction kits specifically designed for kids have a long and rich history (Blikstein, 2013; Martin et al., & Berg, 2000; McNerney, 2004). Blikstein (2013) traces the history of these toolkits back to the same research center from which Constructionism grew: MIT's Media Lab and the LEGO group. Early robotics toolkits include LEGO bricks (Sargent et al., 1996), which served as an antecedent to the popular line of Lego Mindstorms construction kits still in use today. As part of this work, designers sought to give learners more direct access to the underlying computational capabilities, moving beyond the "black box", making underlying processes more visible and accessible (Resnick et al., 2000).

As robotics advanced, construction kits became cheaper and more diverse and were designed to meet different goals and audiences. Examples of this diversification includes kits designed for younger learners (Yu & Roque, 2018), open-hardware kits seeking to democratize learning with construction kits (Sipitakiat et al., 2004), new ways to integrate robotics with traditional crafting practices (Eisenberg, 2002; Kafai et al., 2014; Rusk et al., 2008), and new designs seeking to welcome learners who historically have not been a part of the robotics and construction community (Buechley et al., 2008; Holbert, 2016). Central in this thread of work is the emphasis on learners as empowered builders and exploring the ways that robots and physical computing devices can facilitate this form of exploration, expression, and discovery.

Learning with Robots

There have been two pervading approaches that shape existing approaches to the use of robotics in education that have distinct views on the role of the robot. In the first approach, fully functional, prefabricated robots are introduced as tools for learners to explore and program. In this approach, learners investigate ways to take advantage of the robot's existing capabilities, this usually takes the form of programming the robot to carry out a set of instructions. This approach has been called a "black box" approach because instead of allowing learners to "get under the hood" of the robot and explore how it is made and how it carries out commands, it focuses on the actions the robot can take with the existing, already-put-together robot (Alimisis et al., 2007). One concern with this approach is the missed opportunity for construction and creativity with the result being that the robot is a passive tool used by the learner (Mitnik, Nussbaum, & Soto, 2008).

A very different view of the role of robots in learning can be seen in the "white box" approaches which foregrounds the inner workings of robots and how their parts are put together and how, when assembled, the parts can carry out the functions of the robot (Resnick, Berg, & Eisenberg, 2000). There are a large number of robotics construction kits that emphasize the building of robots reflect this perspective. Examples of technologies in this category include Lego Mindstorms, Pico-Crickets kits, and the growing ecosystem of do-it-yourself microcontroller kits like the Arduino and Go-Go Board (Lego Systems Inc, 2008; Sargent et al., 1996; Sipitakiat, Blikstein, & Cavallo, 2004). The white-box metaphor provides opportunities for construction, creative thinking and involving learners of all ages, mainly in informal educational settings (Alimisis et al., 2007). However, there is a limit to the complexity of the kinds of robots children can make and program. As hardware and sensors become smaller and the ways they can be combined becomes more complex and precise, new robots designed for young learners are being introduced whose sophistication is beyond the capabilities of what can be expected of young learners to assemble (e.g. Ozobots and Spheros).

Most studies that focus on programming with robots from inside the Constructionist tradition take the white box approaches - engaging learners in the design and implementation of the robots themselves. We hypothesize that black box approaches to robotics could also engage learners in authentic and productive constructionist learning experiences. Learning experiences with black box robots rarely utilize constructionist learning approaches, instead focusing on more didactic activities (e.g., programming a robot to navigate a predefined maze). We view this as a missed opportunity. Prefabricated robots have an exciting set of capabilities while constructionist design approaches can lead to personally meaningful learning experiences. The Building for Robots approach seeks to leverage creative design and construction with black box robots. In doing so, we endeavor to blend affordances of the two. In investigating this approach, we seek to understand ways that contemporary robots can support the underlying values and philosophies of constructionism - i.e., design, making, personal and epistemological connections (Harel & Papert, 1991; Papert, 1980).

The Building for Robotics Approach

The main focus of the Building for Robots activity was for learners to use common building materials to construct worlds for their robots to explore. In this way, rather than focusing on designing robots or programming existing robots to complete predetermined tasks, the children were tasked with designing and constructing environments and then deciding how they want their robots to move through the world they built. Over the course of the activity, the children iterated on their environmental designs, building increasingly complicated environments and longer, more sophisticated programs to explore them.

The Building for Robots approach presents the learner with common arts-and-crafts construction materials (e.g. cardboard, popsicle sticks, pipe cleaners) and then asks them to construct a world for a programmable robot to explore. Three features are central to this approach. First is the familiarity of the construction materials. This means materials that are readily available to learners and items the learners already know how to use. Second, the building challenge encourages learners to draw on their own lived experiences and daily lives. This is reflected in both the aforementioned materials as well as the types of environments built. This aspect of the activity is both a result of how learners are introduced to the activity as well as a result of the materials used (as will be shown below). Finally, learners are given complete control over the materials, the robot, and how the two are going to interact. In this way, learners are empowered to build what they want, how they want, and they can have their robots navigate their world as they see fit.

Methods

Participants and Context

This research focuses on a series of robot design activities that were included as part of a summer camp at a local church. The camp was created to engage children in the community with educational experiences during the summer break from school and was led by the church's assistant pastor and youth coordinator. There was a nominal fee to participate in the camp designed to ensure full-time attendance. The camp was held Monday through Friday from 9 am – 5 pm for 4 weeks during the summer. The Building for Robots activity was comprised of 2 2.5-hour sessions each week for the 4 weeks. Two researchers led all 8 design sessions. One led the design, building and programming activities while the other researcher took field notes and captured video and audio data.

The camp included 58 kids ranging in age from 7-13, 49 of which participated in the study (2 second graders, 7 third graders, 9 fourth graders, 7 fifth graders, 9 sixth graders, 8 seventh graders, and 7 eighth graders). Child participants were drawn from the local community, characteristics of which can be seen by looking at the neighbouring public school, which is racially diverse (57% Black or African American, 34% Hispanic/Latino, 4% White, 3% Asian) with 61% of its attendees receiving free or reduced-priced meals and 20% English language learners. Along with the young participants, 16 high school student volunteers from the wider region supported facilitators during the program. The 58 camp attendees were split into 10 groups, with 9 of these groups presenting their environments and programs during the final Building for Robots session.

During Building for Robots activities youth constructed worlds to be explored using the Sphero SPRK+ robot (Figure 1). The Sphero is a spherical robot that can be programmed via tablets using either a block-based or a text-based programming interface. The robot includes motors, accelerometers, LED lights, and communicates with the tablet via Bluetooth. The Sphero programming language includes primitives to move the Sphero, respond to external inputs (such as collisions), change colors, make the tablet produce noise, along with containing conventional programming constructs such as iterative logic, conditional logic, variables, and functions.

Figure 1: The Sphero SPRK+ robot.

Data Collection and Analysis

Distributed constructionism serves as the framework we use to both structure and understand the learning space (Resnick, 1996). Video, audio, researcher notes, and learner constructions (both the physical worlds and virtual programs) were collected and analysed. The focus of the analysis was on looking for various ways learners engaged with the Building for Robot task to identify affordances of the construction activity and knowledge and resources learners drew upon as they engaged in the activity. Data analysis was performed based on emerging themes observed and discussed by the research team.

Affordances of Building for Robots

In this section, we present three affordances of the Building for Robots approach that were identified in our analysis of the data collected. These three affordances are: (1) a low barrier entry to building with robots, (2) engagement with robots grounded on existing knowledge and experiences with the immediate environment, and (3) mutually-informing practices of building and programming.

Low-barrier to Entry

A primary affordance of the Building for Robots approach is that it provides a low barrier to entry in two specific but distinct ways. First, as the primary construction activity uses conventional arts-and-crafts materials, it is possible for very young learners to authentically participate in the activity in a way that might not be possible when construction takes the form of soldering or precisely assembling smaller gears. The low barrier can also be seen in the programming phase of the activity. Since running the program involves the robot navigating the physical space being constructed, younger participants can be involved by interacting directly with the robot (e.g., picking it up and returning it to the start position), or by refining and reconstructing the environment after a run (Figure 2). Both of these activities serve as ways for those too young to traditionally be involved in robotics activities to engage in legitimate peripheral participation (Lave & Wenger, 1991). This low-barrier to entry aspect of Building for Robots can be seen in the composition of the groups who participated in the activity and in the designs of specific groups enabled the youngest member to be fully participating members of the group.

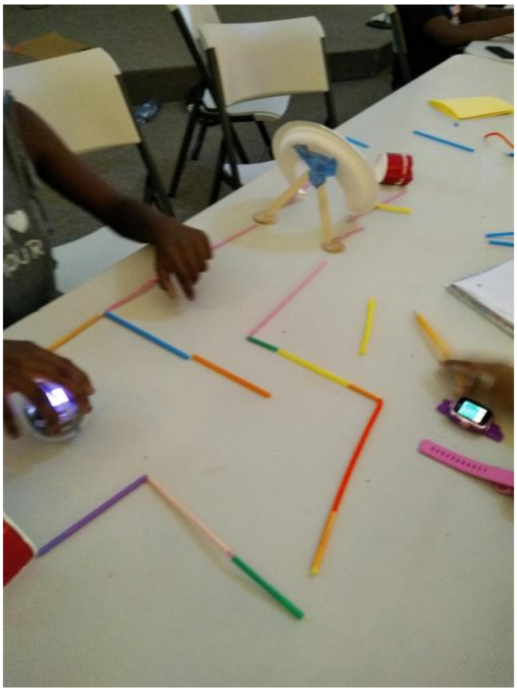
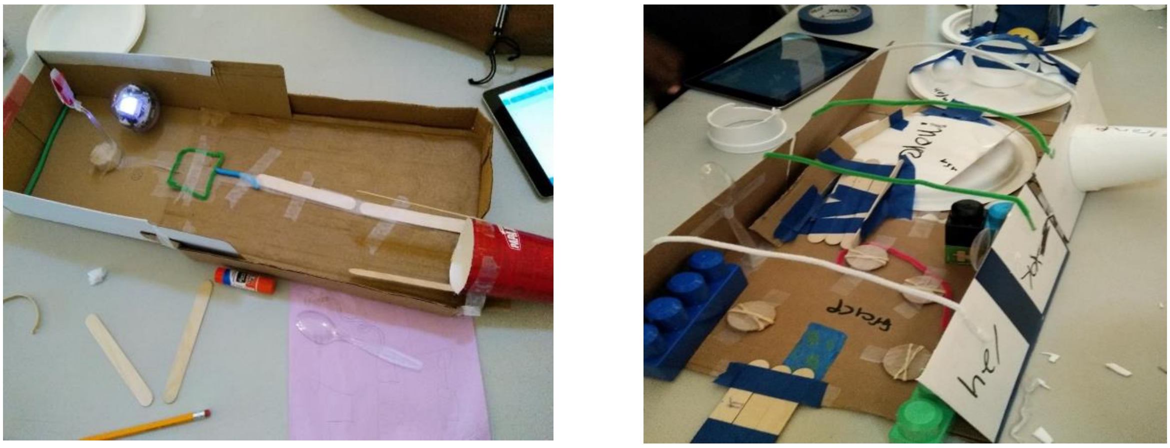


Figure 2: After a run of the Sphero, a young participant moves the robot back to the start position and adjusts the environment, resetting it to its initial starting state.

A second way Building for Robots presents a low-barrier to entry is in the fact that the construction materials used in the activity are inexpensive and readily available. Unlike robot construction kits that can cost significant amounts of money (e.g. Lego Mindstorms EV3 currently retails for $349.99 USD), the materials used when constructing for robots are inexpensive and at times free (e.g., recycled pizza boxes). While a potentially expensive robot is still required for the activity, it is possible to share a single robot among a large group. This also makes it more possible to run robotics activities by relying on temporary access to the robot through programs like library technology checkouts. Since the construction is happening around the robot, the user need not own the robot, this relieves the steep monetary barrier to entry that exists for some robotics tools kits.

This second form of low-barrier to entry was on full display throughout this study. First, an inexpensive church-run summer camp with a very limited budget was able to actively engage 58 children in four weeks of robotics programming activity. The constructions produced relied heavily on inexpensive crafts materials or other materials available, such as leftover pizza boxes and empty packing containers found around the church. Looking across learner constructions, we can see how the low-cost materials were utilized, sometimes in literal ways, like when a cup from a local pizza place was used to represent a pizza restaurant that the robot would visit (Figure 3A) or in more imaginative ways, such as how one group used little wooden tokens taped to a piece of cardboard to serve as hazardous materials for their robot to avoid (Figure 3B).



(A) (B)

Figure 3: (A) Learners use a cup from a pizza restaurant to represent the final destination for the robot and (B) Learners tape wooden tokens on cardboard to represent hazardous obstacles for the robot.

Grounding Engagement with Robotics in Existing Experience

A second affordance of the Building for Robots approach is in how shifting focus to building the environment allows the learners to draw on prior knowledge and experiences as part of engaging in the robotics activity. By this we mean, the vast design space presented by asking them to build something for the robot (rather than the robot themselves) provides an opportunity for them to draw upon a wide array of prior experiences and ideas. This is best demonstrated by looking at the types of environments and challenges that were built by the learners over the course of the workshop. For example, one group built a course for their robot to navigate that mirrored the drive that their parent makes going to and from work, including the introduction of stop signs and stops for errands along the way (Figure 4). There were also projects where the robot played the role of mailman delivering mail, a shopper that needed to scan barcodes as it shopped, and a pizza delivery man out making deliveries (Figure 5). Other constructions included features like car washes in the forms of tunnels and gas stations for the robot to stop at along its journey. Looking across these activities, we see inspiration drawn from daily life while also aspects of creativity and imagination.

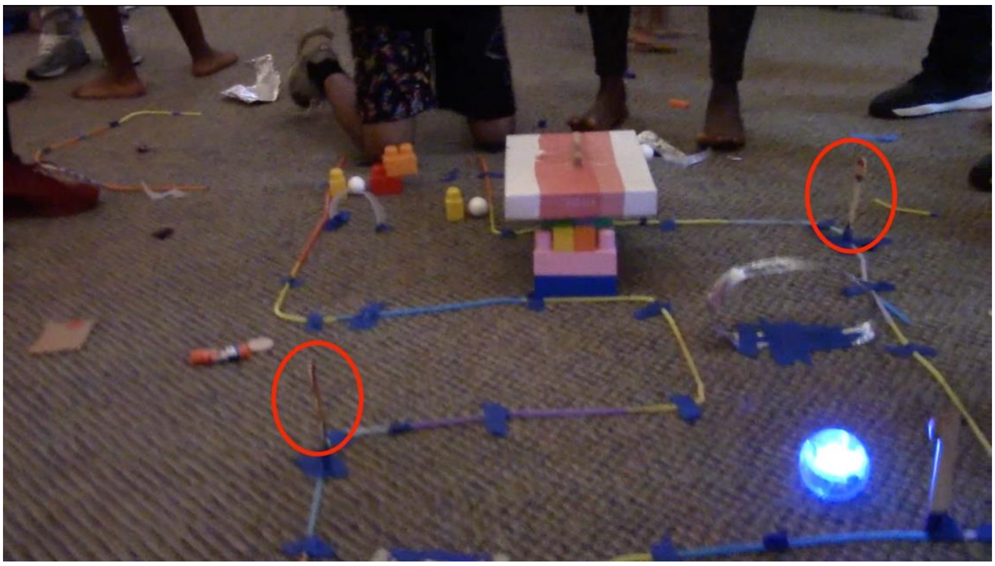


Figure 4: A robot navigates a road including stop signs (circled in red). The robot briefly stops at stop signs and changes colors from blue to red before proceeding.

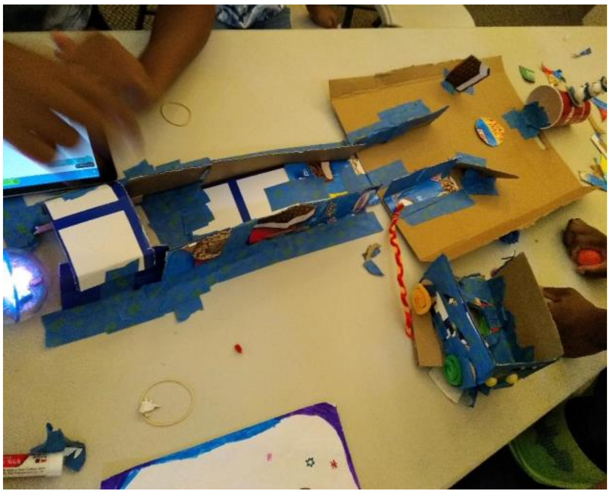


Figure 5: Camp participants design a drive-through track for a robot to navigate to go get a pizza

Mutually-informing practices of building and programming

A third notable affordance of the Building for Robot approach is the unique way that the construction and programming practices become mutually informing. By this we mean, the act of programming informs the construction that is being built while the construction being built informs the program that is being authored. Designing spaces for a programmable robot to navigate provided opportunities for learners to simultaneously work on the designs and program. This is quite distinct from conventional robot construction activities that focus on building the robot because in robot-construction activities programs cannot be run until the robot construction is complete. In contrast, when the construction is happening around an already functioning robot, there is more opportunity to move back-and-forth between programming and constructing. These mutually-informing practices allow for learners to be actively engaged in the process at the same time while also providing generative challenges and solutions throughout the activity. The participants in this activity regularly switched between the physical and the virtual as they refined their designs.

In our analysis, we found instances where constraints encountered while programming the robot shaped the physical construction. For example, one group introduced a traffic light to their environment in the form of wooden discs that were painted red, green and orange. In discussing how the traffic lights would change to allow the robot to stop or move, the learner raised the issue of there being no interactivity between the discs with the robot. As the learners were working through the initial idea, one participant said: "*How are we making the traffic lights? We can color one side of the disc red and color one side green and put them here.*" A discussion on how long to make the robot pause ensued, leading to one learner proposing using a stop sign instead, saying "*When it gets here (pointing to the floor), it has to stop then wait for like 3 seconds just like when you are driving you have a stop sign and [Learner 3] was saying something about traffic light, but we don't really know how to make it work*". Here we can see the issue of not knowing how to program the stoplight behavior shaping the physical construction.

At the same time, there were many instances of the physical construction itself shaping the programs being authored. In fact, the constructions served as the driving motivation for the programming. This interaction could be seen in how learners went about building their environments. Several groups took the approach of incrementally authoring their programs and building their constructions in parallel, meaning they'd build a bit in the physical world, then spent time expanding their programs to account for what was just built, before returning to build the next chunk of their construction. Other cases saw groups complete their physical constructions and then author working programs only to go back and add new obstacles to the course. This type of interactivity and mutually-informing practice produced a generative pattern in which teams kept building more elaborate courses and more sophisticated programs.

Challenges and Limitations of Building for Robots

While there are advantages to the Building for Robots approach compared to the conventional robot construction kit design, there are also drawbacks. First and foremost, among them is that the machine itself remains a black box and learners are not given insight or access to the bits and pieces that make the robot work. Likewise, as the robots are not the focus of the construction, it makes it difficult or impossible to allow learners to personalize the robots themselves. This is quite different from robot construction kits where this type of personalization is an essential part.

A second potential limitation of the Building for Robots approach is that the activity becomes spatially bound. Whereas it is possible (albeit at times inconvenient) to pack up a robot mid-construction, many of the constructions from this project could not be easily moved, especially the larger constructions. Many groups ended up taping things to the floor or having individual components of their machine that couldn't be moved together. This issue was felt acutely by groups who had programs that were close to working on their constructions in one session, only to find out they had to recalibrate all their commands the next session when they reassemble the construction causing distances to be slightly different than what they had been before.

A third drawback to this approach that emerged during this workshop was learners building things that the robot could not navigate or complete, either because the programming required was beyond the learners' abilities or because the physical demands of the course were beyond the capability of the robot itself. We saw this in the form of ramps that were too steep or courses requiring very high levels of precision to complete. This problem was exacerbated by the lack of precision in the running of programs due to external factors, such as the room having carpeting rather than a flooring with a high friction coefficient that is better for the Sphero.

A final drawback of this approach, at least based on this workshop we ran, was that learners explored relatively little of the capabilities of the robots themselves. This can be seen most notably in the contents of the programs written. Most programs used a small subset of the available set of blocks to complete their programs. This mostly consisted of long sequences of consecutive move commands. Few programs included commands associated with logic operators (e.g. conditional logic), robot events (e.g. on collision), variables, or other robot capabilities such as changing colors, producing audio, or reading in values from sensors. This limitation could probably be partially addressed by having more scaffolded construction activities were learners were encouraged to incorporate one feature or another, but those were largely absent from the programs produced as part of the open-ended approach used in this workshop.

Conclusion

This paper introduces the Constructionist learning approach called Building for Robots. This approach retains many aspects common to robot-based learning experiences but introduces some slight tweaks that bring to life a unique set of affordances that can lead to productive engagement. The Building for Robots approach has learners use conventional arts-and-crafts materials to build environments for robots that the learners are programming. Shifting the construction to the surrounding environment rather than the robot itself results in a low barrier to entry and opportunities to draw on learners' prior knowledge and lived experiences. Further, this design opens up new forms of engagement with construction and programming including making programming and construction mutually-informing practices. Collectively, these features provide an alternative way for learners to engage in meaningful constructionist learning with robots.

As new tools, technologies, and forms of engaging with computing emerge, it is important that we continually revisit the types of learning experiences we create around them. Technology on its own does not produce effective learning, it is in the ways that learners take up, use, explore, tinker, and toy with the technologies that meaningful learning occurs. Considering new forms of engagement and alternative ways to support learners in exploring and expressing ideas with technology is as important as designing new technologies themselves. With this work, we explore an alternative way to bring constructionist ideals to robotics and robotics construction kits and hope to encourage others to explore new ways to use robots and robotics toolkits to engage learners. In doing so, we seek to create yet another way for learners to meaningfully engage with the powerful ideas of computing, thus helping them in their path towards fully participating in our increasingly technological world.

Acknowledgments

We would like to thank the kids who participated in this study. This material is based upon work supported by the National Science Foundation under Grant No. 1441523 and Spencer Foundation Grant No. 201900099. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the funding agencies.

References

Alimisis, D., Moro, M., Alfredo Pina, J. A., Frangou, S., & Papanikolaou, K. (2007). *Robotics & Constructivism in Education: The TERECoP project*. 11. Bratislava.

Blikstein, P. (2008). Travels in Troy with Freire: Technology as an agent for emancipation. *Paulo Freire: The Possible Dream. Rotterdam, Netherlands: Sense*.

Blikstein, P. (2013). Gears of our childhood: Constructionist toolkits, robotics, and physical computing, past and future. *Proceedings of the 12th International Conference on Interaction Design and Children*, 173–182.

Buechley, L., Eisenberg, M., Catchen, J., & Crockett, A. (2008). The LilyPad Arduino: Using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 423–432. ACM.

Chambers, J. M., Carbonaro, M., & Rex, M. (2007). Scaffolding Knowledge Construction through Robotic Technology: A Middle School Case Study. *Electronic Journal for the Integration of Technology in Education*, *6*, 16.

Eisenberg, M. (2002). Output devices, computation, and the future of mathematical crafts. *International Journal of Computers for Mathematical Learning*, *7*(1), 1–44.

Freire, P. (1974). *Pedagogy of the Oppressed*. Basic Books.

Harel, I., & Papert, S. (Eds.). (1991). *Constructionism*. Norwood N.J.: Ablex Publishing.

Holbert, N. (2016). Leveraging cultural values and “ways of knowing” to increase diversity in maker activities. *International Journal of Child-Computer Interaction*, *9–10*, 33–39. https://doi.org/10.1016/j.ijcci.2016.10.002

Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., & Lui, D. (2014). A Crafts-Oriented Approach to Computing in High School: Introducing Computational Concepts, Practices, and Perspectives with Electronic Textiles. *Trans. Comput. Educ.*, *14*(1), 1:1–1:20. https://doi.org/10.1145/2576874

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge Univ Pr.

Lego Systems Inc. (2008). *Lego Mindstorms NXT-G Invention System*. Retrieved from http://mindstorms.lego.com

Martin, F., Mikhak, B., Resnick, M., Silverman, B., & Berg, R. (2000). To Mindstorms and Beyond. *Robots for Kids: Exploring New Technologies for Learning*, 9.

McNerney, T. S. (2004). From turtles to Tangible Programming Bricks: Explorations in physical language design. *Personal and Ubiquitous Computing*, *8*(5), 326–337.

Mitnik, R., Nussbaum, M., & Soto, A. (2008). An autonomous educational mobile robot mediator. *Autonomous Robots*, *25*(4), 367–382. https://doi.org/10.1007/s10514-008-9101-z

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic books.

Resnick, M. (1996). Distributed Constructionism. *Proceedings of the 1996 International Conference on Learning Sciences*, 280–284.

Resnick, M, Berg, R., & Eisenberg, M. (2000). Beyond black boxes: Bringing transparency and aesthetics back to scientific investigation. *The Journal of the Learning Sciences*, *9*(1), 7–30.

Resnick, M, Bruckman, A., & Martin, F. (1996). Pianos not stereos: Creating computational construction kits. *Interactions*, *3*(5), 40–50. https://doi.org/10.1145/234757.234762

Resnick, M, Martin, F., Berg, R., Borovoy, R., Colella, V., Kramer, K., & Silverman, B. (1998). Digital manipulatives: New toys to think with. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 281–287.

Resnick, M., & Rosenbaum, E. (2013). Designing for Tinkerability. In M. Honey & D. Kanter (Eds.), *Design, make, play: Growing the next generation of STEM innovators* (pp. 163–181).

Resnick, M., & Silverman, B. (2005). Some reflections on designing construction kits for kids. *Proceedings of the 2005 Conference on Interaction Design and Children*, 117–122.

Rusk, N., Resnick, M., Berg, R., & Pezalla-Granlund, M. (2008). New Pathways into Robotics: Strategies for Broadening Participation. *Journal of Science Education and Technology*, *17*(1), 59–69. https://doi.org/10.1007/s10956-007-9082-2

Sargent, R., Resnick, M., Martin, F., & Silverman, B. (1996). Building and Learning with Programmable Bricks. In *Constructionism in practice* (pp. 161–173).

Sipitakiat, A., Blikstein, P., & Cavallo, D. P. (2004). GoGo board: Augmenting programmable bricks for economically challenged audiences. *Proceedings of the 6th International Conference on Learning Sciences*, 481–488.

Yu, J., & Roque, R. (2018). A survey of computational kits for young children. *Proceedings of the 17th ACM Conference on Interaction Design and Children - IDC '18*, 289–299. https://doi.org/10.1145/3202185.3202738

Characteristics of primary school students' Scratch code

Dimitrios Nikolos, *dnikolos@upatras.gr*
Dept of Educational Sciences and Early Childhood Education, University of Patras, Greece
Vassilis Komis, *komis@upatras.gr*
Dept of Educational Sciences and Early Childhood Education, University of Patras, Greece

Abstract

In the present study we discuss characteristics of primary school students' Scratch code. Characteristics of code that appear in other Scratch code analyses include a) Dead code, b) Initialization problems, c) Extremely Fine-Grained Programming (EFGP), d) Duplicate code and e) Long scripts. Does student code reflect code that is found in large Scratch project repositories or other settings? Does student code have other characteristics? We taught two fifth-grade classes (23 students each) Scratch for 9 hours. Learning Scratch was part of a compulsory 1 hour per week ICT course. After each hour we gathered students' projects that comprised our data (code base). The analysis of the code base showed that many of the characteristics of code that were found elsewhere were also present in the classroom, namely, dead code, initialization problems and EFGP. Duplicate code and long scripts did not appear in the code base in a large percentage. We also discovered that students' projects featured race conditions and code that had no effect. Such codes provide learning opportunities. When students create these types of codes**,** they face a problematic situation and they will have to reflect on their code. Since Scratch makes tinkering easy by design, they may find a way to move forward, otherwise the teacher should be able to assist them.

Keywords

Scratch, code, primary school students, dead code, initialization, EFGP

Introduction

Scratch is a programming language that enables young students to create their own games and other types of interactive media (Resnick et al., 2009). Scratch originates from Logo and aims in enabling students to create meaningful artifacts. Doing so, they engage with a wide range of computational concepts that promote computational thinking (Brennan & Resnick, 2012; Meerbaum-Salant, 2013). These computational concepts can be found in the code of Scratch projects and include sequences, loops, parallelism, events, conditionals, operators, and data.

School teachers may use Scratch to extend student knowledge beyond basic office skills (Crook, 2010). Students face problems while programming with Scratch. Every problem that occurs when students create meaningful artifacts is a learning opportunity. When students overcome such problems, they construct their own knowledge. If they are unable to do so by themselves, teachers can look inside student code and help them. In most cases, teachers will find that some of the code works as intended and some other code has problems or doesn't work at all.

Code that can cause a Scratch project not to work as intended includes:

- Dead code: code that is not executed (Aivaloglou & Hermans, 2016). Dead code includes procedures that are not invoked, unmatched broadcast-receive messages, code that is not invoked and empty event scripts. Static analysis of code can locate dead code and it was found in more than 25% of the analyzed projects in (Aivaloglou & Hermans, 2016). Dr. Scratch, a tool that is used for evaluating computational thinking in Scratch projects also takes dead code into account (Moreno-León & Robles, 2015).
- Problematic initialization: changes in the state of the program that are not restored. An example of problematic initialization is a Scratch project that hides a sprite but never shows it again, not even when the program is restarted. The reason for this is that Scratch stores the state of the program between executions. By convention, a click on the green flag restarts the program (Franklin et al., 2016). Hairball can detect initialization problems using static code analysis (Boe et al., 2013).

Code that may hamper learning programming includes:

- Extremely Fine-Grained Programming (EFGP): Code that is fragmented in small logical blocks (Meerbaum-Salant, 2011).
- Long scripts: Large scripts that are not easily understood (Hermans & Aivaloglou, 2016).
- Duplicate code: Code that is repeated either in the same project or even in the same object (Hermans & Aivaloglou, 2016).

In the present study we determine a) which of the above code patterns can be found when primary school students learn Scratch in the classroom and b) what other problematic code, if any, emerges during the process. In the following section methodology is described, subsequently results are drawn, discussion and conclusions follow.

Methodology

The study was conducted with two fifth-grade primary school classes, during a compulsory ICT course. Each class had 23 students and they worked in groups of 2 or 3 students. The course took place in a primary school computer lab with 13 computers equipped with the Scratch 2.0 software.

The curriculum

We developed a curriculum 9 hours in duration. Students had to create a project that featured a basic Scratch concept in each hour. The Scratch concepts of each lesson were:

1. Introduction: Students navigate in the Scratch programming environment

2. Views: Students learn to change the appearance of Scratch objects and scene.

3. Interaction: Students learn to program interaction with the use of *"if <> then else"*.

4. Messages: Students learn to program broadcasting and receiving messages for synchronizing.

5. Variables: Students learn to add score in a game.

6. Recap: Students create a project with concepts they have already learned.

7. New blocks: Students create a project that features "New blocks" (procedures).

8. Clones: Students learn to program clones of their objects.

9. Students create their own games and finish the course.

After each lesson, students had to submit a project that featured the basic concept.

The projects

During the nine weeks of the course we gathered 152 projects that the students created (Table 1). We did not study the projects of the first week because they were very basic. The course took place in a regular school setting and we had common computer lab problems. Some students did not save their work and some computers malfunctioned during class. That is why there are different number of projects in each class and lesson.

Table 1. Number of projects

lesson	**Number of projects**	
	Class 1	**Class 2**
2. Views	9	8
3. Interaction	9	6
4. Messages	10	13
5. Variables	12	9
6. Recap	12	10
7. New blocks	11	12
8. Clones	6	9
9. Students create their own games	10	6
Total	152	

Some of the projects that the students created may not fit into the group of the projects of their peers. To find these outliers, we scored the projects using Dr. Scratch and created the boxplot of the scores grouped by lesson (Figure 1). The eleven outliers of the box plot were eliminated from the code analysis, leaving 141 projects that comprise the data (code base) of the present study.

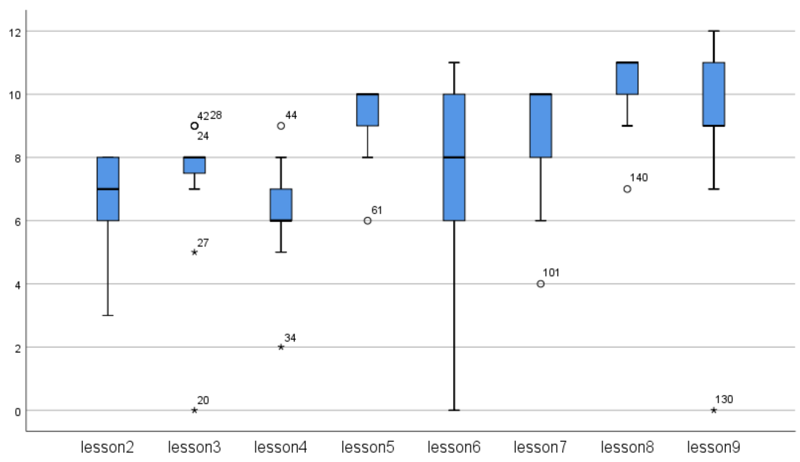


Figure 1. Dr. Scratch score of projects

The projects that the students created are small projects (Table 2). They feature 3.2 sprites on average, and 6.25 scripts per project, as opposed to projects downloaded from the Scratch website that featured 5.68 sprites per project and 17.35 scripts per project on average (Aivaloglu & Hermans, 2016).

Table 2. Description of projects

Lesson	Average Number of sprites per project	Average number of scripts per project
2.Views	2.47	3.70
3.Interaction	3.2	4.90
4.Messages	3.6	3.76
5.Variables	3.05	8.30
6.Recap	4.05	5.91
7.New blocks	1.45	6.14
8.Clones	4.29	9.57
9.Own programs	3.53	8.33
Average	3.2	6.25

Each project was studied in respect to the following questions:

- How many blocks are never executed (dead code)?
- Does the project feature problematic initialization?
- Does the project feature Extremely Fine-Grained Programming?
- Does the project feature long scripts?
- Does the project feature duplicate code?
- Does the project feature other problematic code?

The results of this analysis are described in the next section.

Results

Dead code

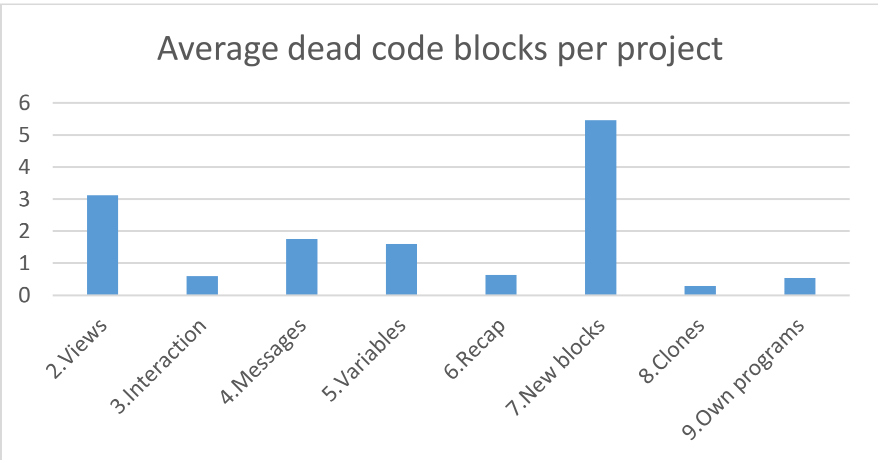


Figure 2. Average dead code blocks per project for every lesson

We counted the blocks that are not executed (dead code) in our code base. Projects contained 1.94 dead code blocks in average. Several projects contained no dead code. The maximum number of dead blocks that appeared in a project was 19. This project was submitted after lesson

7 (New Blocks) where we observed the maximum number of dead code blocks in general (Figure 2).

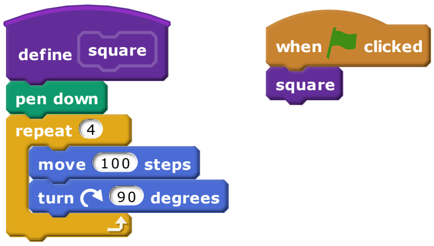


Figure 3. Complete example of "New blocks" usage

"New blocks" is the name of the way that Scratch implements procedures. A complete example of the usage of "New blocks" in a project is shown in figure 3. A new block is comprised of a definition and an implementation. It then needs to be called on an event in order to be executed.

Dead code was present in 18 out of the 22 submitted projects after the "New Blocks" lesson. We documented the reasons for dead code occurrence in relation to the concept of "New Blocks". We found four reasons for dead code occurrence: a) students did not attach the blocks of the implementation to the new block definition, or omitted the definition (4 projects), b) students did not call the new block at all (5 projects), c) students did not call the new block on an event (10 projects) and d) students called the new block in the wrong sprite (1 project).

Problematic initialization

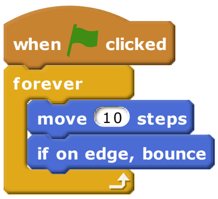


Figure 4. Example of code with no need for initialization

As far as initialization is concerned a project can have either:

a) correct initialization: the state of the program resets (usually on green flag click),
b) problematic initialization: the state of the program does not fully reset, or does not reset at all
c) no need for initialization: the state of the program does not need to reset. An example of a code that changes the state, i.e. the position, of a sprite without need for initialization is shown in figure 4. The sprite is continuously moving, and its starting position is unimportant.

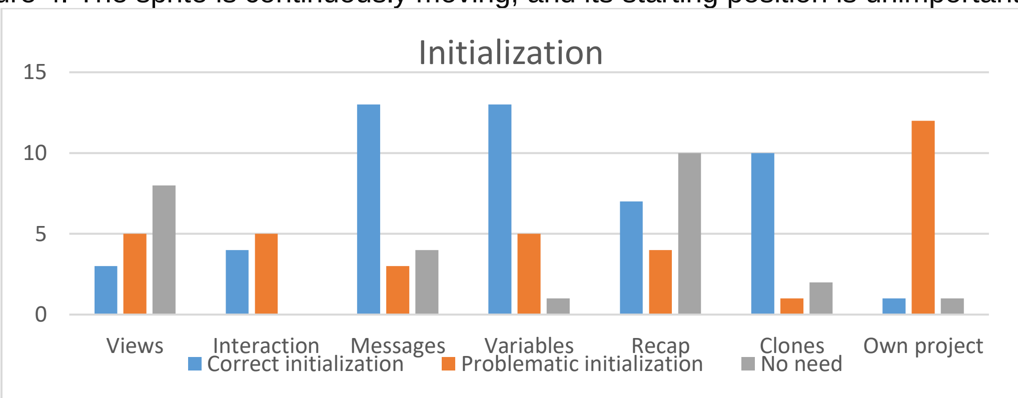


Figure5. Initialization of projects

We counted the number of projects that featured correct initialization, problematic initialization and no need for initialization (Figure 5). We noticed that when students created their own games, they did not implement correct initialization. The reason for the correct initialization in projects of lessons about messages and variables is that initialization was a vital part of these lessons.

Extremely Fine-Grained Programming

When a component of the program, e.g. a variable, is accessed or modified by two or more scripts in the same sprite, the script features Extremely Fine-Grained Programming (EFGP). We counted the occurrences of EFGP (Table 3) and found that around a quarter of the projects featured EFGP.

Table 3. Number of EFGP occurrences

	EFGP Projects	Total projects	Percentage
2. Views	7	17	41%
3. Interaction	3	10	30%
4. Messages	4	21	19%
5. Variables	4	20	20%
6. Recap	0	22	0
7. New blocks	16	22	72%
8.Clones	0	14	0
9. Own games	1	15	7%

While teaching new blocks (lesson 7) students created code like that of figure 6. The left script draws a square and the right script draws a triangle. However, if the sprite is clicked an irregular shape will be drawn because of the concurrent model of execution. Such codes account for the increased number of EFGP projects in lesson 7.

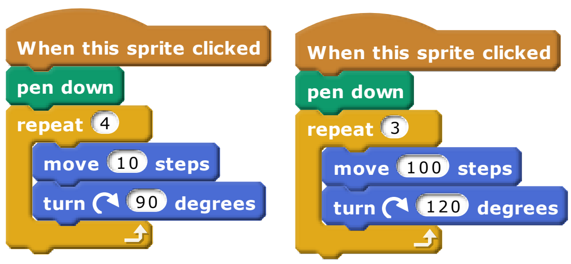


Figure 6. Extremely Fine-Grained Programming after lesson 7

We noticed that when students program their own games, EFGP does not occur (lessons 6 and 9). A possible explanation for this is that students created relatively small projects.

Long scripts

In figure 7 the distribution of projects per maximum script size is shown. Most projects feature maximum script size between 5 and 10 blocks. Only 1 out of the 142 projects featured a very long script with 32 blocks in one script. This script is shown in figure 8. The code checks if a sprite is touching one of the five bad apples that existed in the project. This code can be shortened using logical operators or clones.

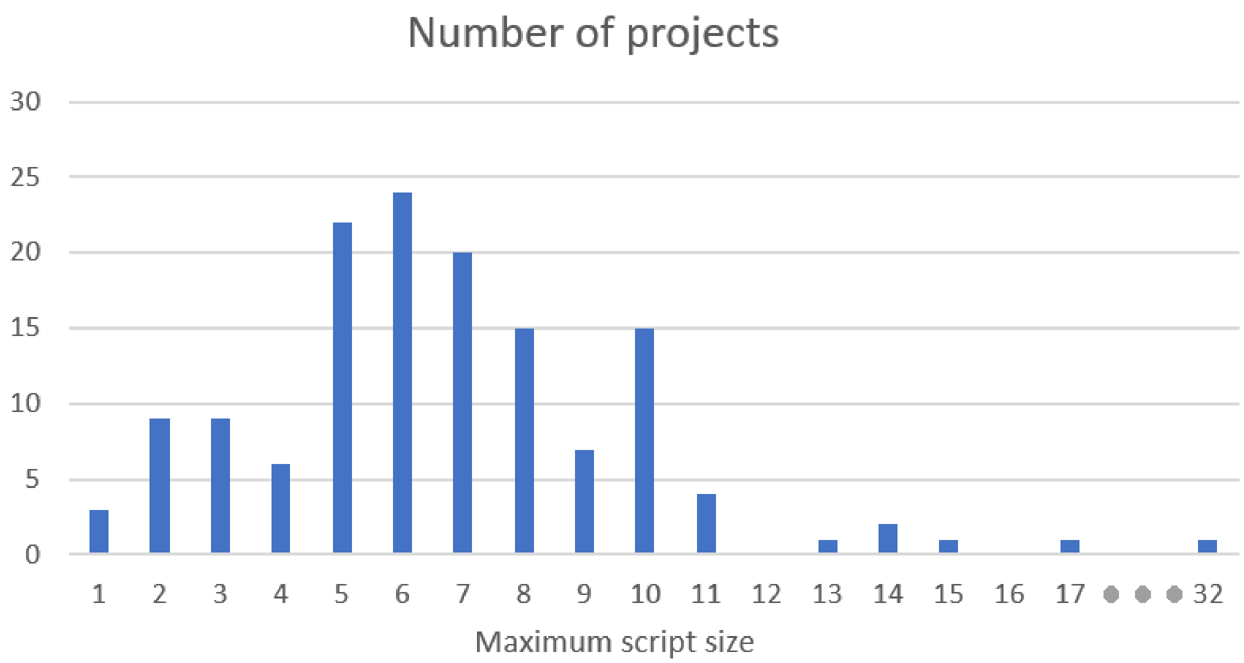


Figure 7. Distribution of number of projects per maximum script size

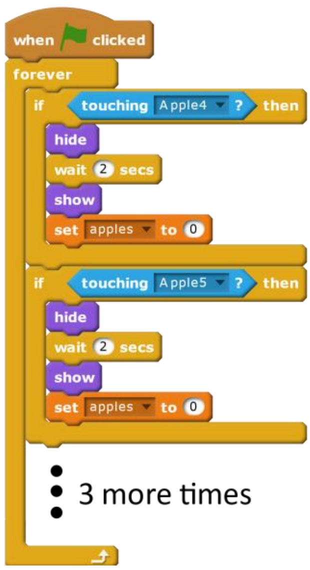


Figure 8. The long script

Duplicate code

Only 5 of the 142 featured projects featured duplicate code. As with long scripts the reason might be that students created small projects in this course. Interestingly, the three of the five projects that featured duplicate code were submitted after the students were familiarized with clones. When the course created the need for multiple sprites, student responded by creating duplicate code.

Other problematic code

We noticed that the already studied problematic code categories (dead code, problematic initialization, EFPG, long scripts, and duplication) were not the only ones that appeared in our code base. Others include a) race conditions and b) code without effect.

Race conditions

When students design games they want to program concurrent events (Kafai, 1995). Scratch provides a concurrent environment that makes these events possible (Maloney et al., 2010). Even though concurrency is intuitive, since we live in a concurrent world, problematic concurrent code emerges in Scratch programming (Meerbaum-Salant, et al., 2011). A case of problematic concurrent code occurs when the program modifies a component, e.g. a sprite attribute, in scripts that are executed at the same time. In that case, the code outcome depends on the way Scratch executes the program. Such codes create race conditions in concurrent programming languages (Netzer & Miller, 1992). In the example of figure 9 there is no way to determine the costume of the sprite by inspecting the code. The student cannot predict the outcome of the code, they can only observe it after execution.

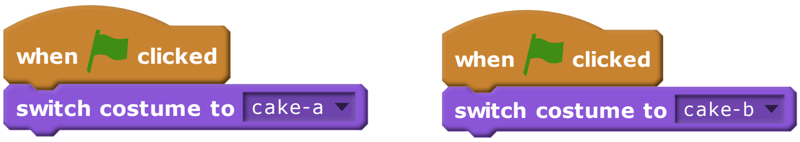


Figure 9. Race condition example

Race conditions occur if fragmented code (EFGP) exists. On the other hand, not all fragmented code leads to race conditions. An example of fragmented code that does not cause race conditions is shown in figure 10.

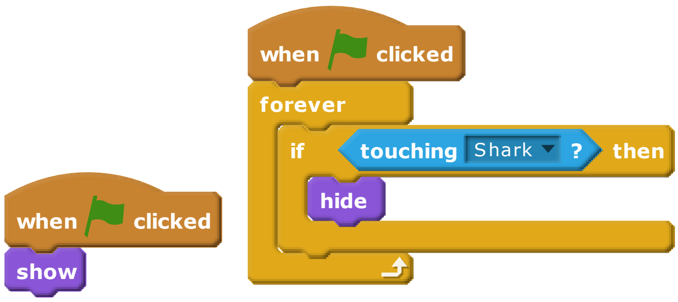


Figure 10. Fragmented code without race conditions

We counted the projects that created race conditions and we noticed that almost all the projects that featured EFGP created race conditions (Table 4).

Table 4. Number of projects with race conditions

	Projects with race conditions	EFGP Projects
2. Views	7	7
3. Interaction	1	3
4. Messages	4	4
5. Variables	0	4
6. Recap	1	0
7. New blocks	16	16
8.Clones	0	0
9. Own games	0	1

Code without effect

Students get immediate visual feedback when they program with Scratch (Maloney et al., 2010). They rapidly get used to anticipating changes in the code to reflect on the program outcome. However, they may create code that runs but has no effect. The *"hide"* block is code without effect in the example of figure 11. Since the *"show"* block is executed immediately after the *"hide"*, students do not observe the execution of the *"hide"* block. Notice that this kind of code does not fall into the dead code category. This code *is* executed but has no effect.

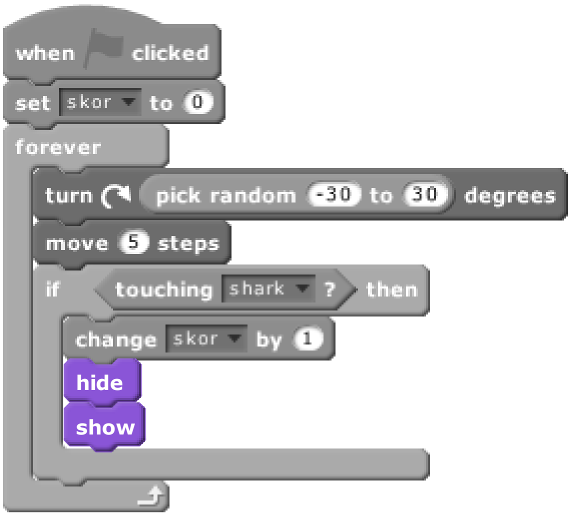


Figure 11. "Hide" block has no effect

We counted the projects that featured codes without effect and found 13 such codes. Other examples of code without effect are shown in figure 12. The *"change [ghosts] by (1)"* block has no effect. On the code on the right, the clone is deleted immediately after its creation.

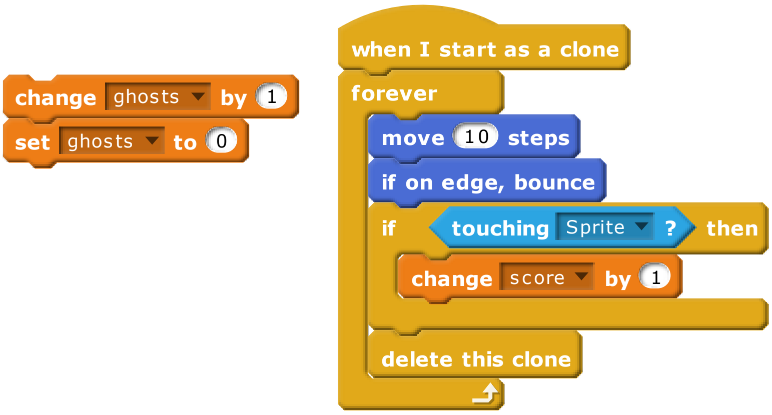


Figure 12. Code without effect

Primary school students created projects that feature dead code, problematic initialization and EFGP. Long scripts and duplicate code did not appear in a large percentage of the projects. During the study other problematic code such as race conditions and code without effect emerged.

Discussion

Following the tradition of Logo (Papert, 1980), Scratch programming language enables students to create artifacts they care about. Students use computer code to program the behaviour of their objects and when an unintended behaviour occurs, a learning opportunity emerges. In the present study, we identified such code.

We found that students leave dead code in their projects. Similar results are found in large Scratch project repositories (Hermans & Aivaloglou, 2016) and dead code is one of the bad smells detected by Dr. Scratch (Moreno-León & Robles, 2015). Dead code is not confusing in all cases. In some cases, students leave dead code for further usage and Scratch allows it by design (Resnick & Rosenbaum, 2013). But it is also possible for students to leave dead code because they have not constructed a mental model for a concept. In our study, we found that the "New blocks" concept led to increased dead code. We noticed that besides code that is not executed, i.e. dead code, students created code that was executed but had no effect. We feel that this type of code may puzzle students since it sabotages the immediate feedback that Scratch provides by design (Maloney et al., 2010).

Initialization problems are found in large Scratch project repositories (Aivaloglu & Hermans, 2016). Initialization is a point of concern in teaching programming either using professional languages, Logo (Lee & Lehrer, 1988) or Scratch (Franklin et al.,2016). In the present study, we found that initialization problems exist when students use Scratch in the classroom.

Most Scratch users make use of concurrent execution of stacks of code (Maloney et al., 2008). When this feature is overused, Extremely Fine-Grained Programming (EFGP) appears (Meerbaum-Salant et al., 2011). We found that EFGP occurs when students learn Scratch in the classroom. Furthermore, we noticed that race conditions appear along with EFGP. Race conditions can be confusing for students since they produce unpredictable results.

We did not find many projects with long scripts or with duplicate code in our code base, though students have been found to develop such code (Meerbaum-Salant et al., 2011). One possible explanation for this is the small size of students' projects. It is possible that both these types of code would appear if students created more complex projects. Code duplication was one of the code smells that was detected in higher numbers in large projects using Dr. Scratch (Vargas-Alba et al., 2019).

Conclusions

Discussing Scratch code is not about bad practices, professional habits, software maintenance or performance. Discussing Scratch code is about what students may find counterintuitive and what learning opportunities appear in the process of creating a project. While Scratch is taught in the classroom, students tinker and try out code in a constructionist approach. But, even with a programming language that is designed for novices, the produced code can be confusing.

Teachers may use the code that was described in this study as grounds for fruitful discussion with the student, ideally when the student stumbles upon it by themselves like our students did. It is possible that manifestations of such code are the result of the student's understanding of Scratch code, further research is needed to provide insights about this connection.

References

Aivaloglou, E., & Hermans, F. (2016). *How Kids Code and How We Know: An Exploratory Study on the Scratch Repository*. *Proceedings of the 2016 ACM Conference on International Computing Education Research* (ICER '16). pp. 53–61.

Boe, B., Dreschler, G., Barbara, U. C. S., Hill, C., Conrad, P., Barbara, U. C. S., & Franklin, D. (2013). *Hairball: Lint-inspired Static Analysis of Scratch Projects*. In *Proceeding of the 44th ACM technical symposium on Computer science education* (SIGCSE '13). pp. 215-220.

Crook, S. (2010). *Embedding Scratch in the Classroom*. International Journal of Learning and Media, 1(4). pp. 17–21.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. Annual American Educational Research Association Meeting. pp. 1–25.

Franklin, D., Hill, C., Dwyer, H. A., Hansen, A. K., Iveland, A., & Harlow, D. B. (2016). Initialization in Scratch: Seeking Knowledge Transfer. SIGCSE ’16. pp. 217–222.

Hermans, F., & Aivaloglou, E. (2016). *Do code smells hamper novice programming? A controlled experiment on Scratch programs*. IEEE International Conference on Program Comprehension, 2016. pp. 1–10.

Kafai, Y. (1995). *Minds in play*. New Jersey: Lawrence Erlbaum.

Lee, O., & Lehrer, R. (1988). *Conjectures concerning the origins of misconceptions in LOGO.* Journal of Educational Computing Research, 4(1), pp. 87-105.

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). *Programming by choice: urban youth learning programming with Scratch*. ACM SIGCSE Bulletin, 40(1), pp. 367-371.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). *The Scratch Programming Language and Environment.* ACM Transactions on Computing Education, 10(4), pp. 1–15.

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2011). *Habits of programming in Scratch.* In Proceedings of the 16th annual joint conference on Innovation and technology in computer science education. pp. 168-172.

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). *Learning computer science concepts with Scratch*. Computer Science Education, 23, pp. 239–264.

Moreno-León, J., & Robles, G. (2015). *Analyze your Scratch projects with Dr. Scratch and assess your computational thinking skills*. In 2015 *Scratch conference*. pp. 12-15.

Netzer, R. H., & Miller, B. P. (1992). *What are race conditions? Some issues and formalizations*. *ACM Letters on Programming Languages and Systems (LOPLAS) 1*(1). pp. 74-88.

Papert, S. (1980). Mindstorms: Mindstorms: Children, Computers, and Powerful Ideas. Basic books: NY.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. B. (2009). *Scratch: Programming for all*. *Commun. Acm*, *52*(11), pp. 60-67.

Resnick, M., & Rosenbaum, E. (2013). Designing for tinkerability. *Design, make, play: Growing the next generation of STEM innovators*, pp. 163-181.

Vargas-Alba, Á., Troiano, G.M., Chen, Q., Harteveld, C., & Robles, G. (2019). *Bad Smells in Scratch Projects: A Preliminary Analysis.* In Proceedings of the 2nd Systems of Assessments for Computational Thinking Learning Workshop (TACKLE@EC-TEL 2019).

Children construct the Cowshed 4.0

Kerstin Strecker, *kerstin.strecker@informatik.uni-goettingen.de*
University of Göttingen, Institute of Computer Science, Goldschmidtstr. 7, 37077 Göttingen, Germany

Abstract

In this article, a concrete learning environment is used to suggest how children of primary school age can get started with algorithms. What is especially important here is that the children can find problems autonomously and solve them with the help of entry algorithms. They invent their own solutions for the problems in this context, which enables them to experience themselves as self-effective in the field of computer science.

Figure 1.: Cowshed 4.0

Keywords

physical computing, primary school, algorithms, self-efficacy, context reference, open tasks

1 The learning environment

Our miniworld is a modern automated cowshed. It consists of LEGO and LEGO Duplo parts, which can be used easily together with the LEGO WeDo technology. Possible problems and the program code of corresponding solutions can be found in chapter 5 of this paper.

In particular, the miniworld contains the following possible stations:

- a big brush that massages and cleans the cows,
- a hatch with which fresh straw can be distributed,
- a conveyor belt that transports new feed into the barn,
- a milking parlor with two barriers which must be programmed individually,
- a lighting system that switches on when there is movement,
- ...

Herper and Röhming show in (Herper, Röhming 2018) with similarly complex programming examples using LEGO-WeDo that children in grades 3 and 4 can write and build the programs possible for this cowshed and the necessary constructions themselves. Also, the work of Hoffmann et. al. (Hoffmann et.al. 2017) supports this observation.

The advantages of "physical computing" as a teaching principle are described in (Przybylla, Romeike 2017), Marco Thomas also refers to primary schools in (Thomas 2018).

2 Intentions

Nadine Bergner et. al. sum up in their study that the children see the computer essentially as a 'toy'. (Bergner et.al. 2017). In our perspective, it is important that children are not just users of software or digital media. In contrast we think that the mere use of media will create a false image of what working with a computer and computer science is. This might especially be true for young children that have no further experience with computers. The learning programs that are widely used in schools, such as vocabulary trainers or mathematical exercise programs, can easily create the image of an "intelligent" machine that knows and can do everything. Pupils can try to solve the problems and are corrected by the machine. In the best case, they can do what the machine can do, otherwise they can do less.

In its strategy paper "Education in the digital world" (Kultusministerkonferenz 2016) for general schools in the competence catalogue 5.5 (Recognising and formulating algorithms), the Standing Conference of the Ministers of Education and Cultural Affairs of the Countries in the Federal Republic of Germany (Kultusministerkonferenz) therefore rightly demands that learners plan and use a structured, algorithmic sequence to solve a problem, i.e. also develop their own algorithms. *Algorithms and computer systems usually have to be formulated very precisely and formally so that a machine can interpret them [...]* (translated from Thomas 2018). In our miniworld, the cowshed, it is new for children to have to adhere strictly to the sequence of actions to determine the parameter values, to identify an event, ... to proceed algorithmically. The children become constructors of software instead of just users. The experience of having to enter the commands into the computer in such detail creates the image of a machine that only executes the children's ideas strictly according to the instructions given. The ideas themselves, however, are in the hands of the children. They are the inventors who use a machine to execute their ideas, which needs to be clearly told what to do.

In order to create a positive and sustainable self-image for the inventor working with the computer, it is not enough to focus solely on the contents of the programming, even if the tool is chosen to be suitable for children. We find support for this argumentation in (Bergner et.al. 2017b), where it says in another context: *Overall, the quantitative evaluation [...] has shown that [...] (the) introduction to graphic programming by means of scratch and the App-Inventor does not* *necessarily contribute to a more positive image of computer science [...]* (translated from Bergner et.al. 2017b).

Additionally, to the construction of software itself, the children have to work in a positive context and experience themselves as self-effective to change their view of how to work with computers.

We will go into this in more detail in the next chapter.

3 Special features of the learning environment

The work of Silke Ladel (Ladel 2018) shows that the context reference in programming is especially important for primary school children. It shows (with examples such as "Ozo in the Magic Forest" and "The Adventures of Harry Potter") how important it is that there is a story around programming and that children can identify with it. Animals have a strong attraction for children, and we believe that cows, unlike horses for example, are interesting for boys and girls alike. Our idea is to instruct the children to create a cowshed in which the cows should lead a "cow worthy" existence. Information technology, with which the miniworld is automated, is helping to achieve this. It is therefore positive for the cows. Thanks to physical computing, the cows can move freely, milk themselves or feed themselves. They can be cleaned if necessary, etc. The owner of the animals also finds the automation to make his work easier, so it helps everyone. The motivation here is to make life beautiful for the animals and easier for the owner. Besides, this corresponds to the "multi-perspectivity" of subject teaching described in (Murmann 2018), in which, for example, *the knowledge of living conditions of selected animals and plants as a basis for appropriate husbandry and care* (translated from Kerncurriculum Sachunterricht Grundschule Niedersachsen 2017) is to be conveyed in the core curriculum of Lower Saxony. In the same curriculum, in the section *Discussion of technical inventions*, it is stated that children reinvent simple technical inventions, build and evaluate this invention and analyse its consequences for everyday life and the environment (Kerncurriculum Sachunterricht Grundschule Niedersachsen 2017).

But the approach is not just about exploring and trying things out in a playful way. Even if smaller children often recreate everyday situations, they also construct something new in their play. In elementary school, children learn how to deal with letters, words and numbers, they learn how to approach the world around them, and they become more and more aware of it. To put it simple, they are explained parts of the world in school lessons. Here, computer science as a constructive science can extend and enrich the teaching of primary school.

In our learning environment the children should not only develop the algorithms independently, they should also find the problems to be solved by themselves. However, this must lead to success in order to strengthen the student's self-esteem. To ensure this, we have created the cowshed in a way that will lead to success in any case. All stations are equipped with a fixed actuator (here: motor, lamp) which can be switched on and off directly. This supports and directs the brainstorming process. With just one command, every learner can create a finished, functioning product. All stations can also be extended with a sensor and linked to the actuators. Thus, the context and the hardware limit the ideas of the children to realizable ideas with which they are successful.

Together, the miniworld and the tool thus provide stories and references to find even simple, successfully solvable tasks. The “product pride” is strengthened, since the invention is regarded as helpful for all participants by the children.

Dorothee Müller (Müller 2018) writes in this context: *Since young people come into contact with computer science systems at an early age - generally before the first computer science lesson - whose basic functions and principles they do not know, they usually only have the ability to use them, cannot cope with new or difficult situations and thus do not develop a positive self-efficacy conviction with regard to computer science. Computer science lessons at primary school must enable pupils to successfully deal with problems by means of computer science*. (translated from Müller 2018).

With our learning environment the pupils can construct the cowshed 4.0 according to their own ideas. They invent the cowshed 4.0. They do not only learn to understand objects and connections, they construct them themselves and they do this successfully in any case, independently from their prior knowledge of computing.

4 Conclusion

The cowshed 4.0 is an example that suggests how to get started with algorithms in primary school. Together with the possibility to achieve something positive, the child's needs are met. Because the problems can be solved by controlling only one actuator, all children are successful. Nevertheless, nobody is underchallenged as there is the possibility of adding sensors and tasks and everyone can be proud of their product in the end. The children experience themselves as self-effective in the field of computer science and learn how to use algorithms to fulfil a selfselected task.

5 Programming examples

Figure 2: Cow with brush

The brush can be programmed either very easily or in combination with a distance sensor (the brush only rotates when a cow nears itselfs). Extensions: Brush stops when the cow leaves; brush stops after a certain time; a lever (tilt sensor) can stop the brush completely or change the speed, ...

Figure 3: Lighting system

In the simplest case, the light in the stable is only switched on, or it turns on when a cow passes by. Extension: it is switched off again after a certain time; it is switched off when the cow has passed, ...

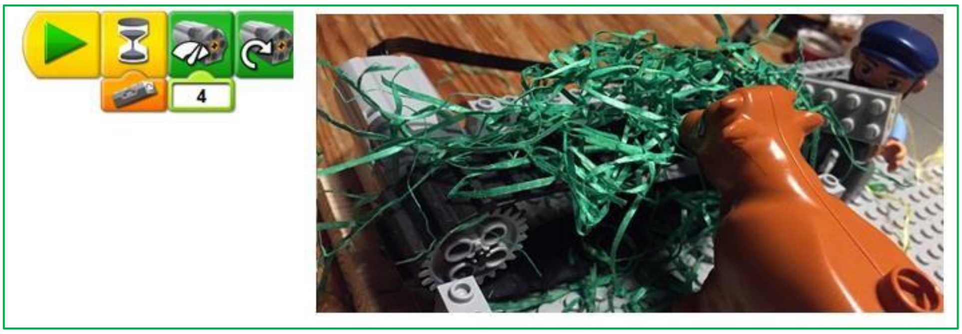


Figure 4: Conveyor belt with feed

The distance sensor can be used at many stations. However, a better monitoring for the farmer is possible, if here (as with the feed) the automatism can be triggered "manually" with the help of a switch (inclination sensor).

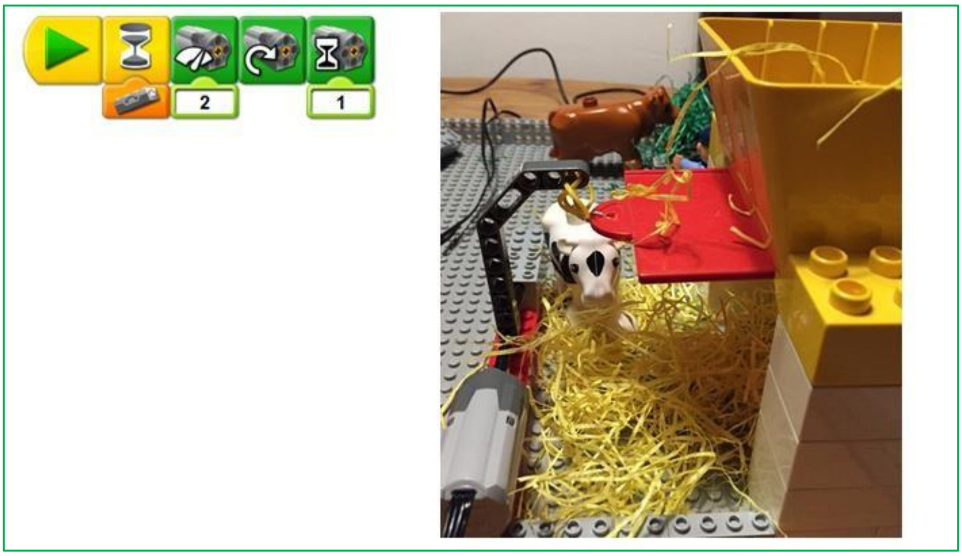


Figure 5: Straw hatch

For the straw hatch and also for the barrier, the combination of motor power and duration must be determined experimentally. For this purpose, it makes sense to remove the modules from the miniworld and test them without the risk of damaging the whole construction.

Figure 6: Barrier milking parlor

The barrier is similar to the straw hatch. At some stations, the complexity can be increased by the fact that the technical system runs continuously, which makes the use of a loop necessary. In this example, the barrier opens when the cow approaches and closes again after a certain time. After that, the program does not end, but the barrier remains ready for the next cow.

References

Bergner, N., Petrut, S., Schroeder, U. (2017): *“Was Grundschulkinder über Informatik wissen und was sie wissen wollen”*, in: Ira Diethelm (Hrsg.): „Informatische Bildung zum Verstehen und Gestalten der digitalen Welt“, GI-Fachtagung Informatik und Schule (INFOS), GI Lecture Notes in Informatics, 2017

Bergner, N., Leonhardt, T., Schroeder, U. (2017b): “*Fünf Argumente für einen grafischen Programmiereinstieg*”, in: Ira Diethelm (Hrsg.): „Informatische Bildung zum Verstehen und Gestalten der digitalen Welt“, GI-Fachtagung Informatik und Schule (INFOS), GI Lecture Notes in Informatics, 2017.

Hoffmann, S., Wendtland, K., Wendtland, M. (2017): *“Algorithmisieren im Grundschulalter*“, in: Ira Diethelm (Hrsg.): „Informatische Bildung zum Verstehen und Gestalten der digitalen Welt“, GIFachtagung Informatik und Schule (INFOS), GI Lecture Notes in Informatics, 2017.

Herper, H., Röhming, M. (2018): “A*ußerschulischer Kompetenzerwerb*“, Zeitschrift LOG IN Heft Nr. 189/190, LOG IN-Verlag, 2018

Kultusministerkonferenz (2016): https://www.kmk.org/fileadmin/Dateien/veroeffentlichungen_beschluesse/2016/2016_12_ 08-Bildung-in-der-digitalen-Welt.pdf, last visited: 13.10.19

Ladel, S. (2018): *S“pielerische Lernumgebungen mit Robotern*“, Zeitschrift LOG IN Heft Nr. 189/190, LOG IN-Verlag, 2018

Müller, D. (2018): “*Informatik? – Das kann ich (lernen)!*“, Zeitschrift LOG IN Heft Nr. 189/190, LOG IN-Verlag, 2018

Murmann, L. (2018): “*Zur Sache Informatik*“, Zeitschrift LOG IN Heft Nr. 189/190, LOG IN-Verlag, 2018

Kerncurriculum Sachunterricht Grundschule Niedersachsen (2017): http://db2.nibis.de/1db/cuvo/datei/kc_su_n-line.pdf, last visited: 13.10.19

Przybylla, M., Romeike, R. (2017): “V*on Eingebetteten Systemen zu Physical Computing: Grundlagen für Informatikunterricht in der digitalen Welt*“, in: Ira Diethelm (Hrsg.): “Informatische Bildung zum Verstehen und Gestalten der digitalen Welt“, GI-Fachtagung Informatik und Schule (INFOS), GI Lecture Notes in Informatics, 2017.

Thomas, M. (2018): “I*nformatische Kompetenz in der Grundschule*“, Zeitschrift LOG IN Heft Nr. 189/190, LOG IN-Verlag, 2018

Code-first learning environments for science education: a design experiment on kinetic molecular theory

Umit Aslan, *umitaslan@u.northwestern.edu*
Learning Sciences, Northwestern University, Evanston, IL, USA

Nicholas LaGrassa, *nick.lagrassa@u.northwestern.edu*
Learning Sciences and Computer Science, Northwestern University, Evanston, IL, USA

Michael Horn, *michael-horn@northwestern.edu*
Learning Sciences and Computer Science, Northwestern University, Evanston, IL, USA

Uri Wilensky, *uri@northwestern.edu*
Learning Sciences and Computer Science, Northwestern University, Evanston, IL, USA

Abstract

Code-first learning entails the use of computer code to learn a concept, and creating computational models is one such effective method for learning about scientific phenomena. Many code-first learning approaches employ the visual block-based programming paradigm in order to be accessible to school children with no prior programming experience, providing them with high-level domain-specific code-blocks that encapsulate the underlying complex programming logic. However, even with the aid of visual clues and the benefit of simpler primitives like “forward” and “repeat,” many phenomena studied in classrooms such as the behavior of gas particles in Kinetic Molecular Theory (KMT) are challenging to describe in code. We hypothesized that code blocks designed from a phenomenological perspective to model the behavior of familiar objects and events would both promote students’ authoring of computational models and their ability to encode and test their beliefs within their models. We created these phenomenological blocks within a code-first gas particle sandbox and integrated it into a KMT lesson plan. Two high school teachers taught this curriculum to 121 students, from which we gathered and analyzed video footage from lesson activities and student focus groups. We found that the phenomenological blocks gave students the ability to start programming right away and to express their intuitive understanding of KMT through computational models. This exploratory study demonstrates the potential for phenomenological programming to broaden the application and accessibility of code-first computational modeling for learning scientific phenomena.

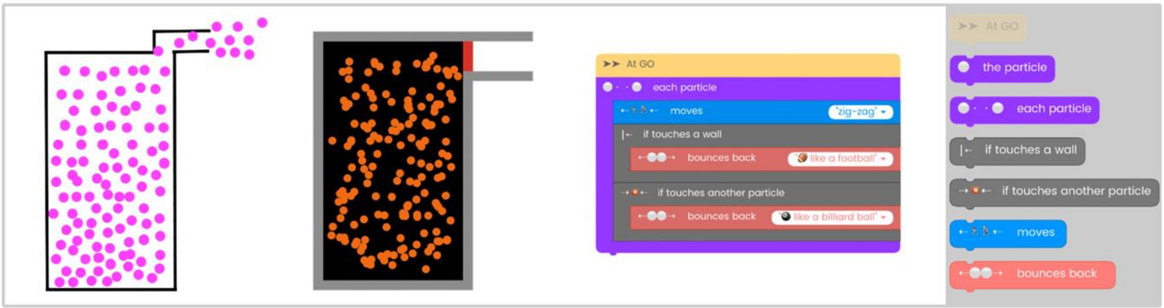


Figure 1. The code-first gas particle sandbox with phenomenological blocks

Keywords

Constructionism, code-first learning environments, agent-based modeling, blocks-based programming, chemistry education, computational thinking

Introduction

Initiatives for integrating computational thinking (CT) into science and mathematics curricula has gained significant momentum in recent years (e.g., Grover & Pea, 2013; Sengupta et al., 2013; Weintrop et al., 2015; Wing, 2006). It is argued that embedding computation across STEM curricula would align science education with contemporary scientific practices, deepen student learning, and promote computational literacy (Wilensky et al., 2014). Theoretical work such as Weintrop et al.'s (2015) CT-STEM taxonomy provide actionable frameworks, but there is also need for significant design innovation in terms of tools and curricular activities. Code-first learning environments aim to promote CT in science education by making the construction of computational models accessible to non-programmer students (Wilkerson-Jerde, 2010; Horn et al., 2014). They achieve this by providing students with domain-specific programming primitives that abstract away the underlying programming logic and formal expressions. Coding their own models affords students the ability to express intuitive ideas about scientific phenomena through a formal computational representation and to "debug" their own thinking along the way (Papert, 1980). Research on code-first learning environments is still in its infancy, but it is accelerating thanks to a recent focus on bringing CT practices into the STEM classroom (e.g., Wilensky, Brady & Horn, 2014; Weintrop et al., 2015), infrastructures such Behaviour Composer (Kahn, 2007), DeltaTick (Wilkerson & Wilensky, 2010), NetTango (Horn & Wilensky, 2012), and early curricular designs such as the Frog Pond (Horn et al., 2014; Guo et al., 2016) and EvoBuild (Wagh & Wilensky, 2017).

In this paper, we present the first iteration of a design-based research experiment (Cobb et al., 2003) in which we followed the examples of Frog Pond and EvoBuild to create a code-first learning environment for the kinetic molecular theory (KMT) as part of a new high school agent-based chemistry unit, adapted from the NetLogo Connected Chemistry unit (Stieff & Wilensky, 2003; Levy & Wilensky, 2007). The original Connected Chemistry (CC'1) unit guided students in model-based inquiry wherein they explored the behavior of different model scenarios. The unit was proved effective in that students made gains on AAAS assessments, and also were able to connect the micro-level physical interactions with the macro-level phenomena (Levy & Wilensky, 2007; 2009). We used Weintrop et al.'s (2015) CT-STEM taxonomy as our unit's design framework. The most challenging aspect of operationalizing the CT-STEM taxonomy was to address the "computational problem solving" category, which included practices such as "troubleshooting and debugging," "programming," and "creating computational abstractions."

We hypothesized that creating a new introductory lesson in which the students were introduced to KMT through a programming activity would be a valuable outcome because it would allow students to express their prior understanding about the behavior of gas particles at the microscopic level by constructing an agent-based model with NetLogo (Wilensky, 1999a). KMT was an important topic for us because it is taught universally, yet research shows that students have difficulty in making sense of the variety of phenomena exhibited. Moreover, students come to the classroom with a number of incorrect conceptions that stay intact despite formal instruction (Lin & Cheng, 2000). Smith et al. (1994) argue that it is essential to bridge students' prior conceptions with formal scientific explanations to facilitate meaningful and robust learning (Smith et al., 1994). Otherwise, students will leave the classroom perhaps able to answer test questions according to the formal scientific explanation, but they will keep relying on their intuitions when making sense of real-world phenomena (diSessa, 1993; 2015). However, it is also not easy to design a code-first learning environment that would enable students to "teach computers how they think" about KMT because designing custom code-blocks for this topic is challenging especially when it comes to particle-particle elastic collisions. Such calculations require command of vector mathematics. Computationally, one needs to know concepts such as variables and collision detection. These skills are typically not expected from learners with minimal-to-no programming experience.

In this paper, we present the preliminary results of a design experiment on developing a code-first gas particle sandbox and supporting learning activities for KMT which culminated in a new blocks-based programming paradigm that we call "phenomenological programming." We designed higher-level code blocks not as procedural commands, but as phenomenological statements that leverage students' intuitive knowledge of real-world events, objects, and patterns. For example, we designed a "bounce" block that can be modified with phenomenological statements such as "like a football" or "like a billiard ball." A particle that bounced *like a football* lost kinetic energy on impact and changed direction randomly, while one that bounced *like a billiard ball* conserved momentum and bounced back at a reflective angle. In what follows, we describe our design experiment in detail, including the preliminary results of a research implementation in which two high school teachers taught KMT to 121 high school students using the code-first gas particle sandbox. We begin with the theoretical underpinnings of our study. We then present the design of our learning activities and the idea of phenomenological programming in detail. Lastly, we describe our first classroom implementation and present vignettes from our data that show how the students engaged with phenomenological programming.

Theoretical framework

Constructionism

Our research is situated within the greater constructionist learning paradigm which maintains that learners construct and learn strong mental models when they engage in constructing personally meaningful, public entities (Papert, 1980; 1991). While constructionist literature is rich with many branches of study, computers tend to play a significant role because they afford learners the opportunity to construct a wide range of dynamic models that can be easily inspected, manipulated, and debugged. Papert and colleagues' design of the Logo programming language (Papert, 1971; 1980) and development environment is the most influential example of a constructionist, code-first learning environment. The primary way to interact with Logo is by programming. The primitives of the language (i.e., commands, branching statements) are designed to be easy to learn by children as young as grade school level. Studies show that Logo promotes powerful learning, especially in mathematics and geometry (e.g., Harel & Papert, 1990; Hoyles & Noss, 1992). Our approach takes inspiration from four constructionist ideas: syntonic learning (Papert, 1980), embodied modeling (Wilensky & Reisman, 2006), code-first learning environments (Horn et al., 2014; Kahn, 2007; Wilkerson & Wilensky, 2010; Wilkerson-Jerde et al. 2015), and blocks-based programming (Bagel, 1996; Bau & Bau, 2015; Resnick et al., 2009; Weintrop & Wilensky, 2015).

Syntonic learning refers to Papert's design of the original Logo turtle, which was *body-syntonic* and *ego-syntonic* (1980). The turtle was controlled by primitives such as *forward* and *right* that relate to children's sense of their own bodily interactions with the physical world. It was also designed to be coherent with children's sense of themselves as people with intentions, goals, and desires. Topics that are counterintuitive when taught formally, such as the definition of a "circle," can be expressed in Logo in a way that is *natural* for children and grounded in their embodied schema (Fig. 2).

Embodied modeling research shows that when learners 'knowledge of individual objects is aligned with their embodied ways of thinking and their own point of view, "they are enabled to think like a wolf, a sheep, or a firefly" (Wilensky & Reisman, 2006). In embodied modeling, learners put themselves in the place of the agents that make up complex systems. For example, when a learner constructs a model of ideal gas laws, they do not solve aggregate-level algebraic equations that are disassociated from the actual underlying real-world events. Instead, they define how each particle behaves autonomously. They take the perspective of a particle and reason that "I would move forward on a straight path," "If I hit a wall, I would bounce back with a straight angle," and so on. Converting embodied agent-rules to a computer simulation using a constructionist agent-based modeling environment such as NetLogo (Wilensky, 1999a) affords learners to see what happens when the same rules are followed by many agents simultaneously. This enables them to connect how micro-level events lead to the emergence of macro-level patterns, properties, and phenomena (Wilensky, 2001). Moreover, they can modify their models easily and test various alternative scenarios, deepening their understanding of scientific phenomena along the way.

“A circle is a plane figure contained by one line such that all the straight lines falling upon it from one point among those lying within the figure equal one another.”	A circle with center (a, b) and radius r is the set of all points (x, y) such that
	
	$(x - a)2 + (y - b)2 = r2$
		`var c = document.getElementById("myCanvas")`
		`var ctx = c.getContext("2d");`
		`ctx.beginPath();`
		`ctx.arc(100, 75, 50, 0, 2 * Math.PI);`
		`ctx.stroke();`
			`TO circle`
			`  repeat 360 [`
			`    forward 1`
			`    right 1`
			`  ]`
			`END`

(a) The Euclidian definition of a circle	(b) The algebraic definition of a circle	(c) With the JavaScript programming language	(a) With the Logo programming language

Figure 2: Comparing formal definitions a circle with the Logo way

Code-first learning environments are constructionist learning environments that make embodied modeling accessible to students with no prior programming experience (Horn et al., 2014; Kahn, 2007; Wilkerson-Jerde et al. 2015). Research shows that well-designed code-first learning environments promote powerful learning by exposing underlying mechanisms of scientific phenomena better than interacting with pre-existing models or simulations (Guo et al., 2016; Wagh et al., 2016). They achieve this goal by providing students easy-to-learn visual programming environments with pre-composed higher-level primitives that abstract away the underlying complex programming logic. For example, the Modeling4All environment comes with an extensive library of small, independent program fragments called *micro-behaviors* that translate into NetLogo code. In DeltaTick (Wilkerson-Jerde & Wilensky, 2010) and NetTango (Horn & Wilensky, 2012), on the other hand, custom modeling primitive libraries are designed for specific topics or phenomena. For example, the Frog Pond code-first learning environment for natural selection has code-blocks such as "*chirp*", "*hop*", "*hunt*" and "*hatch*" to model the behavior of colorful virtual frogs on a virtual lily pad. This allows students to quickly learn programming and create short programs that result in population-level evolutionary outcomes.

Like many code-first learning environments, we employ the blocks-based programming paradigm. Initially developed by Begel (1996) for the Logo Blocks project in 1996, blocks-based programming is a visual paradigm that represents programming constructs (e.g., commands, branching statements, etc.) as visual blocks that resemble physical Lego blocks. Users assemble algorithms by dragging blocks into a code area and attaching them to each other. Blocks-based programming offers some significant advantages for novice programmers compared to traditional text-based programming languages. For example, it is not possible for students to get side-tracked by syntax errors because the user does not type anything. In addition, there is no need to memorize exact commands because they are always present in the blocks library. Many blocks-based languages such as Scratch and PencilCode even implement visual cues to improve their usability, such as assigning categories of blocks to the same color. In PencilCode, motion blocks (e.g., forward, speed) are blue and control flow blocks (e.g., if-else, key-down) (Bau & Bau, 2016).

Connected Chemistry

Our design of the code-first gas particle sandbox also builds on three decades of studies conducted in CCL beginning with Wilensky’s “gas-in-a-box” studies (1999b; 2003) and the subsequent curricular units created for high school chemistry (Stieff & Wilensky, 2003; Levy & Wilensky, 2007;2009; Levy, Novak & Wilensky, 2006; Brady et al., 2014). The study of gaseous matter is particularly suitable for computational modeling because macro-level properties (e.g., temperature, pressure), as well as the scientific laws that describe the relationship between them, are shown to be challenging topics that lead to a multitude of robust misconceptions (Kind, 2004; Lin & Cheng, 2000; Nakhleh, 1992). At school, the study of these relationships often encompasses memorization of equations such as PV = nRT. In contrast, the CCL-developed constructionist units frame these topics in terms of micro-level particle interactions that lead to the emergence of the macro level patterns. Students are guided through computational explorations with agent-based models instead of equation-based problem-solving activities. These units are still actively used by many high school teachers and previous research implementations showed significant learning gains. For example, in one such unit titled Connected Chemistry 1 (CC1; Levy & Wilensky, 2009), students ran NetLogo models to examine the relationship between gas particle behaviors and key variables like the temperature and pressure. They conducted computational experiments and used statistical methods to derive their own versions of the ideal gas laws. Interviews before and after the intervention showed that the students were able to form multi-level explanations of the chemical system at the end of the unit. In pre-interviews, 84% of the students explained gaseous phenomena only using macroscopic terms. In post-interviews, 85% of them explained the same phenomena in terms of the connections between micro-level interactions and the macro-level patterns (Levy et al., 2004; Levy & Wilensky, 2009).

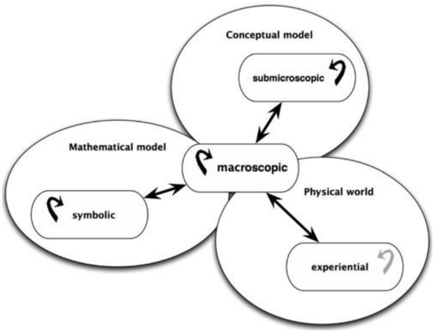


Figure 3: Conceptual framework of the Connected Chemistry 1 (CC1) curriculum (Levy & Wilensky, 2009).

Our code-first gas particle sandbox is inspired by two of these prior units developed at CCL: Levy & Wilensky's Connected Chemistry 1 (CC1) unit and Brady et al.'s (2014) Particulate Nature of Matter (PNoM) unit, part of the ModelSim project (NSF# DRL-1020101). The CC1 unit introduced the idea of a particle sandbox, while the PNoM unit built on CC1 and further developed it to provide students with emergent systems sandboxes (ESSs) (Brady et al., 2014) within which they could construct computational models of dynamic systems that exhibit emergent phenomena without the need for writing any code. For example, in the PNoM unit, students constructed computational models to explore the diffusion of odor in a room when a warm container is opened versus when a cold container is opened. In the sandbox, the students could use a drawing tool to add static walls to represent containers, removable walls to represent valves or doors, and particles that were pre-programmed to move and interact according to kinetic molecular theory (KMT).

Design overview

The design experiment we present here is situated within a greater project to design a new version of the Connected Chemistry Ideal Gas Laws unit, which we call the CC'19 unit (see Aslan et al., 2020a), itself part of the CT-STEM research project (Weintrop et al., 2015; Wilensky et al., 2014). The idea of the code-first gas particle sandbox first emerged as we were designing a brand-new introductory lesson for the CC'19 unit. We had three overarching objectives: a pedagogical objective, a computational objective, and a content-learning objective. Pedagogically, we wanted to bootstrap the rich ideas that students bring into the classroom prior to instruction (diSessa & Minstrell, 1998; Smith et al., 1994). Computationally, we wanted students to be able to express their intuitive understanding of gas particles in terms of simple computer programs. In order to promote chemistry learning, we wanted these activities to build towards the main assumptions of the Kinetic Molecular Theory (KMT) because we wanted students to be able to explain how gas pressure, a macro-level property, emerges from numerous gas particles interaction with each other and the container.

To achieve our pedagogical objective, we designed a beginning activity in which the students examined an air duster can. We chose the air duster because it is a simple real-world object that has a fixed volume and only gas particles inside. The students answered some beginning questions about what happens when the valve is pressed, and they also illustrated their answers by drawing sketches. The teachers projected each student's sketch on the screen and conducted whole-class discussions. These discussions served as benchmark classes (diSessa & Minstrell, 1998) for teachers to survey the students 'intuitive understanding of gas particles and cultivate an exchange of these ideas among students. The discussions also served as an anchor for the idea of computational modeling because the students 'sketches served as static models of the air duster can.

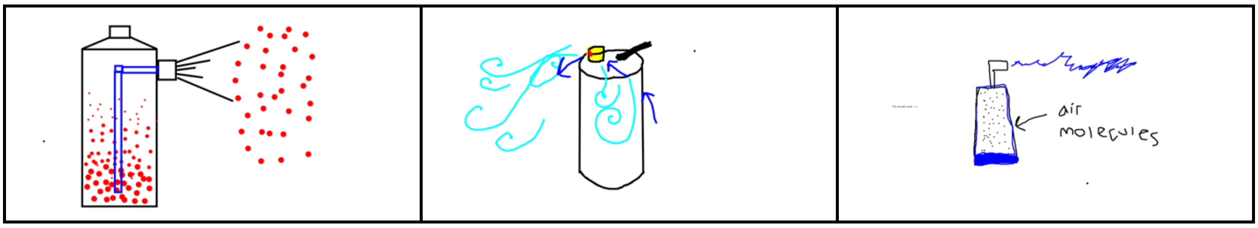
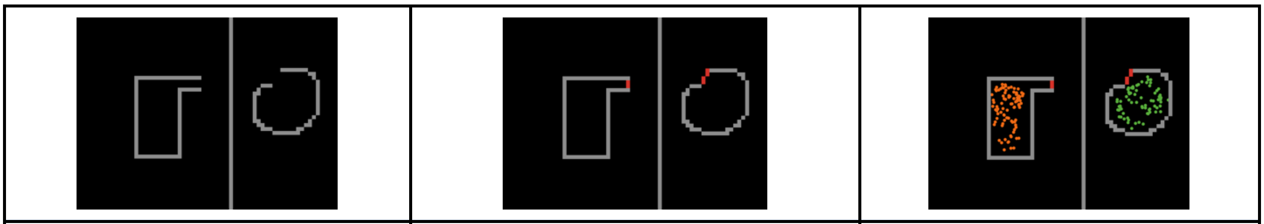


Figure 4: Examples from students' hand-drawn sketches

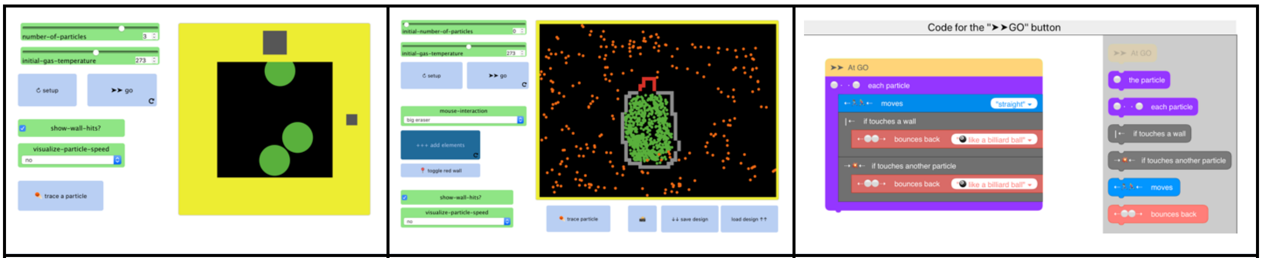
To achieve our CT objective, we designed a multi-step scaffolded activity. First, the students used a static modeling toolkit that resembled the initial sketching activity. They constructed the initial state of a computer model by adding stationary walls, removable walls, green particles, and orange particles. Second, they used a simplified version of the code-first gas particle sandbox to develop a small-scale model of gas particles (max. 4 particles). Lastly, they re-loaded their static air duster models from the first step into the sandbox to see whether their air-duster model behaved as they anticipated. This process allowed them to design and construct a computational model to test their initial hypotheses.



(a) drawing static walls	(b) adding removable walls	(c) adding particles

Figure 5. Sketching a static computational representation of real-world gas containers (step 1).

Designing domain-specific primitives for KMT was a challenging task because we assumed no prior programming experience. A traditional approach would require students to use computational constructs such as variables, vector calculations, and collision detection. Instead, we formulated a new approach that we call *phenomenological programming* (Aslan et al., 2020b). Beside building on the four constructionist ideas that we discussed in the theoretical framework section, phenomenological programming is also partially inspired by diSessa's theory of phenomenological primitives (p-prims in short). By definition, p-prims are "bits of knowledge that contribute to our intuitive 'sense of mechanism;' that is, what kinds of occurrences are natural and to be expected" (diSessa, 1993; diSessa, 2015, p. 34). They are phenomenological because they are encoded non-verbally, probably as images or kinesthetic schemes. The activation of p-prims is instantaneous. They are evident in our daily experience and we see situations in terms of them. They are primitive because we often cannot analyze or justify our p-prims. We hypothesized that code-blocks designed in accordance with students 'p-prims would (1) be easily recognizable for the students, (2) embed implicit assumptions about their function, (3) facilitate easy mental simulation and hence help students express their mental models computationally, and (4) most importantly bridge students 'prior understanding with the formal scientific explorations by facilitating a process of "debugging one's thinking". Furthermore, we hypothesized that code-first learning environments with phenomenological programming could be more approachable for teachers as well because they would not require a lot of teacher training and they would appeal to teachers with little-to-no prior computing experience.



(a) the micro-level sandbox interface (step 2)	(a) the macro-level sandbox interface (step 3)	(b) blocks-based coding view

Figure 6. The main components of the code-first gas particle sandbox.

We designed phenomenological blocks that provide students with procedural templates such as *moves* and *bounces* that can be modified with phenomenologically transparent statements such as "*spinning*" and "*straight*" for the "*move"* block, and "*like a balloon" and "like a billiard ball"* for the "*bounce"* block. Each statement embeds simple assumptions about gas particles. Some of them embed the assumptions of KMT (e.g., move straight, bounce like a billiard ball), while others correspond to potential misconceptions. For example, KMT assumes that particles move straight until they collide another particle or hit a wall. However, we also designed a "*move erratically*" option because research shows that some students might believe that gas particles change direction randomly and haphazardly without any collision (Kind, 2004). This way, students can quickly start programming the particles minimal introduction to programming and without the challenging task of converting their intuitive understanding of gas-particles to formal computer-code. The blocks would be instantly recognizable for them and they can hypothesize about the outcome of the code they put together because it would be easy to mentally simulate the movement of gas particles.

A summary of each code-block we designed for the code-first gas particle sandbox is presented in Table 1. There are only 7 code-blocks in this iteration of the code-first gas particle sandbox (Figure 7). However, combined with the static freehand modeling tools (Figure 5) and the phenomenological sub-statements, these blocks are enough to develop very complex and detailed gas particle simulations. Moreover, they are sufficient to create conflicts between students' intuitive understanding of micro-level gas particle behavior and macro level patterns. For example, if a student chose to make the particles move in circles (i.e. spinning) and collide elastically, it would still generate close-to-expected behavior at the micro-level, yet when tested with an air-duster design at the macro level, the particles would not leave the valve as intended. Similarly, particles that bounce like basketballs may slow down so gradually that students who do not run their micro-level models long enough may not notice the difference initially, but they would notice it when they run their models at macro level with hundreds of particles. Prior research shows that such conflicts can be great opportunities for students to debug their intuitive understanding of gas particles' behavior as well as bridge their intuitive knowledge with the formal scientific explanations (Wilensky, 1999b; 2003; Levy & Wilensky, 2009).

Block	**Explanation**
At GO	Procedural block. Code that is attached to this block is executed in a continuous loop when the GO button of the model is clicked.
the particle 1	Procedural block. The code encapsulated by this block is executed only by the selected particle (e.g., particle 1, particle 10, particle 87).
each particle	Procedural block. The code encapsulated by this block is executed by all particles separately and autonomously.
if touches a wall	Procedural block. The code encapsulated by this block is only executed when a particle is touching a container wall.
if touches another particle	Procedural block. The code encapsulated by this block is only executed when a particle is touching another particle.
moves "straight" moves "spinning" moves "zig-zag" moves "erratic"	Phenomenological block. If a particle is executing this code, it moves 1 unit forward based on the chosen phenomenological statement: Straight: Moves forward 1 unit without changing direction. Spinning: Moves forward 1 unit, changes direction to follow a circular path. Zig-zag: Moves forward 1 unit, changes direction to follow a zig-zag path. Erratic: Moves forward 1 unit, changes direction to follow a path that resembles random walk.
bounces back "like a balloon" bounces back "like a football" bounces back "like a billiard ball" bounces back "like a basketball"	Phenomenological block. If a particle is executing this code, it changes its momentum and kinetic energy based on the chosen phenomenological statement: Like a balloon: Changes direction as if it is an elastic collision. If collides with another particle, exchanges momentum as if it is an elastic collision. Total kinetic energy is decreased significantly. Recalculates its speed based on its kinetic energy. Like a football: Changes direction randomly. If collides with another particle, exchanges momentum as if it is an elastic collision. Total kinetic energy is decreased slightly. Recalculates its speed based on its kinetic energy. Like a billiard ball: Changes direction as if it is an elastic collision. If collides with another particle, exchanges momentum as if it is an elastic collision. Total kinetic energy is preserved. Recalculates its speed based on its kinetic energy. Like a basketball: Changes direction as if it is an elastic collision. If collides with another particle, exchanges momentum as if it is an elastic collision. Total kinetic energy is decreased slightly. Recalculates its speed based on its kinetic energy.

Figure 7. The function of the code-blocks of the code-first gas particle sandbox and the assumptions embedded in the phenomenological blocks

Research implementation

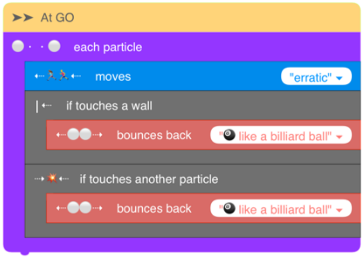
4.1. Participants & Settings: Our study is a design experiment as outlined by Cobb et al. (2003). We designed our code-first gas particle sandbox over multiple iterations. The initial versions were tested by our research team members. Then we met with the two teachers who were going to implement the lesson and updated our design according to their feedback. The first research implementation of our design experiment took place in Spring 2019. Two teachers and a total of 121 high-school regular chemistry students at a U.S. Midwest public high school participated (Table 2). The implementation lasted a total of 10 class periods over the course of 8 days. The students used Chromebooks to access the lesson over the CT-STEM student portal. The final version of this lesson can be accessed through the CT-STEM webpage (https://ct-stem.northwestern.edu/curriculum/preview/513/0/).

4.2. Data collection and analysis: We collected all the open-ended written responses and sketches students posted on the portal. In addition, we asked students to upload screenshots of their static container models and blocks-based algorithms. In order to gain further insight on the students ' thought processes and the interactions between them, we video recorded four focus groups each containing 2 or 3 students. We also attended each class, took field notes, and sometimes even walked around the classroom and asked some non-focus group students to do quick demos of their work on video. Here, we present a preliminary analysis of this data through vignettes from students 'block-codes and excerpts from the video data.

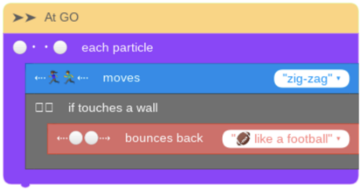
Preliminary findings

We observed that almost all of the students successfully engaged in phenomenological programming. We begin presenting our findings with some examples from the students 'blocks-based algorithms. Figure 8 provides four snapshots from the students 'blocks-based algorithms after the step 2 (micro-level modeling) and their own explanations on why they chose specific code blocks. We chose to ask the students to upload their code at this step, not at the end, because we wanted to observe their assumptions about individual gas particles before they tested their code with hundreds of particles. This data is valuable to show how students 'programming of micro-level gas particles was informed by their intuitive understanding of macro-level phenomena. We also include the students 'own explanations on the right side of Figure 8 to highlight their reasoning behind choosing the specific coding blocks and phenomenological explanations.

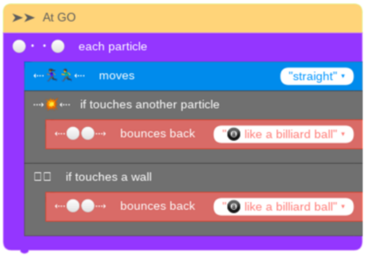
Each example in Figure 8 describes particle movement differently and each student has different reasoning behind their coding decisions. However, we argue that one trend is salient: the students' programming decisions are informed by what kinds of occurrences they found to be natural and expected (sense of mechanism; diSessa, 1993) instead of the formal science terms or explanations. For example, the Student #2 reasons that "*there is not one way each particle moves each time*", while the Student #3 reasons that "*straight*" movement is the "*most realistic*" one. Three of the students believe that particles bounce like billiard balls. One of them specifically reasons that billiard balls do not lose energy when they bounce off of the walls and other particles. Another student simply states that "*they don't slow down*." The last student, on the other hand, neither mentions direction nor speed. In contrast, the Student #2 takes her understanding of a football that they "*don't always have the same direction come back*", while not mentioning the energy exchange at all. Overall, we observe that the students were comfortable in expressing their intuitive understanding of gas particles through the phenomenological blocks. More importantly, there was a great diversity of algorithms created by the students and the underlying student reasoning. Given the apparent difficulty of expressing their intuitive understanding even verbally, it is encouraging that these students could do so computationally.



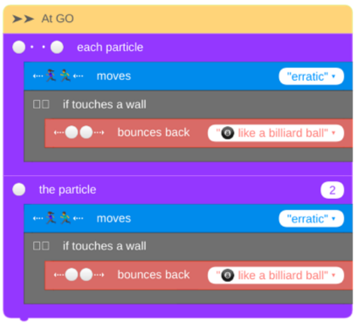
Block	Because ....
Each particle	Because it would program the particles and program how they move
If touches a wall	Because I didn't want the balls to leave the square and disappear
If touches another particle	Because I didn't want the balls/molecules to go through each other because that's not accurate.
Bounce back (2)	I wanted them to move similar to a billiard ball because they don't slow down.



Block	Because ....
Zig-Zag	There's not one way that each blocks moves each time.
Like a football	It bounces back like a football because it doesnt always have the same direction come back



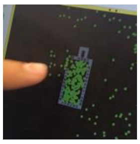
Block	Because ....
moves straight	I used "moves straight" because it was the most realistic out of each one. I wanted them to sorta behave like pool balls. So I based the model off of it.
if touches a wall	Since I based it off the game Pool I wanted them to react like a pool ball. So I chose billiard balls
it touchse another particle	I also chose Billiard balls for this one too. I wanted them to react as realistic as possible. Although I do not know the material of the particles.



Block	Because ....
Moves erratic	particles shake and move around a lot.
if touches wall (square)	If I didn't use it the particles would've bounced out the frame.
bounces back like a billiard ball	When particles touch they bounce off each other.

Figure 8. Three examples from the students' blocks-based algorithms and with their own explanations

The video data from quick student demonstrations also showed that the students 'programming process was highly influenced by their sense of mechanism. Moreover, we were able to observe how students debugged their code after to the conflicts between their assumptions about the micro-level gas particle behavior and macro level outcomes. In the Excerpt 1 below, the Student #4 from the Figure 8 explains her computational modeling process and how in the last step she had to fix two major bugs in her project.

Dialogue	Video Snapshot (cropped)
<u>Researcher:</u> So, this was your original static model, right?	
<u>Student:</u> Yeah. And first I added the red wall.	
<u>Student:</u> And then another change I made was change the direction it moved. *It was erratic for both and then I changed it to straight.* <u>Researcher:</u> Why did you change it? <u>Student:</u> Because I wanted to represent, like, the way the particles move once it comes out the spray can. <u>Researcher:</u> They weren't spraying out as you liked?	
<u>Student:</u> Oh no. *It was just like, spreading out, instead of going towards one direction and then spreading out.* And then, (runs the model with the red wall closed) this is how it was. It was all cramped in there. And then when I took off the red wall, it's all going in one direction and it spreads out, which is what I wanted to show.	

Except 1. A student's explanation of her programming process to the researcher

The first one was a simple design bug: she forgot to design a removable wall to represent the valve. This might also be caused by the activity prompts on the lesson itself or the teachers ' omission. She explains how she solved it quickly. The second one, though, is directly related to her assumptions about the specific particle movement pattern and how it exhibits itself at the macro level. In this quick demo, she explains to the researcher that she initially made her particles move "erratically." As her explanation on Figure 8 shows, she thought that "particles shake and move around a lot." However, in the last step of the activity, when she tried to run her simulation, she noticed that the particles did not behave like they do in real life. In other words, they did not spray in one direction, but they spread out. This prompted her to make the particles move straight instead. In addition, to a question on the portal that asked if their air duster model worked as expected, she responded: "*At first it didn't work as I expected it to, because the particles were spreading into the air which is what I didn't really want to represent. So, I changed my move block from erratic to straight to show the pressure of the air and how they spread into the air.*"

In the end, this student did not only fix her model, but she also debugged her own thinking during the process and learned about how simple micro-level behavior may result in surprising macro-level patterns. She did this all while she did not have to learn formal theories, solve equations, or even articulate her ideas coherently. This is not only a desired outcome for computational modeling, but a very critical idea for learning kinetic molecular theory. We even observe that she uses the term "pressure" in her reasoning about the changes she made in the code.

Discussion and future work

Our design-based research study is still in its early stages and the code-first gas particle sandbox we present in this paper is our first attempt at creating phenomenological code-blocks. The findings we present are preliminary, with further analysis needed to inform the next iteration of our design in order to further study and clarify our findings. Nevertheless, we are encouraged both by the ease with which students were able to start programming with the phenomenological blocks and how well the blocks corresponded to the kinds of occurrences they found natural and to be expected. We are also encouraged by how the student presented in Excerpt 1 was able to *debug* her own thinking as she debugged her model. We argue that such experiences with code-first learning environments help students develop a "feel" of real-world phenomena that is better aligned with correct scientific explanations. In our future work, we will conduct more research implementations and collect more data on how students interact with the code-first gas particle sandbox and the impact of these interactions on their learning. We will continue to improve our design and explore principles for the design of new phenomenological blocks. We are optimistic that if successful, code-first learning environments with phenomenological programming can accelerate the diffusion of computational thinking practices and powerful learning in science and mathematics classrooms.

Acknowledgements

This work was made possible through generous support from the National Science Foundation (grants CNS-1138461, CNS-1441041, DRL-1020101, DRL-1640201 and DRL-1842374) and the Spencer Foundation (Award #201600069). Any opinions, findings, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

References

Aslan, U., LaGrassa, N., Horn, M., & Wilensky, U. (2020a). Putting the Taxonomy into Practice: Investigating Students' Learning of Chemistry with Integrated Computational Thinking Activities. *American Education Research Association (AERA) 2020 Annual Meeting*. San Francisco, CA.

Aslan, U., LaGrassa, N., Horn, M., & Wilensky, U. (2020b, *in press*). Phenomenological Programming: a novel approach to designing domain specific programming environments for science learning. *Paper accepted to the ACM Interaction Design and Children (IDC) conference*. London, UK.

Begel, A. (1996). LogoBlocks: A Graphical Programming Language for Interacting with the World (Doctoral dissertation). Retrieved from https://llk.media.mit.edu/papers/archive/begel-aup.pdf

Bau, D., & Bau, D. A. (2014). A preview of Pencil Code: A tool for developing mastery of programming. *Proceedings of the 2nd Workshop on Programming for Mobile & Touch* (pp. 21-24).

Brady, C., Holbert, N., Soylu, F., Novak, M., & Wilensky, U. (2014). Sandboxes for model-based inquiry. *Journal of Science Education and Technology*, *24*(2-3), 265-286.

Cobb, P., Confrey, J., DiSessa, A., Lehrer, R., & Schauble, L. (2003). Design experiments in educational research. *Educational researcher*, *32*(1), 9-13.

diSessa, A. A. (1993). Toward an epistemology of physics. *Cognition and instruction*, *10*(2-3), 105-225.

diSessa, A. (2015). Alternative conceptions and P-prims. *Encyclopedia of science education*.

diSessa, A. & Minstrell, J. (1998). Cultivating conceptual change with benchmark lessons. *Thinking practices in learning and teaching science and mathematics*, 155-187.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, *42*(1), 38-43.

Guo, Y., Wagh, A., Brady, C., Levy, S. T., Horn, M. S., & Wilensky, U. (2016, June). Frogs to Think with: Improving Students' Computational Thinking and Understanding of Evolution in A Code-First Learning Environment. In *Proceedings of the 15th International Conference on Interaction Design and Children* (pp. 246-254). ACM.

Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive learning environments*, *1*(1), 1-32.

Horn, M. S., & Wilensky, U. (2012). NetTango: A mash-up of NetLogo and Tern. *AERA 2012*.

Horn, M. S., Brady, C., Hjorth, A., Wagh, A., & Wilensky, U. (2014, June). Frog pond: a code-first learning environment on evolution and natural selection. In *Proceedings of the 2014 conference on Interaction design and children* (pp. 357-360). ACM.

Hoyles, C., & Noss, R. (1992). A pedagogy for mathematical microworlds. *Educational studies in Mathematics*, *23*(1), 31-57.

Kahn, K. (2007). Comparing multi-agent models composed from micro-behaviours. *M2M*,165-177.

Kind, V. (2004). Beyond appearances: Students 'misconceptions about basic chemical ideas.

Lin, H. S., Cheng, H. J., & Lawrenz, F. (2000). The assessment of students and teachers' understanding of gas laws. *Journal of Chemical Education*, *77*(2), 235.

Levy, S. T., Kim, H., & Wilensky, U. (2004, April). Connected chemistry—A study of secondary students using agent-based models to learn chemistry. In *annual meeting of the American Educational Research Association, San Diego, CA* (pp. 12-16).

Levy, S., & Wilensky, U. (2007). How do I get there... straight, oscillate or inch? High-school students' exploration patterns of Connected Chemistry. In *2007 meeting of the American Educational Research Association*.

Levy, S. T., & Wilensky, U. (2009). Crossing levels and representations: The Connected Chemistry (CC1) curriculum. *Journal of Science Education and Technology*, *18*(3), 224-242.

Nakhleh, M. B. (1992). Why some students don't learn chemistry: Chemical misconceptions. *Journal of chemical education*, *69*(3), 191.

Papert, S. A. (1971). Teaching children thinking. In *Proceedings of IFIPS World Congress and Education.* Amsterdam, Netherlands.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Papert, S. (1991). Situating constructionism. *Constructionism*, *36*(2), 1-11.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. B. (2009). Scratch: Programming for all. *Communications ACM*, *52*(11), 60-67.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, *18*(2), 351-380.

Smith III, J. P., diSessa, A., & Roschelle, J. (1994). Misconceptions reconceived: A constructivist analysis of knowledge in transition. *The journal of the learning sciences*, *3*(2), 115-163.

Stieff, M., & Wilensky, U. (2003). Connected chemistry—incorporating interactive simulations into the chemistry classroom. *Journal of Science Education and Technology*, *12*(3), 285-302.

Wagh, A., & Wilensky, U. (2018). EvoBuild: A QuickStart Toolkit for Programming Agent-Based Models of Evolutionary Processes. *Journal of Science Education and Technology*, *27*(2), 131-146.

Wagh, A., Levy, S., Horn, M. S., Guo, Y., Brady, C., & Wilensky, U. (2017, June). Anchor Code: Modularity as Evidence for Conceptual Learning and Computational Practices of Students Using a Code-First Environment. In *Computer Supported Collaborative Learning (CSCL)*.

Weintrop, D., & Wilensky, U. (2015). To block or not to block, that is the question: students' perceptions of blocks-based programming. In *Proceedings of the 14th International Conference on Interaction Design and Children* (pp. 199-208). ACM.

Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education*, *18*(1), 3.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127-147.

Wilensky, U. (1999a). NetLogo. Evanston, IL: Center for connected learning and computer-based modeling, Northwestern University.

Wilensky, U. (1999b). GasLab—An extensible modeling toolkit for connecting micro-and macro-properties of gases. In *Modeling and simulation in science and mathematics education* (pp. 151-178). Springer, New York, NY.

Wilensky, U. (2001, August). Modeling nature's emergent patterns with multi-agent languages. In *Proceedings of EuroLogo* (pp. 1-6).

Wilensky, U. (2003). Statistical mechanics for secondary school: The GasLab multi-agent modeling toolkit. *International Journal of Computers for Mathematical Learning*, *8*(1), 1-41.

Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—an embodied modeling approach. *Cognition and instruction*, *24*(2), 171-209.

Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering computational literacy in science classrooms. *Communications ACM*, *57*(8), 24-28.

Wilkerson-Jerde, M., Wagh, A., & Wilensky, U. (2015). Balancing Curricular and Pedagogical Needs in Computational Construction Kits: Lessons from the DeltaTick Project. *Science Education*, *99*(3), 465-499.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

Considering Alternative Endpoints: An Exploration in the Space of Computing Educations

David Weintrop, *weintrop@umd.edu*
College of Education & College of Information Studies,
University of Maryland, College Park, USA

Nathan Holbert, *holbert@tc.columbia.edu*
Department of Mathematics, Science, and Technology,
Teachers College, Columbia University, New York, USA

Michael Tissenbaum, *miketiss@illinois.edu*
Department of Curriculum and Instruction,
University of Illinois, Champaign, USA

Abstract

As more and more countries are pursuing the goal of integrating computing and computer science instruction into curricula and standards, it is important that we carefully consider what the goals and motivations of such programs are and whether or not it is in the best interest of the learners most directly impacted by them. While many national efforts tend to deploy rhetoric elevating economic concerns alongside statements about creativity and human flourishing, the programs, software, curricula, and infrastructure being designed and implemented focus heavily on providing learners with the skills, practices, and mindset of the professional software developer. We contend that computing for all efforts must take the “for all” seriously and recognize that preparing every learner for a career as a software developer is neither realistic nor desirable. Instead, those working towards the goal of universal computing education should begin to consider alternative endpoints for learners after completing computing curricula that better reflect the plurality of ways the computing is impacting their current lives and their futures. Further, we argue that constructionist designs and principles should play a central role in shaping what a computing education might look like that supports these diverse endpoints. In developing this argument, we provide examples of tools and environments that are designed towards alternative, yet equally valid and valuable, endpoints. Central to these alternative endpoints and the tools used to support learners are core constructionist ideas including the centrality of constructing computational artifacts, the importance of pursuing personally meaningful projects, and presenting learners with low-floor/high-ceiling tools with which to work.

Keywords

Constructionism, Computing Education, Computer Science Education, Equity, Access

Introduction

In recent years, there has been a concerted effort to make computing and coding a core educational experience in countries throughout the world (e.g. CSforAll, Make it Digital, Computing at School, etc.). A variety of arguments are given for these large scale efforts ranging from a desire to support young people in being able to “express themselves digitally” (BBC, 2019), empowering them to impact their communities through programming (Bhattacharya, 2017), to providing “the computational thinking skills they need to be creators in the digital economy” (Smith, 2016). These three goals of computing education--creative expression, social justice, and economic opportunity--are frequently cited as primary reasons all students should be exposed to the powerful ideas of computing. In their review of motivations for bringing computing instruction into all classrooms, Vogel and colleagues (2017) identified seven distinct motivations, adding arguments such as creating an informed citizenry and improving general technological literacy to the aforementioned goals. The diversity of these goals speaks to the way in which computing has become a core part of society. Furthermore, these goals highlight the need to make computing education efforts universal--targeting all young people regardless of school, age, or interest.

We think the goal of bringing computer science experiences and practices to all young people as part of their formal educational experience is a worthwhile endeavor and have each in our own way worked to support this effort. We are encouraged to see public rhetoric highlighting the social justice implications of computing education. Likewise, the explicit acknowledgment that computing is a powerful new form of creative expression aligns with a long history of computing education that emerged from early constructionist thought. And while we are hopeful that those participating in computing education are attending to these important goals, the enactment of these initiatives--in the form of curricula, learning environments, tools, assessments, policy, etc.--suggest that the most critical educational decision-makers most directly shaping the enactment of computing for all initiatives are prioritizing place economic concerns first. In other words, while the computing education community claims to attend to social justice and creative expression, the assumed endpoint of computing education seems to be about job preparation--increasing the number of programmers in the workforce so that we can compete in a global market.

Just as Papert explored the idea of alternative possible mathematics educations (1996), here we aim to identify just a few points in a large N-dimensional space that might serve as examples of possible versions of computing education. In this paper, we explore potential endpoints to computing education--what might people do with computing? We argue that computing education should truly acknowledge, and enact, the belief that computing has meaning and value in a host of potential careers and daily experiences. This recognition of alternative endpoints is important and potentially transformative. Identifying alternative potential computing educations invites us to define the dimensions in which this point exists—to, in essence, think about how legitimizing alternative endpoints beyond undergraduate degrees in computer science can bring new tools, practices, and contexts into computer science classrooms and change the narrative around what computer science is and what it looks like to practice it. As alternative endpoints become more central to computer science, characteristics of the curricula, tools, assessments, and projects that live in computer science classrooms can begin to change to reflect these alternative endpoints, and in doing so, can open up the field to those not historically drawn to conventional computer science pathways.

This paper argues that efforts to bring computing to all, and the learners who participate in such programs, would be better served by considering the plurality of endpoints beyond those that prioritize economic interests and career outcomes. Further, we argue that constructionist design and values should play a central role in shaping what a computing education might look like that supports these diverse endpoints. In doing so, various computing for all efforts can better welcome and support the learners they are trying to reach by aligning instruction and learning opportunities with the ideals, values, and goals of the learners. Further, legitimizing and valuing endpoints beyond conventional computer science careers can lead to a more inclusive and welcoming form of computer science, where creative, expressive, and culturally-valued instantiations of computer science ideas are valued alongside the skills that can lead to conventional computer science careers. This work adds to a growing chorus of voices, both in academia and beyond, pushing to rethink the goals, values, and priorities of contemporary computing education (Lewis, 2017; Santo et al., 2019; Vakil, 2018; Vogel et al., 2017).

To demonstrate this idea, this paper lays out three distinct computer science endpoints outside of the conventional computer science pipeline, showing how the consideration of alternative endpoints can shape the tools used, the ways learners are supported in engaging with computer science ideas, and ultimately reshape the computing education landscape and what it looks like for a learner to authentically participate in meaningful computing.

Motivation for Considering Alternative Computing Endpoints

This paper argues for a re-examination of the nature and goals of broad computing education initiatives. Instead of starting with specific values or goals, this work instead begins by considering various desired endpoints of computing instruction and then works backward to reason about what form learning activities might take and what are the underlying values and principles that support learners in reaching these endpoints. The result of this exercise is a push for rethinking the form of contemporary computing education with an eye towards more diverse, equitable, and meaningful endpoints.

Across the literature, a broad array of motivations are provided for computing education. Working with New York City school district stakeholders, Vogel et al. (2017) collected a total of 161 arguments for computer science instruction, and grouped them into seven categories: (1) economic and workforce development, (2) equity and social justice, (3) competencies and literacies, (4) citizenship and civic life, (5) scientific, technological and social innovation, (6) school improvement and reform and (7) fun, fulfillment and personal agency. This plurality of ideas is often not reflected in the nature of the tools, activities, and assessments used as part of classroom instruction. This is especially true with older learners where priorities further shift toward the use of professional programming languages and a prioritization for college and career readiness.

With this work, we introduce three distinct alternative endpoints for computing outside of the conventional computer science pathway as a means of rethinking what forms instruction can or should take.

Constructionism as a means to Reconceptualize Computing Education

A constructionist lens is a particularly powerful means for positioning alternative endpoints to computer education. Through the building of computational artifacts, learners have the opportunity to engage in critical reflection on what they are making and why and how it relates to them personally and to society more broadly (Ratto & Boler, 2014). By focusing on these broader socio-technical aspects of learners' construction (beyond the end-goal of getting a job as a programmer), we introduce opportunities for developing critical consciousness (Freire, 1974; Lee & Soep, 2016) and an understanding how computing shapes the world around them and their ability to create with it for their own goals, identity, construction, and expression (Holbert et al., in press; Tissenbaum et al., 2019).

Papert touted that children learn how to think critically through the process of solving problems that arise while programming computers (1996). Through the creative and investigative processes that are at the core of constructionism, learners begin to understand the multi-faceted ways that computing can and should be a central force for them to personally express themselves, construct their digital and personal identities, and empower them to be critically aware and empowered citizens.

Constructionism's attention to the learner's values and interests make it well suited to support learners in using computational power to explore a diverse range of experiences, practices, phenomena, etc. Whether supporting young people in constructing video games (Harel & Papert, 1991; Kafai, 1991; Weintrop et al., 2012), interactive art (Bontá et al., 2010; Papert & Solomon, 1971), musical instruments (Cavallo et al., 2004; Gorson et al., 2017), e-fashion (Buechley & Eisenberg, 2008; Kafai et al., 2014), or public service announcements (Blikstein, 2008), since the inception of the design paradigm, constructionists have cared deeply about supporting learners as they express their passions, explore their interests, or work to design solutions to real-world problems. We argue that this foundational quality must be present in any effort to broaden participation in computing education. When learners are given the space to construct objects--both digital or physical--that have personal or communal meaning, they have the opportunity to represent these passions in inspectable artifacts that can be viewed, critiqued, extended, or repurposed by others. This not only has powerful cognitive benefits--being able to externalize one's thinking into representational systems that can be debugged, modified, etc.--but also important identity implications. The computer code and resulting artifact can serve as a representation of one's work and one's contributions to the broader computing community. From a constructionist perspective, computing is not just an economically viable way to make things, but a way of *doing* things, with others, to change and impact the world.

Alternative Endpoints for Computer Science

In this section, we lay out three distinct views drawn from our research of alternative endpoints for computing education. The goal of this work is to argue for the importance of computing for all while also providing legitimate and authentic applications of computer science knowledge outside the existing pathway that leads to a computer science industry job.

Endpoint: Impacting local communities and immediate needs

The first endpoint we consider is the development of novices' identities as empowered to address real issues in their own lives, schools, and communities. While traditional computing education was locked to desktop computers, often taking place in computer lab settings, the introduction of mobile technologies (in particular smartphones) has allowed computing education to move out of the classroom and into learners' everyday lives. This ability for the products that students create to be taken out of the computer lab and into the world has allowed students and educators to move beyond simply writing code, instead critically asking *why* and *who* they are building it for, and to what end (Holbert, 2016; Lee & Soep, 2016). By situating computing education directly in students' lives, we open up computing education as a possibility space for impact and empowerment. This is critically important, as a long line of research has shown that the failure to meaningfully connect computing to the personal lives of students contributes to learners feeling computing is not useful or relevant to them (American Association of University Women, 1994; Couragion Corporation, 2018; Margolis & Fisher, 2003). This is particularly true for students underrepresented in computing and engineering careers (Cheryan et al., 2017; Pinkard et al., 2017; Taheri et al., 2019).

In response, we posit that there is a need to re-think the goals of computing education through a lens of *Computational Action* (Tissenbaum et al., 2019), which focuses on three key factors: 1) *Computational identity*, which is a person's recognition of themselves as capable of identifying and creatively implementing computational solutions to issues in their lives, schools, and communities; 2) *Digital empowerment*, which focuses on people's ability to put their computational identity into action in authentic and personally meaningful ways; and 3) *Computational design thinking*, in which learners' can successfully articulate the processes by which they will design and develop their solutions.

In order to support students' engagement in computational action, we need tools that reduce the barriers for them to quickly build, implement, and refine their designs. One example of the kinds of platforms particularly well-suited for such an approach is MIT's App Inventor, a block-based programming language that enables users to build fully functioning, native Android mobile applications. However, it is not enough to provide novice learners with a coding platform and simply let them loose. Supporting computational action also requires the development of scaffolds in the form of support materials (such as design documents) and scripted activities that lead students through the design process. Developing these additional supports is key to ensuring that students progress from ideation to implementation.

To explore how a computational action-focused curriculum can support students in developing meaningful solutions to personally-relevant issues, Tissenbaum, Sheldon & Ableson (2019) implemented a computational action curriculum in an ethnically diverse urban high school in the United States. Tissenbaum and colleagues chose this school as it encompassed a broad spectrum of students, particularly those not traditionally represented in the computing career pipeline. Working with the teacher, they identified an issue that was of interest to students at the school and the broader local community: the pollution of the local river (a major feature that runs through the middle of the city). Working in collaborative teams, students developed their own solutions to increase awareness and investigation strategies for cleaning up the river. To ensure that the students felt their work was meaningful (i.e. to support their *computational empowerment*), they presented their final projects at the school-wide job fair, which included visits from local council members and the mayor.

At the end of the curriculum, many of the students expressed that they never thought they would be able to build an app themselves, let alone build one that they felt had a chance to make real change. Many also expressed excitement towards developing solutions to new problems using the computational tools and knowledge developed during this project.

As this example shows, a computational action approach to computing education has the potential to support students to become, not only programmers but computationally literate, empowered problem-solving citizens.

Endpoint: Means of personal and social creative expression

Many computing initiatives and tools pursue the goal of empowering young people to express themselves digitally. These efforts see the computer and code as a digital canvas, a medium that enables a host of alternative forms of creative expression. Early implementations of Logo, the first true programming language for children, often invited young people to create “computer graphics” similar to those they saw at the arcade and on their video game systems (Harel & Papert, 1990; Kafai, 1996). Scratch, a successor of Logo and the programming environment most widely used to introduce learners to programming, invites children to create interactive stories, games, and animations (Resnick et al., 2009). Similarly, so-called “making,” a popular means of combining computing with fabrication and craft work, invites learners to create personally interesting physical and tangible artifacts (Halverson & Sheridan, 2014).

While many computing education efforts do engage learners in personal and social creative expression, these activities are often used as a means to acquire STEM or computing content knowledge or practices. However, creative construction offers more than just a compelling way to encounter the practices of the software engineer. Here we propose that creative expression itself can be a powerful and worthwhile endpoint of computing education.

The design and creation of compelling artifacts that speak to the experiences, values, or perspectives of society has traditionally been considered the domain of the artist. While artists work with a variety of media and materials, the computer has been a useful tool for artists since its inception to ask questions about the nature of humanity, the role of technology in society, and to reflect on social and governmental structures and systems.

Afrofuturism is a genre of art, music, and literature that has used the practices, affordances, and implications of computing to great effect to imagine future societies and worlds that center the experiences and values of people of color (Anderson & Jones, 2015; Dery, 1994). These perspectives can be found in the costumes and pageantry of Parliament-Funkadelic, the stories sung by Janelle Monea’s android alter ego Cindi Mayweather, and the futuristic technologies created by Shuri in the Black Panther. In the Remixing Wakanda project, Holbert and colleagues leveraged the Afrofuturist aesthetic and design genre to invite young people to reflect on the current state of the world and to use computational tools and practices to create artistic artifacts that construct a future that represents their values and perspectives (Dando et al., 2019; Holbert et al., in press).

In this project, making and computing became tools for Black teens to critically examine their experiences as young people of color in a large American city. Working with professional comic book artists, learning scientists, designers, and local activists, participants designed and ultimately constructed futuristic artifacts or societies that imagined futures that valued harmony between diverse groups of people and between humanity and nature. In these constructions, participants used computational tools, sensors, and circuitry to creatively merge aesthetic considerations with functionality to highlight humanity's problematic relationship with the environment and to acknowledge and respond to their experiences with racism and inequality (Figure 1). For example, one participant designed a fashionable cloak that hid the wearer from prying eyes--eyes that she said, "make you feel like you're alien [...] like you some art exhibit or something." While this cloak offered protection in the form of anonymity as well as a battery of sensors that monitored the health and wellbeing of the wearer (at one point the designer also considered including pepper spray as a built-in feature), the cloak also elevated a distinctly African aesthetic. Using textiles and patterns from her native Senegal, as well as a hypothetical technology that could morph into personally meaningful 3D iconography, this participant created a computing-rich artifact that proudly displayed her heritage.

Figure 1. Learners constructing artifacts as part of the Remixing Wakanda project.

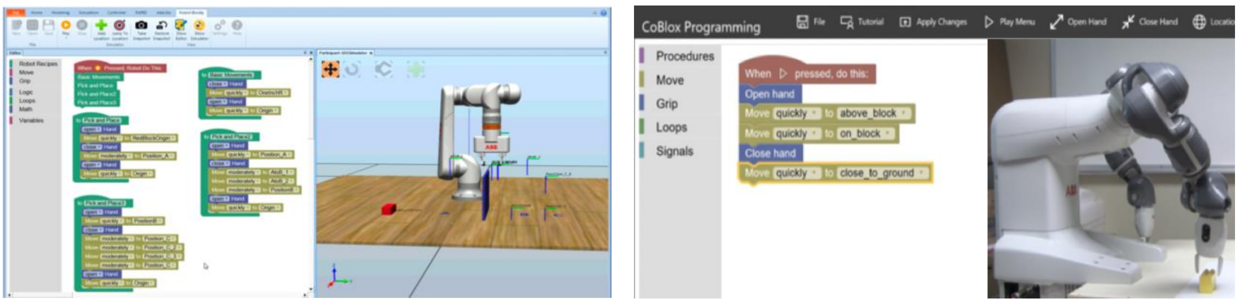
Another participant reflected on her personal frustration about litter and trash in the city. More than a visual blight, this trash often caused disruption in public transportation that impacted her ability to move through her city. As a response, this participant designed an aesthetically appealing trash receptacle that included a futuristic technology that would directly convert trash to energy that could be used to power street lights (a safety concern for those that work late) or serve as a charging station. She then went on to program a microcontroller to illustrate the principle behind this imagined technology.

In each case of the Remixing Wakanda project, computing serves as a tool for critically reflecting on the current state of the world and for creating representations of a possible future that might initiate change today. While participants did encounter coding, and potentially came away with new knowledge about computing concepts or practices, these experiences are themselves means towards the end of creative expression, of creating artifacts and representations the center their anxieties and fears as well as their hopes and dreams. While one implication of this work may be that the construction of critical artifacts may appeal to a broad range of learners currently underrepresented in computing domains, this is far from the only possible purpose of such design experiences. Rather, in the Remixing Wakanda project, computing is a tool for engaging in critical reflection on inequitable societies, unsustainable energy practices, and systems of oppression. Here, computing is a means of agitating for change--itself a powerful and important endpoint (Holbert, in press).

Endpoint: Blue Collar Computing

The third endpoint we present is the closest to the conventional endpoint of a profession in a computer science-related field but challenges the notion of what a computer science-related field looks like. As the technologies that enable automation become cheaper and their capabilities expand, the nature of manual labor is shifting. An example of this can be seen with collaborative robotics where humans and robots work side-by-side in a complementary capacity (Colgate et al., 1996; Kock et al., 2011). Where the robots excel at tasks that require precision or repetition, humans are more efficient at decision-making that requires judgment, adaptability, and creativity (Blank et al., 2006). Responding to this trend of the introduction of automation into new contexts, the skills and knowledge one needs to succeed in this setting relies on an understanding of foundational computing concepts in order to program and re-programming industrial robots. One way to prepare workers for the new computational landscape is to integrate computing across the K-12 landscape in an effort to prepare all learners to write and modify programs written in complicated industrial robotics programming languages. That approach aligns with much of the existing CS for all rhetoric which seeks to prepare all students for a future as a software developer. An alternative approach to address this issue is to redesign the tools at hand, in this case, the industrial robotics programming interface, so as to make it more intuitive, accessible, and draw more directly from the existing knowledge and expertise of today's worker.

Towards this end, Weintrop and colleagues set out to re-envision what it might look like to program industrial robots and investigated training approaches to help adult novices successfully author useful routines. The result of this work was a programming environment called CoBlox (Weintrop et al., 2018) which allows users with little or no prior programming to program virtual (Figure 2a) or physical (Figure 2b) robots. Drawing from prior work on the design of accessible and intuitive programming environments (Bau et al., 2017; Weintrop, 2019), CoBlox uses the block-based programming modality to situate the robotics programming task. Block-based programming using a programming-command-as-puzzle-piece metaphor to provide visual cues as to how and where a command can be used in the construction of a program (Maloney et al., 2010; Weintrop & Wilensky, 2015). CoBlox also leverages several features of block-based programming environments, such as natural language expressions within individual commands, predefined templates for common routines, and integration with both virtual and physical robots (Weintrop et al., 2017).



(a) (b)

Figure 2. Virtual (a) and physical (b) implementations of the CoBlox *programming Environment.*

The goal of highlighting the CoBlox design is to showcase what it looks like for computer science knowledge to be used in professional settings historically not considered within the purview of computer science. In documenting the ways that knowledge and practices clearly within the bounds of the discipline of computer science (programming in this case) can be enacted outside of what is typically viewed as a computer science endpoint, we show the importance of the consideration for alternative endpoints. As the skills and concepts from the field of computer science continue to impact a wider and more diverse set of professions, it is important for the narrative motivating and arguments justifying computer science to reflect this new plurality. With CoBlox, we see an authentic and legitimate professional endpoint in which computer science knowledge is valued but not typically included in the narrative around why computer science is important. By including endpoints that reflect the larger swath of professions impacted by computer science, students who do not see themselves as future software developers may come to recognize the utility of learning computer science.

Conclusion

The increasingly digital nature of our world requires that all learners feel empowered to understand and meaningfully participate in computational practices. The last decade has seen those from the computer science community lead the effort in designing the tools, creating curricula, and crafting the policy that will shape the form this instruction takes for future generations. While it is important for computer science to have a seat at the table, it is just as important that the ideas, values, and goals of those beyond the field also participate to reflect the growing role of computing in the world. Through envisioning and valuing alternative, yet equally valid and important, endpoints, this work seeks to start a conversation about the nature of the dimensions that might make up alternate computing educations--to re-evaluate the current tools and curricula to prepare learners for a future of active and empowered computing-literate citizens. In rethinking the goals of computing education, we see the ideas and principles of constructionism as having much to contribute towards realizing a form of computing education that more fully reflects the plurality and diversity of computing endpoints.

Motivating computing instruction solely based on economic outcomes does not accurately reflect the role of computing in the current and future society. In some contexts, the economic motivation for teaching computer science lives alongside goals such as preparing learners to be informed digital citizens or to prepare learners for 21st-century jobs beyond those in the technology sector. While such framing more accurately reflects the influence of computing in society, the programming languages, computer science curricula, and larger computing pathways still are largely designed as on-ramps intended to lead learners into the computing industry. In this way, the existing educational infrastructure is structured with computer science degrees and the industry jobs that will follow as the endpoint of computing instruction. While it is important that such pathways exist, it is equally important that instruction help learners recognize alternative endpoints computer science can lead to and support those learners for whom computer science can be a generative and valuable skill outside of professional contexts. By recognizing the value of these alternative endpoints, and hypothesizing diverse forms of computing educations, we open up the computing education landscape to be more inclusive, adaptive, and empowering for all, rather than for the select few who choose programming jobs as their educational endpoint.

References

American Association of University Women. (1994). *Shortchanging Girls, Shortchanging America*. AAUW Educational Foundation.

Anderson, R., & Jones, C. E. (2015). *Afrofuturism 2.0: The Rise of Astro-Blackness*. Lexington Books.

Bau, D., Gray, J., Kelleher, C., Sheldon, J., & Turbak, F. (2017). Learnable programming: Blocks and beyond. *Communications of the ACM*, *60*(6), 72–80. https://doi.org/10.1145/3015455

BBC. (2019). *Make It Digital—The BBC micro:bit*. BBC. https://www.bbc.co.uk/programmes/articles/4hVG2Br1W1LKCmw8nSm9WnQ/the-bbc-micro-bit

Bhattacharya, A. (2017). *What happens when girls in one of the world's largest slums start coding and building apps*. Quartz India. https://qz.com/india/1032018/dharavi-diary-what-happens-when-girls-in-one-of-the-worlds-largest-slums-start-coding-and-building-apps/

Blank, D., Kumar, D., Meeden, L., & Yanco, H. (2006). The Pyro toolkit for AI and robotics. *AI Magazine*, *27*(1), 39.

Blikstein, P. (2008). Travels in Troy with Freire: Technology as an agent for emancipation. *Paulo Freire: The Possible Dream. Rotterdam, Netherlands: Sense*. http://tltl.stanford.edu/publications/papers-or-book-chapters/travels-troy-freire

Bontá, P., Papert, A., & Silverman, B. (2010). Turtle, Art, TurtleArt. *Proceedings of Constructionism 2010 Conference*.

Buechley, L., & Eisenberg, M. (2008). The LilyPad Arduino: Toward wearable engineering for everyone. *Pervasive Computing, IEEE*, *7*(2), 12–15.

Cavallo, D., Papert, S., & Stager, G. (2004). Climbing to understanding: Lessons from an experimental learning environment for adjudicated youth. *Proceedings of the 6th International Conference on Learning Sciences*, 113–120.

Cheryan, S., Ziegler, S. A., Montoya, A. K., & Jiang, L. (2017). Why are some STEM fields more gender balanced than others? *Psychological Bulletin*, *143*(1), 1–35. https://doi.org/10.1037/bul0000052

Colgate, J. E., Edward, J., Peshkin, M. A., & Wannasuphoprasit, W. (1996). *Cobots: Robots For Collaboration With Human Operators*.

Couragion Coroporation. (2018). *Altering the Vision of Who Can Succeed in Computing*. Oracle Academy.

Dando, M. B., Holbert, N., & Correa, I. (2019). Remixing Wakanda: Envisioning Critical Afrofuturist Design Pedagogies. *Proceedings of FabLearn 2019*, 156–159.

Dery, M. (1994). *Flame Wars: The Discourse of Cyberculture*. Duke University Press.

Freire, P. (1974). *Pedagogy of the Oppressed*. Basic Books.

Gorson, J., Patel, N., Beheshti, E., Magerko, B., & Horn, M. (2017). TunePad: Computational Thinking Through Sound Composition. *Proceedings of the 2017 Conference on Interaction Design and Children*, 484–489.

Halverson, E. R., & Sheridan, K. (2014). The maker movement in education. *Harvard Educational Review*, *84*(4), 495–504.

Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive Learning Environments*, *1*(1), 1–32.

Harel, I., & Papert, S. (Eds.). (1991). *Constructionism*. Ablex Publishing.

Holbert, N. (In Press). Constructionism as a Pedagogy of Disrespect. In N Holbert, M. Berland, & Y. Kafai (Eds.), *Designing Constructionist Futures: The Art, Theory, and Practice of Learning Designs*. MIT Press.

Holbert, N, Dando, M. B., & Correa, I. (In Press). Afrofuturism as critical constructionist design: Building futures from the past and present. Journal of Learning, Media, and Technology. *Journal of Learning, Media, and Technology. http://dx.doi.org/10.1080/17439884.2020.1754237*

Holbert, Nathan. (2016). Leveraging cultural values and “ways of knowing” to increase diversity in maker activities. *International Journal of Child-Computer Interaction*, *9–10*, 33–39. https://doi.org/10.1016/j.ijcci.2016.10.002

Kafai, Y. (1996). Learning Design by Making Games: Children’s Development of Design Strategies in the Creation of a Complex Computational Artifact. In M. Resnick & Y. Kafai (Eds.), *Constructionism in Practice* (pp. 71–96). Lawrence Erlbaum. https://www.taylorfrancis.com/books/e/9780203053492/chapters/10.4324/9780203053492-11

Kafai, Y. B. (1991). Learning design by making games. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 71–96). Ablex Publishing Corporation.

Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., & Lui, D. (2014). A Crafts-Oriented Approach to Computing in High School: Introducing Computational Concepts, Practices, and Perspectives with Electronic Textiles. *Trans. Comput. Educ.*, *14*(1), 1:1–1:20. https://doi.org/10.1145/2576874

Kock, S., Vittor, T., Matthias, B., Jerregard, H., Källman, M., Lundberg, I., Mellander, R., & Hedelind, M. (2011). Robot concept for scalable, flexible assembly automation: A technology study on a harmless dual-armed robot. *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)*, 1–5. https://doi.org/10.1109/ISAM.2011.5942358

Lee, C. H., & Soep, E. (2016). None But Ourselves Can Free Our Minds: Critical Computational Literacy as a Pedagogy of Resistance. *Equity & Excellence in Education*, *49*(4), 480–492. https://doi.org/10.1080/10665684.2016.1227157

Lewis, C. M. (2017). Good (and Bad) Reasons to Teach All Students Computer Science. In S. B. Fee, A. M. Holland-Minkley, & T. E. Lombardi (Eds.), *New Directions for Computing Education: Embedding Computing Across Disciplines* (pp. 15–34). Springer International Publishing. https://doi.org/10.1007/978-3-319-54226-3_2

Maloney, J. H., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, *10*(4), 16.

Margolis, J., & Fisher, A. (2003). *Unlocking the clubhouse: Women in computing*. The MIT Press.

Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*, *1*(1). https://doi.org/10.1007/BF00191473

Papert, Seymour, & Solomon, C. (1971). *Twenty things to do with a computer*. http://18.7.29.232/handle/1721.1/5836

Pinkard, N., Erete, S., Martin, C. K., & Royston, M. M. de. (2017). Digital Youth Divas: Exploring Narrative-Driven Curriculum to Spark Middle School Girls' Interest in Computational Activities. *Journal of the Learning Sciences*, *26*(3), 477–516. https://doi.org/10.1080/10508406.2017.1307199

Ratto, M., & Boler, M. (2014). *DIY citizenship: Critical making and social media*. MIT Press.

Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., & Silver, J. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60.

Santo, R., Vogel, S., & Ching, D. (2019). *CS for What? Diverse Visions fo Computer Science Education in Practice*. CSforALL.

Smith, M. (2016). *Computer Science For All*. Whitehouse.Gov. https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all

Taheri, M., Ross, M., Hazari, Z., Weiss, W., Georgiopoulos, M., Christensen, K., Solis, T., Chari, D., & Taheri, Z. (2019). Exploring Computing Identity and Persistence Across Multiple Groups Us-ing Structural Equation Modeling. *American Society for Engineering Education (ASEE) Conference Proceedings*.

Tissenbaum, M., Sheldon, J., & Abelson, H. (2019). From computational thinking to computational action. *Communications of the ACM*, *62*(3), 34–36.

Vakil, S. (2018). Ethics, Identity, and Political Vision: Toward a Justice-Centered Approach to Equity in Computer Science Education. *Harvard Educational Review*, *88*(1), 26–52.

Vogel, S., Santo, R., & Ching, D. (2017). Visions of Computer Science Education: Unpacking Arguments for and Projected Impacts of CS4All Initiatives. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education - SIGCSE ’17*, 609–614.

Weintrop, D. (2019). Block-based Programming in Computer Science Education. *Commun. ACM*, *62*(8), 22–25. https://doi.org/10.1145/3341221

Weintrop, D., Afzal, A., Salac, J., Francis, P., Li, B., Shepherd, D. C., & Franklin, D. (2018). Evaluating CoBlox: A Comparative Study of Robotics Programming Environments for Adult Novices. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 366:1-12. https://doi.org/10.1145/3173574.3173940

Weintrop, D., Holbert, N., Wilensky, U., & Horn, M. S. (2012). Redefining constructionist video games: Marrying constructionism and video game design. In C. Kynigos, J. Clayson, & N. Yiannoutsou (Eds.), *Proceedings of the Constructionism 2012 Conference*.

Weintrop, D., Shepherd, D. C., Francis, P., & Franklin, D. (2017). Blockly goes to work: Block-based programming for industrial robots. *2017 IEEE Blocks and Beyond Workshop*, 29–36. https://doi.org/10.1109/BLOCKS.2017.8120406

Weintrop, D., & Wilensky, U. (2015). To Block or Not to Block, That is the Question: Students’ Perceptions of Blocks-based Programming. *Proceedings of the 14th International Conference on Interaction Design and Children*, 199–208. https://doi.org/10.1145/2771839.2771860

Constructing what? Prospects for the weak makerspace in strong accountability regimes

Michael Tan *michael.tan@nie.edu.sg*
Office of Education Research, National Institute of Education, NTU, Singapore

Jemuel Ong *jemuel_ong@moe.edu.sg*
Beacon Primary School, Ministry of Education, Singapore

Abstract

Constructionism as a learning theory is based on the idea of constructivism, which itself is a metaphorical extension of the quotidian experience of construction. Because of the ubiquity of manufactured artefacts in our daily existence, it is perhaps inevitable that we perceive construction as a *hylomorphic* process. Hylomorphism refers to the ancient Greek notion that artefacts are the consequence of abstract form impressing itself upon a passive matter. Hylomorphism drives the popular labelling of products as 'Designed in P, Manufactured in Q', and hylomorphism is also supported by the mind-body distinction. While hylomorphism may describe the manufacture of mass produced artefacts by distributed and specialised teams, the making, or more precisely, creation of artefacts by individuals does not proceed in this way. Careful anthropological studies have revealed the impossibility of hylomorphism because of the manner in which materials respond to human agency. Even for experienced artisans working with known materials, subtle variation can mean that the process of making is never identical across different artefacts, with in-the-moment adjustments to adapt to changes. For experts, such minute variation falls below the threshold of novelty, and is merely passed off as repetitive.

If the constructive metaphor is mistaken for materials, what more agentic human beings who can refuse instruction? In constructionist approaches to education, especially in circumstances where technology is involved, the temptation to demonstrate effectiveness can lead toward hylomorphic interpretations of learning interaction. There is an apparent tension here: as with the distinction between manufacturing and making (or creating), there needs to be a certain repeatability below a threshold that teachers can manage. On the other hand, mechanistic efficiency should not become an ideal for human interaction. Makerspaces have become emblematic contexts for the constructionist learning approach in recent years. Promising hands on experiences underwritten by learning theories that suggest that learning will be fun, engaging, and educational, schools and community organisations such as libraries and museums now consider makerspaces an educational addition. It is in the ecological context of makerspaces in interaction with funding agencies and state organisations of education that this research is situated. A makerspace directed toward more 'manufacturing' type of experiences for its participants is more compatible with a managerial desire to strongly control outcomes. On the other hand, greater recognition of the undesirability of mechanistic control of humans leads one to *weak* interpretations of creativity, making, and makerspaces. This research takes place in Singapore, with strong accountability measures and outcomes-based interactions the norm in most public schooling contexts. Using a case study approach, we detail how a weak makerspace can exist within such a strong context. While using the boundary of the makerspace to signal a break with mainstream schooling, the teacher responsible for this makerspace does boundary maintenance work in subtly violating taken-for-granted norms of schooling. In essence, we find that weak makerspaces require strong boundary work. Implications for the wider context are drawn.

Keywords

Makerspaces, Contingency, Case Study, Post-constructivism, Contextual factors, Weak makerspace, Boundary object

Introduction

As learning theories go, constructivism has had a fairly successful run. It takes into account elements typically found in learning interactions, and organises it in a manner that is amenable to our common sense notions of building. Yet, just as our intuitive sense of physics can be mistaken in fully understanding motion, our metaphorical extension of construction may be inadequate for understanding learning. For instance, a well known misconception in physics is that of 'motion equals force': we perceive objects that are in motion to be acted upon by a force. However, appropriate scientific understanding tells us that it is acceleration that is related to force. Constructionism, built atop constructivism, accepts as its premises all conclusions of the learning theory, and hence is susceptible to criticisms such as by Roth (2015) makes the case that we need to seriously consider 'post-constructivist' theories of learning. Among other things, Roth suggests that the constructive metaphor is inadequate because "as theorized by means of the object/motive concept of cultural-historical activity theory, which implies knowledge of the activity-orienting motive—we inherently cannot know the object of future knowledge that is only revealed to us as a consequence and end product of learning" (p. 38). In contrast to typical constructionist approaches which assume that particular learning sequences lead with high certainty to knowable outcomes, Roth argues that what is possible are only *a posteriori* accounts of having learnt something; one cannot predict in advance the transition from not knowing to knowing something. Neither can one predict what knowledge is to be constructed: just as craftsmen, poets, and artists cannot know in advance what it is they want to create until they succeed in creating it, learners cannot know what it is they are to understand until such time that they succeed. There is subtlety in this distinction: classical constructivist theory presupposes the existence of whatever is supposed to be interpreted, and that instruction makes use of prior knowledge as if bricks to create a new knowledge structure. In contrast, Roth suggests that there is "no evidence of some thing being in consciousness that the learner interprets. Instead, it is only at the end of the movement from unawareness to being aware that something—a new object, a new sign, a new idea—comes to stand out" (p. 45).

While Roth's critique has been specific to constructivism, this general principle of not being able to completely foretell what is to come has been applied to other fields of study. Rorty (1989) has proposed the notion that all of language is contingent; besides truth claims of individual statements that can be adjudicated by reference to an external reality, the truthfulness, or even general utility, of language systems to describe reality is limited because we fail to grasp the relationship between what exists, and our representations of it. That there exists a reality independent of our mental states is unproblematic; however, to assert that there exists truthful descriptions of reality is impossible:

> Truth cannot be out there—cannot exist independently of the human mind—
> because sentences cannot so exist, or be out there. The world is out there, but
> descriptions of the world are not. Only descriptions of the world can be true or false.
> The world on its own—unaided by the describing activities of human beings—
> cannot. (p. 5)

Crucially for our argument, Rorty further asserts an anti-Platonic position that denies the possibility that some vital essence, some intrinsic nature of things or people exists, and that our linguistic or other representational resource can be said to be more or less congruent to such a nature. The consequences for such a line of reasoning influence both curriculum and instruction: it would be overly cynical as educators to take the extreme position that truth does not exist and hence anything goes; on the other hand, to assert that there is one 'right' answer to any problem is similarly mistaken. Using Rorty's scepticism of the existence of Platonic forms, we can assert for instance that there is no essentiality to the concept of 'cup'. This does not mean that cups don't exist, but that there is no such thing as an ideal 'cup' from which one can model a *prototype* cup against. While this may sound borderline outrageous, this situation is a lot more reasonable if we consider the creation of a hitherto novel artefact or knowledge structure. To give a particularly cogent example, consider a chef inventing a dish. Even if she follows well-known methods for working with ingredients, these items can vary in quality. One has to be attentive to the *contingencies* that emerge from the working with the material, and modify one's intentions as work proceeds, leading to a situation where one cannot ever be sure what it is that one is creating, until the creation is complete. The exception to this may be in the process of duplication of a known form, but even that process can be fraught with uncertainty, as I detail as follows.

The ideal of artefactual fabrication as a result of the impression of form onto passive matter has been termed hylomorphism. After the ancient Greek terms for abstract form (*morphe*) and matter (*hyle*), the hylomorphic metaphor for fabrication has taken hold as a result of the reductionist tendencies of STEM and industrial *manufacturing* techniques. Although efficient, such techniques obscure the process of *making*, especially the process of creation which it should be clear should be the appropriate metaphor for the education of students. The example of the early stone handaxe is particularly instructive in establishing this distinction. Archaeologists were caught in a double bind trying to explain the ubiquity of the handaxe across space and time: "If on the one hand, the form of the biface is tied to the body plan, then we can account for its constancy but not for the apparent intelligence of the design. If, on the other hand, we regard the biface as a product of a complex intelligence, then we can account for its design but not for the constancy of form" (Ingold, 2013, p. 37). This paradox is resolved if we do not suppose that the form of the handaxe must have existed, complete, in the minds of its makers, or that abstract representations could be used to communicate the idea and specifications of the handaxe for individuals to reproduce. The making of an handaxe requires skill, attention, and intention, but not a kind of prior intention: "the intentionality of the skilled practice inheres in the action itself, in its qualities of *attentiveness* and *response*, whether or not any prior intentions are affixed to it." (Ingold, 2013, p. 43, emphases added) The artefact of the handaxe is an *emergent* property of the process of knapping the stone. The skill of expert knappers is not reducible to either mental capacity or bodily biomechanics alone, but of a responsive interaction between the maker and the artefact. The stone is not some inert material whose role is merely to receive and be shaped by the maker; stored within its structure is a complex sum of tensions and compressions from its geological formation. The maker's role in the making is to become attentive to the potential, and respond to it in a manner that is harmonious to the nature of the rock such that their intent may be obtained. Making, then, is not an imposition of a complete design in a stepwise itinerary of assembly, but a mutual itineration between the maker and their artefact, "a passage along a path in which every step grows from the one before and into the one following, on an itinerary that always overshoots its destinations." (*ibid*., p. 45). Making is not a process of imposing one's will onto material, but rather one of 'surrendering' to the material and the 'following where it leads' (Deleuze & Guattari, 2004, in Ingold, 2013).

Congruent to these perspectives, Biesta (2016) proposes that educators need to be more mindful of the inherent risk of education. At the level of state organisations of public schools in many jurisdictions, there have been increasing pressures for schooling to adopt a more mechanistic control of the process. Increasingly, it is possible to recognise that classroom interactions are being scripted in ways to respond to the desire for control;

> But that does not mean that an educational technology, that is, a situation in which there is a perfect match between "input" and "output," is either possible or desirable. And the reason for this lies in the simple fact that if we take the risk out of education, there is a real chance that we take out education altogether.

For Biesta, the tension lies between what he terms as 'strong' versus 'weak' approaches to creativity and education. While it may be possible, even perhaps desirable, at the level of state organisations of public schooling to 'black box' the individual human interactions of schooling, treating school as a rational machine that guarantees particular 'outputs', the problem arises when teachers adopt aspects of this discourse and seek mechanistic ways of ensuring measured 'outputs', as will be likely in contexts with strong accountability measures in place.

The core problem for this paper arises at the interface of constructionism as a learning theory and its application into mainstream public schooling contexts. While education research continues presenting good results from constructionist activities such as those involved in makerspaces, there will be increasing pressures to incorporate these ‘proven’ activities as ideal learning in more contexts. The manner in which educators respond to these demands will determine the difference between an authentic educative experience, or, as has already been recognised within the constructionism community, epistemological dilution (Resnick, 2017). In makerspaces, observers have already noted the ‘keychain syndrome’ as contemporary version of such dilution (Blikstein, 2013). While constructionism as learning theory may explain how idealised learning contexts can lead to desirable outcomes, the complexity of schooling requires more companion theories that can explain the interaction. In this paper, we present a case study of how a teacher has managed to ‘shield’ his makerspace from the excesses of the strong demands of schooling, continuing to offer his students opportunities to be authentically creative despite pressures to deliver standardised experiences. In this regard, there appears pressure to amplify the gains of learning practices such as constructionism. We suggest through this case that there is a lack of models of how constructionism can ‘scale up’, and the industrial model that dominates relies on a faulty metaphor that narrows its possible gains.

Context and Method

We relate the case of Able Primary School (APS), an elementary school in Singapore. In most conventional measures of schooling, the school system has been performing well, leading league tables of international comparison. On other measures, the creative outputs of the nation appears low in relation to its investments, with the poor organisational culture suspected to be responsible for this lack of efficiency (The Economist Intelligence Unit & Asian Development Bank, 2014). In schools, this has taken the shape of strong accountability measures, and very uniform pedagogy across most schools that has an overwhelming emphasis on test preparation (Hogan et al., 2013). At grades 6, 10, and 12, particularly high stakes national examinations are conducted, which often set the tempo for classroom pedagogy: in order for students to do well in these mandated examinations, teachers conventionally reverse engineer desired outcomes and teach to the test. In this regard, APS is no different from schools in its immediate vicinity. The local discourse of ‘meritocracy’ is defined strictly in terms of achievement on these standardised tests, and as a result, despite school attempts to the contrary, parents exert a disciplining force to ensure compliance to the state regime of testing.

We made use of an autoethnographic method to provide the data for this paper. Second author Jemuel has been a teacher for nine years, six of which were at APS, teaching mathematics and physical education. As teacher overseeing special projects in his school, he had been responsible for the setting up of the school makerspace since 2017, which he has been managing since its inception. Jemuel, as initiator and manager of the makerspace, has been intimately familiar with the cultural patterns of behaviour in the space. In his interactions with teachers who request time in the space for their students, and in helping teachers manage expectations and plan lessons in the makerspace, he has extensive knowledge of the various purposes to which teachers put his makerspace to use. In other words, Jemuel is a well placed observer to comment on the typical goals and activities that users of the school makerspace.

Findings

We found that of teachers who made use of the makerspace for learning activities, there appeared to be two predominant purposes. Using the constructive metaphor, we identify these positions as making and manufacturing. Making, as we review above, is a fundamentally creative process, and in schooling, takes the form of constructionist practice that is open and attentive to student initiative. In contrast, manufacturing is a more teacher-centred approach, concerned with mass production of predetermined learning objectives. The tension between student and teacher centredness is hardly novel, with Dewey (1902/1963) arguing over a century ago against either end of the spectrum. What ought to be desirable is the artistic tension of attending to student needs for growth, but directing such growth in ways that are cognisant of significant cultural achievements. In this regard, we believe that good instruction is not reducible to a recipe or a programmable if-else sequence of instructions. More significantly, the interactions between school leadership and teachers forms a boundary across which the interpretations of terms offers either opportunity or threat to the educative value of the learning experience of the student.

In APS, as with many schools, the makerspace constitutes a significant investment. Besides the financial cost, scheduling time for making activities represents an opportunity cost. While classroom pedagogy is predominantly concerned with test preparation, central planners at the Ministry of Education (MOE) have already recognised the need for structures that deliberately open up space for schools to have the latitude to experiment. Called the Applied Learning Program (ALP), this program was designed to be within the non-assessable curriculum grey areas, occupying the liminal space between high stakes curriculum and *de facto* irrelevant learning objectives. In many other schools, ALP had taken the form of enrichment activities, usually undertaken by private education vendors, with staff not formally trained in instructional methods, and with a curriculum influenced by contemporary topics such as programming, design and creative problem solving, or robotics. The ALP was marketed among schools as an opportunity for students to get involved in 'engaging' activities that put meaning to the abstract concepts encountered in classrooms. Many of these private education vendors conduct ALP lessons in lecture mode, with practical lessons done in a recipe following method. In contrast, Able Primary decided to have the makerspace as the centrepiece of their ALP offering, got their own teachers to manage the curriculum for their ALP, and have their own teachers instruct their students. While there was reliance on vendors to provide lessons on specific skills the teachers believed they lacked, the overall organisation was primarily school led.

APS' program was conceived on two levels: on an entry level, maker education was conceived to give as many students as possible access to the basic techniques of work in the makerspace. On a more advanced level, maker culture signified for students particular mindset shifts about dependency on teachers for assistance in developing their design intentionalities. Ultimately, students will require supervision and guidance on the use of particular technologies, but the development of a maker culture was valued for its ability to have students develop a particular phronetic independence (Flyvbjerg, Landman, & Schram, 2012). Simply, the makerspace team felt it was important for students to develop projects of their own desire. This was felt as an antidote to the mainstream assessable curriculum where students had little say in what it is they were supposed to learn. With maker culture, instruction shifted from *telling* students what it is that they must make, to *coaching* students on how they could achieve what they wanted, with advice on the appropriateness of particular intentions as needed.

The makerspace and the ALP have been interesting boundary objects (Bowker & Leigh Star, 1999; Leigh Star, 2010) used by both teachers and administrators alike to communicate across very different social worlds. As making as a learning activity was not strictly defined under the rubrics of a standardised examination, teachers could interpret making in ways which met their own interpretations of what making meant. For instance, both science and art teachers could meaningfully design a project to satisfy their individual disciplinary requirements. At the same time, school administrators had the flexibility of reporting the ALP program as being successful as conventional measures could be creatively adapted to demonstrate effectiveness. To be sure, the interpretive flexibility afforded by the ALP allowed teachers to perform a wide range of activities, including some that may not represent the best uses of makerspaces. For instance, teachers who are used to a product-centric manner of thinking about learning can and do focus on making sure their students have a completed project, often shortchanging their students of the thinking necessary in creative problem solving. Somewhat depressingly, these teachers can and do interpret their instruction as 'constructivist' in nature; such identification is made more prominent when materials are organised to build artefacts.

Discussion and Conclusion

Making and manufacturing are two seemingly incompatible interpretations of the possible processes that can take place in makerspaces, not only on the tangible level, but also at the level of learning and instruction. Whether or not teachers make or manufacture in the makerspace is not necessarily a function of the learning design, and certainly not a function of the learning space. Teachers can make subtle shifts in discourse to privilege one aspect over the other. Even when ostensibly engaged in creative problem solving, teachers who have a manufacturing mindset can easily sabotage the potential learning gains by offering models from which students may duplicate from. Constructionism as learning theory may inadequately describe and explain the complex interactions that exist in schooling environments where, for instance, pressure from a habitual desire to demonstrate learning gains narrowly defined may cause teachers to adopt shifts in pedagogy toward a manufacturing orientation. The concept of a boundary object is a useful construct to theorise the possibilities and risks that are posed by makerspaces, the curriculum goals surrounding it, and the learning theories that undergird its perceived utility. The interpretive flexibility of constructionism—as making or manufacturing—can offer its users a means of escape from the overly reductive and restrictive classifications. Instead of a mind-body dichotomy, privileging cognitive process over 'manual labour', constructionism allows making to become a valuable activity in its own right. Nonetheless, researchers should also be wary that such flexibility may also serve to impede a more humanistic understanding of construction-as-making, with its contingent, proliferating sense of possibilities. We return ultimately to Rorty's understanding of the contingent nature of meanings: while theory demands the fixedness of definition, of firm boundaries, between male and female, sacred and profane, making and manufacturing, everywhere we look, we can always find ways to upset established orders, to become more aware of the contingent nature of things-in-themselves. In this sense, makerspaces and other sites can continue to innovate and offer interesting pedagogies until such time that the bureaucratic urge to standardise and define overtakes their ability to resist definition. In the light of international efforts to demarcate and control, here's to more ambiguity and contingency!

References

Biesta, G. (2016). *The beautiful risk of education*. Abingdon, Oxon: Routledge.

Blikstein, P. (2013). Digital fabrication and "making" in education: The democratization of invention. In J. Walter-Herrmann & C. Büching (Eds.), *FabLabs: Of Machines, Makers and Inventors* (Vol. 4, pp. 1–21). Bielefeld: Transcript Publishers.

Bowker, G. C., & Leigh Star, S. (1999). *Sorting things out: Classification and Its consequences*. Cambridge, MA: MIT Press.

Deleuze, G., & Guattari, F. (2004). *A Thousand Plateaus* (B. Massumi, trans.). London: Continuum.

Dewey, J. (1963). *The child and the curriculum*. Chicago: The University of Chicago Press. (Original work published 1902)

Flyvbjerg, B., Landman, T., & Schram, S. (Eds.). (2012). *Real social science: Applied phronesis*. Cambridge, UK: Cambridge University Press.

Hogan, D., Chan, M., Rahim, R., Kwek, D., Maung Aye, K., Loo, S. C., … Luo, W. (2013). Assessment and the logic of instructional practice in Secondary 3 English and mathematics classrooms in Singapore. *Review of Education*, *1*(1), 57–106. https://doi.org/10.1002/rev3.3002

Ingold, T. (2013). *Making: Anthropology, archaeology, art and architecture*. Abingdon, Oxon: Routledge.

Leigh Star, S. (2010). This is Not a Boundary Object: Reflections on the Origin of a Concept. *Science, Technology & Human Values*, *35*(5), 601–617. https://doi.org/10.1177/0162243910377624

Resnick, M. (2017). The Seeds That Seymour Sowed. *International Journal of Child-Computer*. Retrieved from http://web.media.mit.edu/~mres/papers/IJCCI-seeds-seymour-sowed.pdf

Rorty, R. (1989). *Contingency, irony, and solidarity*. Cambridge, UK: Cambridge University Press.

Roth, W.-M. (2015). Becoming aware: towards a post-constructivist theory of learning. *Learning: Research and Practice*, *1*(1), 38–50. https://doi.org/10.1080/23735082.2015.994256

The Economist Intelligence Unit, & Asian Development Bank. (2014). *Creative productivity index: Analysing creativity and innovation in Asia*. Asian Development Bank.

Constructionism and AI: A history and possible futures

Ken Kahn, *toontalk@gmail.com,*
Dept of Education, University of Oxford, Oxford, UK

Niall Winters, *niall.winters@education.ox.ac.uk,*
Dept of Education, University of Oxford, Oxford, UK

Abstract

Constructionism, long before it had a name, was intimately tied to the field of Artificial Intelligence. Very soon after the birth of Logo, Seymour Papert set up the Logo Group as part of the MIT AI Lab. Logo was based upon Lisp, the first prominent AI programming language. Many early Logo activities involved natural language processing, robotics, artificial game players, and generating poetry, art, and music.

In the 1970s researchers explored enhancements to Logo to support AI programming by children. In the 1980s the Prolog community, inspired by the Logo community, began exploring how to adapt logic programming for use by school children. While there has been over forty years of active AI research in creating intelligent tutoring systems, there was little AI-flavoured constructionism after the 1980s until about 2017 when suddenly a great deal of activity started.

Among those activities were attempts to enhance Snap! with new blocks for speech synthesis, speech recognition, image recognition, use of pre-trained deep learning models, and word embeddings, as well as blocks to enable learners to create and train deep neural networks.

We close with speculations about possible futures for AI and constructionism.

Keywords

Artificial intelligence, constructionism, deep neural networks, machine learning, Snap!, logic programming, Logo

AI was there from the beginning

The core constructionism idea is that learning “happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity” [Papert & Harel 1991]. To “consciously engage” one needs to bring to bear concepts while introspecting. Computational concepts are an important well-known part of this within constructionism but here we focus on the learner’s ideas about how they learn, represent things, solve problems, and create. AI-oriented constructionist projects may help learners acquire deeper and more effective ways of learning and creating.

These ideas influenced the design of the Logo language [Solomon et al 2020] and early project ideas. The paper “Twenty Things to Do with a Computer” [Papert and Solomon 1971] suggests on the first page that artificial intelligence should deeply affect our thinking about “computers in education”. Several of the twenty project suggestions were AI projects (as it was conceived in the 1970s) such as programming robots with sensors and effectors, composing music, and generating poetry. These were soon followed by projects such as generating grammatical sentences and programming AIs to play simple games like *Tic Tac Toe* and *Nim*.

In “Three Interactions between AI and Education” [Kahn 1977] AI plays three roles: (1) students use AI programming tools in their projects, (2) students create artefacts by interacting with AI programs, and (3) and the use of computational theories of intelligence and learning in the design of learning activities. In the 1970s research prototypes were built upon Logo to support student projects involving natural languages and semantic networks. LLogo [Goldstein et al 1975], an implementation of Logo in Lisp, was particularly well-suited for this.

[Kahn 1977] lists a few reasons why children should create and interact with AI programs:

1. “Children are encouraged to think explicitly about how they solve problems. Hopefully the children will thereby improve their ability to describe and understand their own thoughts”
2. “The problem domain to which the AI programs are applied is learned, and in a new and perhaps better way”
3. “If children are to program, then AI can an interesting open ended problem domain for that programming”
4. “The children will learn about AI which is a subject … that is as important as spelling or history”.

Regarding the first point consider the role self-reflection plays in learning according to Marvin Minsky [Minsky 2019; Minsky 2009] (original emphasis):

- An adequate theory of learning should also cover the “reflective skills” that people use to recognize exceptions to generalizations, to eliminate tactics that waste too much time, and more generally, to make longer-range plans and form broader perspectives.
- But I’m convinced that these “self-reflective” processes [the methods that people use for *thinking about what we’ve been thinking about*] are the principal ones that people use for *developing new ways to think*.

Efforts in the 1980s to bring AI programming into schools

The articles in the book *New Horizons in Educational Computing* [Yazdani 1984] in the Ellis Horwood Series on Artificial Intelligence describe many efforts to make AI programming tools accessible to children. The logic programming language Prolog inspired two very different approaches: (1) providing children with a simple syntax for using a powerful deductive engine and (2) giving children a powerful symbolic programming language that provides support for building semantic networks and doing backtracking searches. Prolog programs have declarative and procedural readings and different research groups focussed on each of these different aspects.

“Logic as a computer language for children” [Kowalski 1984] describes the declarative approach. Students can build deductive, typically symbolic, databases. Ten-year-old children constructed semantic networks using a front end to Prolog with a very simple syntax. [Sergot 1984] instead built a meta-interpreter for Prolog for use by children. While much slower, it was able to provide explanations for any deductions it made. [Nichol and Dean 1984] describes how students could use Prolog to explore history.

An example of using Prolog as a powerful procedural tool is described in [Kahn 1984]. A grammar kit was built upon LM-Prolog that enabled students to build grammars to both parse and generate sentences. It relied upon a “backtracking turtle” that erased its trail when backtracking. This visualisation helped students understand how their programs and queries worked and to fix bugs.

POP-11 [Sloman 1984] is an AI language similar to Lisp that supports semantic databases. Development of POP-11 started in 1974 and was heavily influenced by the Logo efforts at MIT and the University of Edinburgh. The goals of Pop-11 were to enable students to, instead of “making toy cranes or toy aeroplanes, or dolls”, to make “toy minds”. The POP-11 research team wanted “people to experience themselves as computation”. Accessible AI programming tools were developed with these goals in mind.

POP-11 later incorporated Prolog into their system (renaming it Poplog) after seeing the successes others were having introducing Prolog to children. From 1983 to 1998 Poplog was an expensive commercial product but after it became free in 1998 it was once again used by school students for AI programming. Examples of Poplog projects include puzzle solving, chatbots, analogical reasoning, and planning [Sloman 2012].

Everything changes in 2017

The AI research community, the general public, and educators view of AI began changing in 2012 as neural networks began attaining impressive performance in image recognition, machine translation, transcription of speech, game playing (Atari games, Go, and chess), and natural language processing. Deep neural networks are what most people think of today when they hear AI. Deep learning combined with big data and special hardware for doing tensor operations now dominates the field.

2017 saw the appearance of several excellent machine learning tools and resources for children. They include

1. April 2017 Dale Lane created *The Machine Learning for Kids* website. Learners can create neural networks for images, text, or numbers, train them, and use them in Scratch programs. The site includes a long list of well-designed project suggestions [Lane 2020].
2. May 2017 Stephen Wolfram added a machine learning chapter to the *Mathematica* online programming textbook. He published a blog describing the AI capabilities of the Wolfram Language. He suggests a range of projects suitable for middle school students [Wolfram 2017].
3. May 2017 Google announced its first AIY project where students can build standalone artefacts capable of speech synthesis and recognition. This was followed by a similar kit for image recognition [Google 2020].
4. September 2017 Kahn and Winters released the eCraft2Learn project’s Snap! blocks that access web services for speech synthesis, speech recognition, and image recognition [Kahn & Winters 2017]. This was followed by Snap! blocks for using pre-trained TensorFlow models, and word embeddings, [Kahn & Winters 2018]. Subsequently blocks for creating, training, and using neural nets were added a year later [Kahn et al 2020].
5. October 2017 Google released its Teachable Machine which is a web page where one can train a model to classify three kinds of images [Google 2017]. Unlike the other 2017 machine learning tools this one does not have a programmer interface. While it is a good introduction to an aspect of machine learning it lacks the open-ended nature of most constructionist tools. It did, however, directly inspire similar projects in Snap! using the eCraft2Learn blocks.

This explosion of activity has continued. AI programming by children projects have since started at many universities. The AI for K-12 Initiative is growing and has organised two international workshops [Touretzky 2020].

A closer look at one AI programming for children project

eCraft2learn was a European research project from 2016 to 2018. It focussed on bringing ideas and technology from the Maker Movement to STEAM education. As part of the project the University of Oxford developed Snap! blocks for AI programming. They have continued to develop the blocks, guides, project suggestions, and sample projects after the eCraft2Learn project finished. They recently added blocks for defining neural networks, training them, and using them for predication and classification [Kahn et al 2020].

eCraft2Learn partners in Greece and Finland field tested the blocks and learning resources. Workshops using the blocks have been run in Sri Lanka, Singapore, and Indonesia. Collaborators at the Beijing Normal University have begun efforts to test these resources with both school children and non-CS major university students.

A unique aspect of this set of programming tools and resources is that it is completely web-based and largely server-free. There is no need to register or log in. By relying upon Google's TensorFlow.js [Smilkov 2019] the machine learning occurs in the user's device (laptop, tablet, or phone). It runs at competitive speeds to other technologies since it can accelerate computations by using the device's graphical processing unit (GPU). Nothing needs to be installed. No data needs to be transmitted to servers thereby enhancing privacy. It can run solely upon the local file system when Internet connections are not available or reliable. The Snap! blocks that use pre-trained neural networks and those for creating and training models all rely upon TensorFlow.js to perform all computations locally. Apps using the blocks can be very responsive since they avoid any latency due to communication overhead with servers.

A response to a criticism

[Ames 2018] is a critique of constructionism. She apparently rejects good-old fashioned symbolic AI and a computational view of the mind. She seems to view learning as primarily a social activity. Consequently, it is not surprising she is so critical of constructionism. Most relevant to this paper she wrote "Papert described cybernetics rapturously in Mindstorms as a potent framework for understanding learning by thinking about brains like we think about computers. This was not unusual: at the time, there was widespread belief that human brains were particularly sophisticated computers; Minsky, in fact, famously called them 'meat machines.' While the problems with this equivalence eventually contributed to the collapse of cybernetics and to the 'artificial intelligence winter' during which the field stagnated for decades, fragments of the belief persisted."

One can disagree with this view of AI history (and we do). The important question regarding the value of introducing AI programming to children is whether there really are "problems with this equivalence". Indeed, if one rejects the idea that thoughts are computations and brains are computational machines then one might question the value of helping children to construct AI programs. If thoughts are something other than computations (contrary to most research in cognitive science and neuroscience), then students may acquire a false mechanical concept of their thought processes. But what models of thinking might children acquire instead? Papert considered the alternative to be harmful "Pop-Ed theories" [Papert 1971]. These are the theories that children do acquire about thinking. One is the "blank-mind" theory that what one should do is make your mind a blank and wait for an idea to come. Another Papert named "getting-it" where one expects understanding to come in a flash, the opposite of the Piagetian process of accommodation. Another common theory that interferes with learning Papert called "faculty". For example, some people have the faculty of mathematics and some don't. Failure is because the problem is too hard to address given one's innate faculties. Our hope is that by engaging in AI programming students might acquire models of their thinking and learning that, while crude, can help them become better thinkers and learners.

Possible Futures

The early history of AI and constructionism was focussed on symbolic AI ("good-old-fashion AI"). More recently the focus has shifted greatly towards machine learning with neural networks. This is good since it provides students with very powerful capabilities enabling them to creatively build very capable artefacts. Students now have the tools to build apps that have the potential to make a positive difference in their community or the entire world. [Tissenbaum, Sheldon & Abelson 2019]. Students can acquire a first-hand experience with a technology that is rapidly changing the world. It is a technology they can use to become scientists, engineers, business executives, artists, musicians, doctors, or simply engaged citizens.

But something has been lost. Aaron Sloman wrote of students making "toy minds". Minsky and Papert spoke of acquiring concepts to make one better at reflection and learning. But with neural nets students are now making "toy brains". They are not programming at the level of concepts, symbols, and plans, but instead are relying upon crude approximations to how neurons work in brains.

An ideal future would involve a synthesis of these two schools of AI. Students could not only choose between computational building blocks that are symbolic or neural but use hybrids (a current topic of AI research). When symbolic-neural computation is well-enough understood we should provide accessible hybrid building blocks to learners. Studies are needed to explore more closely whether, indeed, learners' hands-on exposure to AI and machine learning concepts has any effect on their self-reflection and learning skills. And whether the effects are evident regardless of whether the AI is symbolic or neural. Another open question is what the minimum level of intellectual ability a child needs to benefit from AI programming.

AI is also being used in non-constructionist ways in schools. Automatic grading systems are proliferating. Learning analytics are applied to data generated from online courses. Research on intelligent tutoring systems continues to improve.

Currently these uses of AI in teaching work best when the software knows what the students were assigned to do. Maybe someday AI systems will be able to help guide students doing their own projects. Maybe AI systems will be able to provide suggestions and nudges that help students do projects without providing so much guidance that students are deprived of the joy and passion of self-directed creation of artefacts.

References

Ames, M. G. (2018). Hackers, Computers, and Cooperation: A Critical History of Logo and Constructionist Learning. Proceedings of the ACM on Human-Computer Interaction, 2(CSCW), 18.

Goldstein, I., Lieberman, H., Bochner, H., & Miller, M. (1975). LLOGO: An implementation of LOGO in LISP. MIT Artificial Intelligence Lab Memo Number 307a.

DGoogle (2020) Do-it-yourself artificial intelligence. https://aiyprojects.withgoogle.com/

Google (2017). Teachable Machine https://experiments.withgoogle.com/ai/teachable-machine

Kahn, K. (1977) Three interactions between AI and education. In Elcock, E. and Michie, D., editors, *Machine Intelligence 8: Machine Representations of Knowledge*. Ellis Horwood Ltd. and John Wylie & Sons.

Kahn, K. (1984). A Grammar Kit in Prolog. In Yazdani, M., editor, *New Horizons in Educational Computing*. Halsted Press.

Kahn, K. & Winters, N. (2017) Child-friendly Programming Interfaces to AI Cloud Services, *Proceedings of the EC-TEL 2017 Conference*, Tallinn, Estonia, September.

Kahn, K. & Winters, N. (2018). AI Programming by Children, *Constructionism Conference*, Vilnius, Lithuania, August.

Kahn, K., Lu, Y., Zhang, J., Winters, N. and Gao, M. (2020). Deep Learning Programming by All. *Proceedings of Constructionism Conference 2020*, Dublin, Ireland.

Kowalski, R. A. (1984). Logic as a computer language for children. In Yazdani, M., editor, *New Horizons in Educational Computing*. Halsted Press.

Lane, D. (2020). Machine Learning for Kids. https://machinelearningforkids.co.uk/#!/about

Minsky, M. (2009). OLPC Memo 5: Education and Psychology. https://web.media.mit.edu/~minsky/OLPC-5.html

Minsky, M. (2019). *Inventive Minds: Marvin Minsky on Education*. Edited by Solomon, C. and Xiao, X., MIT Press.

Nichol, J. & Dean, J. Pupils (1984). Computers, and History. In Yazdani, M., editor, *New Horizons in Educational Computing*. Halsted Press.

Papert, S. (1971). Teaching Children Thinking. MIT Artificial Intelligence Lab Memo Number 247.

Papert, S., & Solomon, C. (1971). Twenty Things to Do with a Computer. MIT Artificial Intelligence Lab Memo Number 248.

Papert, S. & Harel, I. (1991) *Constructionism*. Ablex Publishing Corporation.

Sergot, M. (1984). A Query-the-user Facility for Logic Programming. In Yazdani, M., editor, *New Horizons in Educational Computing*. Halsted Press.

Sloman, A. (1984). Experiencing Computation: A Tribute to Max Clowes. In Yazdani, M., editor, *New Horizons in Educational Computing*. Halsted Press.

Sloman, A. (2012). Examples of General and AI Programming Teaching Materials Illustrated using Poplog/Pop-11. http://www.cs.bham.ac.uk/research/projects/poplog/examples/

Smilkov, D., Thorat, N. Assogba, Y., Yuan, A., Kreeger, N., Yu, P., Zhang, K. (2019) "TensorFlow.js: Machine Learning for the Web and Beyond." *arXiv preprint arXiv:1901.05350*.

Solomon, C., Harvey, B., Kahn, K., Lieberman, H., Miller, M., Minsky, M., Papert, A. and Silverman, B. (2020). History of Logo. *Proc. ACM Program. Lang. 4, HOPL*, Article 79 (June 2020), 66 pages. https://doi.org/10.1145/3386329

Tissenbaum, M., Sheldon, J., & Abelson, H. (2019). From Computational Thinking to Computational Action. *Communications of the ACM*, 62(3), 34-36.

Touretzky, D. (2020). Ai4k12 wiki. https://github.com/touretzkyds/ai4k12/wiki

Wolfram, S. (2017) Machine Learning for Middle Schoolers. https://writings.stephenwolfram.com/2017/05/machine-learning-for-middle-schoolers/

Yazdani, M. (1984). Editor, *New Horizons in Educational Computing*. Halsted Press.

Constructionism and Creative Learning: supporting the design of technology-based learning activities

Flavio Nicastro, *flavio.nicastro@ic.unicamp.br*
Institute of Computing, University of Campinas, UNICAMP, Brazil

Maria Cecília Calani Baranauskas, *cecilia@ic.unicamp.br*
Institute of Computing, University of Campinas, UNICAMP, Brazil

Ricardo da Silva Torres, *ricardo.torres@ntnu.no*
Department of ICT and Natural Sciences, Faculty of Information Technology and Electrical Engineering, NTNU - Norwegian University of Science and Technology, Ålesund, Norway.

Abstract

The implementation of Educational Technology in school spaces brings the need of continuing education of teachers in contemporary approaches to teaching and learning, such as those proposed in Constructionism and Creative Learning. The proposal of tools to support the design and conduction of activities based on these approaches, are hardly discussed in the mainstream literature. In this work, we present a case study evaluating a framework conceived to support the design of educational activities based on the principles of Constructionism and Creative Learning, in which hands-on/heads-in activities underlie. The objectives of this case study were: to analyse the perceptions of educators about the framework; to identify whether the workshops conducted with educators, introducing the framework, enabled them to design and conduct an activity based on the principles of constructionism and creative learning; and, to understand the level of satisfaction of children when participating in an activity designed with the support of the framework. The case study involved 29 educators and 161 students (5-15 years old). The results revealed that the evaluated framework was appreciated by the educators and facilitated their design of activities. Among the students, they demonstrated to be very motivated in participating in the activity based on the framework, and the majority of them stated that this kind of activity facilitates the understanding of curricular concepts and brings more engagement and satisfaction to all.

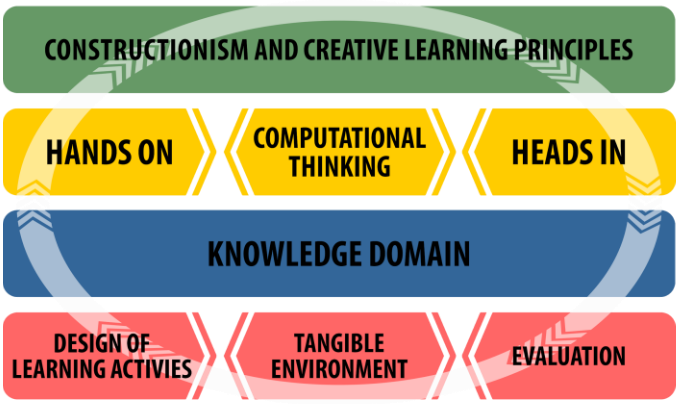


Figure 1. A framework for designing educational environments based on Constructionism and Creative Learning. (Nicastro et al., 2019).

Keywords

Constructionism; Creative Learning; Computational Thinking; Tangible Artifacts; Tangible User Interface; Educational Methods.

Introduction

The Common National Curriculum Base (Brazil, 2017) proposes the use of Educational Technology in school spaces. This makes the continuing learning process of educators necessary and timely, preparing them with contemporary approaches to teaching and learning, such as those proposed by the Constructionism and Creative Learning framework (Nicastro et al., 2019).

Based on "learning by doing," Seymour Papert proposed the concept of Constructionism, arguing that when one is consciously involved in the construction of something, the construction of knowledge takes place more efficiently (Papert and Harel, 1991). Currently, the world is experiencing a growth in hands-on activities and they are becoming more present in educational contexts (Resnick and Rosenbaum, 2013). Hands-on activities and tasks involving computer programming are an excellent way of achieving construction of knowledge, as proposed by Papert. Consequently, more and more educators have shown interest in adopting these approaches. However, they hardly find tools to assist them in designing educational activities based on the principles of constructionism.

In this paper, we present a case study with the application of a framework (Figure 1) conceived to assist educators in designing educational scenarios based on the principles of constructionism and creative learning. This framework enables the educator to design environments with hands-on/heads-in activities based on tangible technological artifacts and programming tasks (Nicastro et al., 2019). This approach not only favors knowledge construction in a more meaningful way (Papert, 1980), but also promotes the development of creativity, student protagonism, teamwork and skills associated with computational thinking (Papert, 1993; Resnick and Robinson, 2017).

The case study was composed of five workshops involving 29 educators who worked with two segments of Basic Education at Vinhedo-SP/Brazil. The main goal of the case study was: to analyze the educators' perceptions of the framework; to identify whether the activities conducted with the educators, based on the framework, enabled them to design and conduct an activity based on the principles of constructionism and creative learning, working with a curricular subject; and, to understand the level of the children's satisfaction when participating in an activity. At the beginning of the process, the educators were not confident in their ability to design an activity based on something new, such as the technologies and the concepts proposed. As the workshops unfolded, the more hands-on the educators became, the more they reflected on the presented concepts. They also gained confidence and designed very interesting activities. These activities were put into practice with 161 students within three days of activities: the first day with 54 children from kindergarten (5 years old), the second with 41 students from 7th grade (12-13 years old), and the last with 67 students from 9th grade (14-15 years old).

At the end of each workshop, the educators' affective states were assessed by the Self-Assessment Manikin SAM instrument (Bradley and Lang, 1994), apart from a form containing questions about the relation with the concepts presented and the applicability in their curricular practices. A similar method was used with the students. The results show that the vast majority of the educators were very motivated to participate in the workshops based on the evaluated framework, and indicated that the framework-based learning helped them gain confidence to design an activity based on the principles of constructionism through the use of technology and hands-on activities that provide reflection. Furthermore, our findings show that the students felt very motivated while participating in an activity based on the framework, and most of them stated that this kind of activity facilitates the understanding of curricular concepts and brings more engagement and satisfaction to all.

The evaluated framework and its bases

In the following, we synthesize the elements of the evaluated framework (Nicastro et al., 2019). This framework is intended to assist the design of a learning environment supported by tangible artifacts, through hands-on practices, while also encouraging reflection, developing the abilities associated with computational thinking, creativity, teamwork, and child protagonism.

Constructionism and Creative Learning Principles

Constructionism is knowledge construction based on the realization of a concrete action, which results in a tangible product that concerns the person who constructs it. Based on Papert's constructionism, Mitchel Resnick and Ken Robinson proposed the creative learning approach (Resnick and Robinson, 2017). As stated by them, learning occurs through a creative process when one creates something that is meaningful to them by playing, in collaboration with peers, thus allowing them to grow as a creative thinker. Creative learning is based on four pillars, presented as "The Four P's of Creative Learning": Projects, Passion, Peers and Play. Apart from the four P's, creative learning inherits Papert's idea of Low Floor and High Ceiling. Resnick adds Wide Walls to these points, in the sense that the activity should provide multiple pathways from low floor to high ceiling. Hence, people with diverse profiles and backgrounds will be most engaged, and learn the most, because they will work on projects that are personally meaningful to them (Resnick and Silverman, 2005; Resnick, 2016). Complementing this approach, Resnick and Robinson present the spiral of creative learning, displaying the learning process as a continuous endeavour of *imagine*, *create*, *play*, *share*, *reflect*, *imagine*, and so on.

Hands-on

Papert advocates the need to learn through creative processes, as well as discovering knowledge rather than receiving it passively. Hands-on provides this different perspective in the learning process, giving learners the opportunity to have control over their own knowledge, rather than being passive receivers. Hands-on can be understood as "getting involved in building something." It does not necessarily have to do with a physical object (a chair, for example); it could be something tangible (digital, conceptual) that has shape and meaning, such as a chair, but also a poem, a song, a computer program, or even a way of thinking.

Heads-in

Not only does hands-on guarantee learning, but also the process of reflection that it provides is of huge importance for the construction of knowledge. Seymour Papert (Papert, 1980, 1993; Papert and Harel, 1991), Edith K. Ackermann (Ackermann 1995, 1996, 2001, 2004) and Mitchel Resnick (Resnick and Silverman, 2005; Resnick and Rosenbaum, 2013; Resnick and Robinson, 2017); clearly bring this concept into their works. Edith reveals that hands-on activities remain essentially undirected and non-controllable, blind, and meaningless, when not fully reflected upon. In line with Edith's work, heads-in requires us to evaluate what we have done in light of what we would like to achieve.

Computational Thinking

Computational thinking is not a new concept. In the 1970s, Seymour Papert was a pioneer of the LOGO language and studies concerning the learning processes of children mediated by the use of programming language and tangible artifacts (Papert, 1980). Since then, several works have been done on computational thinking. Activities which aim at the development of skills associated with computational thinking are an excellent way to practice both hands-on and heads-in. Hands-on: the construction of a computer program; Heads-in: thinking about commands for the desired action and debugging.

Knowledge Domain

Knowledge Domain relates to any kind of knowledge, skill, or competence that educators wish to develop, for example, working with a curricular concept.

Design of Learning Activities

In the process of designing a learning environment, the educator should take all the elements of the framework into account and define the ways in which each one will be addressed. In order to assist educators in this task, the framework is accompanied with a workflow that presents three moments: *thinking*, *tinkering*, and *reflecting*. The last topic in this section presents this workflow.

Tangible Environment

In his 1980 Mindstorms book (Papert, 1980), Seymour Papert already advocated the idea that children learn when they manipulate objects. Studies using tangible interaction to build computer programs began in the 1970s with the creation of physical programs for the Logo language. Tangible User Interfaces (TUIs) are one way of exploring activities based on constructionism principles. TUIs are user interfaces that employ physical objects, instruments, surfaces, and spaces; they augment the real physical world by joining digital information with everyday physical objects and environments (Ullmer and Ishii, 1997; Ishii and Ullmer, 1997). An environment permeated with tangible artifacts is essential for the type of activity that will be designed through the framework.

Evaluation

This element of the framework works on the importance of defining how the learning environment will be assessed. It comprises which items will be evaluated and how; and which evaluation dimensions to use: development of skills associated with computational thinking, development of creativity, protagonism and teamwork of the learners and, in general, evaluation of the effectiveness of the designed environment. Besides this, it is important to evaluate the affective states of the participants during their participation in the activities. While assessing, it should be taken into consideration whether these points are being provided by the designed environment.

A Workflow for Designing Educational Practices

Accompanying the framework, a workflow to guide the design of an educational environment is proposed (Figure 2). This workflow is composed of three moments: *Thinking*, *Tinkering, and Reflecting (TTR)*. This workflow can be used to assist in the designing phase of a learning environment and also during the conduction of activities with students (Nicastro et al., 2019).

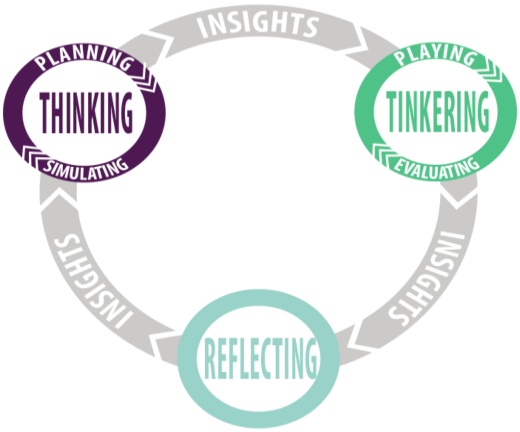


Figure 2. Workflow to put the framework into practice (Nicastro et al., 2019).

Case Study

Based on the presented theoretical framework, the continuing education of the educators was elaborated with the objective of enabling them to design and conduct educational activities based on the Constructionism and Creative Learning Framework. During the process, the elements of the framework were presented, and the educators could experiment with the concepts and reflect on relations with their curricular practices. The activities were thought to provide several moments of interaction and sharing of ideas, with hands-on/heads-in activities, using low cost technologies (leds, batteries and small motors), manipulation of tangible technological artifacts and involving programming tasks. The main idea behind the activities' design was to cover all the elements of the framework and provide the three moments, Thinking, Tinkering and Reflecting (TTR). Table 1 shows the workshops organization as well as which element was covered in each workshop, while Table 2 shows how TTR moments were provided.

Table 1. Organization of conducted workshops.

Workshops Organization		
Workshop	**Goal**	**Element Worked On**
1	To present Constructionism and Creative Learning concepts through a hands-on/heads-in activity.	Constructionism, Creative Learning, Hands-on, Heads-in and Tangible Environment
2	To present the concept of Computational Thinking and Tangible Artifacts through a hands-on/heads-in activity involving programming tasks.	Previous elements plus Computational Thinking
3	To present the design process of a Creative Learning activity, through a hands-on/heads-in practice, while also designing an activity considering local reality, and working on a curricular subject.	Previous elements plus Design of Learning Activities, Evaluation and Knowledge Domain
4	To simulate the designed activity, verifying the steps and seeking points of improvement and attention.	All the Elements
5	To conduct the activity with Students	All the Elements

Participants

Twenty-nine educators from the municipal school system participated in the workshops, including teachers from different backgrounds, and school coordinators. In the workshops with students 161 children participated in three days of activities: the first day with 54 children from kindergarten (5 years old), the second day with 41 students from 7th grade (12-13 years old) and the last with 67 students from 9th grade (14-15 years old).

The instantiated framework

In order to put the framework into practice, the elements were instantiated as follows:

Hands-on/Heads-in: activities based on those provided by the Brazilian Network of Creative Learning (www.aprendizagemcriativa.org - As of April 2020), with some adaptations to suit the local context and to be consistent with the used framework.

Computational Thinking: activities involving unplugged computing (Bell et al., 2009; csunplugged.org), programming through Scratch Environment (Resnick et al., 2009) and tangible programming (Horn and Jacob, 2007).

Tangible Environment: we used three kinds of tangible artifacts:

- Cubetto: a wooden robot that moves on a carpet through commands placed on a programming board (Marinus et al., 2018).
- TaPrEC: A Tangible programming environment for children, where wooden blocks fit into each other, like puzzle pieces, in order to construct a Scratch Program (Carbajal and Baranauskas, 2015).
- Micro:bit: a small computer board that connects to Scratch and extends programming functionality to the physical world through motion and temperature sensors, buttons and a LED panel (microbit.org - As of April 2020). Through Micro:bit it is possible to make the program tangible by bringing manipulation into the program, for example by making a sword with recycled materials that interacts with a character on the screen.

Table 2. Providing TTR Moments.

How was TTR provided?			
Workshop	**Thinking**	**Tinkering**	**Reflecting (Key Questions)**
1	Debating the principles of constructionism and creative learning.	Construction of a Village using low-cost technological materials and unstructured material.	- How do hands-on activities provide the approach to creative learning? - How were the 4Ps presented? - Has the learning spiral emerged in the process?
2	Discussing skills associated with CT and the role of tangible artifacts in learning.	Exploring stations where CT skills and tangible programming can be experienced.	- How can these activities be materialized in the educational context and associated with the curriculum? - How do tangible artifacts contribute to this process and how can they be incorporated?
3	Debating the importance of a proper design of a learning activity.	Designing an activity, integrating hands-on, CT, heads-in and Tangible Artifacts.	- What are the challenges for both designing an activity and putting it into practice? - How do you meet students' interests while meeting curriculum goals? - How can this activity be evaluated? - How do I know whether I was successful in what was proposed?
4	Reviewing the designed activity and analyzing improvement points.	Simulating the activity.	- Are all elements of the framework present? - Has TTR been provided?
5 by Educators	The design process of the activity itself	Conducting the activity with the children, following closely and making on the fly adjustments when needed.	- Which point raised attention? - What worked? - What went wrong? - Were the goals achieved? - What are the proposals to refine the activity?
5 by Students	Debating the subject to be worked on and elaborating the first idea to be implemented.	Getting their hands dirty with the activity, having the freedom to explore their own ways and taking into consideration that error is part of the learning process.	- How did they feel about doing something different, with more freedom of exploration? -Does this type of activity bring greater engagement and understanding of the curriculum themes?

Evaluation: we used the Self-Assessment Manikin (SAM) instrument (Bradley and Lang, 1994). SAM is a nonverbal technique of self-assessment of emotions, specifically measuring the pleasure, arousal, and dominance associated with a person's affective reaction to a wide variety of stimuli. According to the authors, through these three dimensions, it is possible to describe any emotion. SAM consists of a pictographic representation where one can register levels of emotions. For each of the three dimensions of valence presented by SAM, the scale of responses ranges from 1 to 9, in which 1 represents the lowest level (of pleasure, arousal, or dominance) and 9 represents the highest level. Besides the use of SAM, we also used a form containing questions about the relation between the concepts presented and the applicability in their curricular practices.

Based on these definitions, the continuing education was organized into the following workshops:

First-day Workshop: Creative Village

The concepts of constructionism and creative learning were worked on in this workshop through a hands-on activity, using various unstructured materials (plastic bottles, various papers, play dough, hot glue etc.) and low-cost technological materials (leds, batteries and motors). Reflections and sharing of ideas were provided with some key questions, such as: Can you see how this type of activity can be inserted into your curriculum context? Does this kind of activity give you another perspective on what the teacher's role in the classroom might look like?

The development of the proposal was very interesting. Figure 3 shows a few of their creations, and moments with educators.

Figure 3: Creative Village Creations with the Educators (Source: authors').

Second-day Workshop: Computational Thinking and Tangible Artifacts

The purpose of this workshop was to present the concepts of Computational Thinking and Tangible Computational Artifacts. Four stations were assembled with different technologies that addressed the following themes: 1-unplugged computing; 2-tangible programming with Cubetto and TaPrEC; 3-Scratch environment; and 4-Scratch with Micro:bit.

The educators organized themselves into groups of six people in order to tinker at each station for about twenty minutes. Subsequently, a reflection on the subject was provoked through questions, such as: "How do you see the role of this type of activity associated with a curricular and conventional class?"; or "Did you feel engaged and willing to stay at the station even after the 20 minutes passed? How do you get this engagement from the students?"

Following this, we started to chat about skills associated with computational thinking and how these skills were present at the stations. Also, we talked about the importance of tangible computational artifacts in the Creative Learning process. Figure 4 shows moments of the educators at the stations.

Figure 4: Stations, in order: Unplugged, Cubetto, Scratch, and Scratch with Microbit (Source: authors').

Third-day Workshop: Design of Learning Activities

This workshop presented the design process of a Creative Learning activity. The goal was to place the educator in the role of a designer, providing elements to assist him/her in this process. Starting from the experience of the educators, the process leads them to design an activity considering the principles of Creative Learning, creating new practices or adapting existing ones.

The educators were organized into groups and guided in the design process in which they should, by mutual agreement, define the theme to work on. Before starting each step, a short moment of sharing and reflection was proposed. The work was acute, and many interesting ideas came up. One example includes the development of games for handling concepts related to water management systems. The moments of reflection were enriching and helped the educators in understanding the design process. At the end, the initial outline of an activity had been proposed and was the basis for the following activities. Figure 5 illustrates some moments from the workshop.

Figure 5: Some of the educators in a brainstorm session during the design process (Source: authors').

Fourth-day Workshop: Simulating an Activity

Based on the initial outline drawn up in the third workshop and the provided templates, the groups continued the work and designed a proposal to be conducted with their students. This workshop was intended to simulate and validate the activity, checking whether everything was according to the plan. The educators reviewed the proposal, simulating amongst themselves the times of each stage and checking whether all the materials and spaces defined were in compliance with what they had planned. Through this process, they were able to identify some adjustments and refinements to the process that were made and passed on to the activity document, as well as some definitions that were still pending. Figure 6 presents some moments of this workshop.

Figure 6: Educators engaged in the simulation of the proposed activity. In order: Reflection upon conducted activities; Design of an unplugged activity; Example of a designed Cubetto board; Simulation of activities using TaPrEC (Source: authors').

Fifth-day Workshop: Putting the activity in practice

As a result of the whole process described so far, we had 3 days of activities with the students. The structure applied to the activities was: to organize the children into three groups in order for each group to be able to experiment with one of the three stations. At the beginning, the students listened to the goals of the activity and received different challenges to be addressed at each station, however they were asked to work on the same curricular subject defined for the activity. The three stations were:

Creative Village Station: a place where it is possible to create artifacts while using unstructured materials and low-cost technologies such as LEDs, batteries, and small motors.

Unplugged Computing Station: challenges involving unplugged computing.

Technology Station: divided into substations: programming tasks with Scratch; tangible programming with TaPrEC and Cubetto; and with kindergarten, ScratchJr on Tablets and, for 7th and 9th grade, Scratch with Micro:bit.

Table 3 shows a summary of these activities and Figure 7, 8, and 9 show a few moments of the activities with students.

Table 3. Activities Designed by Educators

Activities Designed by Educators to be Conducted with Students			
Grade	Kindergarten	7th	9th
Age	5 years old	12 years old	15 years old
Children	54	41	67
Educators	17	6	6
Curricular Subject	- Storytelling with the story "Three Little Pigs" - Development of laterality and localization.	Mathematics working on the concept of polyhedra	Geography working on the water treatment process
Challenge on Creative Village Station	Create artifacts to protect their homes from the big bad wolf while using low-cost and unstructured materials.	Create components of a village, such as trees and houses, based on Polyhedron.	Create components of a water treatment system to be placed in a village.
Challenge on Technology Station	Explore programming tasks to move a pig or a wolf according to a proposed challenge.	Explore programming tasks while creating algorithms that draw polyhedra and / or capture polyhedra that "fall" onto the screen.	Explore the creation of scenarios that deal with the water treatment process.

Challenge on Unplugged Station	Follow commands given by a colleague in order to save the three little pigs.	Create a sequence of commands in order to capture a determined polyhedra.	Create a sequence of commands in order to solve some challenges related to the water treatment process.

Figure 7: Moments with the Kindergarteners (Source: authors').

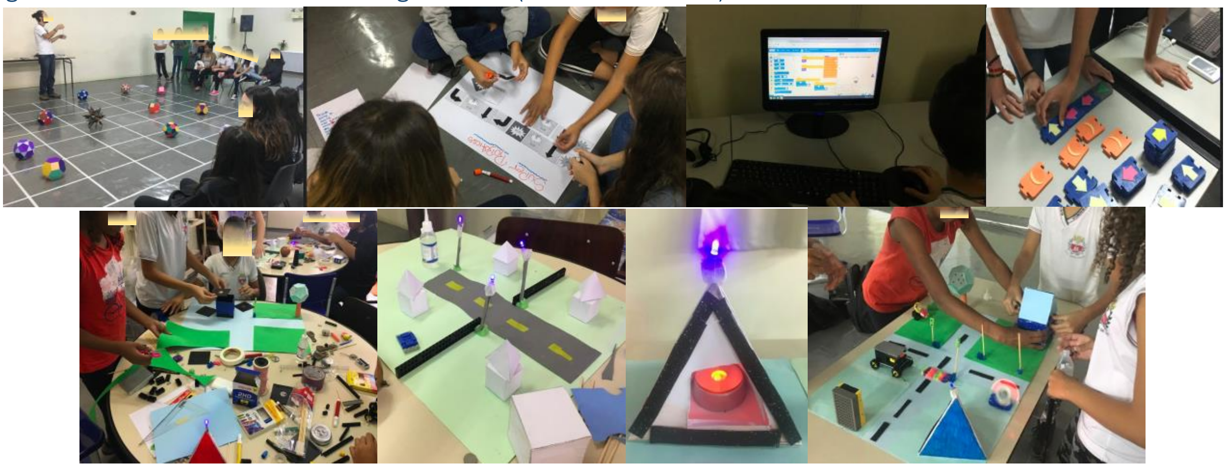


Figure 8: Moments with the 7th graders (Source: authors').

Figure 9: Moments with the 9th graders (Source: authors').

Results and Discussion

The workshops were well appreciated by the educators. In general, the educators initially showed some insecurity and wariness in their ability to work with unfamiliar concepts and practices. As the workshops unfolded and the more engaged they were, everyone felt very motivated and participated enthusiastically. At the end, the activities designed to be conducted with the students, working on a curricular subject, became very attractive and well structured.

We noticed that all the elements of the evaluated framework were covered in the design of the activities, and the TTR moments were provided with the students, revealing that the educators had a good understanding of the principles of constructionism and creative learning, and, especially, of how to put the elements of the framework into practice.

Some findings are worth mentioning from the answers filled out in the forms during the workshop. In the first three workshops, we asked two key questions: "Would you recommend this activity to a colleague?" For this question, 100% of educators answered "Yes"; and the second question was about the relation between the concept and/or practice in the workshop and the curricular practice. For this question, 100% answered "Yes, I can see this relation" in the first workshop; about 70% in the second; and 83% in the third workshop. Among the students, our findings show that they felt very motivated, and the majority of them stated that this kind of activity facilitates the understanding of curricular concepts and brings more engagement and satisfaction to all.

Figures 10, 11, and 12 show the results of the SAM with educators in the first three workshops and Figures 13 and 14 show the result of SAM with students of 7th and 9th grade. In all scenarios, positive assessments are provided. Instead of using SAM with the Kindergarteners, a ludic instrument of evaluation was employed, with three options: the color green representing "I liked it very much," "I liked it" associated with the color yellow and "I didn't like it," associated with the color red. Figure 15 shows how this was implemented and Figure 16 presents the positive feedback provided.

Taking all this into account, we can point out that the evaluated framework has great potential to facilitate the design of learning environments, based on the principles of constructionism and creative learning, even by educators without any previous experience with these kinds of activities or technologies.

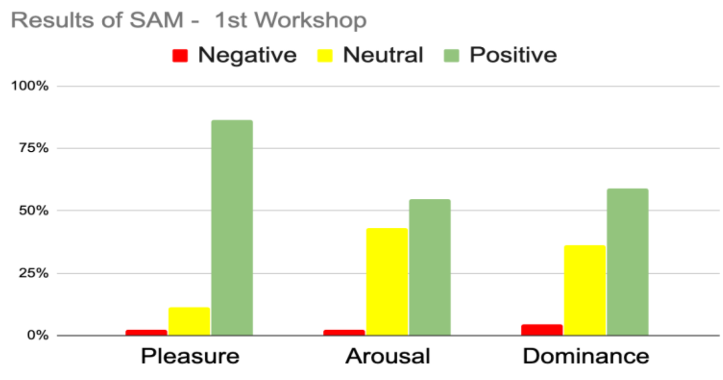


Figure 10. SAM - Educators – 1st Workshop.

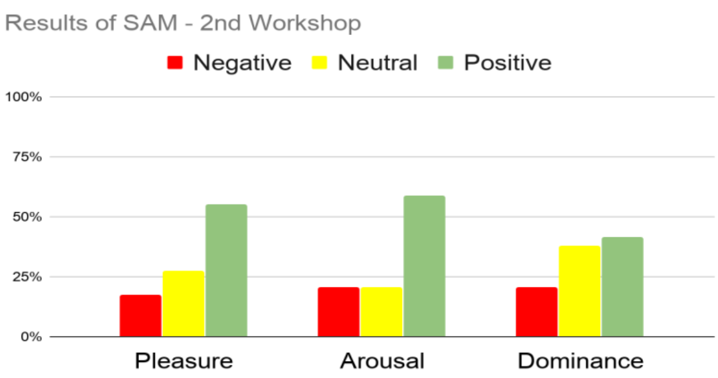


Figure 11. SAM - Educators – 2nd Workshop.

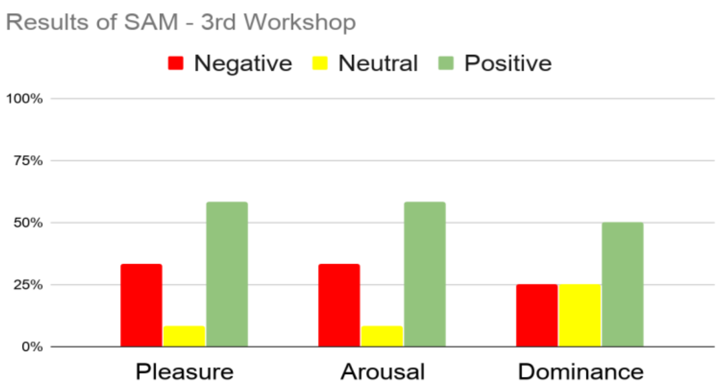


Figure 12. SAM - Educators – 3rd Workshop.

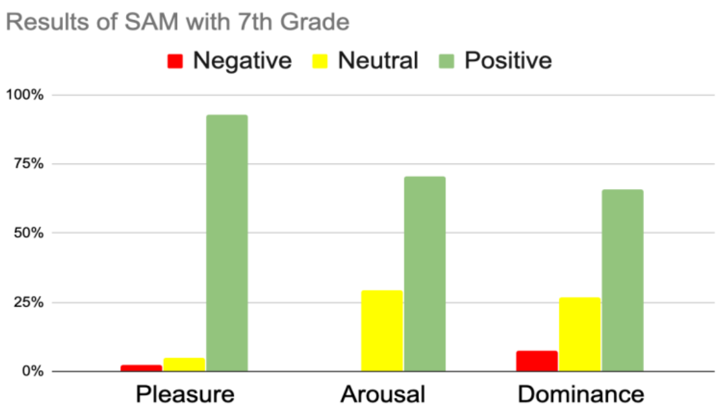


Figure 13. SAM - 7th Grade Students – 5th Workshop.

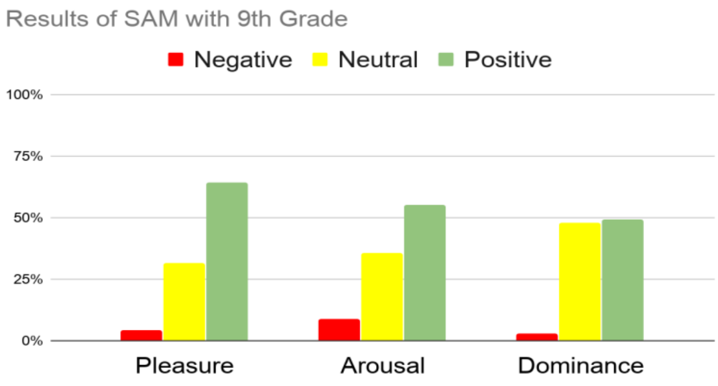


Figure 14. SAM - 9th Grade Students – 5th Workshop.

Figure 15. Evaluation Tools - Kindergarteners.

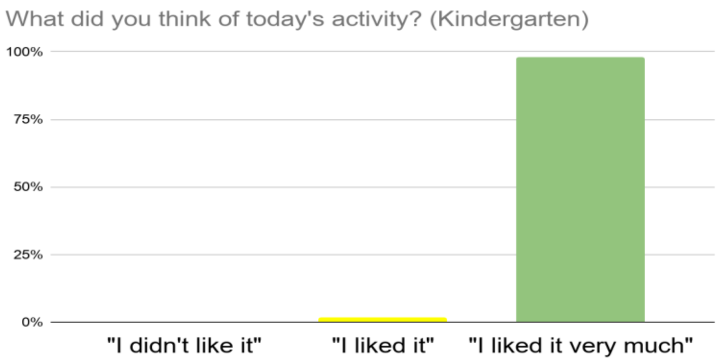


Figure 16. Perceptions of the Kindergarteners.

Conclusion

Novel educational policies from around the world have been proposing the use of Educational Technology in school spaces, as The Common National Curriculum Base in Brazil (Brazil, 2017). Hence, continuing professional development (CPD) for teachers working on contemporary approaches, such as those proposed by the Constructionism and Creative Learning, has been piquing the interest of educators. However, they hardly find the tools to assist them in the process of designing educational activities based on the principles of constructionism.

In this paper, we presented a case study involving 29 educators and 161 children in which we evaluated the application of a framework conceived to assist educators in designing educational scenarios based on the principles of constructionism and creative learning.

Initially the educators felt insecure and worried about their ability to design and conduct an activity with their students based on Creative Learning. As the workshops unfolded and the more engaged they were, they created very compelling activities. This shows that the adopted process helped them to overcome their initial feelings, understand the concepts and put them into practice. Among the students, they felt very motivated, and stated that this kind of activity facilitates the understanding of curricular concepts and brings more engagement and satisfaction to all.

Finally, we point out that the role of the educator experiencing the same process as their students is crucial. They should create strategies to foster the engagement of all and the reflection on the results achieved, the problems faced, and the solutions encountered. At this point, encouraging students to be conscious about the three-moment workflow when they are solving problems can be valuable. This work revealed that exposing educators to this experience was essential to develop their confidence to create good experiences for their students too. The evaluated framework supported this process.

In further research, we plan to develop a platform for educators to publish designed activities based on the evaluated framework, with the possibility of sharing designs, experiences, validations, and evaluations, in a 'living' curriculum for creative learning.

Acknowledgement

This research is supported by São Paulo Research Foundation (FAPESP), grant #2015/16528-0 and #2015/24300-9. Additional support was provided by the National Council of Technological and Scientific Development (CNPq) under grants #306272/2017-2 and #307560/2016- 3 and by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

Ackerman, E. (1995). Construction and transference of meaning through form. In Steffe, L & Gale, J (Eds), Constructivism in education (pp. 341-354). Hillsdale, NJ: Lawrence Erlbaum.

Ackermann, E. (1996). Perspective-Taking and object Construction. In Constuctionism in Practice: Designing, Thinking, and Learning in a Digital World (Kafai, Y., and Resnick, M., Eds.). Mahwah, New Jersey: Lawrence Erlbaum Associates. Part 1, Chap. 2. pp. 25-37.

Ackermann, E. (2001). *Piaget's constructivism, papert's constructionism: What's the difference*. Future of learning group publication, 5(3):438.

Ackermann, E. K. (2004). Constructing knowledge and transforming the world. In M. Tokoro & L. Steels (Eds.), *A learning zone of one's own: Sharing representations and flow in collaborative learning environments* (pp. 15–37). IOS Press.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. The New Zealand Journal of Applied Computing and Information Technology, 13(1), 20-29.

Bradley, M. M. and Lang, P. J. (1994). *Measuring emotion: the self-assessment manikin and the semantic differential*. Journal of behavior therapy and experimental psychiatry, 25(1):49–59.

Brasil, Ministério da Educação. Secretaria de Educação Básica. Base Nacional Comum Curricular, (2017). In: <http://basenacionalcomum.mec.gov.br>. Access: oct-18-2019.

Carbajal, M.L., Baranauskas, M.C.C.: TaPrEC: desenvolvendo um ambiente de programação tangível de baixo custo para crianças. In: Anais XX Congreso Internacional de Informática Educativa, Santiago, Chile (2015).

Horn, M. S. and Jacob, R. J. (2007). *Tangible programming in the classroom with tern*. In CHI'07 extended abstracts on Human factors in computing systems, pages 1965–1970. ACM.

Ishii, H. and Ullmer, B. (1997). *Tangible bits: Towards seamless interfaces between people, bits and atoms*. In Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI ’97, pages 234–241, New York, NY, USA. ACM.

Marinus, E., Powell, Z., Thornton, R., McArthur, G., and Crain, S. (2018). *Unravelling the cognition of coding in 3-to-6-year olds: The development of an assessment tool and the relation between coding ability and cognitive compiling of syntax in natural language*. In Proceedings of the 2018 ACM Conference on International Computing Education Research, ICER ’18, pages 133–141, New York, NY, USA. ACM.

Nicastro, F., Baranauskas, M. C., and Torres, R. (2019). Supporting educators in the design of constructionist learning environment scenarios: a framework. Proceedings of CBIE - Congresso Brasileiro de Informática na Educação, Brasília, DF, Brasil, 2019. DOI: http://dx.doi.org/10.5753/cbie.wie.2019.286

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Papert, S. (1993). *The Children’s Machine: Rethinking School in the Age of the Computer*. Basic Books, Inc., New York, NY, USA

Papert, S. and Harel, I. (1991). *Situating constructionism*. Constructionism, 36(2):1–11.

Resnick, M. (2016). Designing for Wide Walls. *Design Blog: MIT Scratch Team, MIT Media Lab*.

Resnick, M. and Robinson, K. (2017). *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. MIT Press.

Resnick, M. and Rosenbaum, E. (2013). *Designing for tinkerability*. Design, make, play: Growing the next generation of STEM innovators, pages 163–181.

Resnick, M. and Silverman, B. (2005). *Some reflections on designing construction kits for kids*. In Proceedings of the 2005 conference on Interaction design and children, pages 117-122, ACM.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. and Kafai, Y. (2009) Scratch: programming for all. Commun.ACM, 52, 11, p. 60–67

Ullmer, B. and Ishii, H. (1997). *The metadesk: Models and prototypes for tangible user interfaces*. In Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology, UIST ’97, pages 223–232, New York, NY, USA. ACM.

Cultural Diversity and Scratch Programming: Enhancing Integration of International Students using Interactive Posters

Loice Victorine Atieno, *atienomunira04@gmail.com*
Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary

Richard Kwabena Akrofi Baafi, *richbaafius@gmail.com*
Faculty of Education and Psychology, Eötvös Loránd University, Budapest, Hungary

Abstract

Academic mobility has become a global norm. The movement from one's social milieu to another has significant effects on the students' social and academic life. This is attributed more to cultural diversity. Eötvös Loránd University, Hungary, admits international students for various programmes. The students are faced with various challenges such as language barriers, cultural differences and integration. Various intervention programmes have been put in place to facilitate the integration process. Unfortunately, most of the programmes focus on the host culture posing challenges in building a cohesive and vibrant classroom environment that elicits interaction and collaboration.

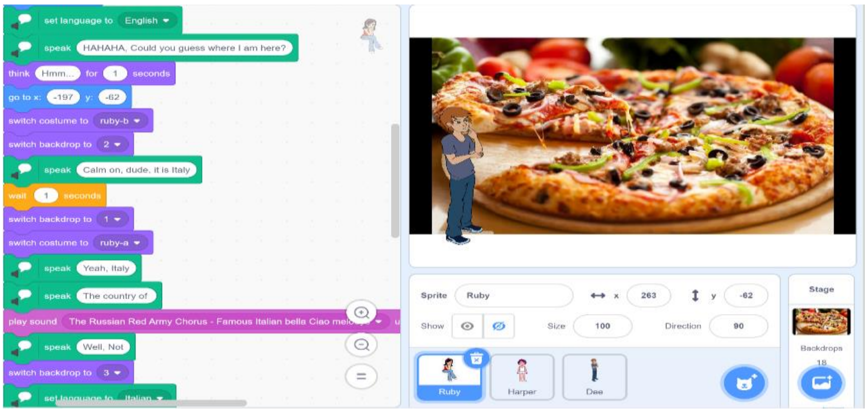


Figure 1: Using various ways of data representation to share culture

Strategies that incorporate integration into the learning process need to be adopted. To enable effective integration, emphasis needs to be laid also on different cultures of the international students. Two PhD students together with the lecturers at the faculty of Informatics have embarked on a project dubbed "Show and Share your Culture" aiming at tackling integration challenges due to cultural diversity experienced in a learning environment. This paper, therefore, demonstrates the findings of the project's preliminary phase.

The study was carried out among the zero-year Computer Science international students of the faculty of Informatics at Eötvös Loránd University. This involved creating interactive posters using Scratch programming in self-introduction and culture sharing. Data was collected using survey and projects. Findings showed that visual programming languages such as Scratch can be used in learning and facilitating integration process in learning environment.

Keywords

Cultural diversity, Scratch programming, Interactive posters, Integration, International students

Introduction

Education is a unifying force for both teachers and students across the globe. Connecting higher education to global perspectives that seeks to integrate international students into global learning diversities is important (American Council on Education, 2012). The rapid transformation of the world and expectations of a constantly changing post-secondary education have significantly impacted the process and outcomes of higher education leading to internationalisation (Knight, 2004). Internationalisation is the integration of global dimensions such as culture, policies and programmes into higher education's functions to meet the needs of students (Haji, Moluayonge, & Candidates, 2015; American Council on Education, 2012; Altbach & Knight, 2007; Knight, 2004). It is therefore important in selecting the content and objectives of universities' curriculums (Van Der Wende, 2010).

Internationalisation has become a global phenomenon enabling students who aspire for higher education leave their home countries for study abroad (Paulo, Karine, Ester, & Elena, 2009; Altbach & Knight, 2007). A study about higher education established that "the increase in student mobility is associated with: the introduction of a more international curriculum in many subject areas in universities across the world; greater diversity of student bodies in many higher education institutions; new cross-national friendships among others" (Brooks & Waters, 2011, p. 137). This is driven by advancement in information communication and technologies coupled with the growth of English language as a common language in most parts of the world (Haji et al., 2015; Healey, 2008; Knight, 2004).

In facilitating international students' integration process, host institutions initiate programmes geared towards making students feel welcomed (Altbach & Knight, 2007). Because of the complex nature of cultures and the struggles associated with integration, international students are found building new relationships that help them fit into their host environment. The transition period is often challenging to most students due to the complexity of internationalisation (Knight, 2004). This comes as a result of cultural diversity, differences in language and previous educational experiences (Prud'Homme van Reine & Blom, 2017). The new learning environment that international students must acclimatise to requires integration (Castro, Barrera, & Steiker, 2010), hence the need for programmes that can accelerate its process. The experiences the students make in the new environment impact on their lifelong learning and participation in society (Denson & Zhang, 2010).

A common danger that retards integration process is the phenomenon of resilience to accept others' beliefs leading to cultural blindness (Kim, Maleku, Lemieu, Du, & Chen, 2019). The students tend to group creating a niche of comfort for themselves according to their nationalities and cultures leading to divisiveness between international and home students (Spiro, 2014). To create an inclusive learning environment universities are mandated with ethical and educational duties to incorporate various diversities into the teaching and learning process (Guo & Jamal, 2007). Universities therefore need to be conscious of their multicultural nature putting in place interventions that help students develop skills and behaviours that make them fit as global citizens (OECD, 2018). Cultural diversity should improve the image which international students bring into the university communities as a blend of cultures (Plessis & Bisschoff, 2007). Thus engaging international students in courses that bridge cultures will foster active participation and collaboration in the learning process creating harmonised cultural cohesiveness (Prud'Homme van Reine & Blom, 2017).

Cultural Diversity

Human beings attribute meanings to their environment finding common grounds for daily actions. The idea may be connected to the concept of culture which can be defined as set of practices among people that provide common understanding, beliefs, behaviours and objects (Gill, 2013). Culture therefore can be viewed as inherited artefacts that facilitate achievement of everyday life (Pacheco & Gutiérrez, 2009). Admission of international students into higher learning institutions have made the institutions ethno-culturally diverse (Guo & Jamal, 2007). International students are viewed within the learning environment as active replicas of many cultural systems since they come from different cultural setups. Apart from building a rich academic environment, this increase in the heterogenous learning community has led to challenges associated with cultural diversity such as colour blindness and fear of diversity (Prud'Homme van Reine & Blom, 2017).

There is growing effort to understand how culture enhances the process of learning (Nasir & Hand, 2006). Understanding the creation of a good teacher, student and effective learning is attributed the variations in academic culture - beliefs, expectations and cultural practices on academic achievement (Prud'Homme van Reine & Blom, 2017). The learning process provides opportunity of sharing in each other's cultural identity. Irrespective of individual's cultural traits, international students can showcase interactive cultural sharing as they learn. Generally international students have performed well in both academic integration and achievement, but a deeper insight shows that non-western international students still have problems adapting to their first year at the university (Mesidor & Sly, 2016). Understanding the struggles students go through in their new environment demands awareness of cultural processes and artefacts defining their cultural backgrounds (Nasir & Hand, 2006). Providing an outlet that communicates understanding can help the students transform their mindset for effective integration.

With ever-increasing international students in universities across the world, dealing with cultural diversity has become a great course (Woods, Jordan, Loudoun, Troth, & Kerr, 2006). Research has shown Europe as the most affected in terms of pronounced language diversity, variety in educational systems and cultural heterogeneity (House, Hanges, & Javidan, 2006). As the result of these diversities international students are confronted with several challenges in the learning environment requiring intervention (Abdul Rahman & Alwi, 2018). Five dimensions on cultural differences inform the structure and strategies employed by universities in different countries (Hofstede, Hoftstede, & Minkov, 2010):

Power distance: acceptance and expectation of less powerful on the discharge of authority

Uncertainty avoidance: tolerance to ambiguity and uncertainties

Individualism versus collectivism: extent of individual integration

Masculinity versus femininity: gender roles and affirmative actions

Long-term versus short-term orientation: rewards verses society needs and desire

Universities are also environments where effective cultural integration takes place. Therefore international students' integration needs to be tackled with flexibility to allow cohesive integration (Woods et al., 2006). Cultural exchange is a flexible mechanism for addressing challenges of cultural diversity (Prud'Homme van Reine & Blom, 2017). Culture influences learning and communication process, relationship between students and teachers, and motivation towards learning (Guo & Jamal, 2007). There is need therefore for creating cultural awareness and heterogenous classrooms (Andrade, 2006).

Cultural Integration into our Classrooms

Existence of international students calls for changes in the teaching and learning processes, acquisition of competencies and capabilities enhancement that accommodate cultural diversity

(Abdul Rahman & Alwi, 2018). The effectiveness of integration depends on how well the students cultural background blends with the culture of the host institution (Guo & Jamal, 2007). Ignoring the uniqueness of students' cultural disposition could have consequential effects on transforming learning diversity into personalised learning (Prud'Homme van Reine & Blom, 2017).

Classroom is viewed as a cultural community mirroring what is taught and learnt (Vaccarino, 2009) demanding effective and culturally responsive pedagogical approaches. Classroom made up of international students can be described as an international classroom. It constitutes a learning environment where students from various international backgrounds meet and exchange ideas, experiences and acquire competencies for participation in the global community (de Vrieze-McBean, 2014). This calls for culturally responsive pedagogical approaches. These approaches are student-centred and support the integration of cultural diversity in the learning environment and are of three dimensions (Richards, Brown, & Forde, 2004):

Institutional: administrative procedures and policies

Personal: learning abilities and teachers' affective engagement in supporting student cultural integration in the teaching and learning process

Instructional: tools, strategies and resources for enhancing teaching and learning.

The dimensions need to be aligned with the various students' cultural backgrounds if their needs are to be met. Teachers possess the power to create an impact on students' achievements therefore must be culturally responsible in utilising tools and engaging in practices that can create a cohesive learning environment (Delpit, 1988). It is also important for teachers to understand students' individual differences such as culture and language as contributing to students' behaviours and attitudes (Richards et al., 2004). They need to employ strategies that enable students create (design and construct artefacts), personalise (work on tasks based on prior knowledge, questions and passions), share (collaborate and exchange ideas) and reflect (Brennan, 2014). An international classroom provides a learning environment that enables students with various backgrounds co-operate and exchange ideas as they develop and enhance their competencies (de Vrieze-McBean, 2014). Sharing of one's culture in the classroom through interactive posters enable students create, personalise, share and reflect on their projects leading to constructionism.

Constructionism in the Classroom

Constructionism approach to learning emphasises on creation of new ideas while developing artefacts that can be reflected on and shared (Brennan, 2014). Papert in describing constructionism emphasised that even though learning takes place in the students' mind it can only happen if they take part in meaningful tasks outside their mind (Passey, Dagienė, Atieno, & Baumann, 2018). Teaching in an international classroom requires incorporating activities that accelerate learning and integration. Papert, in the constructionist paradigm, acknowledges that affective engagement is key in developing and enhancing students' competencies (Sacristán, Kaminskiené, Michaela, & Baafi, 2018). Conducive learning environment supports students in creating and reflecting on their own artefacts, providing them with objects-to-think-with (The Turtle) that constitutes culture, competencies and personalising artefacts (Papert, 1980, p. 11). Technology facilitates the creation, personalisation and sharing of the artefacts. For example, in the Scratch programming class students used Scratch for self-introduction and sharing of culture with their peers. Technology such as visual programming language like Scratch can be used to nurture the development of a new generation of creative, systematic thinkers who thrives in using programming to express their ideas (Resnick, Maloney, Andrés Monroy-Hernández, et al., 2009). To be able to put the ideas across technology became handy even though it was not the central focus. As Papert would describe "… the role I give to the computer is that of a carrier of cultural "germs" or "seeds" whose intellectual products will not need technological support once they take root in an actively growing mind" (Papert, 1980, p. 9). Scratch, therefore, provides an ideal tool for use in a constructionism classroom as it allows for the students with the capabilities to design, personalise, express, share and reflect on their works. These aspects according (Brennan, 2015) are essential in the designing of constructionist learning environments.

Scratch Programming

Many learning tools continue to assist teachers and students express their ideas or increase interactions using interactive media. Having a programming language that appeals to students who have never programmed is excellent way to make them interested in programming. Scratch is an example of such programming languages. Scratch is inspired by the idea of yearning to make programming easy for all despite the age, experiences, and interests (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010; Resnick, Maloney, et al., 2009). These help the learners in developing their own digital interactive stories, games, animations, and simulations; and share their works with others.

Programming with Scratch is grounded on the constructionist philosophy of developing projects and artefacts based on personal engagement and motivation (Papert, 1980; Resnick, Maloney, Andrés Monroy-Hernández, et al., 2009). Scratch is considered a powerful programming tool which supports 21st-century skills[13]. These skills include creative thinking, clear communication, systematic analysis, effective collaboration, iterative designing, and continuous learning which can be developed and enhanced through Scratch programming. The activities carried out in the class provided the international students with a platform for learning through sharing and also help develop computational thinking, creativity, critical thinking and problem-solving skills.

Sharing various cultures using Scratch programming contributed to constructing visualised interactive media that helps the students to integrate faster and also develop and enhance different competencies. Therefore, the course in this regard remained a learning tool which had the purpose of assisting the students convey meaningful ideas and artefacts through their posters. The experience in learning from peers is symbolic since it serves as a means of empowering students to collaborate and support the building of creative ideas. According to (Nasir & Hand, 2006), one of the ways of enhancing learning is by using artefacts and new learning tools. Thus, such tools are an essential means of introducing students, especially international students, to new learning environments because they are capable of transforming beliefs and cultural traits into positive learning goals.

Computational Thinking (CT) with Scratch in a culturally Diverse Classroom

According to Brennan & Resnick (2012), learning with Scratch is boosted by the phrase computational thinking, hence programming with Scratch presents an opportunity in contributing towards its definition. The authors further define CT by deriving its meaning based on three key dimensions:

Computational concepts: concepts used by learners when coding - sequences, loops, parallelism, events, operators, conditional and data.

Computational practices: ways of solving problems developed and used while programming - being experimental and iterative; testing and debugging; reuse and mixing, and abstraction and modularisation.

Computational perspectives: elements that allow Learners' understand themselves and the world around them - expressing, connecting and questioning.

Therefore CT is viewed as a catalyst in the conceptualisation of knowledge creation and acquisition that take place in Scratch (Brennan & Resnick, 2012). While the students were creating their projects, various CT skills were developed and enhanced. The whole process took the constructionism approach where learning was learner-centred; through design approach to learning, each learner was expected to have a project; and through sharing of already known knowledge, the learners were able to acquire new knowledge. From constructionist point of view,

[13] http://www.21stcenturyskills.org

effective learning experiences stem from creation artefact that are personal and socially meaningful (Brennan, 2015).

The Goal of Our Research

The capabilities of Scratch makes it not strong enough to be viewed as a programming language for creating professional programmer but it has proved to be a powerful tool for nurturing the development of a new generation of creative, systematic thinkers who thrives in using programming to express their ideas (Resnick et al., 2009),. Scratch allows the users to express their ideas with a lot of ease and in a captivating manner hence can be used to solve day-to-day problems and cultural diversity is not an exception. Culture gives identity and being able to create an understanding of one’s cultures requires elaborative description devoid of ambiguity. The expressive nature of Scratch, coupled with its ease of use, has made it a target tool for use in problem-solving prompting its use in the creation and sharing of different cultures among international students through interactive posters. This therefore forms the basis of our research.

Methodology

Scratch programming is a course that was introduced as an intervention course to help redeem the academic performance of year one BSc Computer Science international students who had performed poorly in their programming courses. A survey carried out among the affected students, attributed the poor academic performance largely to their inability to juggle between their academics, integration into their new environment and getting to understand their peers to be able to work together as a team. The course was then pushed to zero-year level and made compulsory. It is taught using a design-based approach, and the end-product of each activity is a project. The course activities are divided into i) character and name animation; ii) digital storytelling; iii) gaming; and iv) own choice project. At the beginning of the course there is the introduction session which is important as it allows the parties involved to get to know each other and create a rapport between them. This session normally has no predefined way of doing it.

During the Scratch programming course first lecture meeting, the introduction session was carried out in a different away by tasking the students to come up with a project introducing themselves and their culture. The students were expected to create interactive posters using Scratch and share their projects with their peers and teachers. This enabled them gain a deeper understanding of each other leading to a faster and effective integration process. The preliminary phase was conducted over four weeks with students having 4 contact hours with the facilitators each week.

Participants

The preliminary research was conducted among 23 zero-year Computer Science international students facilitated by 2 PhD students. The group comprised of students from various parts of the world as shown in table1.

The 2 PhD students facilitating the course were from Kenya and Ghana, and they also participated in the project as a way of motivating the students.

	Frequency	Percent
Azerbaijan	6	26.1
China	3	13.0
Egypt	3	13.0
Hungary	2	8.7
Kosovo	1	4.3
Kyrgyzstan	2	8.7
Macedonia	1	4.3
Moldova	1	4.3
Morocco	1	4.3
Pakistan	1	4.3
Yemen	2	8.7
Total	23	100.0

Table 1. Nationality distribution of Participants

Process

The introduction project dubbed "Show and Share your Culture" was divided into two phases. The first phase, which is also the preliminary phase comprised of: i) creating an interactive poster using Scratch: ii) Sharing the postcard with peers in the class. The second phase (full project) comprises of i) replicating the project beyond the faculty; ii) pinning the postcard on to the visualised world map to share it with the rest of the world, and iii) viewers voting for the best project (s) and awarding of the prizes.

Creating an Interactive Poster

The preliminary study of the "Show and share your culture" project was conducted among zero-year BSc Computer Science international students of the faculty of informatics. As part of their first activity in the Scratch programming course class, the students were expected to create an interactive poster as a way of introducing themselves and their culture. To be able to accomplish the above task, the students were introduced to Scratch. First, they were expected to download the offline Scratch 3.0 editor that they could use even without network[14]. They were then taken through Scratch window and the use of various Scratch blocks. They were also furnished with various resources that would enable them to work on their project such as:

Tutorials on how to get started with Scratch[15]

Various projects created using Scratch[16]

Video tutorial on how to create interactive posters using Scratch[17].

Project Development

In this section, the students were expected to come up with their respective projects in a systematic way. Apart from introducing themselves and showcasing their culture, the following learning objectives were to be realised:

Systematic documentation of the process of developing the project

Sharing and collaboration among the students

[14] https://scratch.mit.edu/download
[15] https://scratch.mit.edu/explore/projects/tutorials/
[16] https://scratch.mit.edu/explore/projects/all/
[17] https://www.youtube.com/watch?v=vB_yBP0BFSA

Demonstration of creativity and problem-solving skills

Acquiring and development of debugging skills

Expressing of one's feelings

Development and enhancement of computational thinking concepts such as sequencing, decomposition, generalisation, data representation among others

To achieve the above objectives, the students were exposed to the snippet of projects created by the facilitators as a guide to enable them to jam start their projects. A guideline on how to develop their documentation was also provided which comprised of: i) a brief project description, and ii) step-by-step process of creating the project. This process took two weeks. The students were encouraged to collaborate with each other, and the facilitators were also available for consultation. This resulted in the constant iteration of the projects to factor in new ideas and refined the projects.

Project Sharing

In this section, the students were given a chance to showcase their projects. Each student presented his/her project to their peers while their peers were expected to critic their projects at the same time learn from the projects. The following learning objectives were to be achieved:

Demonstration of computational thinking perspectives such as collaboration and sharing

Presentation skills

Variety of projects were presented ranging from simple animated projects to complex projects which consisted of a combination of games and animation. The students demonstrated their creativity by portraying the unique cultures of their countries in various ways. They used a combination of animated images on different backdrops, audio and music.

Discussions

Carrying out this project revealed a lot about new students. All students have a vibrant but hidden potential that needs to be triggered and nurtured. At the beginning of the exercise, most of the students were quiet and withdrawn to the extent that they had to be coerced even to say their name. This is more pronounced among international students due to cultural blindness (Kim et al., 2019) and fear of diversity (Prud'Homme van Reine & Blom, 2017). When the project was launched coupled with the demonstration from the facilitators, which showed a collaborative project showing cultures of both Ghana and Kenya. The move triggered the students, and each one of them was eager to showcase their culture. Teaching international students calls for needs fulfilment, enabling cultural integrations and exchange of ideas through collaboration and consultation (Mantiri, 2013). The research was grounded on the four constructionism beliefs:

Create: substantial learning experiences stem from designing and creating artefacts (Bruckman, 2012; Papert, 1980). Nonetheless, not all students had it easy developing a project on their culture and show it using Scratch programming without difficulties. While some of the students quickly excelled in this, others achieved their goals but with challenges. The biggest hurdle that most students exhibited was how to package their ideas for others to understand what they intended to communicate. This challenge was easily solved through collaboration and sharing with their peers. In constructionism, students are able to realise not only their ways of solving problems but also understand the various degrees of the problem (Holbert, Penney, & Wilensky, 2010).

Personalise: projects created should be personalised and have social meaning (Bruckman, 2012; Papert, 1980). Culture influences one's identity, feeling, behaviour, beliefs and practices enabling adherence to society's customs and morals (Gutiérrez & Rogoff, 2003). The project gave the students an opportunity to share their identity, feelings and beliefs allowing their peers understand them better. Thinking about differences in individuals' learning styles and self-concepts, and recognizing that there is not one way or style of learning (Brennan, 2015). The uniqueness in the projects were well portrayed in various ways by the students. For example, in Figure 2 the student used his own image to narrate and walk the peers through their culture.

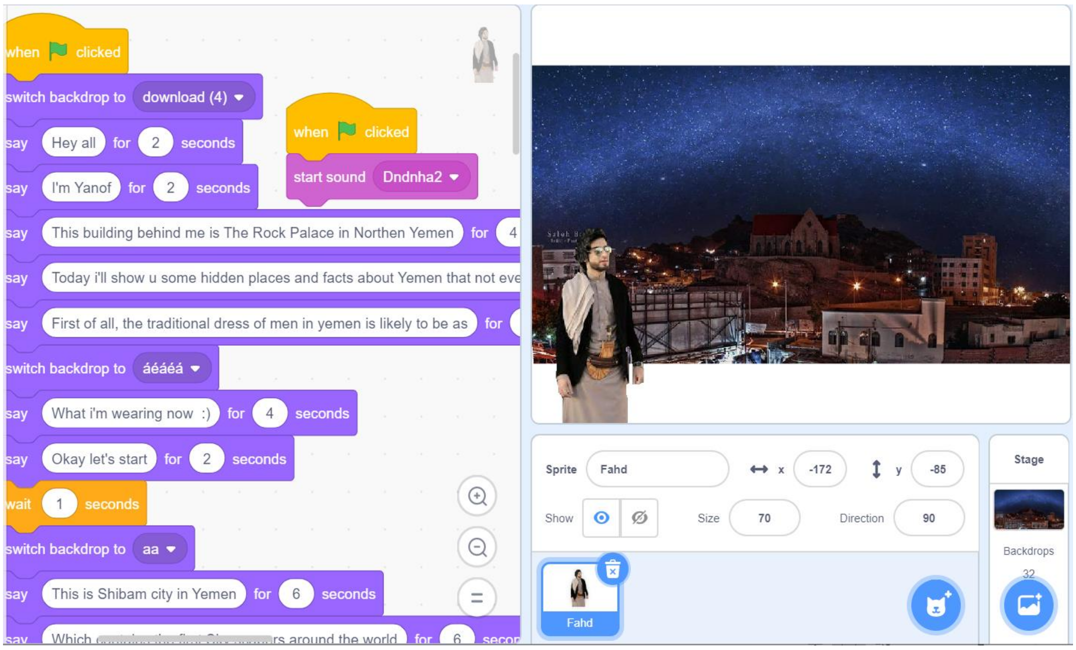


Figure 2: Personalising and Project

Share: exchange of ideas through sharing, questioning and collaboration (Papert, 1980). Despite the uniqueness of the individual projects, the students realised the commonalities in the projects, especially when it came to attires, foods, religion, among others. These commonalities helped in bridging the gap created by culture blindness making the students see each other as part of a shared community. Apart from sharing their culture, a lot of learning also took place. The students developed various computational thinking skills which were well portrayed in their projects and during the process of creating their projects. For example, the codes behind one of the projects showed the use of concepts such as repetition and events, as shown in figure 3:

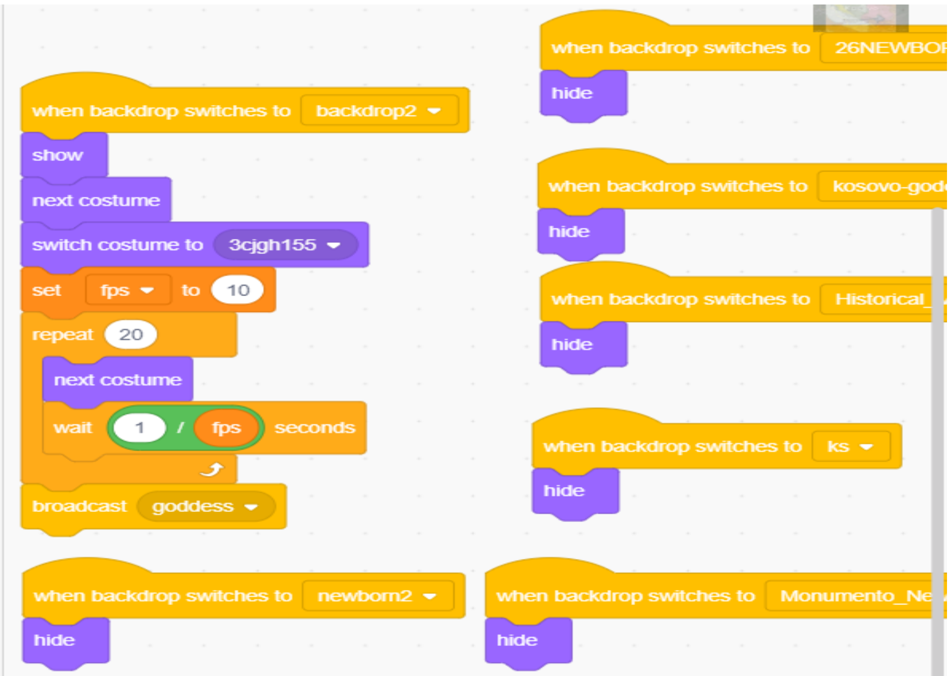


Figure 3: Project codes showing repetition and events

Being the first project, most of the students displayed the use of simple computational thinking skills such as sequencing as shown in figure 4.

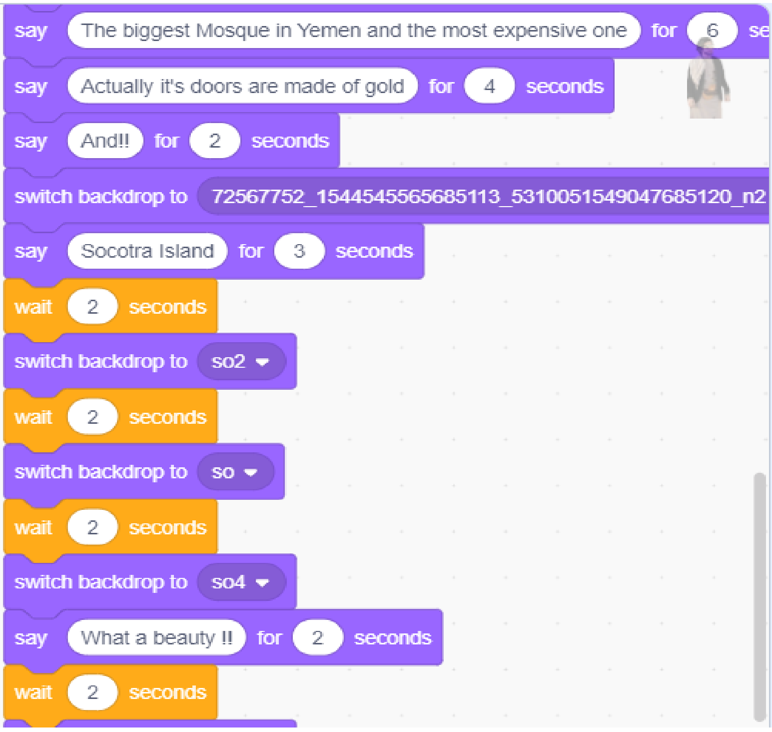


Figure 4: Project codes showing the sequencing

The students also engaged in a lot of sharing and consultations. This led to frequent interactions, testing and debugging of the projects to come up with their final projects.

Reflect: 'support thinking about one's own thinking (Kolodner, Camp, Crismond, Fasse, & Gray, 2003; Papert, 1980). As the students create, personalise and share their projects various questions are formed in their minds based on what they are doing and how they are thinking presenting them with opportunity to reflect on their projects (Brennan, 2015).

Conclusion and Way Forward

The principal target of this project was to tackle the challenge of integration of international students due to cultural differences enabling them to focus on their studies and be able to realise excellent grades. Scratch provided the students with a suitable programming platform for them to communicate their ideas. The power of visualisation presented the students with various ways of data presentation making it easier for them to create the interactive posters. The results show that culture is an element closer to people's hearts and most if not all, handle it with a lot of passion hence the quality of the projects. Working in a constructionist classroom boosted the whole process as the students were able to create personal and meaningful projects, personalise, share and reflect on their projects. Apart from effective integration, the students were able to master various computational thinking and problem-solving skills and also demonstrate a high level of creativity.

The next step will be to replicate the project in other faculties within the university and beyond the university. In this second phase, the students will create postcards. Being that the postcards will be created from various places, the projects will be pinned onto a visualised world-map as a way of showing and sharing their culture with a broader community. This research is intended to be replicated in different universities within Hungary as a way of helping international students adapt smoothly.

References

Abdul Rahman, S., & Alwi, A. (2018). The Inclusion of Cultural Diversity in Higher Education Curriculum Design. *International Journal of Contemporary Educational Research*, *5*(2), 87–94. https://doi.org/10.33200/ijcer.479054

Altbach, P. G., & Knight, J. (2007). The internationalization of higher education: Motivations and realities. *Journal of Studies in International Education*, *11*(3–4), 290–305. https://doi.org/10.1177/1028315307303542

American Council on Education. (2012). Mapping Internationalization on U . S . Campuses : 2012 edition. *Center for Internationalization and Global Engagement*.

Andrade, M. S. (2006). International students in English-speaking universities: Adjustment factors. *Journal of Research in International Education*, *5*(2), 131–154. https://doi.org/10.1177/1475240906065589

Brennan, K. (2014). Constructionism in the Classroom: Three experiments in Disrupting Technocentrism. *Constructionism*, 1–8.

Brennan, K. (2015). Beyond Technocentrism Supporting Constructionism in the Classroom. *Constructivist Foundations*, *10*(3), 289–304.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *AERA*.

Brooks, R., & Waters, J. (2011). Student mobilities, migration and the internationalization of higher education. In *Student Mobilities, Migration and the Internationalization of Higher Education*. https://doi.org/10.1057/9780230305588

Bruckman, A. (2012). Learning in Online Communities. *The Cambridge Handbook of the Learning Sciences*, pp. 461–472. https://doi.org/10.1017/cbo9780511816833.028

Castro, F. G., Barrera, M., & Steiker, L. K. H. (2010). Issues and Challenges in the Design of Culturally Adapted Evidence-Based Interventions. *Annual Review of Clinical Psychology*, *6*(1), 213–239. https://doi.org/10.1146/annurev-clinpsy-033109-132032

de Vrieze-McBean, R. (2014). Addressing Cultural Diversity in the International Classroom: A Challenge or an Opportunity? *The Second European Conference on Education*.

Delpit, L. D. (1988). The Silenced Dialogue: Power and Pedagogy in Educating Other People's Children. *Harvard Educational Review*, *58*(3), 280–298.

Denson, N., & Zhang, S. (2010). The impact of student experiences with diversity on developing graduate attributes. *Studies in Higher Education*, *35*(5), 529–543. https://doi.org/10.1080/03075070903222658

Gill, T. G. (2013). Culture, complexity, and informing: How shared beliefs can enhance our search for fitness. *Informing Science*, *16*(1), 71–98. https://doi.org/10.28945/1778

Guo, S., & Jamal, Z. (2007). Nurturing Cultural Diversity in Higher Education: A Critical Review of Selected Models. *Nurturing Cultural Diversity in Higher Education: A Critical Review of Selected Models.*, *37*(3), 27–49.

Gutiérrez, K. D., & Rogoff, B. (2003). Cultural Ways of Learning: Individual Traits or Repertoires of Practice. *Educational Researcher*, *32*(5), 19–25. https://doi.org/10.3102/0013189X032005019

Haji, S. A., Moluayonge, G. E., & Candidates, P. D. (2015). Internationalization of Higher Education: Promotion, Implementation and Improvement Strategies in Korea. *Print) International Journal of Social Science and Humanities Research*, *3*(April), 60–65. Retrieved from www.researchpublish.com

Healey, N. M. (2008). *Is higher education in really ' internationalising '?* (September 2006), 333–355. https://doi.org/10.1007/s10734-007-9058-4

Hofstede, G., Hoftstede, G. J., & Minkov, M. (2010). *Cultures and Organizations* (3rd Ed.). The McGraw-Hill Companies.

Holbert, N., Penney, L., & Wilensky, U. (2010). Bringing Constructionism to Action Game-Play. *Constructionism 2010*, 1–7.

House, R. J., Hanges, P. ., & Javidan, M. (2006). *Culture , Leadership , and Organizations : The GLOBE Study of 62 Societies [review] / House , R . Organizations : The GLOBE*. *1*(1), 55–71.

Kim, Y. K., Maleku, A., Lemieu, C., Du, X., & Chen, Z. (2019). Behavioral health risk and resilience among international students in the united states: A study of sociodemographic differences. *Journal of International Students*, *9*(1), 282–305. https://doi.org/10.32674/jis.v9i1.264

Knight, J. (2004). Internationalization Remodeled: Definition, Approaches, and Rationales. *Journal of Studies in International Education*, *8*(1), 5–31. https://doi.org/10.1177/1028315303260832

Kolodner, J., Camp, P., Crismond, D., Fasse, B., & Gray, J. (2003). {…} meets case-based reasoning in the middle-school science classroom: Putting Learning by Design{\texttrademark} {…}. *Journal of the Learning Sciences*, *12*(4), 495–547. Retrieved from http://www.jstor.org/stable/1466914

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education*, *10*(4), 1–15. https://doi.org/10.1145/1868358.1868363

Mesidor, J. K., & Sly, K. F. (2016). Factors that contribute to the adjustment of international students. *Journal of International Students*, *6*(1), 262–282.

Nasir, N. S., & Hand, V. M. (2006). Exploring sociocultural perspectives on race, culture, and learning. *Review of Educational Research*, *76*(4), 449–475. https://doi.org/10.3102/00346543076004449

OECD. (2018). Preparing our youth for an inclusive and sustainable world: The OECD PISA global competence framework. Retrieved November 10, 2019, from https://www.oecd.org/education/Global-competency-for-an-inclusive-world.pdf

Pacheco, M., & Gutiérrez, K. (2009). Cultural-historical approaches to literacy teaching and learning. *Breaking the Silence: Recognizing the Social and Cultural Resources Students Bring to the Classroom*, 60–77. Retrieved from http://dm.education.wisc.edu/mapacheco/intellcont/PachecoGutierrezCHT_2008-1.pdf%5Cnpapers2://publication/uuid/09C716BA-40D2-495F-9766-5C6BAAE007F1

Papert, S. (1980). *Mindstorms Children, Computers, and Powerful Ideas*. New York: Basic Books, Inc., Publishers.

Passey, D., Dagienė, V., Atieno, L. V., & Baumann, W. (2018). Computational practices, educational theories, and learning development. *Problemos*, *2018*, 24–38. https://doi.org/10.15388/Problemos.2018.0.12346

Paulo, S., Karine, T., Ester, B., & Elena, A. (2009). Tertiary Education for the Knowledge Society. *Tertiary Education for the Knowledge Society*, Vol. 2. https://doi.org/10.1787/9789264063518-hu

Plessis, P., & Bisschoff, T. (2007). Diversity and complexity in the classroom: valuing racial and cultural diversity. *Educational Research and Reviews*, *2*(9), 245–254.

Prud'Homme van Reine, P., & Blom, H. (2017). Handling cultural diversity in higher education. In R. Coelen, K.-W. van derHoek, & B. Herman (Eds.), *Valorisation of Internationalisation* (pp. 111–137). Retrieved from https://www.researchgate.net/publication/322235404_Handling_cultural_diversity_in_higher_education

Resnick, M., Maloney, Andrés Monroy-Hernández, J., Rusk, N., Eastmond, E., Brennan, K., Rosenbaum, E., … Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, *52*(11). https://doi.org/10.1145/1592761.1592779

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., … Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67. https://doi.org/10.1145/1592761.1592779

Richards, H. V., Brown, A. F., & Forde, T. B. (2004). What is Culturally Responsive Pedagogy. *Addressing Diversity in Schools*, 3. Retrieved from www.nccrest.org

Sacristán, A. I., Kaminskiené, L., Michaela, S., & Baafi, R. A. K. (2018). Constructionism 2018. *Constructionism in Upper Secondary and Tertiary Levels*, 925–939. Retrieved from http://www.constructionism2018.fsf.vu.lt/

Spiro, J. (2014). Learning interconnectedness: Internationalisation through engagement with one another. *Higher Education Quarterly*, *68*(1), 65–84. https://doi.org/10.1111/hequ.12031

Vaccarino, F. (2009). Learning and teaching in culturally diverse classrooms: A case study exploring teachers' and students' perceptions. *International Journal of Learning*, *16*(4), 105–118. https://doi.org/10.18848/1447-9494/CGP/v16i04/46231

Van Der Wende, M. (2010). Internationalization of higher education. *International Encyclopedia of Education*, *15*, 540–545. https://doi.org/10.1016/B978-0-08-044894-7.00836-8

Woods, P. R., Jordan, P. J., Loudoun, R., Troth, A. C., & Kerr, D. (2006). Effective teaching in the multicultural business classroom. *Journal of Teaching in International Business*, *17*(4), 27–48. https://doi.org/10.1300/J066v17n04_03

Debugging by Design: Students' Reflections on Designing Buggy E-Textile Projects

Deborah A. Fields, *deborah.fields@usu.edu*
College of Education and Human Services, Utah State University, Logan, UT, USA

Yasmin B. Kafai, *kafai@upenn.edu*
Graduate School of Education, University of Pennsylvania, Philadelphia, PA, USA

Abstract (style: Abstract title)

Much attention has focused on designing tools and activities that support learners in designing fully finished and functional applications such as games, robots, or e-textiles to be shared with others. But helping students learn to debug their applications often takes on a surprisingly more instructionist stance by giving them checklists, teaching them strategies or providing them with test programs. The idea of designing bugs for learning—or *debugging by design*—makes learners again agents of their own learning and, more importantly, of making and solving mistakes. In this paper, we report on our first implementation of "debugging by design" activities in a classroom of 25 high school students over a period of eight hours as part of a longer e-textiles unit. Here students were asked to craft buggy circuits and code for their peers to solve. In this paper we introduce the design of the debugging by design unit and, drawing on observations and interviews with students and the teacher, address the following research questions: (1) What did students gain from designing and solving bugs for others? (2) How did this experience shape students' completion of the e-textiles unit? In the discussion, we address how debugging by design contributes to students' learning of debugging skills.

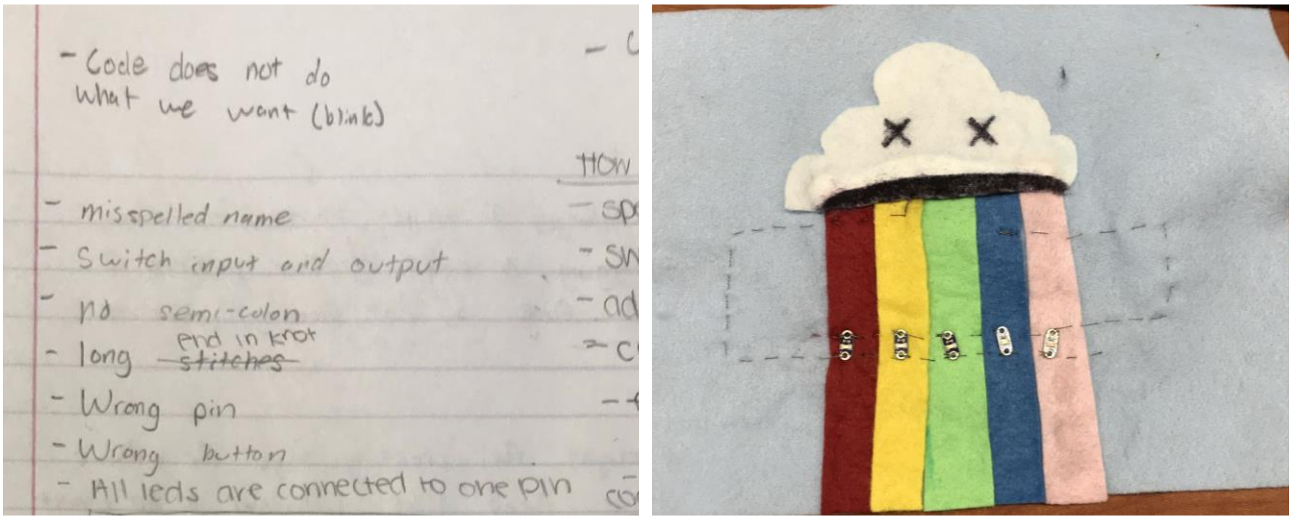


Figure 1. Example of Students' Buggy Project Design: Bugs (left) and E-textile (right)

Keywords

Programming, Debugging, E-Textiles, Computer Science Education, Physical Computing

Introduction

Much attention in Constructionism has focused on designing tools and activities that support learners as creators of fully finished applications—games, stories, robots or sandcastles—to be shared with others (Papert, 1980). Prior studies have provided substantial evidence for the benefits of learning programming in the context of designing personally relevant and complex applications rather than in writing short pieces of code (Kafai & Burke, 2014). When students design such complex applications, they often make mistakes—or bugs—of various types which hinder program completion. These bugs can range from simple syntactic problems such as forgetting commas or making typos (problems overcome by visual programming languages like Scratch) to more complex challenges that involve dealing with thorny run-time errors or logic design (McCauley et al., 2008). Helping students debug their programs often takes on a surprisingly instructionist stance: explicitly telling learners how to read and understand compiler messages or creating projects for them with specific bugs to identify and solve.

While these approaches to teaching debugging can provide valuable learning experiences, they ignore promising opportunities afforded by putting learners themselves in charge of designing buggy artifacts for their peers to solve. This approach builds on earlier successful work where students were asked to become designers of software (or games) to teach younger students in their school about academic content, such as mathematics or science (Harel, 1990; Kafai, 1995). Here we propose a paradigmatic twist to this idea by having students intentionally design *buggy* (rather than functional) software, games, robots, electronic textiles (e-textiles, see below), or other computational artifacts to teach them about bugs and the debugging process. The idea of designing bugs for learning—or "debugging by design" (DbD)—builds on the core tenet of constructionism that learners are agents of their own learning and construct mental models when they create things; in this case mistakes. Debugging by design provides a promising pedagogical application of constructionism.

In this paper, we examine the feasibility of DbD for learning and teaching about debugging. We present findings from a classroom study in which students created and then solved buggy projects for each other. The study took place in a 12-week long e-textiles unit in a 9th grade high school introductory computing class in the United States. E-textiles involve stitching circuits with conductive thread to connect sensors and actuators to microcontrollers (Buechley et al., 2013). Making an e-textile involves not only designing functional circuits but also writing code that controls interactions—thus providing myriad opportunities for bugs in crafting the physical artifact and designing the circuits and programming (Litts Lui, Widman, Walker, & Kafai, 2017). We implemented DbD in a classroom of 25 high school students over a period of 8 hours about three-quarters of the way through the larger e-textiles unit. In this paper we introduce the design of the debugging by design unit and, drawing on observations and interviews with students and the teacher, address the following research questions: (1) What did students gain from this experience? (2) How did participation in DbD shape students' completion in the e-textiles unit? In the discussion, we address opportunities and challenges that debugging by design provides in student learning.

Background

In general, teaching debugging strategies has focused on succinct screen-based problems with one bug in a piece of code. Yet bugs that arise in open-ended design situations are often much more complex, with multiple problematic bugs overlapping and difficult to identify. While there is significant research around tools and programming environments designed to support learners through the process of debugging (e.g., Sorva, Lönnberg, & Malmi, 2013), there is little research about how to support learners dealing with multiple debugging challenges in their designs. Understanding and designing learning environments that support debugging is especially important since the complexity associated with debugging demands not just programming skills but also other skills such as decision-making, emotional intelligence, and perseverance (e.g., Patil & Codner, 2007).

In addition, in physical computing contexts of e-textiles or robotics, bugs may occur both on-screen and off-screen and even in ways that stretch across both spaces, creating even more challenging situations for students seeking to isolate, identify, and fix problems. Students must not only attend to a variety of code-based errors, but also to construction errors such as incorrect placement of a sensor or incorrect circuitry, and even crafting issues in an e-textile project (Fields, Jayathirtha, & Kafai, 2019). Thus, identifying, debugging, and solving these problems is at the crux of being able to design functional computational and engineering projects. Researchers have taken these insights to design problem sets for physical computing to teach students about programming concepts and debugging. For instance, Sullivan (2008) presented students with a carefully designed set of robotics dilemmas and examined how the intricate inquiry skills students demonstrated. Fields, Searle and Kafai (2016) developed identical e-textile problem sets that students collaboratively solved. More recently, Jayathirtha, Fields and Kafai (2019) carefully studied students' debugging strategies and practices in navigating the multi-representational problem space of carefully designed buggy e-textile projects, particularly the system-level strategies students employed as they worked across spatial, material, and programmed issues. These studies provided early insights that researcher-designed problem sets in physical computing contexts are useful as meaningful assessment tools and for teaching students about debugging problems.

In this study, we approached teaching of debugging skills from a different perspective: rather than using researcher-designed problem sets, we engaged students themselves as designers of problem sets, or buggy e-textiles, for their peers to solve. A critical impetus for this approach to teaching debugging came directly from constructionist theory itself, which emphasizes the need to provide more agency to students as learners. Furthermore, we saw complex debugging situations as opportunities, or contexts for "productive failure." The concept of productive failure, introduced by Kapur (2008, 2012, 2015), highlights the counterintuitive notion that failure precedes later success in learning. Kapur's idea for designing effective learning activities was to provide student groups with carefully designed tasks in which early failures help them later in completing the problems more successfully. Today's extensive research on productive failure focuses on better understanding the role of multiple representations and solutions, their role in activating prior knowledge, and the nature of peer support during the problem generation phase to identify which dimensions are most productive for which students and under what conditions (Kapur & Rummel, 2012; Kapur & Bielaczyc, 2012). Most of these studies, however, have focused on getting students to solve well-defined canonical problems rather than on the role that failure plays in solving open-ended design problems more common in software and engineering applications such as e-textiles (Searle, Litts, & Kafai, 2018). Moreover, researchers or teachers designed these problems that provided students with experiences of productive failure. In our constructionist approach to productive failure, the learners themselves become the designers of challenging problem sets. Our research focused on understanding how such design experiences of productive failure can be realistically implemented by students within a classroom setting, how the teacher facilitated the design, and what students thought about their experiences.

Methods

Participants

The participating school was located in a large metropolitan school district in the southwestern United States. This particular class included 11 girls and 14 boys from 14-18 years old: 72% speaking a language other than English at home, 80% with no prior computer science experiences prior to the course, and 20% with no family members with college experience. The class was racially diverse, with 48% Latino, 36% Asian American/Pacific Islander, 8% White, 4% Other, and 4% race not reported. The teacher had three prior years of teaching the e-textile unit and helped co-develop the DbD unit.

Data Collection and Analysis

Data for our analyses was drawn from daily fieldnotes and teacher reflections during the DbD unit, student reflections (n=24 students) written after the unit, and post-interviews with pairs of students (n=21 students) and the teacher, Ben, after the entire e-textile unit was completed,. Analysis primarily consisted of two-step, open coding of transcribed interviews and reflections (Charmaz, 2014), with several meetings amongst the research team (four people) to interpret, revise, and re-apply the coding scheme consistently. We compared student interview and reflection analysis with the teacher's daily reflections and field note observations to provide a fuller picture of the experience and expand on design recommendations.

Design & Context

Broader Educational Context: E-Textiles Unit for ECS

The DbD unit took eight hours, about three-fourths of the way through the e-textiles unit designed for *Exploring Computer Science* (ECS), a year-long course providing an introduction to computing with equity-focused and inquiry-based teaching (Goode, Chapman, & Margolis, 2012; http://exploringcs.org/e-textiles). The e-textiles unit took place over 12 weeks and consisted of a series of four projects that allow increasing flexibility in design and personalization in the context of learning challenging new technical skills: 1) a paper-card using a simple circuit, 2) a wristband with three LEDs in parallel, 3) a classroom-wide mural project completed in pairs that incorporated two switches to computationally create four lighting patterns (the only collaborative project), and 4) a project that used handmade sensors to create conditions for lighting effects (see Kafai & Fields, 2018).

Debugging by Design Unit

The DbD unit contained several intentional characteristics. First, it took place in the latter half of the larger e-textile unit, allowing students to build on their experiences of errors in designing their buggy projects (or DebugIts) and to apply their experiences with DbD in their final projects. Second, the unit began with partner and group discussions where students named all of the problems that had come up in their designs by a given point and then categorized these problems. This promoted class-wide transparency of problems arising across students' prior projects. Third, students had to receive teacher approval on their DebugIt designs before they could construct them. This requirement was added in the moment during the unit, after the teacher noticed that many groups were proposing identical types of problems. The approval process enabled the teacher to challenge students to either make problems more interesting and creative or consider whether the problems they created were potentially solvable within a single class period. In other words, students needed feedback on both challenging and containing the level and number of problems they introduced. Fourth, after students exchanged and solved each other's problems, they presented their solutions to the class, letting the designers see to what degree and how their peers had solved the problems they designed. Finally, the class engaged in a reflective journaling and discussion about how they felt about designing and solving DebugIts, and what kinds of strategies they employed in solving problems. Table 1 shows the timeline of debugging by design.

Table 1. Debugging by Design Unit.

Class 1	**“Hall of Problems”:** As partners then as a whole class, students list e-textile problems. Then they categorized these problems into groups, which are written on posters on the classroom walls.
Class 2	**DebugIt Design:** Students plan their DebugIts, turning in a list of problems with solutions as well as a circuit diagram showing any circuitry bugs. Designs had to be approved by the teacher. Most groups revised their designs after teacher feedback, which continued into Class 3.
Classes 3-5	**DebugIt Construction**: After receiving teacher approval on their design, students created their DebugIts, sewing, and coding their projects.
Classes 6-7	**DebugIt Solving:** Directed by the teacher, students exchanged projects and had 1.5 class periods. Students then reflected on what the best, most frustrating, and surprising parts of the entire debugging by design experience were.
Class 8	**Reflection on Problem-Solving Strategies:** Individually then as pairs and as a class, students reflected on the kinds of strategies they used to solve DebugIts.

DebugIts had to contain at least six problems: at least two problems had to involve code, with one being undetectable by the Arduino compiler. This latter constraint was added after earlier pilots of DbD where we noticed that students tended to focus on simple syntax problems in their DebugIt designs. The requirement to create a bug undetectable by the compiler generally led to more challenging coding problems. The DebugIt had to involve either a switch or a sensor to ensure a level of coding challenge similar to the interactive mural project (i.e., with conditionals and functions). Finally students had to include a description of how the project should act when complete. This allowed for the inclusion of design errors (or “intention errors” as the class named them) where a project might function but not as desired. The final DebugIt design included: a list of problems and solutions, a circuit diagram showing any circuitry errors, code, and a statement of how the DebugIt should work.

Findings

Students’ Reflections as Buggy Designers

There was a dramatic difference in students’ feelings about the DbD process immediately after the experience and several weeks after their final projects were complete. In written reflections after solving their peers’ DebugIts most students in the class explicitly expressed frustration. They noted that the errors were difficult to detect, often involved a lot of cutting and re-sewing to fix, and generally did not have enough time to solve everything. As Avery clearly expressed, “It felt weird debugging someone else's project because they INTENTIONALLY (sic) put more bugs than a normal project would have so fixing obvious mistakes were easy but if the stitching was perfectly fine but it was crossed or flipped over I felt uncomfortable and MAD”. Some students also found it interesting or even fun to debug, but the most-repeated word (by 18 students) was “frustrating”.

However, after they had completed e-textile unit, students expressed feelings of comfort and competence with solving and designing problems in their interviews. All students interviewed said that the DbD unit should be done again next year because it was such a good learning experience for them. Some even asked for it to be done earlier and more often! As for what they appreciated about the unit, many students remembered making problems as fun and mischievous: creating problems “challenging enough to stress someone out is kind of funny and good” (Nicolás). They claimed that this gave them a “new perspective on coding” (Liam) that was the “opposite” from what they normally experienced, making it both challenging and interesting. Further, with some retrospective distance from solving DebugIts, many students also felt that solving others’ bugs “was fun” because in figuring out the problems they learned how to “fix it ourselves” (Noah). Related to this, many students felt DbD helped them to feel more comfortable with problems in general. As Evelyn said, “I think it helped me realize I knew if I saw the errors in the next one, I knew how to fix it.” Being more comfortable with problems helped students feel better able to ask for help from others since they became more aware that “a lot of people make mistakes” (Camila). This sense of capability or power over problems seems to have been most apparent after they were able to apply their learning from the DbD experience to a later project.

Not only did students feel more comfortable with problems, they also reported learning several important aspects about problems and problem-solving from the DbD unit. All students interviewed claimed to have applied their DbD experience to their final projects. They explained this in several ways. Many students described becoming more familiar with a range of problems in e-textiles. As Nicolás stated, “It helps them get a better understanding of the type of errors there are, like how errors can be prevented and caused and everything.” He and others said that they knew more problems: “common issues, common errors, more hard-to-see errors” (Liam). Knowing more problems also enabled students to avoid problems in their final project creation. Gabriel explained this succinctly:

> “[P]utting in your own bugs into a project and fixing them from another group… makes you more aware that the next time you're creating a project, that you know that those bugs exist and… you know whether you need to watch to make sure you don't make that mistake in the project.”

Of course, avoiding some problems in their final projects did not mean that no errors occurred. All students described problems that arose in their final projects. However, one thing that was different for many students was that they were better able to identify problems more easily. Some students were able to see coding errors more easily because they were more familiar with interpreting compiler feedback on syntax errors. Others learned processes for detecting errors more effectively: “learning how to spot [errors]” (Logan), using “the process of elimination just see which part is a problem, is it the stitching or the code?” (Nathan), or simply “testing” thread lines or code lines to see if they worked (Lucas). Overall, nearly all students interviewed reported several ways that they applied new knowledge about problems and problem solving from the DbD unit in their final project.

Two other interesting areas of learning came up in students’ interviews about what they learned from the DbD unit. First, many students said that they actually learned new code. Most often this arose from debugging projects where three groups of students used coding techniques that had not been introduced to the entire class: fading lights, playing music, and using a light sensor (instead of a switch). Each of the students in the groups who received their projects reported learning new content about code, and some of the students who designed those projects also claimed to learn new content because of this design opportunity. In part these introductions of new code resulted from the teacher’s constructive feedback on designs, where he challenged certain groups to go further than they had in the mural project in the debugging project, demonstrating the value of this one-on-one design feedback in the mini-unit. Second, three students went further than others in describing an actual method of problem solving that they could apply to future e-textile projects. This often came up in how they helped other students after the DbD unit. As Liam described:

> “So when I was helping other people, I picked out important parts that we should look at first for them, and so I said, "look at the front of your code where you named everything." They should've named these correctly, and then after that, we looked at the threading.”

Here Liam described a methodical approach to identifying and solving problems in an e-textile project: looking first at the beginning of the code where students often mis-declare variables, then looking at the crafted circuits if the problem has not already been fixed. The methods these three students described demonstrate the beginnings of a systematic approach to debugging that is rare amongst novice coders (Simon, et al., 2008). The teacher had further insights into these findings.

Teacher Reflections on Classroom Implementation

The teacher’s daily and post-unit reflections, mirrored students’ experiences during the DbD unit and several weeks afterwards. Ben, the teacher, observed that students were highly engaged and interested in designing buggy projects for each other. Overall, "[the students] seem very focused. I did not see a single group goofing off or stuck on an idea" (DbD Day 2). In fact, one of the students who previously tended to show disinterest in the class, was more involved than usual because he liked to be mischievous. Ben also suggested that designing bugs made problems less "painful" because "it's like a puzzle and so you're creating puzzles for people" (DbD Day 2). This made bugs into a "designed experience" rather than an accidental one that interrupted one's personal design. Further, he found that designing bugs was a very valuable activity because "actually having to like think through all of the different things that could go wrong and trying to plan them in advance made some sort of like click in, in some of them" (post-interview). In other words, designing bugs forced students to think through them in a more intentional way than simply dealing with bugs as they came up in their personal projects.

At the same time, the teacher reflected that students' early designs featured simple problems. To counter this tendency, he used constructive criticism during design approval to challenge students to think more creatively and to consider the recipients, or audience, of their buggy designs. Indeed, this thoughtfulness about audience was reflected in many students' interviews (noted in the prior section) where they mentioned curtailing their bug designs to consider whether their peers would be able to fix them within a class period or whether the problems would be too many or too difficult for them to find.

When it came to debugging each others' projects, Ben noted the "excitement" and "hussle bussle" amongst the students. At the same time, he noticed that students often used naive strategies to debug projects, taking more difficult approaches than necessary to solve problems. In contrast, *after* DbD he observed that students had a stronger knowledge base for constructing their final project. Even more, after the entire e-textiles unit was complete and students had moved on to the next unit, Scratch, Ben commented that for the first time ever, "[students] have a system of debugging. It might not be complete, you know... they're still learning how to build that system. But at least it now exists. Because it didn't before." He expounded on this point in describing the ways that students were more independent in tackling problems in the Scratch unit than they had been in prior years, "meticulously and methodically trying to figure out what the bugs are," making hypotheses for what might be wrong and applying those systematically to identify and fix bugs. This suggests that, at least in Ben's class, the DbD experience helped students not only to apply knowledge about bugs and debugging to their e-textile designs but also supported them in more systematically debugging in another programming environment.

Discussion

In this paper we outline a new way to engage and empower students in debugging—by designing bugs for others to solve. Analysis of student and teacher reflections on the unit demonstrates high levels of interest and engagement with the DbD unit as well as benefits for after the unit. These benefits included a broader knowledge base of types of problems that can arise in designing and crafting e-textiles, improved abilities to detect and fix problems and collaborative problem solving with others, and finally greater confidence in debugging. The DbD approach may support students in near-transfer of debugging in what the teacher described as the beginning of a more systematic approach to debugging in at least one other domain after the e-textiles unit.

Our analyses of student and teacher reflections further suggests several important aspects of the design of the DbD unit, including constraints on the number and types of problems that students design, teacher feedback and approval process for the designs, and class-level reflections about types of problems, solutions, and problem-solving approaches. Attending to the audience of the designed bugs (i.e., for other students) as well as challenging students to be original in the problems they designed was important. Without these supports, students tended to create simplistic problems or problems that were too time-consuming to solve (i.e., that required substantial sewing to fix). Teacher constructive criticism allowed these naive conceptions of problem design to be a starting point but not an end point for students' bug designs. Further, it was very important that this unit took place within a larger unit where students had significant time to develop baseline knowledge of the domain (e-textiles) and problems that could arise as well as an entirely new project after the DbD mini-unit so that they could follow through on what they had learned. Thus the DbD mini-unit was a mid-point in the e-textiles unit, rather than a starting or an endpoint.

This paper represents our pilot efforts to design, understand, and study this new approach to debugging education. In upcoming months, we plan to study more closely the problems that students designed, their processes of design, and their processes of debugging. Further, our pilot implementation was conducted by one highly experienced teacher. Later studies need to look at how other teachers apply the DbD unit in different classrooms, the variances that occur, and best practices across multiple teachers for this approach. Debugging by design could also be applied to other domains in physical computing and general software design. We look forward to the continued explorations in empowering students to design bugs in projects in mischievous and creative ways and thus ultimately "becoming more articulate about one's debugging strategies and more deliberate about improving them" (Papert, 1980, p. 23).

Acknowledgments

This work was supported by a grant from the National Science Foundation to Yasmin Kafai (#1742140). Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF, the University of Pennsylvania, or Utah State University.

References

Buechley, L., Peppler, K., Eisenberg, M., & Yasmin, K. (Eds.) (2013). *Textile Messages.* New York, NY: Peter Lang.

Charmaz, K. (2014). *Constructing grounded theory*. Sage publications, Thousand Oaks, CA.

Fields, D. A., Kafai, Y. B., Nakajima, T. M., Goode, J., & Margolis, J. (2018). Putting making into high school computer science classrooms: Promoting equity in teaching and learning with electronic textiles in *Exploring Computer Science. Equity, Excellence, and Education,* 51(1), 21-35.

Fields, D. A., Searle, K. A., & Kafai, Y. B (2016). Deconstruction kits for learning: Students' collaborative debugging of electronic textile designs. In *FabLearn '16*, *Proceedings of the 6th Annual Conference on Creativity and Fabrication in Education*, ACM, New York, NY, 82-85.

Goode, J., Chapman, G., & Margolis, J. (2012). Beyond Curriculum: The Exploring Computer Science program. *ACM Inroads, 3*(2), 47-53.

Harel, I. (1990). *Children Designers*. Norwood, NJ: Ablex.

Jayathirtha, G., Fields, D., & Kafai, Y. (2018). Computational Concepts, Practices and Collaboration in High School Students' Debugging Electronic Textiles projects. *Proceedings of the International Conference on Computational Thinking Education 2018*. Hong Kong: The Education University of Hong Kong. 27-32.

Jayathirtha, G., Fields, D. A., & Kafai. Y. B. (2020). Pair debugging of electronic textiles projects: Analyzing think-aloud protocols for high school students' strategies and practices while problem solving. In *Proceedings of the 14th International Conference of the Learning Sciences*, Nashville, TN: International Society of the Learning Sciences.

Kafai, Y. B. (1995). *Minds in Play*. Mawhaw, NJ: Ablex.

Kafai, Y. B. & Burke, Q. (2014). *Connected Code.* Cambridge, MA: MIT.

Kafai, Y. & Fields, D. (2018). Some Reflections on Designing Constructionist Activities for Classrooms. *In Proceedings of the Constructionism Conference 2018*. Vilnius, Lithuania, 601-607.

Kafai, Y., Fields, D., & Searle, K. (2014). Electronic textiles as disruptive designs: Supporting and challenging maker activities in schools. *Harvard Educational Review*, *84*(4), 532-556.

Kapur, M. (2008). Productive Failure. *Cognition and Instruction, 26*(3), 379-424.

Kapur, M., & Kinzer, C. K. (2009). Productive failure in CSCL groups. *International Journal of Computer-Supported Collaborative Learning*, *4*(1), 21-46.

Litts, B. K., Lui, D. A., Widman, S. A., Walker, J. T., & Kafai, Y. B. (2017). Reflections on Pair E-Crafting: High School Students' Approaches to Collaboration in Electronic Textiles Projects. In *Proceedings of the International Conference of Computer Supported Collaborative Learning* 2017, Philadelphia, PA (2) 569-572.

McCauley, R., Fitzgerald, S., Lewandowski, G., Murphy, L., Simon, B., Thomas, L., & Zander, C. (2008). Debugging: a review of the literature from an educational perspective. *Computer Science Education*, *18*(2), 67-92.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.

Patil, A., & Codner, G. (2007). Accreditation of engineering education: review, observations and proposal for global accreditation. *European Journal of Engineering Education*, *32*(6), 639-651.

Searle, K. A., Litts, B. K., & Kafai, Y. B. (2018). Debugging Open-Ended Designs: High School Students' Perceptions of Failure and Success in an Electronic Textiles Design Activity. Thinking Skills and Creativity.

Simon, B., Bouvier, D., Chen, T.-Y., Lewandowski, G., McCartney, R., & Sanders, K. (2008). Common sense computing (episode 4): debugging. *Computer Science Education, 18*(2), 117-133.

Soloway, E., & Spohrer, J.C. (Eds.). (1989). Studying the novice programmer. Hillsdale, NJ: Lawrence Erlbaum.

Sorva, J., Lönnberg, J., & Malmi, L. (2013). Students' ways of experiencing visual program simulation. *Computer Science Education*, *23*(3), 207-238.

Sullivan, F. R. (2009). Robotics and science literacy: Thinking skills, science process skills and systems understanding. *Journal of Research in Science Teaching*, 45(*3*), 373-394.

Deep Learning Programming by All

Ken Kahn, *toontalk@gmail.com,*
Dept of Education, University of Oxford, Oxford, UK

Yu Lu, *luyu@bnu.edu.cn, Advanced Innovation Center for Future Education, Beijing Normal University, Beijing, China*

Jingjing Zhang, *jingjing.zhang@bnu.edu.cn, Big Data Centre for Technology-mediated Education, Beijing Normal University, Beijing, China*

Niall Winters, *niall.winters@education.ox.ac.uk, Dept of Education, University of Oxford, Oxford, UK*

Ming Gao, *mgao519@126.com, Big Data Centre for Technology-mediated Education, Beijing Normal University, Beijing, China*

Abstract

We describe an open-source blocks-based programming library in Snap! [Harvey and Mönig, 2010] that enables non-experts to construct machine learning applications. The library includes blocks for creating models, defining the training and validation datasets, training, and prediction. We present several sample applications: approximating mathematical functions from examples, attempting to predict the number of influenza infections given historical weather data, predicting ratings of generated images, naming random colours, question answering, and learning to win when playing *Tic Tac Toe*.

Keywords (style: Keywords)

Machine learning, Snap!, visual programming, constructionism, neural nets, artificial intelligence

Introduction

This work builds upon the work of Kahn and Winters [Kahn and Winters, 2018] who developed a visual programming library designed to be used by high school students in building AI applications. They first created Snap! blocks that connect to AI cloud services for speech and image recognition as well as speech synthesis [Kahn and Winters, 2017]. They then provided block-based interfaces to several pre-trained deep learning models for services such as transfer learning, pose detection, style transfer, and image labelling. Here we present new additions to this Snap! library that support the definition of the architectures of deep learning models, their loss functions, their optimization methods, training parameters, and obtaining predictions [eCraft2Learn, 2020a]. In addition to the library itself, there are learning resources including interactive guides and sample projects

Motivations

In the last decade machine learning systems have demonstrated very impressive capabilities including image recognition, translation, autonomous driving, speech recognition, interpretation of medical imagery, transcription, recommendation generation, robot control, game playing, and much more. It is likely to continue greatly affecting nearly every aspect of modern life. Students who experience in a hands-on manner the possibilities, strengths, and weaknesses of this technology are likely to obtain a deeper understanding than those who simply study the technology. This viewpoint reflects on Dewey's (1938) 'learning by doing' theory, which emphasises the value of experience and engagement. While we expect a small fraction would go on to become AI researchers or engineers, we expect the rest will be better prepared to contribute, in an informed manner, to a society rapidly changing due to the impact of AI. Such students are likely to be better equipped to deal with the social, economic, and ethical issues that are arising from the use of machine learning.

We aim to provide high-school and non-computer science undergraduate students with limited programming abilities with tools for acquiring experiences designing, training, testing, and using deep machine learning models. We believe that our reliance upon a blocks-based language has many advantages over simply providing machine learning “wrappers” in JavaScript or Python. Students have fewer distractions such as syntax errors. Their cognitive load is less since they rely upon drag and drop of blocks instead of needing to remember the names of primitives. The blocks can be very readable without the usual cost involved in entering verbose instructions. The Snap! blocks provide an intuitive interface for asynchronous functions, a construct that many learners find difficult in textual languages. And a great number of students have familiarity with Scratch [Resnick et al. 2009] upon which Snap! was based. While these blocks have yet to be used in studies with our intended audience, we expect the kinds of success we have seen with other parts of our Snap! AI library [Kahn and Winters, 2018; Kahn et al, 2018; Loukatos et at, 2019] when we run trials of these machine learning blocks.

The students who master our machine learning library can become empowered to build impressive apps that listen, see, predict, and more. This may motivate them to create innovative applications that match their interests and passions. Furthermore, in the process they may acquire the ability to reflect more deeply upon how they perceive, reason, and act. While deep learning neural nets are very different from brains, and how they perform and are structured is different from minds, they are still useful models of cognition. And perhaps students who acquire concepts for more effectively thinking about thinking may become better learners [Papert, 1980; Minsky, 2019; Kahn and Winters, 2020].

Related Research

While there is a great deal of support for building deep neural networks in Python and JavaScript, we are focused on supporting learners lacking the technical skills to effectively use those resources. Mathematica’s Wolfram Language has a good deal of support for machine learning including learning resources aimed at middle and high school students [Wolfram, 2017a; Wolfram, 2017b]. Our efforts differ from this in that we are building upon the ease-of-use and familiarity of blocks-based languages such as Scratch [Resnick et al., 2009] and Snap!. Furthermore, Snap! and our library are open-source and run in modern browsers without any installation requirements.

There are other efforts to integrate machine learning with blocks-based languages including the Machine Learning for Kids website [Machine Learning for Kids, 2020] and the Cognimates project [Druga, 2018]. These systems, like the earlier work of Kahn and Winters, offer blocks that provide easy-to-use access to various AI cloud services. Unlike our current efforts, they do not provide a programmatic interface for constructing neural networks – the programmatic interfaces they provide are only for training and using neural nets.

SnAIp is a project that aims to implement machine learning techniques in Snap! [Jatzla et al, 2019]. Unlike the Snap! blocks described in this paper, there are no black-box implementations in JavaScript, and no reliance upon complex APIs or cloud services. Enabling students to see how machine learning works in terms of blocks they are familiar with clearly has advantages. But technically it is very difficult to achieve the speed and scale that our blocks are capable of. Also our blocks provide access to very powerful APIs that would be a tremendous effort to fully replicate in Snap!. Ideally students should have access to both of these Snap! libraries so they can incorporate both transparent functionality and very capable functionality into their projects as appropriate.

Google’s Teachable Machine [Google, 2020a] is a web page where users can train the system to classify images. The TensorFlow Playground [Google, 2020b] is a web page where one can interactively define, train, and test a deep learning neural net. While wonderful learning resources, these systems do not provide a programmer interface.

Snap! Blocks for AI Programming

Snap! [Harvey and Mönig, 2010] is a blocks-based programming system that closely resembles the immensely popular Scratch programming system [Resnick et al. 2009]. Snap! is a much more powerful and expressive language than Scratch because of its thorough support for first-class functions and lists. Also, crucial to our efforts, new blocks can be defined either in Snap! or in JavaScript. All the blocks in our library are either defined in JavaScript or in terms of other blocks that ultimately rely upon our JavaScript-defined blocks. Note that this implies that we didn't touch the source code of Snap! in implementing our extensions to Snap!. Student projects relying upon our blocks can be loaded into an ordinary Snap! web page.

Snap! itself is implemented in JavaScript and hence can run in any modern web browser without the need for any extensions or plugins. (Note that the neural network blocks work in Chrome and Firefox but currently not in most other browsers.) Snap! is open source, well-documented, supported by an active community, and under continual further development.

A blocks-based programming system has several advantages over text-based languages. Because blocks only click together if they are syntactically compatible, they eliminate the need to learn the syntax and, perhaps more importantly, eliminate the possibility of syntax errors or misspelled commands. A block can also be any mixture of text, input parameters, and icons, making it more readable. And blocks can easily be displayed in languages other than English.

Blocks are organized into palettes that enable users to browse for appropriate blocks. Consequently, users needn't memorize large numbers of language primitives. Well-designed learning activities could aid students to use blocks to reduce cognitive load during the learning process [Çakiroğlu et al., 2018]. While this relieves memory demands on users and facilitates the discovery of new functionality, it can be significantly slower than typing. Snap! addresses this by providing a keyboard method for searching for blocks.

In Figure 1 the block for obtaining a prediction from a model will either call the "say" block with the prediction for the input (36) or else the "think" block will be called with the error message. Note that while this block expects success and errors continuations (also known as callbacks) it does so in a manner accessible by beginning programmers.

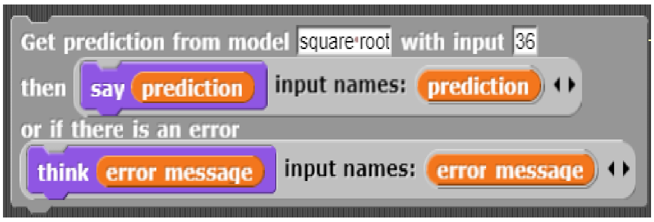


Figure 1 - Prediction block with two embedded blocks

Deep Neural Networks

The neural networks that can be built with our library consist of a large number of connected artificial neurons. These neurons are loosely based upon biological neurons. The connections between neurons have weights that encode how much a neuron influences another neuron. In our work, as is typical for deep learning models, the neurons are organized into layers. The first layer receives the input and the last layer produces the output. The layers in-between, called "hidden layers", typically produce successively higher-level features or interpretations of the data. Our library includes blocks for defining these layers and their connections.

Neural nets work exclusively with numbers. To work around this for classification tasks, numbers are used to encode labels. E.g., if the sentiment of some text is either "negative", "neutral", or "positive" this can be encoded as 0, 1, and 2. The training blocks accept text labels as outputs and converts them to "one-hot encodings" (vectors with one 1 and the rest 0). Image input is usually converted to a list of pixel values (either black and white (0 and 1), grayscale (0.0 to 1.0), or values for the red, green, and blue intensities). There are various schemes for converting words or sentences into a vector of numbers [Mikolov, 2013]. Other parts of our library provide blocks that can be used to convert images and text into vectors of numbers well-suited for machine learning.

The weights associated with neurons are randomly initialized. These weights are updated as the model is trained on data. In supervised learning the data includes examples of outputs associated with inputs. During training the system adjusts the weights in an attempt to reduce the difference between its predictions and the desired outputs given in the datasets. Our Snap! library provides blocks for controlling many aspects of this training phase.

Many of the concepts underlying neural nets are over fifty years old [Minsky & Papert, 1969] but only began leading to many thousands of useful systems in the last decade [Deepindex, 2020]. This recent success is usually attributable to much more powerful computation engines and the availability of large amounts of data. Computation engines typically exploit the graphical processing units (GPUs) found in most computers, tablets, and smartphones. These accelerators often decrease the time it takes to train a model or use it for predictions by a factor of one hundred or more.

In our work we are able to exploit the speedups from using GPUs due to the arrival of TensorFlow.js [Smilkov, 2019]. This is an implementation of TensorFlow, a popular machine learning API, in JavaScript. It can access the GPU of a laptop, desktop computer, or phone using the WebGL interface [WebGL, 2019] that is supported by all modern web browsers.

Training that relies upon big data can be a problem for students using our library. Students are likely to have problems acquiring millions of labelled images and importing them into a browser. And models built by professionals can take weeks to be trained on huge collections of images, even when using a large number of state-of-the-art GPUs or other accelerators. While many tasks are consequently impractical for students to attempt, fortunately, there remain many interesting tasks that don't require huge datasets or computing resources.

Most neural net applications run on servers that accept data from a client and respond with predictions. This enables the service providers to host their models on very powerful servers, sometimes on special hardware for accelerating machine learning. Many business models rely upon providing functionality via servers.

There are drawbacks however. Many people are concerned about privacy concerns when using these services. Voice and video are often transferred to the servers. The cost of using a server is often an obstacle. Another disadvantage of running the models on servers is that applications cannot be as responsive as ones that run locally on the user's devices. Also, applications that rely upon servers work only when the client has a fast and reliable network connection.

By running on a user's device in a web browser these drawbacks are avoided. One can download Snap! and our library and then run everything without an Internet connection.

A Library of Machine Learning Blocks

The Snap! library exists in two forms: (1) a set of blocks that can be imported into any Snap! instance or (2) a Snap! project that includes illustrative instances of the use of the blocks together with informative comments. One can simply click on any of the instances to run them.

Typically, we provide at least two versions of any block: (1) the simplest usable version and (2) a full-featured version.

Model Creation

The simple block for creating a model is illustrated in Figure 2. It creates a model named 'guess relation' with a single input that is connected to 100 neurons. Each of those neurons is connected in turn to 50 neurons that are connected to a single output neuron. When the system has finished creating the model the "say" block is run.

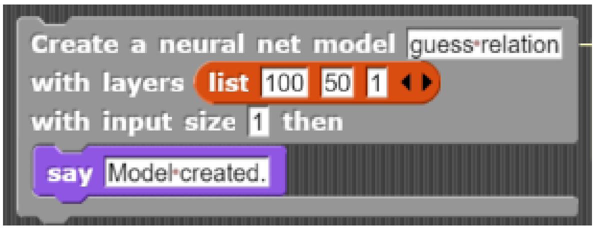


Figure 2- A simple block for creating a model

Figure 3 shows an example of the full-featured version of this block. It differs from the example in Figure 2 by specifying which optimization method and loss function are desired. Note that these are indicated by using pull-down menus for ease of use. Documentation of these methods and functions is provided via the help menu item and in the programming guide for greater detail. Examples of using this block for more difficult tasks are described later.

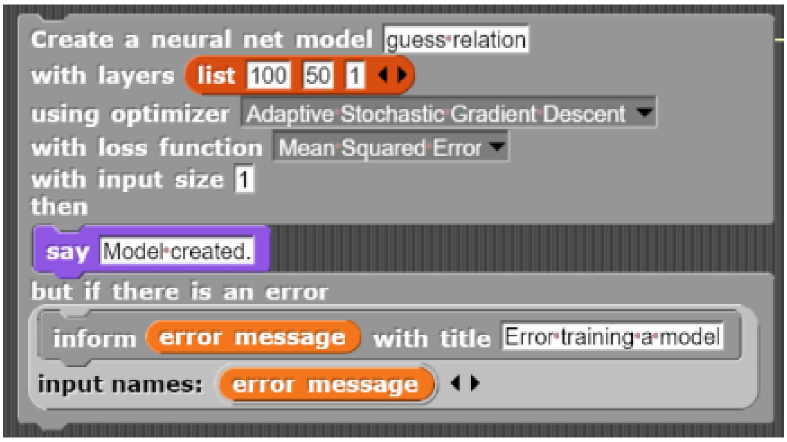


Figure 3 - A full-featured block for creating a model

Training

Once a model is created one can begin training it. First one specifies the training dataset and optionally the validation dataset. A dataset consists of two Snap! lists with the same number of elements: the input and the output. The lists can consist of numbers, or lists of numbers (e.g. coordinates or red-green-blue intensity triples), or any number of levels of lists. The output can also contain text strings that are converted to numbers internally. The validation dataset, if provided, does not influence the weights during training and is used to provide predictions free of the risk of overfitting for evaluating the model. An alternative to providing a validation dataset is to request that a specified fraction of the training data be set aside for validation.

In Figure 4 a dataset containing the first five positive integers is used as input and the output is computed using 2*n+1. (This is the machine learning analogue of the “Hello World” program – an extremely simple example.) This block can be used to either completely define the dataset or to provide data to be added to the current dataset. Datasets can be available for all models to use or be associated only with a specified model.

Figure 4 - Specifying the training dataset

After specifying the dataset one can begin training. Figure 5 illustrates the simplest block for initiating training. It requests 50 iterations of the training step and then displays the statistics from the training as shown in Figure 6.

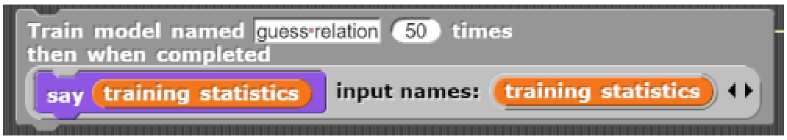


Figure 5 - A simple block to initiate training

loss	0.0014190125511959
accuracy	0
duration in seconds	3

Figure 6 - The resulting training statistics

Figure 7 is an example of using the full-featured version of the training block. It specifies a learning rate of .001, that the data should be shuffled (to avoid any artefacts resulting from the order), that none of the data should be used for validation. It also responds to any errors that arise.

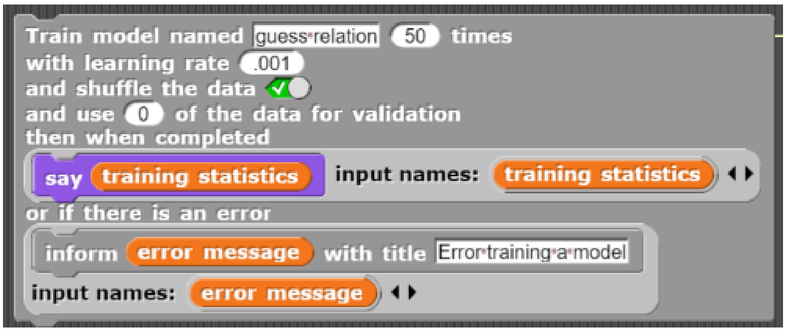


Figure 7 - A full featured block for initiating training

Prediction and Classification

In Figure 8 we see a block requesting that the model predicts the output given 10 as the input. For this example, the "say" block will be passed a number close to 21 (i.e., 2*10+1). There is a version of this block that accepts a list of inputs and replies with a list of corresponding predictions. If the model has been trained to label the input, then the output is a list of pairings of labels and confidence scores.

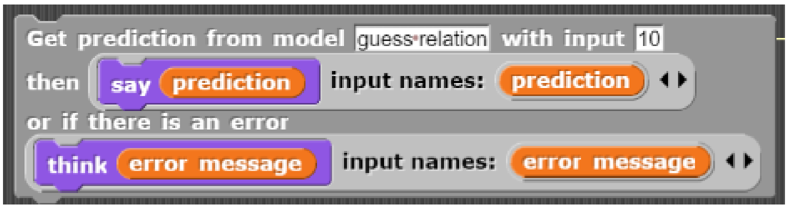


Figure 8 - A block for getting predictions from a model

A Graphical Interface for Creation, Training, and Prediction

One of the challenges in creating good deep learning models is that some architectures (number of layers and number of neurons in each layer) can be quickly trained and produce accurate predictions, while others are difficult to train or produce poor results such as bad predictions or classifications. Deciding the size of the training data is challenging. Small datasets lead to fast training but only sometimes produce good predictions. Similarly, it is hard to know what are good values for hyper-parameters such as the learning rate, number of training iterations, loss function, and optimization method.

One way we address this is to provide an optional graphical interface for setting all these parameters as illustrated in Figure 9. Students can use it to quickly try different parameters and teachers can use it to accompany a demonstration.

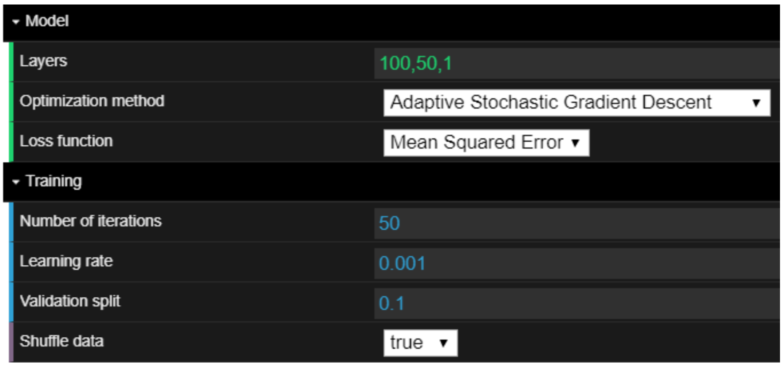


Figure 9 - A graphical interface for exploring architectures and hyper-parameter values

The graphical interface provides buttons for creating, training, and prediction. The training section provides real-time graphs of the training progress as seen in Figure 10. The x-axis is the number of training steps performed and the blue line shows the drop in the difference between predictions and the correct answers from the training data. The red line shows the difference for the validation dataset.

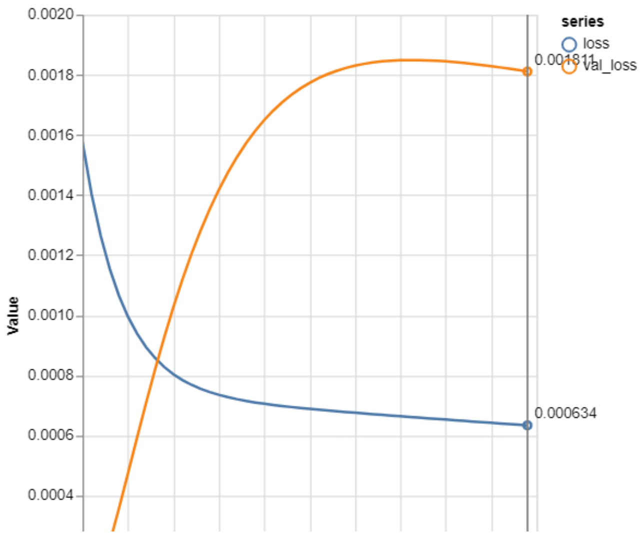


Figure 10 - a graph of the loss function during training

Hyper-parameter optimization

An alternative to tuning the model interactively is to let the computer search for good architectures and hyper-parameter values. While one can implement such a search in Snap*!* we also provide a block that interfaces to a TensorFlow.js hyper-parameter optimization library [Stoyanov, 2018]. Ambitious projects may find good settings much faster using this block than "manual" experimentation. We believe, however, that students can learn a good deal from some manual experimentation but it may become tedious for big projects.

Figure 11 shows a block that starts a search exploring 50 different parameter settings. As each experiment is performed the parameter values are displayed. The best settings are captured when the search completes, and another block can be used to create and train a model using those settings. Boolean switches are used to indicate what hyper-parameters to explore. Finally, weights are provided to guide the search towards the desired trade-off between accuracy, training time, and model size/speed.

The parameter search starts with the current settings and uses various heuristics to try values close to the currently most promising ones. The search can be customized in the graphical interface.

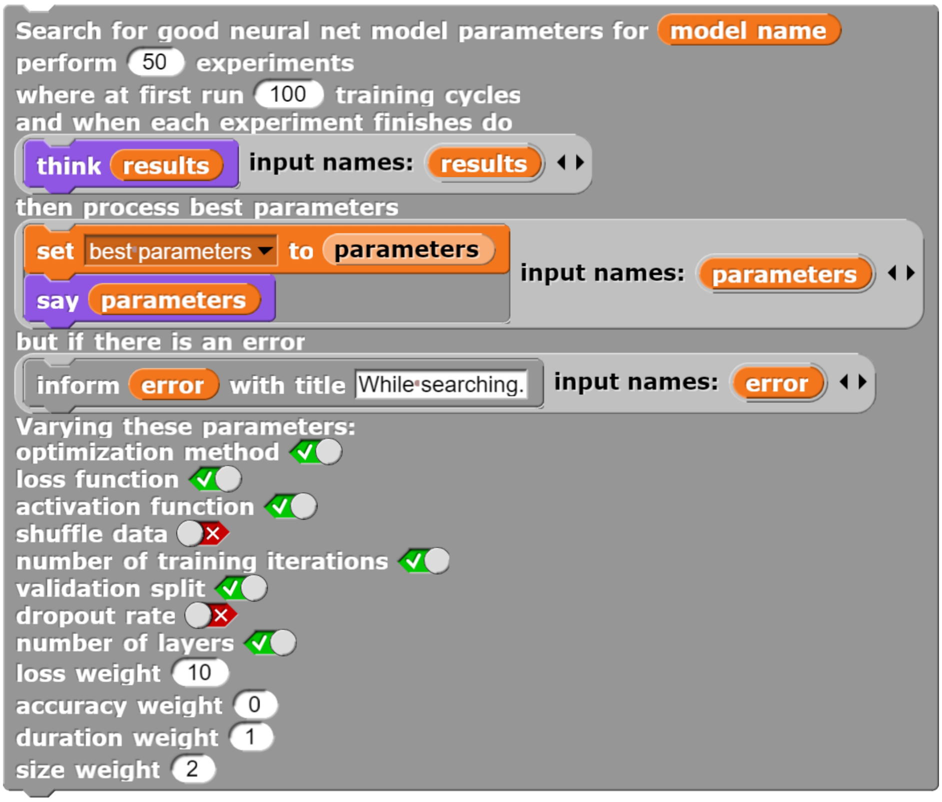


Figure 11 - Launching a search for good architectures and hyper-parameters

Illustrative Projects

Our library does not expose all the functionality of TensorFlow.js (which, while very powerful, supports only a subset of the full TensorFlow API). We decided to construct our library with only the most common and easiest to understand building blocks of deep learning programming. Despite this there is a wide variety of projects that it can support. Here we discuss several classes of projects that we developed, in increasing order of complexity.

Approximating Mathematical Functions

A pedagogically simple exercise using the deep learning blocks is to learn to approximate a mathematical function given sample input and output values. The early example of predicting 2x+1 is so simple that many alternatives to deep learning can work as well. A more interesting example is to provide a list of numbers as output and the square of those numbers as input. Attempting this one can discover how the architecture of the model and number of training examples strongly influences how well it can "predict" the square root of test numbers. Furthermore, if trained on numbers, say, between 1 and 100, one can explore how well it can approximate the square root of 1000 or 1/100. Figure 12 displays the approximations for 2, 1, 49, 900, and 0.01. Applying deep learning to mathematical functions need not be constrained to functions of one argument nor functions that produce a single output value.

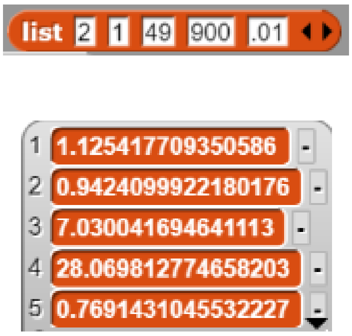


Figure 12 - A model's "predictions" of the square root of five test numbers

(notice how poorly it estimates the square root 0.01)

Some students may enjoy exploring the limits of this kind of technology. Can it reliably identify integers as prime or non-prime? Can it learn the prime factors of numbers? How well can it approximate transcendental functions?

Discovering Real-world Data Relationships

There are many freely available datasets that can be used as training data. Google Dataset Search is a convenient way to find them [Google, 2019c]. The example we explored starts with two datasets: the Global Historical Climatology Network-Daily Database [NOAA, 2019] and the Epi Data Surveillance Information from the World Health Organization [WHO, 2019]. The idea is that perhaps recent temperatures and precipitation can help predict subsequent occurrences of influenza. As is well-known, cases of influenza increase in the winter. Perhaps colder weather leads to more influenza in the following weeks. If there is such a relationship can deep learning discover it?

The first step is to import data into Snap! directly as CSV and JSON files. Alternatively, one can import the contents of a data file as a string and programmatically parse it. Snap! has all the mathematical and list processing primitives needed to support “data wrangling”. We experimented with input data being temperatures from the previous week and the subsequent number of reported flu infections. The temperature data was normalized to be the ratio of the temperature to the average temperature for each location for each time of year.

While this is an illustrative example of the kinds of explorations a typical high school student should be able to perform, we have yet to find any predictive value in knowing the previous week’s weather. Students can explore other relationships such as predictions from the previous several weeks. But negative results can be instructive as well. We remain optimistic that students will follow their interests in applying machine learning to data. There are also plenty of opportunities to tie these explorations to their other studies be it science, history, social studies, athletics, or language. Many machine learning projects can be developed to answer questions about epidemics such as covid-19.

Learning to Win (at *Tic Tac Toe*)

Deep learning has had several impressive accomplishments in game playing. DeepMind built a system that learned to play dozens of Atari video games [Mnih et al., 2013]. They later created AlphaZero that learned to play *Go* and *Chess* at world-class levels [Silver et al., 2017]. We have explored how students can do this themselves for simple games such as *Tic Tac Toe*.

We provide a Snap! implementation of *Tic Tac Toe* since our focus is on using machine learning to discover winning ways of playing and we are ignoring the challenge of learning the rules of play. We frame the problem as one of predicting the probability of winning given a specific board. A board consists of 9 squares that can be X, O, or empty. One possible input to a neural net can be a vector of 9 instances of 0, 1, or 2. However, these numbers don't work as well as using a vector of 27 instances of either 0 or 1. Each board square is either <0, 0, 1>, <0, 1, 0> or <1, 0, 0>.

As a game is played a list of successive board positions is recorded. When the game ends then the outcome is encoded for each board that occurred during the game: 0 for a tie, 1 for a win, and -1 for a loss. After training, a model can be used to make moves by considering the boards resulting from all possible next moves. It can then choose the move that is most likely to lead to a win or instead choose a random move based upon the probability of that move winning.

It is not difficult to seed this process by collecting moves from games where a purely random player plays against another random player. After using this game play data as training data, one can begin to have a model play itself, a random player, another trained model (perhaps by a different student), or a human player. The moves from these games can then be used for further training.

We also provide a web page where one can experiment interactively with deep learning for *Tic Tac Toe*. Unlike the Snap! *Tic Tac Toe* program, the webpage does not yet support human players. It does, however, facilitate large scale experimentation since it can play hundreds of games per minute.

This approach to learning to win at a game relies upon the fact that the state of the game is known to all players and can be concisely and simply represented and that the number of possible moves on each step is small. Consequently, many popular games are too complex to learn to play them well in a manner similar to *Tic Tac Toe*. There are, however, several games that could use this approach. *Connect the Dots* and *Nim* are good examples.

More sample projects

We provide other examples of machine learning using the Snap! blocks. One generates random images and learns to rate them. Another asks the user to name randomly generated colours and then learns to predict the name of additional colours. Another project is a question answering system that can answer questions about the Snap! AI blocks library.

Learning Resources

For many students a library of deep learning programming blocks is not enough. They need tutorials, guides, sample programs, and clear documentation. In addition to the examples and documentation provided by the library presented as a Snap! project, we provide an interactive web-based guide [eCraft2Learn, 2020b]. Within the guide are instances of Snap! that enable readers to explore the blocks and examples on the same page as they are reading the guide. The guide by default also contains sections describing the underlying ideas, history, project ideas, links to videos and further information, and societal impacts. For students who are focused on programming these can easily be hidden.

Current Status and Future Developments

This paper reports on the design, implementation, and motivations behind our deep learning programming library and its associated learning resources. At the time of this writing we have tested this with only two high school students and two university students. We learned that unless the students understand the larger context and challenges of machine learning they are mystified why the computer can't more easily and accurately figure out, for example, square roots. Or even why one would need to train it since "it already knows how to compute square roots". Toy examples can be pedagogically valuable but only if presented as learning exercises and not as example of serious machine learning.

We have plans to test our library and learning resources with non-computer science university students in at least two classes. Given how well students do using Snap! libraries for speech synthesis and recognition, using pre-trained neural nets, and word embeddings, we are optimistic students will master our deep learning library.

The source code, libraries, guides, sample projects, and additional documentation are all freely available online [eCraft2Learn, 2020a].

Currently our library has only been thoroughly tested in Chrome on personal computers. We have yet to succeed in getting it to run reliably on tablets or smartphones.

Ideally high-level building blocks should support not only construction but "deconstruction", i.e. transparency in how they function. Users ideally should be able to open up our blocks to see how they work and modify them. While we see that is as very valuable it is very challenging in this case and beyond the scope of this project. But [Jatzlau et al 2019] demonstrate this is possible in some cases.

There are many technical enhancements we are considering. All of the models that can be created now are a sequence of fully connected layers. Convolutional layers are not yet supported; nor are recurrent networks and reinforcement learning. The ultimate challenge is to support all the functionality in TensorFlow.js in a manner that is accessible to people who are not expert users of some textual programming language such as Python or JavaScript.

Acknowledgments

Initially the work reported here was supported by the eCraft2Learn project funded by the European Union's Horizon 2020 Coordination & Research and Innovation Action under Grant Agreement No 731345. Subsequently it is partially supported by the National Natural Science Foundation of China (No. 61702039) and the Fundamental Research Funds for the Central Universities.

References

Çakiroğlu, Ü., Suiçmez, S. S., Kurtoğlu, Y. B., Sari, A., Yildiz, S., & Öztürk, M. (2018). Exploring perceived cognitive load in learning programming via Scratch. *Research in Learning Technology*, 26.

Deepindex (2020) https://deepindex.org

Dewey J. (1938) Experience and Education. Macmillan, New York, NY.

Druga, S. (2018) Growing up with AI: Cognimates: from coding to teaching machines., PhD diss., Massachusetts Institute of Technology.

eCraft2Learn (2020a) https://ecraft2learn.github.io/ai/

eCraft2Learn (2020b) https:// ecraft2learn.github.io/ai/AI-Teacher-Guide/chapter-6.html

Google (2020a) https://experiments.withgoogle.com/teachable-machine

Google (2020b) https://playground.tensorflow.org

Google (2019c) https://toolbox.google.com/datasetsearch

Harvey, B. & Mönig, J. (2010) Bringing "No Ceiling" to Scratch: Can One Language Serve Kids and Computer Scientists? *Constructionism 2010 Proceedings*, Paris, France.

Jatzlau, S., Michaeli, T., & Seegerer, S. (2019). It's not Magic After All - Machine Learning in Snap! using Reinforcement Learning. https://www.semanticscholar.org/paper/It-%E2%80%99-s-not-Magic-After-All-Machine-Learning-in-Snap-Jatzlau-Michaeli/6ffb305d63d33d9f1530db6883083e6f1a8e1ca3

Kahn, K. & Winters, N. (2017) Child-friendly programming interfaces to AI cloud services, *Proceedings of EC-TEL 2017: Data Driven Approaches in Digital Education*, 10474, 566-570.

Kahn, K. & Winters, N. (2018) AI Programming by Children, *Proceedings of Constructionism 2018*, Vilnius, Lithuania, 2018.

Kahn, K, Megasari R., Piantari, E. & Junaeti, E. (2018) AI Programming by Children using Snap! Block Programming in a Developing Country, Proceedings of the *EC-TEL Conference*, Leeds, UK, September 2018.

Kahn, K. & Winters, N. (2020) Constructionism and AI: A history and possible futures, *Proceedings of Constructionism 2020*, Dublin, Ireland, 2020.

Loukatos, D., Kahn, K. & Alimisis, D. (2019) Flexible Techniques for Fast Developing and Remotely Controlling DIY Robots, with AI flavor. Proceedings of the *Edurobotics 2018 Conference*, Rome, Italy, October 2018. In Moro, M., Alimisis, D. and Iocchi, L., 2019. *Educational robotics in the context of the maker movement*, Springer.

Machine Learning for Kids (2020) machinelearningforkids.co.uk.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). "Distributed representations of words and phrases and their compositionality." In *Advances in neural information processing systems*, pp. 3111-3119.

Minsky, M. (2019). *Inventive Minds: Marvin Minsky on Education*. MIT Press.

Minsky, M. & Papert, S. (1969). *Perceptrons: An introduction to computational geometry*. MIT press.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). "Playing Atari with deep reinforcement learning." arXiv preprint arXiv:1312.560.

National Oceanic and Atmospheric Administration (2019). https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/readme.txt

Papert, S. (1980) *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. and Kafai, Y. (2009), Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67 DOI=http://dx.doi.org/10.1145/1592761.1592779.

Silver, D., Schrittwieser, J., Simonyan, K. Antonoglou, I., Huang, A, Guez, A., Hubert, T. (2017). "Mastering the game of go without human knowledge." *Nature,* 550, no. 7676: 354.

Smilkov, D., Thorat, N. Assogba, Y., Yuan, A., Kreeger, N., Yu, P., Zhang, K. (2019) "TensorFlow.js: Machine Learning for the Web and Beyond." *arXiv preprint arXiv:1901.05350*.

Stoyanov, M., (2018).HPJS: Hyper-parameter Optimization for JavaScript, *Medium*, https://medium.com/@martin_stoyanov/hpjs-hyperparameter-optimization-for-javascript-8f78aa7a3368.

WebGL (2019) https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API

World Health Organization (2019) http://apps.who.int/flumart/Default?ReportNo=16

Wolfram, S. (2017a) *An Elementary Introduction to the Wolfram Language, Second Edition*. Wolfram Media.

Wolfram, S. (2017b) Machine Learning for Middle Schoolers. Stephen Wolfram blog. http://blog.stephenwolfram.com/2017/05/machine-learning-for-middle-schoolers/.

Developing pre-service teachers' understanding of computational thinking: a constructionist lens

Deirdre Butler, *deirdre.butler@dcu*
Institute of Education, Dublin City University, St Patrick's Campus, Dublin 9, Ireland

Margaret Leahy, *margaret.leahy@dcu.ie*
Institute of Education, Dublin City University, St Patrick's Campus, Dublin 9, Ireland

Abstract

While it has been argued that computational thinking is most effective when introduced at primary and early secondary education (Fletcher & Lu, 2008), the development of computational thinking at primary school level is an area of research still in its infancy (Angeli et al., 2016). Irrespective of what approach is taken, the teacher is central to ensuring that the children they work with develop computational thinking. It is therefore essential that they are adequately prepared to include computational thinking as part of their pedagogical classroom practices (Yadav et al., 2011; Lye & Koh, 2014). Moreover, this preparation should begin at pre-service level (Barr & Stephenson (2011) so that pre-service teachers not only develop understandings of computational thinking but are also introduced to ways they can design learning opportunities for their students to develop computational thinking (Wing, 2016).

Adopting the stance that computational thinking should be developed as part of subject areas other than computer science, this paper presents and discusses the findings from a study which investigated preservice teachers' understandings of computational thinking on completion of a specialism in digital learning as part of their Bachelor of Education (B.Ed) degree at the Institute of Education of Dublin City University (DCU IoE). The design of the specialism was informed by constructionist principles and sought to develop student understandings of computational thinking in a progressively developmental manner and where the final activity entailed using computational tools with children in the classroom within the context of the extant primary school curriculum.

18 pre-service teachers who had completed the Major Specialism in Digital Learning participated in the study which was qualitative in its design. Data collection methods included group interviews and document analysis.

Two broad themes are presented as findings, (i) development of the pre-service teachers computational thinking and (ii) the importance of an immersive classroom experience as part of this development. The pre-service teachers demonstrated the development of a "computational thinking language" (Lu & Fletcher, 2009) with an ability to define and explain the concepts of computational thinking pointing to specific examples they observed while working with children in the classroom and relating their observations to extant research literature. The importance of the classroom experience emerged as key towards deepening the preservice teachers' understandings. They all strongly emphasised that working with children in the classroom brought the theory 'to life, deepening their personal understandings of constructionism and what computational thinking looks like "in action". The insights gained from this study are particularly relevant for the design of teacher preparation programmes indicating how CT can be effectively embedded to combine theory and practice. This will ensure that CT concepts are not developed in a decontexualised manner but are embedded within the prescribed curriculum across a range of subject content in a relevant and meaningful manner.

Keywords (style: Keywords)

Constructionism. Computational Thinking, Pre-service Teachers, Learning Design

Introduction

While it has been argued that computational thinking is most effective when introduced at primary and early secondary education (Fletcher & Lu, 2008), the development of computational thinking at primary school level is an area of research still in its infancy (Angeli et al., 2016). Neither is there any agreement as to how computational thinking should be introduced at primary level. Current debates centre on whether computational thinking becomes a battle cry for coding in K12 education (Kafai, 2016)? Whether there should be a computer science curriculum for primary level with an explicit focus on computational thinking (Angeli et al.; Fluck et al., 2016)? Or whether young people should be able to develop and use computational thinking concepts in their problem-solving activities as part of subject areas other than computer science (e.g. ISTE/ CSTA, 2011)?

Irrespective of what approach is taken, the teacher is central to ensuring that the children they work with develop computational thinking. It is therefore essential that they are adequately prepared to include computational thinking as part of their pedagogical classroom practices (Yadav et al., 2011; Lye & Koh, 2014). Moreover, this preparation should begin at pre-service level (Barr & Stephenson (2011) so that pre-service teachers not only develop understandings of computational thinking but are also introduced to ways they can design learning opportunities for their students to develop computational thinking (Wing, 2016).

Adopting the stance that computational thinking should be developed as part of subject areas other than computer science, this paper presents and discusses the findings from a study which investigated preservice teachers' understandings of computational thinking on completion of a specialism in digital learning as part of their Bachelor of Education (B.Ed) degree at the Institute of Education of Dublin City University (DCU IoE). The design of the specialism was informed by constructionist principles and sought to develop student understandings of computational thinking in a progressively developmental manner and where the final activity entailed using computational tools with children in the classroom within the context of the extant primary school curriculum.

Computational Thinking in a Constructionist Learning Environment

Computational Thinking

The concept of computational thinking originates in the work of Papert (1980) when he introduced the "idea of the computer being the children's machine that would allow them to develop procedural thinking through programming" (Dede, Mishra & Voogt, 2013, p. 2). However, it was not until 2006 and the publication of Wing's seminal article that the concept of computational thinking came to prominence. Describing computational thinking as a "fundamental skill" for everyone, Wing defined it as the thought process of formulating and solving problems by "drawing on the concepts fundamental to computer science" (p.33) when "equipped with computing devices" (p.35). Within this broader context, she outlined the central components of computational thinking, including algorithms, abstraction, decomposition and automation, all of which can be found in many contexts and disciplines and which assist learners in developing problem solving skills.

Researchers, have continued to put forward a number of definitions of computational thinking as they built on the work of Wing but have failed to agree an accepted definition of computational thinking (e.g. Barr & Stephenson, 2011; International Society for Technology in Education (ISTE) & Computer Science Teachers Association (CSTA) (2011) Dede, Mishra & Voogt, 2013; Grover & Pea, 2013; Selby & Wollard, 2014; Voogt, Fisser, Good, Mishra & Yadav, 2015). However, it is broadly accepted that computational thinking is a thought process that utilises the elements of abstraction, generalisation, decomposition, algorithmic thinking and debugging i.e. detection and correction of errors (Angeli et al., 2016).

Element	Definition
Abstraction	Reducing unnecessary details, highlighting the relevant details to make the process simpler and easier to understand
Algorithmic thinking	Devising a step by step solution to a problem
Decomposition	Breaking down complex problems into manageable smaller problems
Generalisation	Looking for a general approach to a class of problems
Debugging	Skill to identify, remove, and fix errors

Table 1. Core Elements of Computational Thinking

A range of dispositions or attitudes have been identified which some have claimed are integral to the development of computational thinking. Brennan & Resnick (2012) for example, refer to these dispositions as computational practices and computational perspectives. Computational practices are the "problem solving practices that occur in the process of programming" (p.53) and include: iterative and incremental, testing and debugging, reusing and remixing, and abstracting and modularising. Computational perspectives relate to the "student's understandings of themselves, their relationships to others, and the technological world around them" (p. 53).

A number of implementation frameworks have also been put forward. While most of these frameworks focus on post-primary and third level, a small number have been presented for primary level (e.g. Angeli et al. (2016), Brennan & Resnick (2012), Curzon, Dorling, Selby & Woollard (2014)). Across these frameworks, one of the most frequent methods of providing the opportunity to engage in computational thinking in primary classrooms is through the use of programming languages such as Scratch (e.g. Brennan & Resnick, 2012).

Although scholars advocate the introduction of computational thinking early at primary school level (Buitrago Flórez et al. 2017), to date, little attention has been accorded to preservice primary school teachers. While there have been some examples of planned structured teacher preparation programmes (Hodhod et al., 2016; Lodi 2017; Yadav et al., 2017); and it is widely accepted that the development of computational thinking for pre-service K-8 teachers should be integrated with pedagogical content knowledge (Yadav et al. 2017); framework or models that focus explicitly on pre-service teachers are not yet available (Zhao et al. 2019).

Computational Thinking in a Constructionist Learning Environment

From a constructionist perspective, computational tools can be a powerful medium for creating contexts for constructing knowledge and computational thinking can be thought about in much the same way as Papert viewed computer programming; that is computational thinking is both a skill to learn and a way to learn –"to create, discover, and make sense of the world, with digital technologies as extensions and reflections of our minds". (Cator et al., 2018, p.21). However, in keeping with Papert's idea of engaging with "powerful representations", what is essential to consider when designing a learning environment is not so much what programming language and/or computational materials to use, but what personally meaningful ideas the programming language and materials can enable the learners develop and how those ideas will develop computational thinking and form new ideas about the subject area (e.g. mathematics, science). Furthermore, activities and learning situations should be developmentally appropriate for the learners and grounded in meaningful contexts (Butler, 2007).

Drawing on these ideas, the authors wanted to design the Major Specialism modules to provide immersive learning experiences for the pre-service teachers to develop computational thinking. The challenge was complex, as we needed to develop their understandings of the implications of learning design, constructionist principles and computational thinking, the use of a range of expressive computational tools and also how to make connections to the content of the primary school curriculum

The design of the major specialism was anchored in the central tenet of constructionism which is that learning is facilitated by constructing tangible artefacts or objects, which can then be shared and discussed with others (Papert & Harel, 1991). The preservice teachers (learners) were accordingly viewed as active builders of their own knowledge as they engaged in constructing artefacts using a range of computational materials. These artefacts became their “objects to think with” (Papert, 1980, p.12) and supported the development of concrete ways of thinking and learning about computational thinking concepts and practices (Brennan & Resnick, 2012). The ability to manipulate these objects, to repeatedly make adjustments and refinements or experiment with them to see how they work lent itself to a concrete style of reasoning (Turkle & Papert, 1992). This, as argued by Papert changed the process of learning to one which is iterative and cumulative, embracing both planning and bricolage styles.

Constructionism also draws attention to the social nature of learning, noting that activities such as making, building or programming through which the learner produces artefacts that others can see and critique provide a rich context for learning. The artefacts are a means by which others can become involved in the thinking process while at the same time; the learner’s thinking benefits from multiple views and discussions (Butler, 2007). Consequently, while the preservice teachers were continuously encouraged to articulate their thinking; peer feedback and group discussions were also a key feature of these workshops, thus enabling them to understand and incorporate the perspectives of others. In this way, the artefacts or ‘objects to think with’ provided a link between sensory and abstract knowledge (hands on workshops coupled with learning theory / computational thinking literature), and between the individual and the social worlds (preservice teachers and the classroom). Shared knowledge was constructed when artefacts and shared understanding were coupled through cycles of representing and interpreting (Ostwald, 1996). Using an incremental spiral approach, engaging in conversation around their own or another’s artefact in each cycle, the development of a shared understanding was enabled and the foundation for new understandings were cultivated in progressively deeper ways. (Ackermann, 2001). Turkle and Papert (1990) refer to this ‘validity of multiple ways of knowing and thinking’ as “an epistemological pluralism” (p.129).

In summary, the design of the specialism was underpinned by constructionism principles which embraced a *process* of building, both in the sense of building artefacts and building new understandings. As module facilitators, the authors continuously worked alongside the students using a range of pedagogical strategies and practices to help them construct and reflect on their emerging understandings of computational thinking .

Overview of the Major Specialism

A Major Specialism on the BEd (Primary) degree comprises of five modules (5 * 5 ECTS) taken over years two, three and four of the programme. The overarching aim of a specialism is that students develop deep subject knowledge and leadership skills in a specific area which will allow them to make a strong contribution to curriculum development and innovation in schools on graduation (BEd Accreditation Document, 2012). The Major Specialism in Digital Learning accordingly targets the deepening of student teachers’ pedagogical content knowledge and curricular knowledge in digital learning. More specifically, the primary focus of the specialism is the development of preservice student teachers’ understanding of the theoretical and practical concepts of computational thinking and coding. Our approach is to expose students to the concepts of CT using a range of computational materials; enable them to build their own interpretation based on experience, literature, class discussions, and hands-on learning experiences; and finally, to make learning concrete and relevant through the completion of assignments that relate to their future teaching and the primary school curriculum.

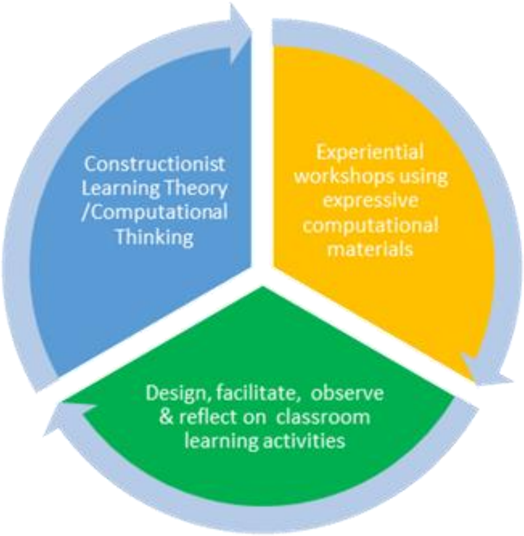


Figure 1. Conceptualisation of the Major Specialism

Modules are designed to include a range of computational materials (e.g. BeeBots, Kibo, Scratch Junior, Scratch, Lego WeDo) and contexts to explore strategies that support interest driven, project-based, collaborative approaches to learning. In addition, an incremental spiral approach is used across the modules in which the key ideas and concepts in relation to computational thinking and constructionism are returned to, in increasingly deeper and more complex ways (Bruner, 1960). For example, initial modules focus on understanding the key ideas in computational thinking whereas by the final module, students are expected to to put their understandings of computational thinking "into action" in classroom settings. Personal reflection, peer feedback and group discussions are key features of modules as we believe it is through engaging in conversation around their own or another's artefact (be it a programme in Scratch or a Lego robot) that, the development of a shared understanding is enabled and the foundation for new understandings laid.

Module	**Key Concepts introduced**	**Computational Materials**
Design & Build to Learn	Introduction to computational thinking skills, Introduction to constructionism	BeeBots, Kibo, Lego WeDo
Gaming to Learn	Computational thinking in coding	Scratch; Minecraft
Creative Construction	Development of understanding of constructionism	Lego WeDo
Designing & Learning with Digital Technologies	Embedding Constructionist principles and Computational Thinking within the Primary classroom Participant Observation	Lego WeDo

Table 2. Conceptual Development in the Major Specialism in Digital Learning

The final module, Designing & Learning with Digital Technologies, is designed to enable students to translate their learning into practice (Teaching Council, 2017) and also afford students an opportunity to engage in research, thus providing "the foundation of their practitioner-based enquiry stance in the future" (Teaching Council, 2017, p. 23).

As part of the module; students in groups of three, are required to design and implement a series of three two-hour workshops in which they introduce coding and computational thinking concepts using Lego WeDo robotic materials to 4th or 5th class children. (aged 10-11 years) in local schools. The series of workshops must be designed within the context of the Primary School Science Curriculum (DES, 1999).

In order to document the experience, students use participant observation as a data gathering mechanism using a pre-designed template. This template was designed by the authors and included themes identified from the literature presented throughout the specialism. Analysis is completed as a class activity and students write up the research as the module assignment. In addition to documenting their findings, students are required to reflect on the computational thinking they saw in action in the classroom as well as on their own understandings of computational thinking.

Preparation of the classroom-based activity is carried out as part of workshops with each group discussing, sharing and giving feedback on the proposed range of possible learning activities to be implemented over the 3 weeks. These learning activities become the preservice teachers' "objects to think with" as they share and discuss how they can draw on their understandings of constructionist learning principles, to help them design learning activities that embed opportunities for the development of computational thinking and which leverage the primary science curriculum.

Students are also introduced to participant observation as a data collection method, the deductive approach to data analysis and the ethics of classroom-based research.

Finally, consistent with the iterative and incremental nature of constructionism, the preservice students' classroom experiences, in turn become their "object to think with" as they engage in the final set of reflective group discussions at workshops. These discussions are structured to enable students to connect theory with practice and to further deepen their understandings of both computational thinking and constructionist learning theory.

Research Design

Participants and context

18 pre-service teachers who had completed a Major Specialism in Digital Learning as part of their BEd degree at the Institute of Education, Dublin City University participated in the study

As part of the final module in the specialism, Designing & Learning with Digital Technologies, the pre-service teachers in groups of three worked with pupils in 4 local schools

- School A: 2 x Fourth Classes (64 boys)
- School B: 1 x Fourth Class & 1 Fifth Class (60 girls)
- School C: 1 x Fourth Class (30 boys & girls)
- School D: 1 x Fifth Class (25 girls)

The pre-service teachers worked in groups of three for two reasons; first, it enabled them to plan together, discussing and refining their understandings of how they were going to develop science content in tandem with computational thinking. Second, when implementing the project, one could take the lead facilitation role while the other two could engage in observation of the learning process.

Methodology

The aim of the study was to investigate preservice teachers' understandings of computational thinking on completion of a specialism in digital learning as part of their BEd degree. A qualitative approach was adopted in the research. Data collection methods included document analysis and group interviews:

- Each of the 18 students completed reports of their classroom-based research. The students gave the authors permission to use these reports as part of the data corpus.
- Group interviews with student teachers took place at the end of the semester. Each group interview, approximately 20 minutes with four to five students, was carried out by authors. The aim of the interviews was to probe the students' experiences, understandings and reflections in relation to computational thinking.

A typological analysis framework (Hatch, 2002) drawing on a literature review previously completed by the authors was used for initial analysis. Typologies were generated from the main themes in the review and related to computational thinking and constructionist learning theory. Initial data processing took place within these categories. A typological analysis can be problematic in that starting analysis with predetermined categories can constrain the analysis process and crucial data might be overlooked. The categories were therefore re-examined after coding to ensure that they were justified by the data or if the data excluded contained insights contrary to that proposed. Overall, decisions were driven by the data and, where necessary, new categories of adjustment added. This iterative process of sifting, analysing, and winnowing a collection of data helped reduce it to a small set of themes that then lent themselves to the final narrative (Cresswell, 2007).

Findings

Two broad themes are presented as findings. The first relates the development of the pre-service teachers computational thinking and the second to the importance of an immersive classroom experience towards this development.

Development of Computational Thinking

Analysis demonstrated not only the development of a "computational thinking language" among the pre-service teachers (Lu & Fletcher, 2009) but also an ability to identify the development of computational thinking concepts and practices in the children that they worked with across the three weeks. More specifically, the most prominent concepts and practices observed and discussed by the students across the reports included decomposition, logical thinking and debugging. Algorithmic thinking, abstraction and generalisation were elaborated to a lesser extent although this was not entirely surprising given the duration of the project. Across all the reports, students demonstrated an ability to define and explain the concepts of logical reasoning, pointing to specific examples they observed while working with children in the classroom and relating their observations to extant research literature. The skills of logical reasoning and abstraction are presented as examples of this.

<u>Logical Reasoning</u>

In her essay, Student 4 noted and tracked the development of logical reasoning across the three weeks stating at the outset that "there was variance in the children's ability to think logically. The majority of children employed basic strategies to solve problems that arose when building and programming their models". However, by the end of the project, she documents the children using strategies of logical reasoning such as predicting, analysing, creating and correcting their algorithm and the build of their model.

In the final week the children were challenged to make their frogs move faster adapting both their builds and code. One child responded with … *"Well, if you want to get a car to go faster you need more speed… the speed comes from the acceleration so…. we could try giving him (robot) more acceleration and he might go faster."* This child analysed the problem linking it with their own knowledge and experiences "to predict the behaviour of…. programs" (Csizmadia, 2015, p.9). The child broke down the problem and thought logically about the elements that affect the speed of a car and transferred this knowledge to their own code. By saying, "might", this insinuated that this solution may not work and the child would have to think of an alternative solution, this implies that the child would be tinkering with the problem. (Essay Student4)

Figure 2 Extract from the essay of Student 4

Abstraction

For the most part, where students identified abstraction in operation, it related to the models they built. Through questioning and observation of the children as they worked, the pre-service teachers were able to identify ways the children filtered superfluous features of the models in order to get them to debug them and/or make them more efficient.

> One child remarked how some parts of their model were just there for decoration, so he was going to remove them in order for his model to be lighter and move faster (Student 15, Essay).

" We are after adapting our floodgate and making it too heavy for its purpose as we have built walls all around it, this will put too much pressure on the opening of the gate and make it efficient for purpose."

Figure 3. Student 2, observation notes week 3

One student, (Student 2) indicated that abstraction was beginning to emerge in the children's coding, noting that "by the third week it was evident that some groups were thinking in terms of how to make their code more efficient and shorter". However, the example she included does not fully elaborate abstraction. She states "the children combined their understanding of the blocks with the codes they had used the previous week, and in turn used blocks like 'the repeat block' to *remove unnecessary detail*" – thus indicating an understanding of abstraction. . Although the data extract in in Figure 3, suggests that the children had begun to generalise from specific instances (Wing, 2011), further evidence is needed. Despite this, the data she uses to support her observation demonstrates a developing sophistication in coding (algorithmic thinking/pattern recognition) rather than abstraction i.e. there is no evidence provided of removing unnecessary detail.

- *"Why not use repeat block like last week with our car so that our code is not too long and complicated, it*

Figure 4. Student 2, observation notes week 3

The Importance of the Immersive Classroom Experience

The importance of the classroom experience emerged as key towards deepening student understanding of computational thinking and constructionism. It was strongly emphasised by all students that the experience of working with children in the classroom not only brought the theory 'to life' but helped them deepen their personal understandings. It led them to reflect more deeply on the fundamentals of computational thinking and constructionism as well as develop their understandings of what computational thinking looks like "in action". It also served to highlight the critical role of the teacher in designing and facilitate learning environments so that their pupils in turn develop computational thinking.

> …it brought it to life I think, for me anyway… even when we were revising at the beginning just the different components of the computational thinking and being like oh yeah, that's what this is. But then going into the classroom and seeing, like linking the theory of it with children; actually putting it into practice really got me to understand the definition of it more. (Focus Group 1)
>
> …. it was my first chance to see computational thinking alive in a classroom. Instead of learning about it in college or discussing it with peers, I had the opportunity to identify different components in a classroom and use my expertise to progress these skills further. (Essay Student 3)

Prior to the classroom experience, the preservice teachers had been concerned as to how they could implement the "Digital Learning Framework" (DES, 2017), and accommodate the development of computational thinking into the curriculum. However, based on their classroom experiences, the preservice teachers began to understand that they have a pivotal role in determining what or if the learner developed CT and that it centres on anchoring the design of learning experience within the existing primary curriculum.

> …to think of computational thinking as just problem solving, is a simplistic view … the teacher has a pivotal role in enabling children to develop this higher order skill-set, thus planning through subject integration is pivotal as suggested by the Irish DLF (Student 11)
>
> The importance of the teacher as a model to use correct language and progress children's thinking … deepening the children's thoughts guiding them on what to look out for and discuss, by using prompt questions (student 3)
>
> A key point to note is computational thinking skills will not develop in isolation, and they need the initial teacher scaffolds to allow children to communicate what they are doing (student 1).

Conclusion

To conclude, working with the children in the classroom helped the preservice teachers deepen their own understandings of computational thinking and in turn, reflect more deeply on how to develop computational thinking in their own classroom practice as qualified teachers. Their competence and confidence in using the computational materials developed and they were able to connect principles of constructionism and inquiry based science with an informed understanding of the necessity to plan for the development of computational thinking in primary classrooms. Having the opportunity to engage in this module rooted in classroom practice enabled these preservice teachers understand how computational thinking can be developed by embedding it within existing curricula, leading to the realisation that it is not a case of one or the other (i.e. CT or Subject Content) but a means of combining both.

The insights gained from this study are particularly relevant for the design of teacher preparation programmes indicating how CT can be effectively embedded to combine theory and practice. This will ensure that CT concepts are not developed in a decontexualised manner but are embedded within the prescribed curriculum across a range of subject content in a relevant and meaningful manner.

References

Ackermann, E. (2001). Piaget's constructivism and Papert's constructionism: What's the difference? In *Constructivisms: Usages et Perspectives en Education* (Volume 1 et 2.) Geneva: SRED/Cahier 8. pp. 85-94.

Angeli, C., Voogt J., Fluck, A., Webb, M., Cox, M., Malyn- Smith, J., & Zagami, J. (2016). A K-6 computational thinking framework-implications for teacher knowledge. *Educational Technology & Society*, 19 (3), 47-57.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. *ACM Inroads*, *2*(1), 48-54.

BEd Accreditation Document (2012). Dublin: St Patrick's College of Education. Unpublished document

Brennan, K. & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking.* In AERA2012-Annual Meeting of the American Educational Research Association. Vancouver, Canada.

Bruner, J. (1960). *The Process of Education.* London: Harvard University Press.

Buitrago Flórez, F., Casallas, R., Hernández,M., Reyes, A., Restrepo, S.,& Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research*, 87(4), 834–860.

Butler, D. (2007). A constructionist view of what it means to be digitally literate–Learning in 21st century classrooms. *Nordic Journal of Digital Literacy*, *2*(02), 61-77.

Cator, K., Angevine, C., Weisgrau, J., Waite, C. & Roschelle J. (2018). *Computational thinking for a computational World.* Digital Promise.

Creswell, J. W. (2007). Qualitative inquiry and research method: Choosing among five approaches.

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational thinking-A guide for teachers.

Curzon, P., Dorling, M., Ng, T., Selby, C., & Woollard, J. (2014). *Developing computational thinking in the classroom: a framework.*

Dede, C., Mishra, P. & Voogt, J. (2013, October). Working group 6: Advancing computational thinking in 21st century learning. EDU Summit 2013: International summit on ICT in education. San Bruno.

DES (1999). Irish Primary School Science Curriculum. Government Publications Office. Dublin, Ireland.

DES (2015). Digital strategy for schools: Enhancing teaching, learning & assessment.

Department of Education and Skills (2017). Digital Learning Framework for primary schools. Dublin: Author. Online at https://www.education.ie/en/SchoolsColleges/Information/Information-Communications-Technology-ICT-inSchools/digital-learning-framework-primary.pdf

Fletcher, G. H., & Lu, J. J. (2009). Education Human computing skills: rethinking the K-12 experience. *Communications of the ACM*, *52*(2), 23-25.

Fluck, A. E., Webb, M., Cox, M. J., Angeli, C., Malyn-Smith, J., Voogt, J., & Zagami, J. (2016). Arguing for computer science in the school curriculum. *Educational Technology & Society*, *19*(3), 38-46.

Grover, S., & Pea, R. (2018). Computational Thinking: A competency whose time has come. Computer Science Education: Perspectives on teaching and learning in school. London: Bloomsbury Academic, 19-37.

Hatch, J. A. (2002). Doing qualitative research in education settings. Suny Press.

Hodhod, R., Khan, S., Kurt-Peker, Y., & Ray, L. (2016, February). Training teachers to integrate computational thinking into K-12 teaching. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 156-157). ACM.

ISTE, CSTA. (2011). Computational Thinking in K–12 Education leadership toolkit. Computer Science Teacher Association:http://csta.acm.org/Curriculum/sub/CurrFiles/471.11 CTLeadershipt Toolkit-SP-vF. pdf adresinden alındı.

Kafai, Y. (2016). From computational thinking to computational participation in K-12 education. *Communications of the ACM,* Vol. 59 No. 8, Pages 26-27.

Lodi, M. (2017, August). Growth Mindset in Computational Thinking teaching and teacher training. In *Proceedings of the 2017 ACM Conference on International Computing Education Research* (pp. 281-282). ACM.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior*, *41*, 51-61.

Lu, J. J., & Fletcher, G. H. (2009, March). Thinking about computational thinking. In *ACM SIGCSE Bulletin* (Vol. 41, No. 1, pp. 260-264). ACM.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. New York: Basic Books, Inc,.

Papert, S. & Harel, I.(Eds). (1991). Situating constructionism. In *Constructionism.* New York: Ablex Publishing.

Selby, C., & Woollard, J. (2014). Refining an understanding of computational thinking.

Teaching Council (2017). Initial Teacher Education Criteria and Guidelines. Ireland

Turkle, S., & Papert, S. (1990). Epistemological pluralism: Styles and voices within the computer culture. *Signs: Journal of women in culture and society*, *16*(1), 128-157.

Turkle, S., & Papert, S. (1992). Epistemological pluralism and the revaluation of the concrete. *Journal of Mathematical Behavior*,*11*(1), 3-33.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715-728.

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Wing, J. (2011). Research notebook:Computational thinking-What and why. *The link magazine*, *6*

Wing, J. M. (2016). Computational thinking, 10 years later. *Microsoft Research Blog*, *23*. /

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011, March). Introducing computational thinking in education courses. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 465-470). ACM.

Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational thinking in teacher education. In *Emerging research, practice, and policy on computational thinking* (pp. 205-220). Springer, Cham.

Zhao, S., Jin, Y., Moore, P., & Gaston, J. (2019). Hopscotch into Coding: Introducing Pre-Service Teachers Computational Thinking. TechTrends, 1-12.

Developing STEM Literacy with young learners using a blend of Bridge 21 and playful pedagogy

Mairéad Holden, *mairead.holden@gmail.com*
School of Education, Trinity College Dublin, the University of Dublin, Dublin 2, Ireland

Abstract

This paper presents a vignette and accompanying reflections which describe the author's recent experience of using a constructionist pedagogy, Bridge 21, as part of a 21st Century STEM learning experience with a small group of young learners aged 5-7 in a primary school setting. The broad aim of the learning experience was to develop pupil participants' STEM Literacy, as described by Liston (2018) and Kelley and Knowles (2016). The learning experience sought to meet specific objectives derived from the Irish Primary Science Curriculum (Department of Education and Skills (DES), 1999) and Digital Learning Framework-Primary (DES. 2017). An additional aim of the learning experience was to enable pupils to develop the 21st century skills of critical thinking, creativity, communication and collaboration (Dede, 2010).

The paper offers some background on the concept of STEM Literacy and relevant pedagogies before presenting a vignette which details the planning, teaching and assessment of a learning experience. Using an adapted version of the Bridge 21 Model (Conneely, Girvan, Lawlor & Tangney, 2015; Roberts, 2016), pupils were facilitated to explore the concept of gravity through engaging in play-based activities with slopes made from recycled cardboard kitchen roll tubes, marbles and Lego© blocks. Pupils then built on their play experiences to design and make a mini marble run using the same materials. One pupil also used Minecraft to create a digital marble run. Pupils recorded, described and shared their learning with verbally annotated photographs and videos, which they uploaded to Seesaw, an e-portfolio tool.

In order to support the reflective process, the author drew from Brookfield's concept of critical lenses (1995; 2017), Schön's (1987) concepts of reflection-in-action and reflection-on-action and Rolfe, Freshwater and Jasper's (2001) reflective framework. The author's reflections are used to derive implications for their own classroom practice and implications for other educators who may be interested in using a 21st Century constructionist approach as part of their work.

Figure 1. Exploring STEM concepts with young learners using Bridge 21 Model

Keywords

STEM Pedagogy; Bridge21; Playful learning; Reflective Practice

STEM Education & STEM Literacy

Be it “STEM”, “STEAM” or “STREAM”, the STEM acronym (and its associated variations) remains contentious.

On its surface, “STEM” is the acronym of Science, Technology, Engineering, and Mathematics. However, when you pull that first layer away, you reveal the most elaborate puzzle in the education world. Most educators know what STEM stands for, but how many really know what it means?

Gerlach (2012, p.1).

STEM Education is not simply Science, Technology, Engineering and Maths, but a cross-curricular approach focusing on activities relevant to all four areas (Kelley & Knowles, 2016; Rosika, 2016) defines. In the context of Irish education, one of the purported goals of STEM Education is STEM literacy (DES, 2016). STEM literacy can be described as the knowledge, skills, and dispositions that pupils acquire and develop as a result of participating in STEM education (Kelley & Knowles, 2016; Liston, 2018). STEM-literate pupils are problem solvers, inventors, innovators and logical thinkers. They can work with others but are also self-reliant and technologically competent (Huling & Speake Dwyer, 2018; Kettler, 2019). An integrated approach to STEM Education enables learners to develop STEM literacy, that is, to build and apply knowledge, deepen their understanding and develop creative and critical thinking skills within authentic contexts (DES, 2017; Kelley & Knowles, 2016; Liston, 2018).

Some theorists argue that Engineering is “the glue” that serves to integrate Science, Maths and Technology (Bagiati & Evangelou, 2015; Liston, 2018; Mooney & Laubach, 2002; & Moore et al., 2014). In considering the Irish primary STEM context, Liston (2018) suggests that the Engineering Design Process offers a “systematic, orderly, open-ended way of approaching problems and designing solutions for those problems” (p.1). Engineering is not explicitly mentioned in the Irish Primary Science Curriculum (PSC) (DES, 1999). However, the “Design and Make” section of the PSC espouses the drawing together of each the STEM elements, stating that “designing and making should be developed in association with and through visual arts, science and mathematics” (p.8). The process of designing and making in the PSC (1999) involves four steps: Explore, Plan, Make and Evaluate.

Not simply confined to digital technology, the “T” strand of STEM can be interpreted in a much broader sense (Kelley, 2010). Presenting a similar view to that taken in the PSC (1999), Jolly (2017) suggests that technology is “any innovation or device created by people for the purpose of meeting a human need or want” (p.6). Vasquez (2015) uses the example of a pen as “technology” to illustrate this view of technology. Jolly (2017) posits that during authentic integrated STEM activities, technologies are created when pupils make prototypes and products in response to problems. Digital technologies offer an important resource to support pupils learning. Digital technology tools such as video can enhance analysis and add rigour to investigations. There are many useful digital tools to support recording and communication of findings, as well as e-portfolio tools which can support pupils in formatively assessing their work (DES, 2017).

Pedagogical approaches

In line with the ideas put forward by Beatham and Sharpe (2013), in order to develop pupils’ 21st century skills, a 21st century pedagogical approach is advocated. Bridge 21 is a teaching and learning model initially developed for use with post-primary students to develop the 21st century skills of critical thinking, creativity, collaboration, cooperation and communication (Bridge 21 Handbook, 2013, p.1). Bridge 21 involves an active student-centred approach in order to enable acquisition of curriculum content while developing and operationalising 21st century skills (Tangney et al., 2015). The model is underpinned by constructionist principles of active and social learning and the creation of artefacts as part of the learning process (Tangney et al., 2015; Papert, 1993). Some key features of Bridge 21 include a teaching approach which is facilitative rather than didactic, the use of student-led project based learning with pupils in groups in order to engage with thematic cross-curricular content, the use of technology as an integral tool in the learning process as well as the incorporation of meaningful learner reflection as part of learning (Tangney et al., 2015).

There appear to be a number of commonalities between the Bridge 21 model, the Engineering design process (EDP) and the PSC Design and Make process. These are summarised in Figure 2.

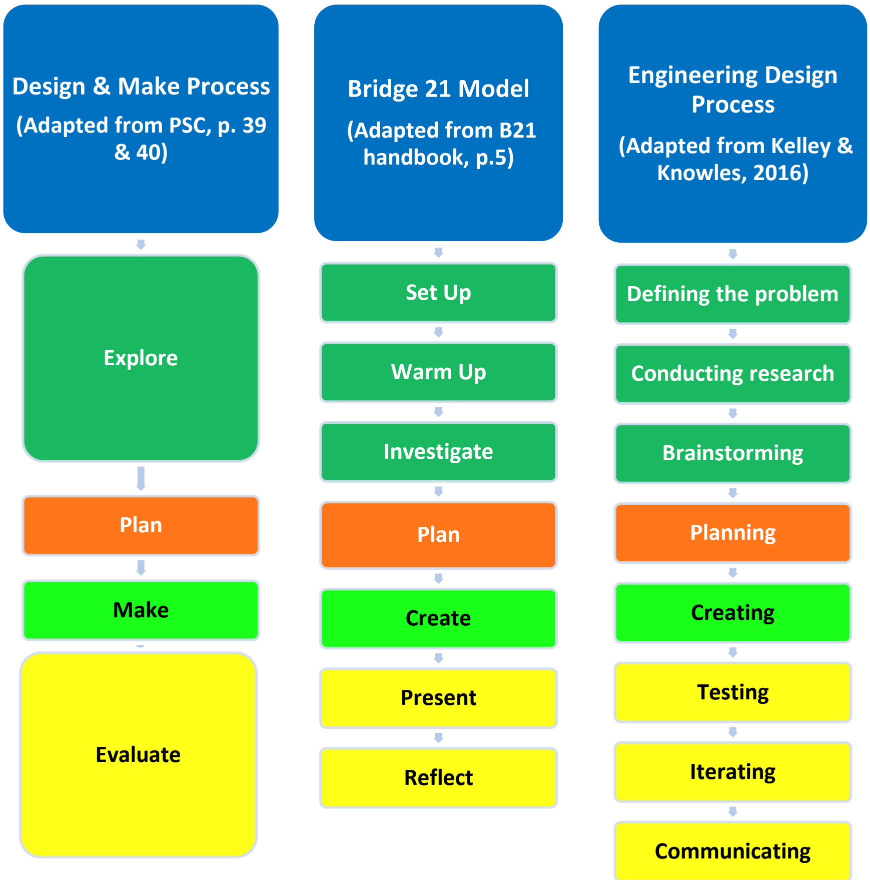


Figure 2. Comparative overview of Design & Make Process, Bridge 21 Model & Engineering Design Process (Adapted from Bridge21, 2013; Kelley & Knowles, 2016; & PSC, 1999)

The positive impact of the use by educators of the Bridge 21 approach on post-primary student outcomes relating to teamwork, motivation and personal responsibility for learning has been well documented in literature (Byrne et al., 2015; Fisher et al., 2015; Johnston et al., 2016; Lawlor et al., 2015; Lawlor et al., 2018; & Tangney et al., 2015). However, understanding as to the potential effectiveness of the use by educators of the Bridge 21 model on similar outcomes relating to early years learners and primary pupils has yet to be established. For the purpose of this learning experience, the Bridge 21 model was adapted to fit within a developmentally appropriate approach, Roberts (2016) 3-Stage Project Based Model, for the young learners taking part. This will be explained in further detail in the next section.

Playful Pedagogy

Roberts (2016) makes the important point that young children's natural curiosity may be dampened by overly structured approaches to STEM activities. Due to the age profile of the learners (5-7 years) involved in this learning experience, playful pedagogy (as described by the NCCA Aistear Framework, 2006) was used as a complement to the Bridge 21 model. The importance of play as a context for young children's learning and development has been well established (Broadhead et al, 2010; Van Oers, 2010; & Vukelich,1994). Theory suggests that playing is central to a child's physical, social, emotional, cognitive and language development (Froebel, 1826; Piaget, 1976; Vygotsky, 1962).

In the Irish early years and primary context, the Aistear framework (NCCA, 2006) advocates a balanced use of both teacher-led and child-led play. The categories of play this learning experience focused on were creative and physical play (NCCA, 2006, p. 54). Initial exploration of materials was predominantly child-led, with teacher scaffolding pupils' use of STEM specific language. The design and make element (teacher-led) built on the pupils' initial play experience, in line with Vygotsky's Zone of Proximal Development (ZPD) Model (1987), and extended this knowledge by applying it to a context (building a marble run).

The design of the learning experience drew from Roberts' (2016) suggested teaching approach for STEM activities with young learners (birth to 8 years). Her approach is based on a Reggio Emilia concept "progettazione" meaning "project based approach" (Edwards, 1994; Rinaldi, 2004). Projects should be derived from children's interests. Roberts' approach suggests three stages (see Figure 3) to projects with pupils with pupils engaging in different types of activities at each stage.

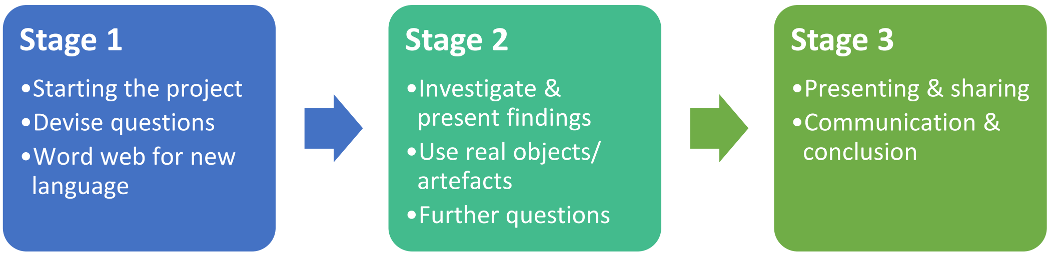


Figure 3. Overview of Roberts' (2016) 3 Stage Project-Based Model

Roberts' (2016) model is broadly similar to the models previously outlined in Figure 2. However, she appears to suggest omitting a stand-alone "plan" stage, as this is incorporated into the activities of Stages 1 and 2. The choice of topic and use of artefacts of interest to learners and the active hands-on nature of Roberts' approach connects strongly with Papert's (1987) principles of constructionism. While constructionist approaches are often associated with the creation of digital artefacts, Papert (1991) also suggested a broader interpretation to include the designing and making of physical artefacts, for example, sandcastles, sculptures and art projects. He suggests that the process of designing and making, both physical and digital, can be useful in supporting learners' construction of knowledge.

Components of a STEM learning experience?

So far, I have highlighted a number of key considerations as to what components an appropriate STEM learning experience for young learners might involve. Jolly (2017) offers a useful "STEM Design tool" with a number of suggested components (see Figure 4) for STEM learning experiences. The purpose of this tool is to guide and support teachers in designing STEM learning experiences. This checklist also acknowledges constructionist principles, particularly in relation to the authentic and learner-centred nature of suggested components and the foregrounding of active, hands-on and socially mediated learning processes.

Component	**Prompt questions for reflection**
Authentic problem	Does the learning experience present a real problem (an engineering challenge)?
Pupil-centred	Will pupils relate to the problem?
Open-ended	Does the learning experience allow pupils multiple and creative approaches and solutions for successfully solving the problem?
Cross-Curricular	Does the learning experience integrate and apply science and maths curriculum content and skills?
Design & Make/ Engineering design process	Does the learning experience: • clearly use the engineering design process as the approach to solving problems? • lead to the design and development of a model or prototype?
Hands-on learning	Does the learning use a child-centred, hands-on teaching and learning approach?
Technology	Is the role of technology in the lesson clear to the pupils?
Teamwork	Does the learning experience successfully engage pupils in purposeful teamwork?
Evaluation & iteration	Does the learning experience include testing the solution, evaluating the results, and redesigning to improve the outcome?
Communication	Does the learning experience involve students in communicating about their design and results?

Figure 4. STEM Learning Experience Reflective Checklist (adapted from Jolly, 2017)

The checklist in Figure 4 informed the design of this learning experience and also supported the follow-up reflections on the learners' engagement with the planned activities.

Description and organisation of the learning activity

Project Stage	Bridge 21 Phase	Activity	Purpose for pupils
Stage 1	**Set Up**	*Seesaw plenary*	• Introduce new tool to pupils
		Tumble tracks & Wooden marble run	• Language development • Trigger for curiosity
	Warm Up	*Free play with project materials*	• Enjoyment • Language development • Trigger for questions • Motor skills
Stage 2	**Investigate**	*Structured (teacher-led) play with slopes*	• Language development • Conceptual understanding • Trigger for questions
	Create	*Structured brief presented to pupils, pupils use materials from earlier to create marble runs*	• Creativity • Critical thinking • Motor skills development • Language in context
Stage 3	**Present**	*Pupils present their artefacts (marble run) & photograph/video for Seesaw*	• Communication • Language in context • Celebration of learning
	Reflect	*Pupils engage in verbal reflection using Seesaw*	• Critical thinking • Self-assessment

Desired learning intentions

As a result of engaging in this learning experience, the pupils were enabled to:

1. Develop their understanding of how forces act on objects through observing and investigating the movement of marbles on level and inclined surfaces
2. Record the results of their own investigations using verbally annotated photographs
3. Use the e-portfolio tool Seesaw to collect evidence of, evaluate and reflect on the artefacts (Marble runs) they have created
4. Develop the 21st century skills of critical thinking, creativity and communication through the process of a hands-on STEM-based task

Success criteria: Content & Skills development

- Pupils to show their understanding through a combination of explanation, photographs and demonstration-by changing the slope of a ramp, they can affect the distance and direction travelled by a marble
- Pupils to use a combination of media (photographs, videos and audio recordings) to communicate the findings of their investigations on slopes
- Pupils to create their own unique STEM artefact/prototype of a model marble run

- Pupils to upload the recorded media of their marble run and reflections on their work to their Seesaw portfolio

Figure 5. Resources for Free play: Tumble Trax© Marble activity.

Figure 6. Pupils engaging in free play with wooden marble run.

Assessment of learning

Pupils' learning was assessed using teacher observation during play as well as their Seesaw uploads. The age of the learners meant that their capacity written communication was limited. The Seesaw tool allowed the pupils to record and reflect on their learning verbally. The assessment rubric in Figure 7 was devised and used to categorise the pupils work samples. Some selected samples are shown in Figure 8a, 8b and 8c.

	Below average	**Average**	**Above average**
Understanding of forces (the effect of slope on the trajectory of the marble)	Unable to demonstrate understanding	Able to demonstrate understanding through physical demonstration but verbal description was limited	Able to demonstrate their understanding in a broad variety of ways including physical demonstration and verbal description
Digital technology	Did not upload any evidence of their artefact	Uploaded photograph only with some limited annotation	Uploaded and effectively annotated selection of suitable photographs of artefact
21st century skills- Communication	• Difficulty communicating ideas about their work • Used little or no STEM language in their communication	• Communicated some ideas about their work • Used some new STEM language	• Clearly communicated ideas about their work in variety of ways (verbal, photo & physical demonstration)
Creativity	• Tendency to copy other pupils' work without adding or augmenting it	• Built on the ideas of others' when engaging in their own work	• Built on others ideas and weaved these into their own ideas when engaging in their work
Critical thinking	• Difficulty stating what they might do differently next time	• Gave one suggestion on what they might do differently next time	• Gave multiple sensible suggestions on what they would do differently next time and backed these up with evidence from their work

Figure 7. Rubric to assess pupils' development of STEM literacy

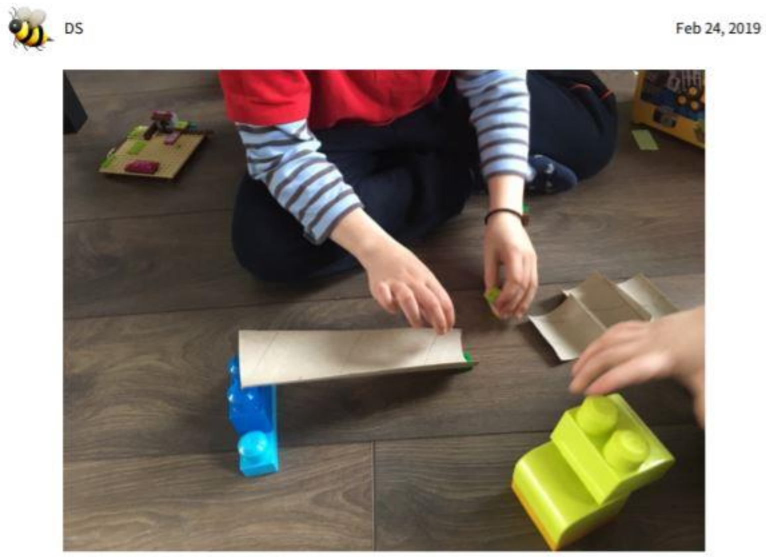


Figure 8a.Pupil work sample which meets “Below average” criteria.

In Figure 8a, the pupil has used the teacher-suggested design unchanged. They have taken a picture of this, but have not added any annotation or communicated their thinking about how their model works, or how they might improve it or make it more sophisticated.

Figure 8b. Pupil work sample which meets “Average” criteria.

In Figure 8b, the pupil has created a simple physical model, which they have shared using video. The video also acts as evidence that their model works.

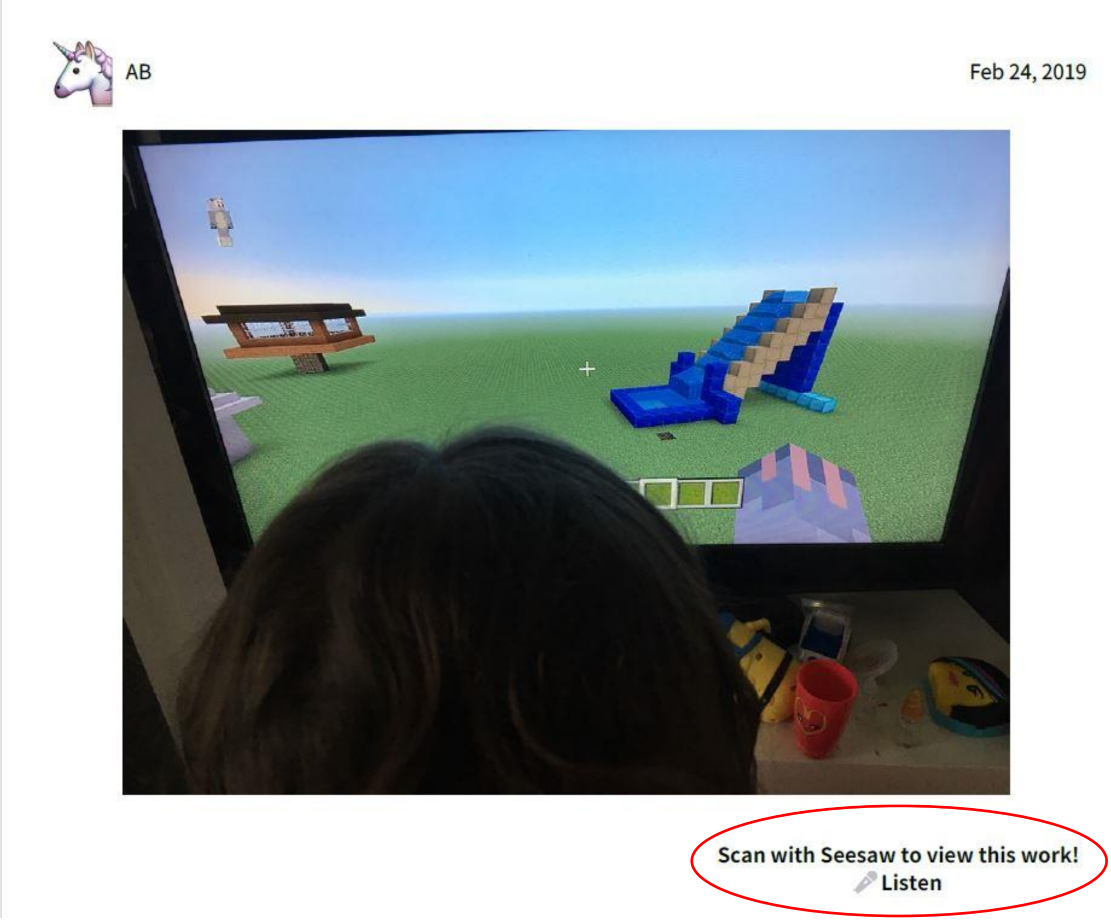
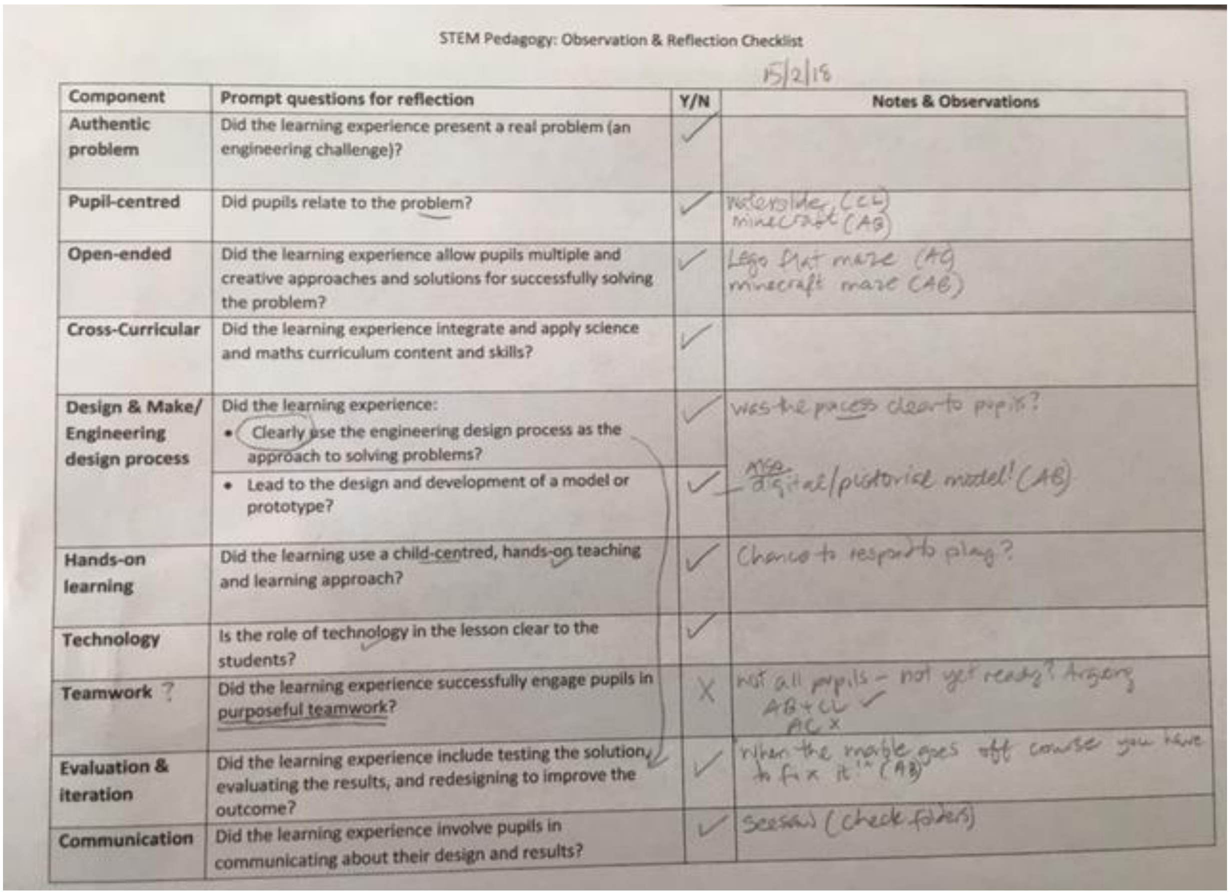


Figure 8c. Pupil work sample which meets "Above average" criteria.

In Figure 8c, the pupil has created both digital and physical models. They have communicated their work using a photograph, which they have annotated clearly and articulately. They have also added their own design features to the digital model, adding water thus turning their marble run into a water slide.

Discussion: Reflecting on the learning experience

During the process of planning, teaching and assessing the learning experience, I sought to structure and deepen my reflections. I drew from Brookfield's concept of "critical lenses" (1995; 2017), with a focus on reflecting through the lenses of self, literature and my pupil participants. I also drew from the work of Schön (1987), in particular his concepts of reflection-in-action and reflection-on-action. I kept field notes during the activities and as earlier mentioned, I used Jolly's STEM Design tool (2017) to design a reflective checklist to support my reflections on the learning experience. Figure 9 shows I was able to answer "yes" to almost all of the checklist questions. However, while offering an insight into how the pupils interacted with learning experience, the simple yes or no style of the checklist did not depict the whole story of what rich learning took place for me during the learning experience. However, it did offer a useful starting point.



STEM Pedagogy: Observation & Reflection Checklist

15/2/18

Component	Prompt questions for reflection	Y/N	Notes & Observations
Authentic problem	Did the learning experience present a real problem (an engineering challenge)?	✓	
Pupil-centred	Did pupils relate to the problem?	✓	waterslide (CL) minecraft (AB)
Open-ended	Did the learning experience allow pupils multiple and creative approaches and solutions for successfully solving the problem?	✓	Lego flat maze (AG) minecraft maze (AB)
Cross-Curricular	Did the learning experience integrate and apply science and maths curriculum content and skills?	✓	
Design & Make/ Engineering design process	Did the learning experience: • Clearly use the engineering design process as the approach to solving problems?	✓	was the process clear to pupils?
	• Lead to the design and development of a model or prototype?	✓	Also digital/pictorial model! (AB)
Hands-on learning	Did the learning use a child-centred, hands-on teaching and learning approach?	✓	Chance to respond to play?
Technology	Is the role of technology in the lesson clear to the students?	✓	
Teamwork ?	Did the learning experience successfully engage pupils in purposeful teamwork?	X	not all pupils – not yet ready? Arguing AB+CL ✓ AC X
Evaluation & iteration	Did the learning experience include testing the solution, evaluating the results, and redesigning to improve the outcome?	✓	"When the marble goes off course you have to fix it!" (AB)
Communication	Did the learning experience involve pupils in communicating about their design and results?	✓	Seesaw (check folders)

Figure 9. Completed reflection checklist (adapted from Jolly, 2017).

I wanted to identify what learning had taken place for me in a sharper way, to allow me to refine my practice going forward. Following consideration of the STEM checklist in Figure 9, I conducted an inductive thematic analysis on my field notes (Braun & Clarke, 2019). The themes I identified allowed me engage in further critical reflection on my practice. I have summarised themes according to three broad areas: What went well, what I found challenging and what I found surprising. This allowed me to consider suggestions as to the "why" underlying my perceptions of certain aspects of the experience, using the reflective lenses of self, literature and the participants to dig deeper (Brookfield, 1995; 2017).

What I felt went well	**Why?**
Pupils' demonstration of: ✓ Curiosity ✓ Creativity ✓ Engagement	The activities were linked to children's interests. This was in line with constructionist principles, which highlight the importance of learning which involves meaningful artefacts (Papert, 1991).The children also got the opportunity to explore the materials for themselves, rather than observing a teacher demonstration. Most of the children were given an appropriate level of challenge within their ZPD (Vygotsky, 1987). The activities were open-ended in nature, allowing pupils to express their creativity with no one "right" solution to the task. I also felt the children's enjoyment linked to Cziksentmihalyi's notion of "flow" (1975), where children were engaged in "hard fun".

What I found challenging	Why?
Creating Opportunities for teamwork	The level of social maturity of the children meant that some pupils found it difficult to work with others. Some of the children showed a preference for working individually rather than working with another child to build a design. I felt this resonated with Piaget's work (as cited in Ginsburg & Opper, 1988) which suggests that children's social development in terms of their capabilities to interact meaningfully with peers beyond parallel play at age 5 can vary. This also suggests that while socially constructed learning makes sense in theory, these principles are challenging to enact in the practical realities of a classroom. Designing inclusive experiences which aim to offer children meaningful opportunities to work and learn together requires careful planning and consideration on the part of the educator, particularly in the choice and set-up of tasks and experiences.
Fine Motor skills/ craft handling skills	The Aistear (2006) Framework recommends using teacher observation to assess children's play. While I was observing, I found it difficult not to intervene when children were experiencing difficulty with handling materials. In one instance, a child was having great difficulty disconnecting two Lego© bricks. I felt what I observed resonated with Fallon's (n.d.) point that it is sometimes difficult to assess children's play without intervening. She suggests it takes time and practice to learn how to do this with a group of pupils. The difficulty experienced by certain children also led me to consider whether I could have made the activity and resources more inclusive for children with limited fine motor skills, for example by offering children the choice to use a chunkier style of construction block than the Lego© ones I had offered.

What was unexpected for me

Variety of marble runs

I had not anticipated the sheer variety of marble run designs that the pupils would come up with. For example, Figure 10 shows a flat plane Lego© maze-type model designed by one of the pupils.

Figure 10. Pupil work sample. Lego© maze design

For me, this showed that my own assumptions coloured my expectations of pupils' creativity. This reminded me of the work of Roche (2007), who points out the importance of hunting out and addressing our own assumptions when seeking to develop our practice.

Minecraft

I had deliberately chosen to omit the "plan" section in my learning experience design as I was drawing from Roberts' (2016) model. However, while engaging in what Schön (1987) describes as reflection-in-action, I felt compelled to respond to unfolding events during the course of the lesson. During the child-led free play segment of the lesson, one of the learners mentioned that the marble run reminded her of a rollercoaster they had built at home in Minecraft. In line with constructionist principles which advocate utilising objects of interest to learners (Beisser, 2005; Papert, 1991), I felt this would be a good opportunity to build on this pupils' interest and prior knowledge, and asked them if they would like to make a plan of their marble run in Minecraft. Their work was shown earlier in Figure 7c. I felt my interaction with and response to this pupil resonated with the work of Sarama and Clements (2009) and Bruner (1964). Both describe the learning trajectories of children, and how with teacher scaffolding, pupils can progress with communicating their conceptual understanding from concrete to pictorial representation.

Implications for practice

What I would do differently next time

In terms of my future practice, I aim to create more opportunities for children to learn from each other. Through carefully choosing engaging, inclusive and stimulating tasks which lend themselves to discussion, I can further align my classroom practice with constructionist ideas, which underline the value and richness of such socially constructed learning (Papert, 1991). Working together on a collaborative project, rather than an individual one (as in this learning experience) would also support development of pupils' collaboration and teamwork skills. This may mean adjusting the design and success criteria of the activity to explicitly develop pupils' teamwork skills.

Engaging in the process of planning, teaching, assessing and reflecting on this learning experience has highlighted the need for me to re-examine my own assumptions and expectations around what I understand as inclusive, creative open-ended tasks. I plan to begin developing a bank of possible open-ended tasks which I could draw from in designing future learning experiences. In line with constructionist principles, the pupils' engagement with this experience allowed me to see the value of using truly open-ended tasks to tap into children's strengths and interests and using these as a springboard for deeper learning.

Broader implications for practice and limitations

Based on the data gathered, the blend of Bridge 21 with playful pedagogical approaches, used in this learning experience appears to be effective in developing young pupils' STEM literacy. However, this paper describes an individual learning experience with a small group of pupils over in a primary school context. The primary purpose of this endeavour was to inform the author's own classroom practice. Further exploration of similar pedagogical approaches in a variety of contexts with a large sample size would be useful in order to ascertain their effectiveness more broadly.

Conclusion

This paper presented a vignette and accompanying reflections which described the author's recent experience of using an innovative constructionist pedagogy as part of a 21st Century STEM learning experience with a small group of young learners aged 5-7 in a primary school setting. The aim of the learning experience was to develop pupil participants' STEM Literacy using a series of hands-on STEM-based tasks. It has highlighted the possible potential of such constructionist-founded pedagogies to build the foundations of STEM literacy, even from a young age.

References

Bagiati, A., & Evangelou, D. (2015). Engineering curriculum in the preschool classroom: the teacher's experience. *European Early Childhood Education Research Journal,* 23(1), 112–128. https://doi.org/10.1080/1350293X.2014.991099.

Beisser, S. (2005). An examination of gender differences in elementary constructionist classrooms using lego/logo instruction. *Computers in the Schools, 22*, (3) 7-19

Braun, V. & Clarke, V. (2019). Thematic analysis. *Handbook of research methods in health social sciences*. 843–860.

Bridge21. (2013). *Activity Handbook for Teachers*. Trinity College Dublin: Dublin.

Broadhead, P., Howard, J., & Wood, E. (Eds.). (2010). *Play and learning in the early years: From research to practice*. Sage.

Brookfield, S. (1995). The getting of wisdom: What critically reflective teaching is and why it's important. *Becoming a critically reflective teacher*, 1-28.

Brookfield, S. D. (2017). *Becoming a critically reflective teacher*. John Wiley & Sons.

Byrne, J. R., Fisher, L. & Tangney, B. (2015). A 21st Century Teaching and Learning Approach to Computer Science Education: Teacher Reactions, In S. Zvacek, M. T. Restivo, J. Uhomoibhi & M. Helfert (Eds.). *Computer Supported Education, 583,* 523-540. Springer International Publishing.

Conneely, C., Girvan, C., Lawlor, J., Tangney, B. (2015). *An Exploratory Case Study into the Adaption of the Bridge21 Model for 21st Century Learning in Irish Classrooms,* in editor(s) Butler, D., Marshall, K., Leahy, M., *Shaping our Future: How the lessons of the past can shape educational transformation*, Dublin, Liffey Press.

Conneely, C., Murchan, D., Tangney, B. & Johnston, K. (2013). *21 Century Learning –Teachers' and Students' Experiences and Views of the Bridge21 Approach within Mainstream Education,* Proceedings of Society for Information Technology & Teacher Education International Conference, March, AACE., New Orleans, USA.

Csikszentmihalyi, M. (1974). *Flow: Studies of enjoyment*. University of Chicago.

Dede, C. (2010). Comparing frameworks for 21st century skills. *21st century skills:Rethinking How Students Learn*, *20*, 51-76.

Department of Education and Science (DES). (1999). Primary Science Curriculum. Stationary Office.

DES. (1999). Primary Mathematics Curriculum. Stationary Office.

DES. (2016). *STEM Education Policy Statement 2017-2026*. Retrieved from https://www.education.ie/en/The-Education-System/STEM-Education-Policy/stem-education-policy-statement-2017-2026-.pdf

DES. (2017).Digital Learning Framework (Primary). Retrieved from http://www.dlplanning.ie/teaching-learning/

Edwards, C. (1993). *The hundred languages of children: The Reggio Emilia approach to early childhood education*. Ablex Publishing Corporation. New Jersey.

Fallon, J. (n.d.). Observing and Assessing Play. Aistear Toolkit. NCCA. Retrieved from https://vimeo.com/ncca/review/53877623/5a82601475.

Fisher, L., Byrne, J. R., & Tangney, B. (2015). *Exploring Teacher Reactions towards a 21St Century Teaching and Learning Approach to Continuing Professional Development Programme in Computer Science.* 7th International Conference on Computer Supported Education (CSEDU), Lisbon, Portugal, 22-31.

Froebel, F. (1826). 1887. *The Education of Man. WN Hailman (Trans.), New York: D.Appleton*.

Gerlach, J. (2012). *STEM: Defying a simple definition*. NSTA Reports. Retrieved from

http://www.nsta.org/publications/news/story.aspx?id=59305

Ginsburg, H. P., & Opper, S. (1988). *Piaget's theory of intellectual development*. Prentice-Hall, Inc.

Huling, M. & Speake Dwyer, J. (2018). *Designing Meaningful STEM Lessons*. NSTA Press.

Jolly, A. (2017). *STEM by Design: Strategies and Activities for Grades 4-8.* New York: Routledge.

Johnston, K., Conneely, C., Murchan, D., Tangney, B. (2015). Enacting Key Skills-based Curricula in Secondary Education: Lessons from a Technology-mediated, Group-based Learning Initiative. Technology, *Pedagogy and Education, 24*, (4), 423-442.

Kelley, T. (2010). Staking the Claim for the T in STEM. *Journal of Technology Studies*, *36*(1), 2-11.

Kelley, T. R. & Knowles, J. G. (2016). A conceptual framework for integrated STEM education. *International Journal of STEM Education*, *3*(1), 1–11.

Lawlor, J., Conneely, C., Oldham, E., Marshall, K., & Tangney, B. (2018). Bridge21: teamwork, technology and learning. A pragmatic model for effective twenty-first-century team-based learning. *Technology, Pedagogy and Education*, *27*(2), 211-232.

Liston, M. (2018). Unravelling STEM: Beyond the acronym of Science, Technology,

Engineering, and Mathematics, *Science*, *53* (3), 28-29.

Margot, K. C., & Kettler, T. (2019). Teachers' perception of STEM integration and education: A systematic literature review. *International Journal of STEM Education*, *6*(1), 2.

Mooney, M. A., & Laubach, T. A. (2002). Adventure engineering: a design centered, inquiry based approach to middle grade science and mathematics education. *Journal of Engineering Education, 91*(3), 309–318. https://doi.org/10.1002/j.2168-9830.2002.tb00708.x.

Moore, T. J., Stohlmann, M. S., Wang, H. H., Tank, K. M., Glancy, A., & Roehrig, G. H. (2014). Implementation and integration of engineering in K-12 STEM education. In J. Strobel, S. Purzer, & M. Cardella (Eds.). *Engineering in precollege settings: research into practice*. Rotterdam: Sense Publishers.

Morrison, J. (2006). Attributes of STEM education: The student, the school, the classroom. *TIES (Teaching Institute for Excellence in STEM)*, *20*.

Murphy, C., Murphy, C., & Kilfeather, K. (2011). Children making sense of science. *Research in Science Education, Vol 41*(2), pp. 283-298

National Council for Curriculum and Assessment (NCCA). (2009). *Aistear: The early childhood curriculum framework*. Dublin: NCCA.

Papert, S. (1987). *A new opportunity for science education.* NSF Grant Application.

Papert, S. (1991). Situating constructionism. In I. Harel & S. Papert (Eds.), *Constructionism.* Norwood, NJ: Ablex Publishing.

Papert, S. (1993). The children's machine: Rethinking school in the age of the computer. New York: Basic Books

Partnership for 21st Century Skills. (2011). P21 Common Core Toolkit: A Guide to Aligning the Common Core State Standards with the Framework for 21st Century Skills. Tucson, AZ, USA.

Piaget, J. (1964). Part I: Cognitive development in children: Piaget development and learning. *Journal of research in science teaching*, *2*(3), 176-186.

Rinaldi, C. (2004). *In dialogue with Reggio Emilia: Listening, researching and learning*. Routledge.

Roberts, P. (2016). STEM in Early Childhood: How to keep it simple and fun. *Presentation to Early Childhood Australia National Conference, 2016. Centre for Research in Early Childhood (CREC). Edith Cowan University*. *Australia.* Retrieved from http://www.ecaconference.com.au/wp-content/uploads/2016/11/Roberts_P_STEM.pdf

Roche, M. (2007). *Towards a Living Theory of Caring Pedagogy: Interrogating My Practice to Nurture a Critical Emancipatory and Just Community of Enquiry.* Unpublished Doctoral dissertation. University of Limerick.

Rolfe, G., Freshwater, D., Jasper, M. (2001). *Critical Reflection for Nursing and the Helping Professions: a User's Guide*. Basingstoke: Palgrave.

Rosicka, C. (2016). *Translating STEM education research into practice.* Australian Council for Educational Research (ACER).

Sarama, J., & Clements, D. (2009). *Early childhood mathematics education research: Learning trajectories for young children.* New York: Routledge.

Schön, D. A. (1987). Educating the reflective practitioner.

Science Foundation Ireland (SFI). (2018). Investigating Slopes. Classroom Planning Guide. *Discover Primary Science and Maths*. Retrieved from http://www.sfi.ie/engagement/discover-primaryscience-and-maths/resources/teacher-created-resources/teachers-planningguide.pdf

Tangney, B., Oldham, E., & Bray, A. (2015). Realistic Mathematics Education, Mobile Learning and the Bridge21 Model for 21st-Century Learning: A Perfect Storm. In *Mobile Learning and Mathematics*. New York: Routledge.

Van Oers, B. (2010). Emergent mathematical thinking in the context of play. *Educational studies in mathematics*, *74*(1), 23-37. Vasquez, J.A. (2015). Beyond the Acronym. *Educational Leadership*, 11-15.

Vukelich, C. (1994). Effects of play interventions on young children's reading of environmental print. *Early Childhood Research Quarterly*, *9*(2), 153-170.

Vygotsky, L. (1987). Zone of proximal development. *Mind in society: The development of higher psychological processes*, *5291*, 157.

Framing agency as a lens into constructionist learning

Vanessa Svihla, *vsvihla@unm.edu*
Organization, Information & Learning Sciences, University of New Mexico, Albuquerque, U.S.A

Tryphenia B. Peele-Eady, *tbpeele@unm.edu*
Language, Literacy, & Sociocultural Studies, University of New Mexico, Albuquerque, U.S.A

Abstract

Problem Statement. Because of prior experiences in instructionist settings, students may struggle to direct their own work in constructionist learning environments that require them to frame problems.

Aim. We introduce the construct *framing agency*, defined as opportunities to make decisions that are consequential to framing problems and learning. Specifically, we sought to ground this construct in constructionist learning experiences across a range of instructional settings.

Research questions. What does student discourse reveal about framing agency? How might framing agency shed light on the ways student engage with constructionist learning experiences?

Methodology. We selected data from previously collected cases and reanalyzed them for this discussion. In the current study, we focus on cases drawn from two sites: (1) student teams in an industry-sponsored, capstone, two-semester biomedical engineering design course at a large research university in the American West; and (2) undergraduate and graduate students from diverse backgrounds in an interdisciplinary research lab at a Hispanic-serving university in the American Southwest. All data were originally collected through Svihla's extended (nine months or longer) participant observation that included audio and video records and field notes of students' classroom interactions. We analyzed multiple cases from each site using sociolinguistic analysis to characterize framing agency. We compare and contrast these cases to illuminate some nuances of framing agency.

Key findings. We identified three key markers of agency in students' talk: hedging, sharing agency, and using verbs that express potential control. In contrast, use of verbs—even minimal use—showing no control over the problem characterized a lack of framing agency. When facing the ambiguity of framing problems, some knew to use their agency to dwell with the problem, and others situated tasks as out of their control and scope of work, even in settings in which it was clearly in scope.

Contribution. Framing agency provides a lens into how prior experiences—dominated by solving archetypical well-structured problems with predetermined solutions—can covertly structure students' engagement with constructionist learning experiences.

Keywords (style: Keywords)

Design learning, Problem framing, Agency, Discourse analysis, Teams

Statement of the problem

Instructors aiming to create constructionist learning experiences are sometimes thwarted by students' expectations about teaching and learning. Students' prior instructionist experiences in school settings, which typically focus on well-structured problems—problems that have a single correct solution and a most efficient solution pathway (Jonassen, 2000)—can lead them to expect more of the same. As a result of such experiences, students may not understand that they need to *frame* the problems they encounter in constructionist learning settings (Crismond & Adams, 2012); instead, they may treat such problems as always having a single right answer (Christiaans & Dorst, 1992; Rowland, 1992). Additionally, instructors must negotiate tensions between overly scaffolding progress, such that students have too few opportunities to make consequential decisions, and providing too little direction, such that students flounder unproductively.

Compared to problem solving and relevant psychological constructs, research has paid much less attention to effectively supporting students in their problem framing process. This is in part because of the nature of this framing, which involves interacting with diverse activities and approaches (Murray, Studer, Daly, McKilligan, & Seifert, 2019; Resnick & Ocko, 1990). Here, we argue that improved understanding of what problem framing looks like and ways to differentiate efforts to frame problems from efforts to maintain a well-structured problem space is needed. To this end, we introduce the construct *framing agency*, defined as making decisions that are consequential to framing design problems and learning through this process. We characterize framing agency by considering discursive exchanges among students working to frame problems in constructionist learning settings.

Theoretical Framework

Constructionist learning settings involve learners in participatory roles, situating them as designers (Resnick & Ocko, 1990). We argue that this aspect makes agency particularly salient for constructionist settings. To theorize framing agency, we build on research on design problem framing and agency, pulling broadly from extant literature on learning, social sciences, and beyond.

Why must design problems be framed and how do designers accomplish framing?

Design problems—compared to well-structured problems that have a single correct answer and canonical solution path—have multiple solutions and solution paths (Jonassen, 2000). As a result, designers must structure the problems they aim to solve (Restrepo & Christiaans, 2004; Schön, 1983). While a design brief may describe client needs, desires and context, the problem still requires framing (Coyne, 2005). As such, framing design problems involves understanding needs, context and requirements, setting boundaries, and exploring tentative solutions (Atman et al., 2008; Morozov, Kilgore, & Atman, 2007). Designers gather information deliberately to clarify ambiguity and rule out untenable solution paths (Basadur, Graen, & Green, 1982), seeking divergent stakeholder perspectives (Daly, McKilligan, Murphy, & Ostrowski, 2017), understanding research shortcomings of existing solutions, and identifying resources available to them (Dominick, 2001). In defining the bounds of the problem (Atman et al., 2008), designers also identify constraints and criteria for success and question the information given to them (Atman, Chimka, Bursic, & Nachtmann, 1999; Dominick, 2001). Even gathering information is itself a contextual and contingent process, as designers seek to address gaps in their own understanding (Tracy, 2005) in relation to tentative solution paths (Rittel & Webber, 1973). Designers therefore make many consequential decisions, both about the problem frame and about how to proceed in framing processes (Dorst & Cross, 2001; Schön, 1983). Problem framing is an agentive process (Hanauer, Frederick, Fotinakes, & Strobel, 2012) that builds a sense of ownership over the problem (Newell & Simon, 1972; Restrepo & Christiaans, 2004; Schön, 1987).

How is agency related to designing and learning?

In social science research, agency is typically defined as making decisions (Alsop, Bertelsen, & Holland, 2006). Agency depends on opportunity structures (Narayan & Petesch, 2007)—that is, whether there are opportunities to make decisions, whether students actually make decisions, and whether they are satisfied with the outcomes of their decisions (Alsop et al., 2006). In instructionist settings, students have limited agency. Even in student-centered classrooms, they typically only make choices about format (poster or presentation) or 'menu' options (e.g., choose an animal, holiday, explorer, to research and report on). Students seldom have choices about what to learn, or how to proceed in their problem framing and solving process in instructionist settings (Resnick & Ocko, 1990). As a result, students may flounder when they have opportunities to make such decisions in constructionist settings. Engeström and Sannino (2010) argued that an outcome of learning should be increased agency, meaning students need opportunities to practice using their agency. The coercive effects of past instructionist experiences can shape the kinds of decisions students make, suggesting the need to consider not just whether students have agency, but also what kinds of agency they have.

Why contextualize agency?

Many have argued that learning and performance are situative (Greeno, 1998; Lave & Wenger, 1991), necessitating contextualized views of constructs. For instance, self-efficacy is typically considered in relation to specific situations—few would claim that if someone had high science self-efficacy, they would also necessarily have high self-efficacy in all subjects. We argue that agency is likewise situated.

Others have proposed different *forms* of agency. For instance, Engeström and Virkkunen (2007) proposed that *transformative agency* involves making deliberate changes in one's work by resisting, considering new possibilities, committing to making changes, making consequential changes (Engeström, 2011), criticizing (Haapasaari, Engeström, & Kerosuo, 2016), and critiquing one's own agency (Heikkila & Seppänen, 2014). In this approach, different forms direct us to consider what constitutes agentive activities and their role in bringing about change.

Scholars who study agency have noted that it may be shared with others, and in design processes, it may also be shared with materials (Knappett & Malafouris, 2008) and envisioned stakeholders. In longer term collaborative work, members' accounts of their decisions as collective (e.g., "We decided to…") can reinforce their sense of shared agency (Tollefsen & Gallagher, 2017). Scholars have debated how simple alignment between two individual's choices becomes collectively shared (e.g., Bratman, 2013; Gilbert, 2009) and have investigated the individuals' capacity to engage in shared tasks. For instance, (Edwards, 2007) proposed *relational agency*—the ability to offer and seek support from individuals in different positions from one's own. Similarly, Kafai and colleagues proposed *collaborative agency*—"the ability to choose collaborators, organize work, and design together in an unstructured context where roles, tasks, and people are not specified" (Kafai, Fields, Roque, Burke, & Monroy-Hernandez, 2012, p. 65). While many have studied similar issues from the lens of self- and co-regulated learning (Law, Ge, & Eseryel, 2016), we appreciate the lens that agency brings, as it focuses on the choices individuals and groups must make, rather than on the actions conjectured to be important (e.g., asking for clarification). We argue this view of agency opens a space to empirically re-evaluate both the ways participants make decisions and the kinds of decisions they make, and both are potentially productive in problem framing. For instance, analysis of collaborative agency highlighted that students made *different* decisions about how to organize their collaborative work as the nature of the work changed across a design project and emphasized that "it is youth themselves who need to make choices about who to work with, how to contribute to work" (Kafai et al., 2012, p. 80).

Other contextualizations of agency have been proposed related to media production: critical agency relates to "learning how to critically engage with digital content and practices" and cultural agency relates to "learning how to navigate cultural identity with digital media production" (Kafai, Fields, & Searle, 2019, p. 2). Such contextualizations highlight that the kinds of decisions and information needed to make those decisions are contingent and situated.

We conjecture that framing agency—making decisions that are consequential to framing problems and learning as a result of those decisions—helps differentiate between learners' engagement in activities as instructionist or constructionist. This is important knowledge for instructors and learners as they negotiate instructional goals and shape praxis knowingly and unknowingly through discourse.

Methodology

Using a qualitative case study lens (Creswell, 2013; Ragin & Becker, 1992), we analyzed discursive exchanges among some collaborating students that Svihla documented through participant observation (Atkinson & Hammersley, 1994; DeWalt & DeWalt, 2010). We aimed to clarify framing agency as a construct and to analyze what the students' discourse revealed about the process of making decisions consequential to framing and learning in a collaborative setting. The following questions guided our analysis:

- What does student discourse reveal about the process of making decisions that are consequential to students' framing and learning?
- What kinds of discourse patterns differentiate between framing agency and other kinds of agency (e.g., deciding to disengage, making decisions that situate learning as instructionist)?
- How might framing agency shed light on student engagement with learning experiences that are intended to be constructionist in nature?

Data Collection, Prior Analysis & Case Selection

Data were gathered from two sites. First, we selected cases from a canonical design field—engineering—in a setting intentionally created to help students learn to design. While not identified as constructionist by the faculty, the course displayed many hallmarks of constructionism. Student teams ranked their choices of industry client and were generally matched to their first or second choice. The teams sought to meet a client need, detailed in a design brief, and they presented their final solutions to their clients at the end of the two-semester sequence. Yet, largely due to their experiences with prior coursework, which emphasized accuracy, some teams treated the process as finding "the right answer." Prior analysis highlighted differences in the teams' navigations of impasses in designing (Svihla, 2010; Svihla, Petrosino, & Diller, 2012). Experts' assessments of the quality and creativity of their early and final work provided an outcome variable, highlighting differences in the paths of each team over the course of many months that led to more and less creative and quality outcomes. We selected cases based in this prior study for this current analysis, focusing on two cases with different paths and outcomes—Tom's team and Steve's team.

Tom's team, mentored by teaching assistant (TA) Shanti, included Cynthia, Addai, and Greg. Their client was a physical therapist from a local hospital, who wanted a means to objectively measure spasticity in patients' limbs. The team planned to design a glove with a pressure sensor and accelerometer. However, Tom, who was adept at thinking in vector space, realized an accelerometer could be moved in ways that would register no movement. Consequently, the team worked to frame and reframe the problem. We argue they maintained an opportunity structure for members to have agency over framing the design problem.

Steve's team, mentored by TA Michelle, included Daniela, Dillon, and Bob. Their client, the director of a local biomedical technology company, wanted a way to measure specific biochemical processes in the body as an early warning system for sepsis following surgery. Steve's team struggled to define this as a design problem and resisted reframing the problem. Instead, they treated the problem as well-structured and their task as finding the right answer.

Second, we selected a case outside the canonical design fields, but in which problem framing was particularly salient: an interdisciplinary research lab focused on the roles of bacteria in caves. Due to the exploratory nature of their research, this team's work had a designerly quality as they found they needed to design new methods of data collection and analysis, make numerous decisions about representations of their results, and craft explanations for these, a practice they labeled as “finding the story in the data.” Prior analysis highlighted the diversity of the lab and ways the principal investigator encouraged student participation, even from new undergraduate members.

Denise’s lab included undergraduate and graduate students from diverse backgrounds (n=16 across four years, with approximately seven students participating at any one time): two Native American students, six Latinx students, one African American student and a majority of students who were first generation college attendees; the gender balance was generally close to even. Students majored in fields like chemistry, biochemistry, biology and geology. Denise explained that she recruited students who were “in the middle academically,” and who might be in “danger of leaving or not considering science careers.”

Data Analysis

We re-transcribed data to allow for a more fine-grained analysis that focused on the discursive exchanges within and across the teams. We first adapted the agency toolkit, an approach to linguistic analysis that directs attention to autonomy (Konopasky & Sheridan, 2016) (Figure 1). Konopasky and Sheridan (2016) developed the agency toolkit based on their analysis of interviews with adults about their decisions to drop out of school as youth and later enter a GED program; their focus included intentional causation and degree of autonomy.

To characterize *intentional causation* in discourse, the researchers considered whether the speaker framed themself as an agent or as someone acted upon by focusing on the use of "I" or "we" as opposed to placing oneself as the object (e.g., "It was assigned to me") and using verbs that involve an action that affects someone else ("I showed him how to solder" versus "I soldered the LEDs"). While Konopasky and Sheridan (2016) argued that "I" and "we" could be treated interchangeably, we posited that in team design settings, it is important to differentiate between these. Likewise, our past work on design teams has made clear that materials are salient actors in design process (2018), as designers have reflective conversations with the materials (Schön, 1992).

Konopasky and Sheridan (2016) characterized the *degree of autonomy* using hedging and mitigation, such as evidenced by using the generic "you" in which the speaker places themselves amongst many, reducing their agency or using a clause that offloads agency onto another or the environment ("I used a pipe cleaner because it was all I had"). We sought to differentiate between those cases and situations in which designers justified their decisions ("I used a pipe cleaner because it is conductive and unexpected"). In addition, because designers aim to remain tentative in early problem framing, we posit that hedge words might reveal this stance as team members negotiated design ideas.

We more carefully focused on situations in which students exhibit a *lack of control* and *potential control*. We characterized this through verb and pronoun use (Figure 1), anticipating that this approach, paired with review of the targets of their comments, would provide clearer differentiation between displays of agency and their connections to framing and non-framing actions.

- **Shared agency marker.** First person plural pronoun (we, we're, we've, we'll)
- **Tentative agency marker.** Speaker modifies statement with diminishing hedge terms (like, actually, perhaps, maybe, kind of, possibly, might, apparently, just, sometimes, etc.)
- **Tentative agency marker.** Verbs that show potential control (could, might, should, can, going to, would, want to, etc.)
- **Low agency marker.** Verbs that indicate a lack of control (told to, have to, need to, must, required, supposed to, etc.)

Figure 1. Markers of agency used to differentiate framing agency from other forms

Reliability & trustworthiness

We used common strategies to ensure trustworthiness in data collection and analysis (Maxwell, 2012): (1) *Purposive sampling*. For the first site, the original team selection emphasized heterogeneity, with input from faculty, teaching assistants, and access to students' course grades and internship participation. For the current analysis, we likewise emphasized heterogeneity in participation and outcomes, allowing us to compare and contrast. For the second site, we reviewed all data previously coded as involving problem framing, as our goal here was to contrast a non-canonical design site with a canonical design site. (2) *Intensive involvement*. In the original data gathering, Svihla spent approximately 100 hours with each team over the course of one academic year from the first site, and approximately 150 hours with the second site. She developed rapport with the participants; and in both sites, members sought her advice on various aspects of their work. Her sustained presence and observations helped reduce the potential for spurious inferences. (3) *Individual coding & debriefing.* For this current work, in effort to reduce bias, we individually (re)analyzed the transcripts, first making our own inferences about them, and then meeting several times to discuss our understandings. (4) *Triangulation.* Given the large data corpus, we also reduced bias by triangulating the findings over time, across individual participants and teams, and across data sources. For discourse data, this involved reviewing their speaking style (e.g., how commonly they used hedge words) across multiple interactions with different individuals. (5) *Member checking.* The original analysis was subject to formative and final member checking by participants. However, due to the timing of the analysis reported here, additional member checking was not feasible. This limitation should be addressed by additional studies.

Results

In the next sections, we present our analyses of the following vignettes to illustrate differences in students' talk as they both enacted and shied away from framing agency.

Tom's team: Tentative, shared problem framing

After receiving the accelerometer, Tom realized that if one were to move in a direction opposite to the direction of gravity at the same velocity as gravity, no motion would be recorded. Later (in mid-February), he spent an hour carefully presenting this anticipated problem to his team. The team members initially seemed concerned there was no way forward. Addai put forward a tentative solution, displaying relatively low agency (Figure 2).

Addai minimized risk associated with introducing his idea by calling it a "first draft." His hedge words and use of the generic "you" mitigated his agency. Tom reacted positively, widening the opportunity for Addai to pursue this line of thinking, which scaffolded Addai to continue reframing the problem. In response, Addai's discourse was less tentative; he shared agency with his team ("we").

Vignette 1: Feb 11

Addai: Instead of taking measurements in three dimensions, this is this is like maybe a first draft. //
Tom: //hm!//
Addai: //You throw away the position information.
Tom: Right.
Addai: And we roll the XYZ coordinates into just one combined vector and that way we've always accounted for your full gravitational contribution.
Shanti: That's a good idea!
Cynthia: Yeah.
Shanti: Like a magnitude (.) of all three of them like a//
Addai: //Exactly. Exactly. So if you roll them all together you can still figure out

Vignette 2: 5 min. later

Addai: Like I said I'm still not sold on it, but. I'm not sold on it, but I like the way it looks.
Tom: mmhmm [positive]
Shanti: Yeah anyway try it out it might work I don't know.
Addai: [quietly] You do lose, uh I think you do lose your position because you rolled all of your axes. [louder] But it would be a much easier way also to keep track of your overall change

Figure 2. Vignettes from Tom's team, color-coded as defined in Figure 1

In vignette 2, although Addai presented his idea as one he was not yet "sold on," Shanti encouraged the group to "try it out" because "it might work." By doing so, Shanti scaffolded Addai's thinking and advanced the team's framing process, without taking an authoritative role, although this is the role afforded her as a TA. Addai acknowledged the team's concerns, but exhibited a firm belief in his idea by increasing the volume of his talk.

Across these and other interactions, Tom and Shanti—both occupying positions of power—maintained opportunities for other members to reframe the problem. They scaffolded Addai to move from throwing forward the earliest draft of an idea to ultimately displaying shared ownership of a solution that eventually came from his reframing of the problem.

Steve's team: Shutting down framing

Steve's team generally displayed agency to solve the problem as given to them (Figure 3). Concerned they were not *designing* anything, TA Michelle encouraged them to "try to have some kind of engineering analysis" and pressed them to explain why their project was "so great." Her concern reflected the instructor's comment, "What can you really uniquely contribute as an engineer?" as she pressed, "Why is there a need for it?" The students explained the potential for saving lives by having a way to detect symptoms of shock, yet they generally sought *right answers*, for instance, investigating whether a sensor performed according to specs in an experiment.

At the beginning of vignette 3, Daniela's interactions showed framing agency as she brought up her concern about the plan, but Steve and Dillon rejected her idea as out of scope. This pattern was common: Steve displayed high agency and Dillon repeated Steve's words as if to amplify them. Steve made decisions, but they were not consequential to the *framing* of a design problem. In this, he and Dillon shut down Daniela's attempts to frame the problem, limiting the impact of her ideas on the discussion. During this exchange, TA Michelle offered no scaffolding, and the problem failed to become a design problem. Daniela expressed frustration multiple times ("something bothers me").

Vignette 3: Feb 4

Daniela: I just thought that something bothers me the fact that (.) yeah we're gonna put the sensor on the stomach (.) right? During surgery? (.) But then (.) we're gonna, the surgery only lasts like one::: to two hours and we're gonna take it off and the patient is gonna be, (.) um well the surgery is gonna be over and there's not gonna be any monitoring afterwards, and I'm thinking (.) Well there's higher chance of sepsis or shock appearing after surgery. So::: should we think about leaving the sensor? or::: (.) 'cause I don't really think it's//

Dillon: //Seriously, that could be like, the next step.

Steve: Yeah.

Dillon: Right.

Steve: I think that—are you talking about like for like in real life? like

Daniela: Yeah. Like what what's the use of it if // you're just gonna

Steve: //I think

Bob: //I thought I thought the problem—the project was to do an internal sensor that it could be left there.

Vignette 4: 2 min later

Steve: I would think that would be something left up to a surgeon or something to be honest I mean likelikeliek our project. I think it's kinda outside the scope of our project our project is//

Bob: //If we left it up to the surgeon and whoever actually designs the sensor.

Steve: Yeah whoever is really doing this.

Bob: 'cause we're not supposed to be designing anything.

Figure 3. Vignettes from Steve's team, color coded as defined in Figure 1

Research lab: Tentative, less shared

In Denise's lab, members commonly contributed to framing, while generally leaving ownership of the problem with the eventual author (Figure 4). Lab members explicitly engaged the interdisciplinary nature of their work, as when Tania talked about getting a new idea by attending a talk outside her field (Vignette 5). Tania's introduction of how she got her new idea showed that she knew outsider points of view were valued within the lab, and she presented this with little tentative talk. But as she began to explain the connection she saw between the "mass of the star" and her bacteria, she became hesitant, using more tentative language as she made her idea clearer, training off with "I don't know." Denise, however, seemed intrigued, evidenced by her first "hmm!" and encouraged Tania to go on; she did, though with a little uncertainty in her voice. Denise affirmed that while she did not have the expertise to evaluate the idea, she recognized the approach as potentially useful. This exchange showcases both Tania's comfort in bringing a new idea forward and Denise's work to make space for new ideas; each individual demonstrated value for other points of view.

In vignette 6, as they began to frame this problem, Nora and Denise negotiated what might be "interesting" to attend to. Denise's idea that the contrast between the acidic surface and basic cave environment could "actually be important" presented the problem as tentative and the possible variables as likewise tentative. This invited Nora, an undergraduate student, to contribute to the team's work using her knowledge from chemistry coursework. Denise showed that she valued this contribution as she built on Nora's explanation that the rate "changes for every temperature." This exchange illustrates how Denise welcomed new ideas and relied on students to contribute substantially during lab meetings.

Vignette 5: Stars talk

Tania So I had this idea. I went to this stars and constellation talk. Totally different. Not my field. But someone, or one of the speakers spoke about how they look at the color of light that is being emitted that would help tell the mass of the star.
Denise: Okay.
Tania: And I am wondering if there is something similar where//
Denise: //hmm!//
Tania: //bacteria, if they are emitting a light or a (.) compound that maybe you could. I don't know.
Denise: There may be something along—
Tania: Maybe something out there that could at least tell us what it looks like?
Denise: That's interesting. This may be some of the techniques, I just don't know, but it would be worth—One of the ways to make progress on something like this is to look at what other fields do.

Vignette 6: Dichotomies

Denise Here's a question. Caves tend to not—What happens if there's a dichotomy? It's very acidic because of pine trees, but the cave is basic. If there's a dichotomy like that, what happens? That could be a question.
Nora: It is—pH and temperature is really important.
Denise: There are wildly different temperature regimes between surface and subsurface.
Nora: Inside the cave is interesting//
Denise: //But that change from surface to cave. (.) Could be really important. At least at [the particular cave location] it is high elevation, juniper, pinion, high enough at that boundary. That may actually be important. Really good stuff.
Nora: pH, temperature will effect it too. Remember in chemistry? The rate limiting step? The slope changes for every temperature.
Denise: It's cold in caves.
Nora: That could slow the process.

Figure 4. Vignettes from Denise's lab, color coded as defined in Figure 1

Discussion

By analyzing design talk, we identified discourse patterns connected to agency in problem framing. Across both cases from canonical design settings, we saw the team leader work within an initial problem frame and another member tentatively put forward an idea. In Tom's team, this idea eventually became part of the final solution, which yielded a reframed problem. Essentially, Tom and Shanti opened spaces for Addai to reframe the problem.

In contrast, when Daniela suggested leaving the sensor in, Steve and Dillon rejected her idea as out of scope. Steve's displays of confidence prevented negotiation of the problem space. While Bob opened a space to reconsider Daniela's reframing, his effort was fleeting. This pattern was consistent when new ideas were put forth, as if the team did not see its role as shaping the problem. We do not know if Daniela's experiences drove her to find design encounters elsewhere, but we see this as a missed opportunity to learn to frame problems and her peers did not get to learn from her through this process.

Our analysis of Denise's lab highlighted similar patterns reflected in Tom's team, with Denise scaffolding students' tentatively proposed ideas, opening space for students to frame without hesitation. In contrast to the design teams, in Denise's lab, much of the framing work was situated individually rather than shared collectively. Yet, even situated as such, we see evidence for distributed constructionism (Resnick, 1996) in members' efforts to contribute to another's problem frame.

While these data and our analysis of them are limited in time and scope, we argue the contrast between Tom's and Steve's teams provides insight about the kinds of experiences that can help learners develop increased capacity for framing agency. Tentative talk—commonly noted as suggesting low agency (Konopasky & Sheridan, 2016)—rather than being a barrier, was actually a resource for Addai to develop his idea because his collaborators nurtured his participation. This aligns with past work showing that responsiveness/politeness helps groups successfully solve well-structured but complex problems (Barron, 2003; Chiu, 2008) when they have authority to work on problems (Brown & Campione, 1998) and act on this authority. By seeking to conceptualize framing agency as a specific skill set within interactional contexts, we bring renewed focus and clarity to the kinds of framing moves that learners might make, and the ways their peers and instructors can support them to develop tentative ideas into solutions over which they feel a sense of ownership. For students to develop capacity to frame problems, they need constructionist experiences and supports that help them move beyond the instructionist settings to which they have become accustomed.

Our future work investigates how this discourse analysis approach may be used to inform interventions that could support the development of framing agency in varied learning settings. Specifically, we wonder about ways to help students recognize the need to frame the problem, and if attending to their talk will help them track their reactions to the ambiguity that problem framing presents.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. EEC 1751369. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Alsop, R., Bertelsen, M. F., & Holland, J. (2006). *Empowerment in practice: From analysis to implementation*. Washington, DC: World Bank Publications.

Atkinson, P., & Hammersley, M. (1994). Ethnography and participant observation. In N. Denzin & Y. S. Lincoln (Eds.), *Handbook of qualitative research* (Vol. 1, pp. 248-261). Thousand Oaks, CA: Sage Publications.

Atman, C. J., Chimka, J., Bursic, K. M., & Nachtmann, H. L. (1999). A comparison of freshman and senior engineering design processes. *Design Studies, 20*(2), 131-152.

Atman, C. J., Yasuhara, K., Adams, R. S., Barker, T. J., Turns, J., & Rhone, E. (2008). Breadth in Problem Scoping: a Comparison of Freshman and Senior Engineering Students. *International Journal of Engineering Education, 24*(2), 234-245.

Barron, B. (2003). When Smart Groups Fail. *Journal of the Learning Sciences, 12*(3), 307-359.

Basadur, M., Graen, G. B., & Green, S. G. (1982). Training in creative problem solving: Effects on ideation and problem finding and solving in an industrial research organization. *Organizational Behavior and Human Performance, 30*(1), 41-70.

Bratman, M. E. (2013). *Shared agency: A planning theory of acting together*: Oxford University Press.

Brown, A. L., & Campione, J. C. (1998). Designing a community of young learners: Theoretical and practical lessons. In N. M. Lambert & B. L. McCombs (Eds.), *How students learn: Reforming schools through learner-centered education* (pp. 153-186). Washington, D.C.: American Psychological Association.

Chiu, M. M. (2008). Flowing Toward Correct Contributions During Group Problem Solving: A Statistical Discourse Analysis. *Journal of the Learning Sciences, 17*(3), 415-463.

Christiaans, H. H. C. M., & Dorst, K. (1992). An empirical study into design thinking. In N. Cross, K. Dorst, & N. F. M. Roozenburg (Eds.), *Research in design thinking*. Delft, The Netherlands: . Delft University Press.

Coyne, R. (2005). Wicked problems revisited. *Design Studies, 26*(1), 5-17.
Creswell, J. W. (2013). *Qualitative inquiry and research design: Choosing among five approaches*. Thousand Oaks, CA: Sage.
Crismond, D. P., & Adams, R. S. (2012). The informed design teaching and learning matrix. *Journal of Engineering Education, 101*(4), 738-797.
Daly, S. R., McKilligan, S., Murphy, L., & Ostrowski, A. (2017). Tracing Problem Evolution: Factors That Impact Design Problem Definition *Proceedings of Design Thinking Research Symposium 2016*: Copenhagen Business School.
DeWalt, K. M., & DeWalt, B. R. (2010). *Participant observation: A guide for fieldworkers*. New York, NY: Rowman Altamira.
Dominick, P. G. (2001). *Tools and tactics of design*: Wiley.
Dorst, K., & Cross, N. (2001). Creativity in the design process: co-evolution of problem-solution. *Design Studies, 22*(5), 425-437. doi:https://doi.org/10.1016/S0142-694X(01)00009-6
Edwards, A. (2007). Relational Agency in Professional Practice: A CHAT Analysis. *Actio: An International Journal of Human Activity Theory*(1).
Engeström, Y. (2011). From design experiments to formative interventions. *Theory & Psychology, 21*(5), 598-628.
Engeström, Y., & Sannino, A. (2010). Studies of expansive learning: Foundations, findings and future challenges. *Educational Research Review, 5*(1), 1-24.
Engeström, Y., & Virkkunen, J. (2007). Muutoslaboratorio–kehittävän työntutkimuksen uusi vaihe. *Teoksessa Ramstad, E. & Alasoini, T.(toim.) Työelämän tutkimusavusteinen kehittäminen Suomessa: lähestymistapoja, menetelmiä, kokemuksia, tulevaisuuden haasteita. Työelämän kehittämisohjelman raportteja, 53*, 67-88.
Gilbert, M. (2009). Shared intention and personal intentions. *Philosophical studies, 144*(1), 167-187.
Gravel, B., & Svihla, V. (2018). *Big problems, real problems: Fostering heterogeneous engineering practices through whole-class design problems.* Paper presented at the AERA, New York, NY.
Greeno, J. G. (1998). The situativity of knowing, learning, and research. *American Psychologist, 53*, 5-26.
Haapasaari, A., Engeström, Y., & Kerosuo, H. (2016). The emergence of learners' transformative agency in a Change Laboratory intervention. *Journal of Education and Work, 29*(2), 232-262.
Hanauer, D., Frederick, J., Fotinakes, B., & Strobel, S. (2012). Linguistic analysis of project ownership for undergraduate research experiences. *CBE-Life Sciences Education, 11*(4), 378-385.
Heikkila, H., & Seppänen, L. (2014). Examining developmental dialogue: the emergence of transformative agency. *Outlines. Critical Practice Studies, 15*(2), 05-30.
Jonassen, D. H. (2000). Toward a Design Theory of Problem Solving. *Educational Technology Research and Development, 48*(4), 63-85. doi:10.1007/BF02300500
Kafai, Y. B., Fields, D. A., Roque, R., Burke, W. Q., & Monroy-Hernandez, A. (2012). Collaborative agency in youth online and offline creative production in Scratch. *Research and Practice in Technology Enhanced*.
Kafai, Y. B., Fields, D. A., & Searle, K. A. (2019). Understanding Media Literacy and DIY Creativity in Youth Digital Productions. *The International Encyclopedia of Media Literacy*, 1-10.
Knappett, C., & Malafouris, L. (2008). *Material agency: towards a non-anthropocentric approach*: Springer.

Konopasky, A. W., & Sheridan, K. M. (2016). Towards a Diagnostic Toolkit for the Language of Agency. *Mind, Culture, and Activity, 23*(2), 108-123. doi:10.1080/10749039.2015.1128952
Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. New York, NY: Cambridge University Press.
Law, V., Ge, X., & Eseryel, D. (2016). The Development of a Self-regulation in a Collaborative Context Scale. *Technology, Knowledge and Learning, 21*(2), 243-253.
Maxwell, J. A. (2012). *Qualitative research design: An interactive approach: An interactive approach*: Sage.
Morozov, A., Kilgore, D., & Atman, C. (2007). *Breadth in design problem scoping: Using insights from experts to investigate student processes.* Paper presented at the American Society for Engineering Education Annual Conference and Exposition.
Murray, J. K., Studer, J. A., Daly, S. R., McKilligan, S., & Seifert, C. M. (2019). Design by taking perspectives: How engineers explore problems. *Journal of Engineering Education, 108,* 248-275. doi:10.1002/jee.20263
Narayan, D., & Petesch, P. (2007). Agency, opportunity structure, and poverty escapes. *Moving Out of Poverty: Cross-Disciplinary Perspectives on Mobility,* 1-44.
Newell, A., & Simon, H. A. (1972). *Human problem solving* (Vol. 104): Prentice-Hall Englewood Cliffs, NJ.
Ragin, C., & Becker, H. (1992). *What is a case?: Exploring the foundations of social inquiry*. New York: Cambridge University Press.
Resnick, M. (1996). *Distributed constructionism.* Paper presented at the Proceedings of the 1996 international conference on Learning sciences.
Resnick, M., & Ocko, S. (1990). *LEGO/Logo--learning Through and about Design*: Epistemology and Learning Group, MIT Media Laboratory.
Restrepo, J., & Christiaans, H. (2004). Problem Structuring and Information Access in Design. *Journal of Design Research, 4*(2), 218–236. doi:10.1504/JDR.2004.009842
Rittel, H., & Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy Sciences, 4*(2), 155-169.
Rowland, G. (1992). What do instructional designers actually do? An initial investigation of expert practice. *Performance Improvement Quarterly*, *5*(2), 65-86.
Schön, D. A. (1983). *The Reflective Practitioner: How Professionals Think in Action*. New York: Basic Books.
Schön, D. A. (1987). *Educating the reflective practitioner*. San Francisco: Jossey-Bass.
Schön, D. A. (1992). Designing as reflective conversation with the materials of a design situation. *Research in Engineering Design, 3*(3), 131-147.
Svihla, V. (2010). Collaboration as a dimension of design innovation. *CoDesign: International Journal of CoCreation in Design and the Arts, 6*(4), 245-262. doi: 10.1080/15710882.2010.533186
Svihla, V., Petrosino, A. J., & Diller, K. R. (2012). Learning to design: authenticity, negotiation, and innovation. *International Journal of Engineering Education, 28*(4), 782-298.
Tollefsen, D., & Gallagher, S. (2017). We-narratives and the stability and depth of shared agency. *Philosophy of the Social Sciences, 47*(2), 95-110.
Tracy, P. (2005). *Design and Problem Finding in High Schools: A Study of Students and Their Teacher in One Queensland School.* (Masters), Griffith University.

Fundamental Concepts of 3D Turtle Geometry

Manuel Riel, *manuel.riel@fu-berlin.de*
Didactics of Computer Science, University of Potsdam, Germany

Ralf Romeike, *ralf.romeike@fu-berlin.de*
Computing Education Research Group, Free University Berlin, Germany

Abstract

Seymour Papert's turtle in Logo has launched a revolution in programming enabling students, even at a young age, to create their own art using elementary geometry. Recent developments allow for creating turtle art in 3D: Visual, block-based programming environments, such as Beetle Blocks, in explicit tradition of Papert's turtle graphics, help students to design their own 3D models and convert them from the virtual to the physical world using 3D printing technology. However, reported experiences from previous teaching attempts involving 3D turtle geometry faced a significant challenge: in order to "draw" such beautiful three-dimensional objects, complicated advanced mathematical methods seemed inevitable, even for creating basic figures. This encouraged us to explore in detail, how artistic figures can be created in 3D using intuitive geometric knowledge suitable for novice programmers. Therefore, as a first step to understand what makes creating 3D turtle art complicated, we identify the underlying conceptual difference between the two-dimensional and three-dimensional space for turtle geometry: the existence of two entirely different coordinate systems. Building on this, we discuss in depth, how traditional turtle geometry can be applied for creating 3D objects, and explain a universal approach to creating a wide range of shapes in 3D turtle geometry. Subsequently, we present our experiences from the classroom gained in our teaching series based on the discussed concepts. In conclusion, when taught in an intuitive way, 3D turtle geometry offers another motivating setting for fostering creativity.

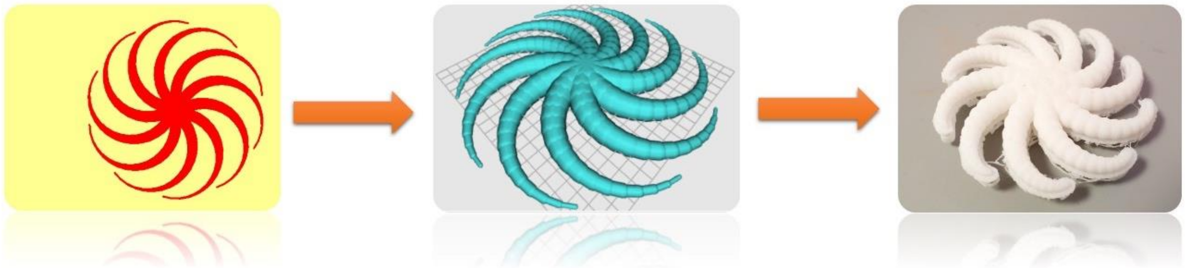


Figure 1. Classic 2D turtle art transformed from 2D to 3D and printed out as physical object. [contains image from turtleart.org[18]]

Keywords

3D printing, 3D model, computing education, programming introduction, Beetle Blocks, turtle art

[18] https://turtleart.org/gallery/imagepage.html?50

1. Introduction

Within the last years, fab labs and makerspaces have shown that modern fabrication processes, which reach from 3D modeling objects in CAD environments to manufacturing them using additive or subtractive technologies, motivate individuals to create small items for personal use. An aspect of this fascinating maker culture can be brought to the classroom in CS education when using 3D printers to build previously self-designed 3D models.

However, professional CAD environments used in fab labs are quite hard to understand for students. Instead, this technique can be applied in the classroom with one of the following approaches:

1) An older, more “traditional” way which utilizes easy-to-use CAD environments, e.g. TinkerCAD (Buehler et al., 2014) or Google SketchUp (Lutz, 2013). Students use pre-defined 2D and 3D shapes, such as circles, cubes, and spheres, to create new objects by combining the given shapes.

2) Constructionist driven *programming* environments which directly follow the ideas of Logo – but add the third dimension: In Beetle Blocks, a beetle – like Logo’s turtle – is controlled by students with simple programming commands. On instruction, it extrudes shapes along the travelled path, hence creating Turtle Art in 3D (Romagosa et al., 2016).

Most educators around the world prefer the latter approach, because it inherently supports the constructionist geometric idea for novice programmers. Furthermore, it seems more intuitive than the usual abstract (de-)composition of objects in conventional CAD environments.

Offering great potential for constructionist approaches, 3D printing technology allows students to convert their 3D models, which have been created in environments like Beetle Blocks, from the virtual to the physical world. However, in the light of experiences with existing teaching attempts for 3D printing in CS education, described by Kastl et al. (2017), one major challenge becomes apparent: Even in higher school classes students faced difficulties due to complicated mathematical methods used for creating even simple figures.

Encouraged by these reports, in this paper we explore in depth, how the intuitive concepts of traditional turtle geometry can be brought to 3D. Therefore, the paper is structured as follows: In section 2 we analyze the underlying conceptual difference between the two-dimensional and three-dimensional space for turtle geometry, which causes the challenge mentioned above: The existence of two different, three-dimensional coordinate systems with two different reference points. We explain further, how this affects the process of constructing 3D objects. Building on this, we discuss in section 3, which concepts of traditional turtle geometry are already appropriate for creating 3D objects or how strategies can be adapted in 3D – always considering novice programmers keeping the mathematical background intuitive. Additionally, we present an easily accessible approach, which supports creating 3D turtle geometry and can be used in a wide range of projects. Based on the concepts discussed, we designed a teaching series for novice programmers and share our experiences from the classroom presenting students’ final projects in section 4: Supported by fundamental concepts in 3D turtle geometry students can create their own 3D objects.

2. Conceptual Differences in 2D vs. 3D Turtle Geometry

Before we can discuss fundamental concepts of turtle art in 3D, further investigation of the geometrical situation in 3D is needed, which differs more from 2D than expected: Not only a third axis is added to the 2D coordinate system, but also *two* different three-dimensional coordinate systems become apparent (see fig. 2).

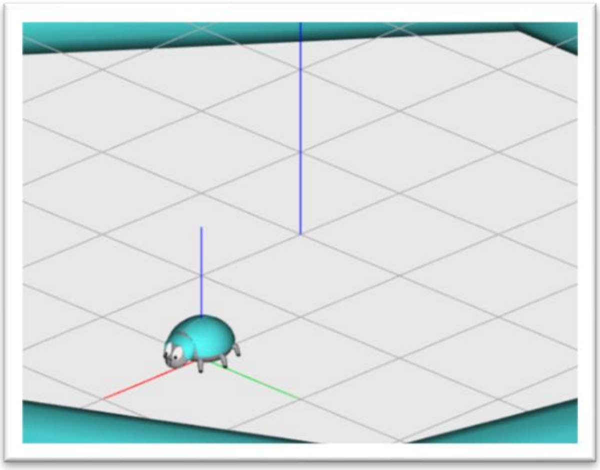


Figure 2. The two coordinate systems in 3D turtle geometry, easily recognizable by the particular blue z-axis.

In this section we explain, why two different coordinate systems are necessary for 3D turtle geometry and how they affect strategies for creating three-dimensional objects. Furthermore, we discuss strategies for helping students to overcome issues when entering the third dimension.

2.1 The Need for Two Coordinate Systems

The existence of two different reference systems is a result of two intuitive and complementary requirements for a 3D design environment. As a consequence, both coordinate systems differ in their reference point as well as in their movability:

1) The world’s absolute coordinate system starts in the world’s origin and thus is fixed there. It allows to name the position of the beetle or other objects in the "traditional" way; e. g. (1, 4, 3) for the point in the world with coordinates x=1, y=4 and z=3.

2) The beetle’s coordinate system arises in the beetle itself and hence is dynamic: It moves together with the beetle’s position and supports programming from the beetle's point of view – which makes 3D turtle graphics possible. This coordinate system is primarily important for turning and moving the beetle within the three-dimensional space.

By using these two coordinate systems, which both seem intuitive at a first glance, inconsistencies can result from mixing up their specific conceptions, which will be addressed in the following section. An additional challenge arises from the fact that Beetle Blocks lacks an explicit separation of their dedicated instructions. Since there is no extra category (like the existing “Motion” or “Control”) it is important to keep the different coordinate systems in mind while programming (see fig. 3).

commands from the view of the world's coordinate system | **equivalent instructions from the beetle's point of view**

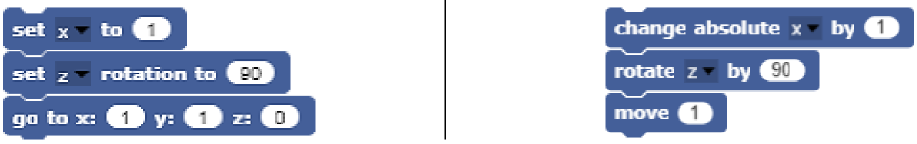


Figure 3. Exemplary comparison of instructions from different point of views.

2.2 Two Coordinate Systems – Two Ways for Creating 3D Objects

The two coordinate systems allow 3D objects to be constructed in two common, but quite different ways, which we will explain in the following: One possibility is to move the beetle like it is "going up stairs" by rotating the beetle around of one of its horizontal axes, i.e. the x- or y-axis. The other option is to create two-dimensional base areas and layering them on top of each other (see fig. 4).

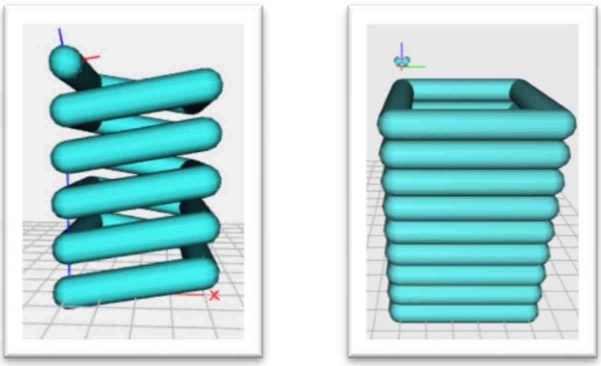


Figure 4. The two ways of creating 3D objects:
"Going up stairs" on the left, layering base areas on the right.

However, even though the first way seems to be the intuitive and turtle-like approach, it poses a significant challenge to the students due to a likely confusion of the coordinate systems, which will be discussed in detail in the next section. In order to overcome such issues, we suggest initially using the second approach, which we will illustrate afterwards.

2.2.1 "Going Up Stairs"

We explain, why "going up stairs" in 3D is complicated, by illustrating an example from the classroom: A student intended to program a spiral, which consists of "shifted" squares, but the attempt failed due to a misconception regarding the two coordinate systems. It is not sufficient to "rotate the beetle upwards" ("rotate y by 10"), as figure 4 illustrates.

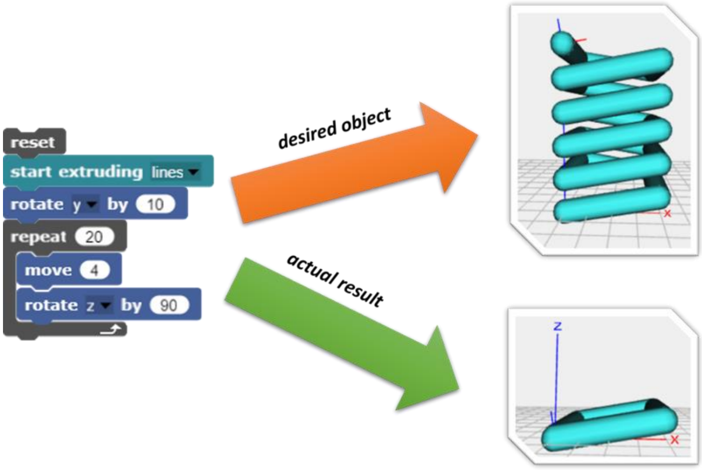


Figure 5. Misconception resulting from the existence of two reference systems

This misconception, which can quickly be adopted by novice programmers, is that the rotational situation of the beetle is not only changed by the block "rotate y by 10", but also by "rotate z by 90" - in relation to its own dynamic coordinate system, not the world's fixed reference system.

Recurring to the spiral in figure 4: When implemented correctly, a *repeated* "rotation" around the y-axis (from beetle's perspective) is necessary[19] in order to achieve the desired result (see fig. 5). This makes the program quite complicated.

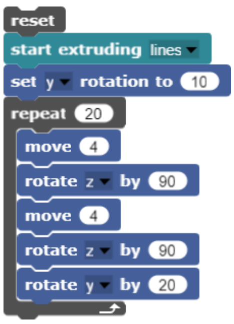


Figure 6. Correct implementation of the spiral.

Generally speaking, rotations around different axes are not even commutative – which makes the geometrical concept even harder to understand. On top of that, horizontal rotations as a strategy for creating three-dimensional objects can hardly be applied systematically to a variety of objects. In conclusion, "going up stairs" as well as horizontal rotations in general most likely should be avoided in CS introductory lessons.

2.2.2 Layering Base Areas

In contrast to "going up stairs", layering up base areas on top of each other seems to be an all intuitive approach for creating three-dimensional objects. At a first glance, it looks like traditional turtle art in 2D can just be layered on top of each other for creating three dimensional figures (see fig. 7).

[19] Note that Beetle Blocks developer, Bernat Romagosa, has made an optional library for movements relative to the beetle's orientation available in Beetle Blocks' forums: This library allows the spiral in fig. 5 to be implemented quickly, but still shares the other issues from two-axes-rotations.

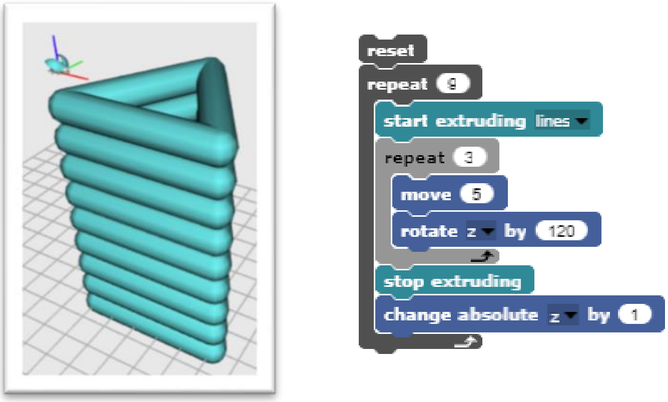


Figure 7. Triangles layered up to create a tower.

However, the situation is more complicated than that: It is essential to keep the vertical axis of the desired 3D object in mind, which goes through the starting point of the beetle when generating the base area. So, while in 2D the turtle's starting point for figures usually does not matter, in 3D the beetle's starting point – correspondingly the starting point for its dynamic coordinate system – will define the vertical axis of the resulting 3D object. We illustrate this phenomenon by using the example of a vase consisting of equilateral triangles in increasing sizes (see fig. 8), similar to the vases published by Kastl et al. (2017).

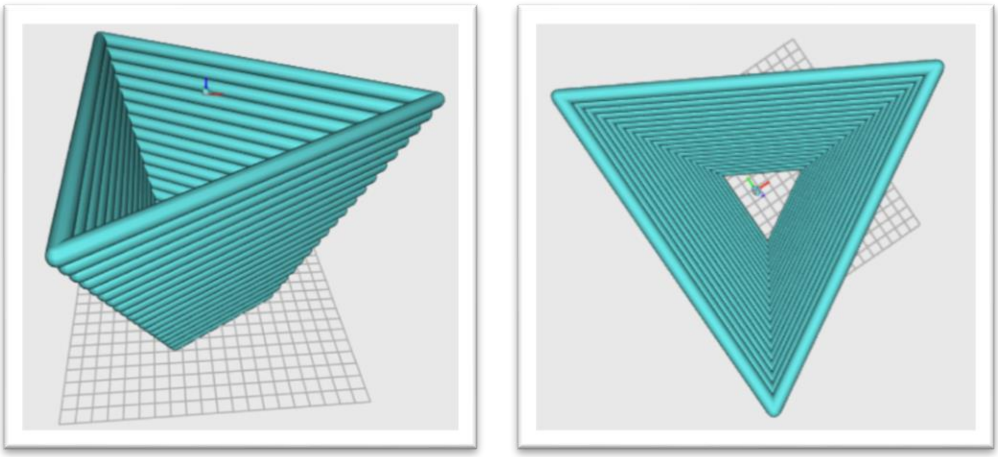


Figure 8. Vase-like object consisting of equilateral triangles.

First, we create – in the typical Logo way – an equilateral triangle as base area for the vase (see fig. 9). The beetle's start position is – as usual in Logo – in one corner point of the polygon.

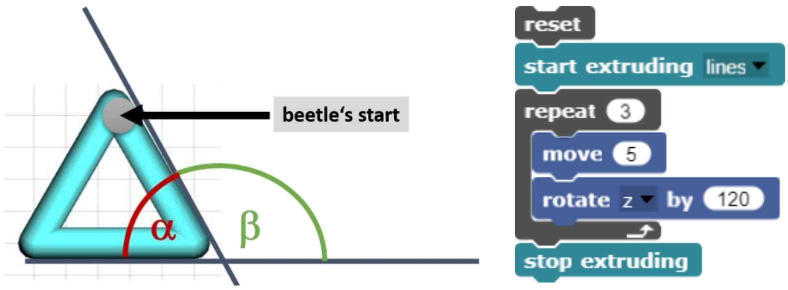


Figure 9. Creating a triangle with side length 5 by rotating the beetle around β = 120°: Just like in two-dimensional Logo.

In the next step, we layer the triangles without varying their size – this will result in the tower already shown in figure 7 – looking good so far.

However, when modifying the program in the final step to continuously increase the triangles' side lengths, a leaning tower will emerge instead of the intended vase (see fig. 10).

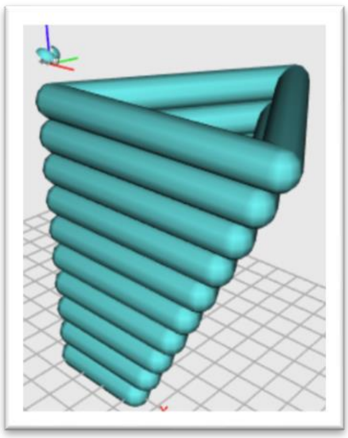


Figure 10. Resulting leaning tower by layering traditional constructed triangles in increasing size.

As explained, this is another effect of the two coordinate systems, particularly the vertical z-axis of the beetle's dynamic coordinate system, and leads us to *the* main challenge in 3D turtle geometry: In order to create the desired vase – or generally speaking: in order to create any 3D object symmetrical to its vertical axis – it is essential to start the creation process of their base area in its center point. In existing teaching attempts this is achieved by advanced mathematical methods, like calculating the coordinates of corners using the Pythagorean theorem (Kastl et al., 2017).

Following the proposal of Kastl et al. (2017) to find simpler mathematical methods for 3D objects like the vase above, we explore possible solutions in the subsequent section.

3. Exploring Concepts of Turtle Geometry in 3D

Our aim is to systematize fundamental concepts of 3D turtle geometry, which support students in creating their own 3D objects: These concepts should be easily accessible even for novice programmers and be (re-)usable in a large variety of different projects. Thus, we explore in this section, which proven concepts of traditional turtle art in 2D can also be applied in 3D, or how they can be adapted to 3D. Subsequently, we put forward a universal approach addressing the vertical axis problem.

3.1 Traditional Turtle Art for flat objects in 3D

While new approaches to base areas are required when creating actual three-dimensional objects, traditional turtle geometry can be applied in 3D without further modifications and achieve appealing results: For example, the swirl from the logo-inspired Turtle Art Gallery[20] can also be implemented in an appealing three-dimensional way (see fig. 11) and even looks good printed out (see fig. 1).

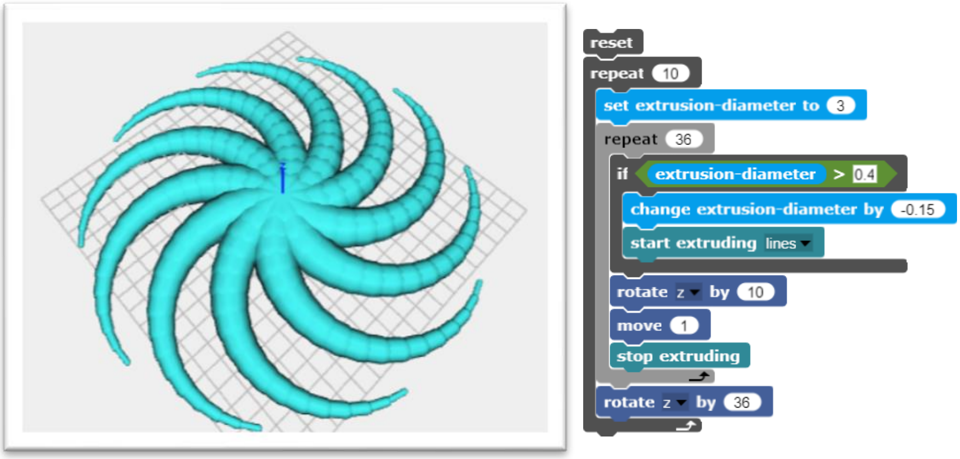


Figure 11. Screenshot and code of Turtle Art Gallery's swirl brought to 3D.

While the swirl is a specific example, a broad variety of flat objects can be created by just rotating polygons (see fig. 12), which have been constructed the traditional way (as seen in fig. 9).

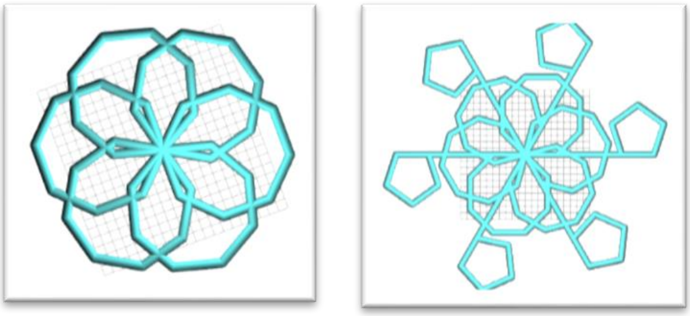


Figure 12. 2D turtle art recreated in Beetle Blocks.

In conclusion, Logo's two-dimensional turtle art can be recreated in 3D programming environments like Beetle Blocks and thus its concepts can also be taught in introductory lessons – as an approach which has been proven effective for decades.

3.2 Regular Polygons as base areas for creating actual three-dimensional Objects

What is fascinating about 3D printing, however, are not flat structures consisting of base areas, but more the creation of real three-dimensional objects. We focus on regular polygons as base area for actual 3D objects, because they represent symmetrical structures, which are considered aesthetic bearing in mind Papert's "poly pictures" (1972). In 3D turtle geometry the main challenge in creating base areas is to start in their center point to obtain symmetrical objects (as explained

[20] https://turtleart.org/gallery/index.html

in section 2.2.2). In the following we address that challenge regarding polygons constructed the traditional way and subsequently suggest a universal approach to constructing base areas.

3.2.1 Tricks for Polygons constructed traditionally

Regular polygons constructed the usual way will result in “leaning towers”: For example, a simple layering (and continuous shrinking) of squares leads to a leaning tower (see fig. 13), because the vertical axis runs through the beetle’s starting point, i.e. a corner, when creating the base area.

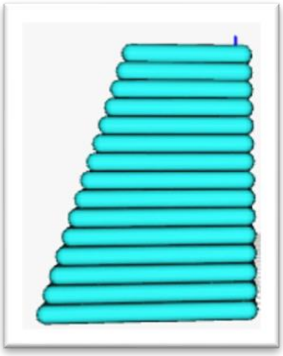


Figure 13. Leaning tower of piled squares with vertical axis through corner of the base area.

However, two intuitive tricks are available for *some* regular polygons, which have been constructed the traditional way: The first trick finds the center point of squares and hexagons, the second one turns leaning towers with a triangular or squared base area into symmetrical vases.

1st trick: Moving the beetle to the center point: For squares and hexagons it is possible to determine the base area’s center with elementary geometry – using the (half) side length and the known inner angle of 90° or 60° respectively. The resulting tower meets the expectations (see fig. 14).

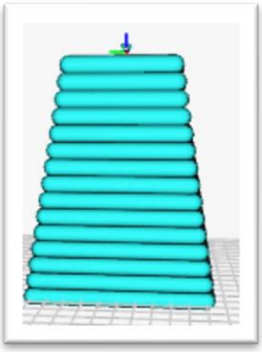


Figure 14. Tower with vertical axis through circumferential center of the quadratic footprint

We compare the source code of the leaning tower and its symmetrical counterpart achieved by this trick in figure 15: The simple layering on top of each other, which is basically realized with the command "change absolute z by 1", is lengthened with further instructions. This makes the formerly easy code much more complicated to read and modify.

Figure 15. Source code of the leaning tower (left) and its symmetrical counterpart, which has been constructed finding the center with simple geometry (right)

2nd trick: Rotation of the *whole* object*:* In order to create a symmetrical vase, leaning towers as a whole can be rotated four times by 90° for squared base areas (respectively six times by 60° for triangular base areas) so that a larger tower with inner bars is formed (see fig. 16). Even though only small changes in program code are necessary, these bars, however, can interfere in a lot of objects, e. g. when designing vases.

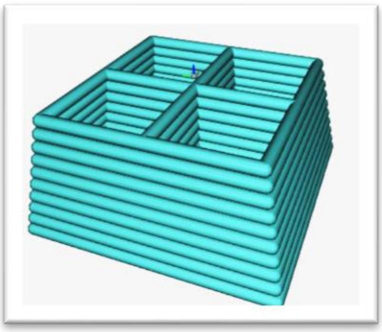


Figure 16. Symmetrical tower created by four-time-rotation of a leaning tower showing inner bars.

3.2.2 Creating Polygons using a Stack Register

For generating base areas with the beetle's start position in their center point, we suggest the algorithm described in the following: The algorithm's idea can be understood by novice programmers and supports the creation of many different 3D objects. The original program code and visualization of the geometric situation is as an example shown for a triangle in figure 17 – nevertheless the idea is easily adapted for *all* regular polygons.

Beetle Blocks offers a hidden, not (yet) visualized stack register for the position of the beetle using the instructions "push position" and "pop position" in the "Motion" category. The beetle's starting point in this alternative method is no longer a corner of the polygon to be implemented, but the (circumferential) center point of the constructed regular polygon. From this point, the beetle moves to a corner and pushes the current position onto the stack. Then it moves backwards to the center of the polygon again, rotates around the angle. The last steps are being repeated for the remaining corners. Eventually, the corner position saved first has to be placed on the stack again. Now the beetle can "beam" itself to the location of a corner by using "pop position" and then connect the corners by extrusion and calling "pop position" again. Finally, the beetle has to stop the extrusion and return to the center of the circle. The latter can be achieved easily, since it is located on the circumferential line and therefore the beetle only has to "go backwards" again.

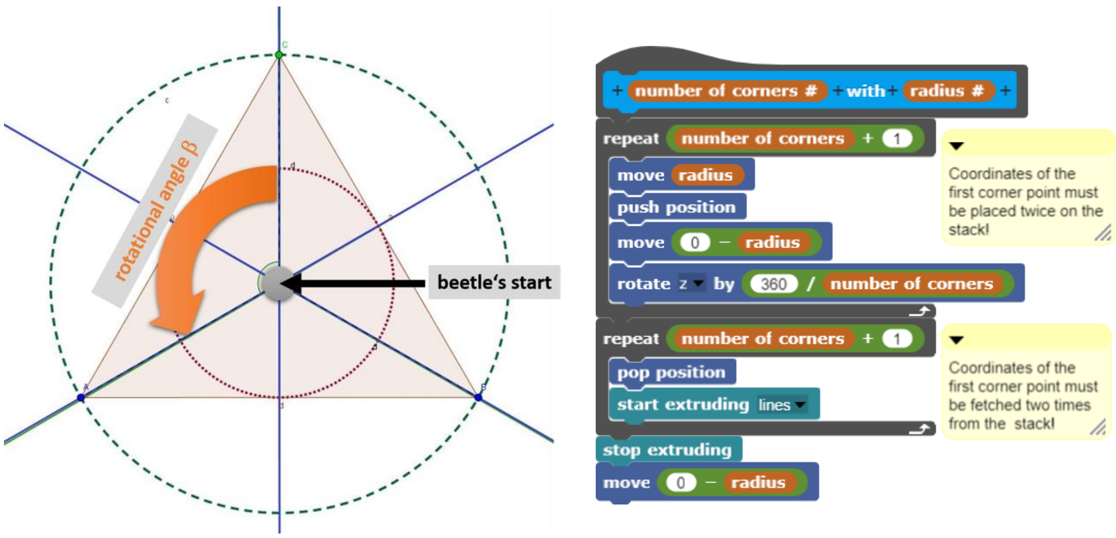


Figure 17. Code of stack register based algorithm (on the right), visualization of the geometric situation for a triangle (on the left).

This approach seems, at first, much more demanding from a student's point of view: Some more geometrical “tricks” are needed and an invisible stack register is used.

A closer look, however, levels the alleged disadvantages: The geometrical situation can still be intuitively understood following the beetle movements. Furthermore, the stack register is a fundamental data structure in computer science, therefore makes it both comprehensible for younger students and valuable for CS education (Schwill, 1997). Considering these aspects, this strategy is appropriate for novices.

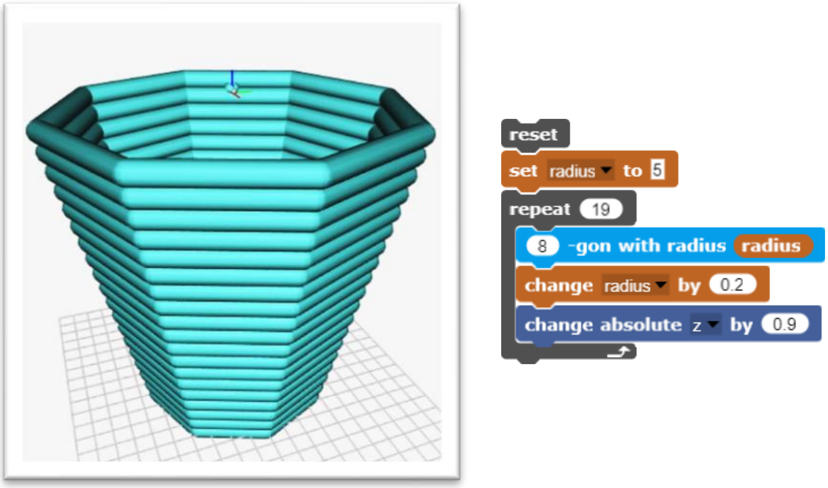


Figure 18. Vase consisting of stacked octagons constructed using the stack register.

In conclusion, in this approach the difficulty of *advanced* school mathematics has been traded in for the need of *fundamental* CS knowledge in comparison to existing teaching attempts. This can be considered as another benefit from an educational perspective as well.

4. Experiences

Based on the concepts identified, we designed a 7 hours long teaching series for twelve-year-old students, who had never programmed before. We present our experiences in the following.

4.1 Introductory Lessons with 2D Turtle Geometry for motivating students

In the two initial lessons we introduced concepts of two-dimensional turtle art to the students and printed – in the lessons itself – their first, mostly flat, objects using a 3D printer in the classroom (see fig. 19). This highly motivated the students, because they could program their first objects quickly, watch how they were built in the 3D printer and immediately take them home after school presenting them to their family members.

Figure 19. Students' results in the first both hours, printed out in a 3D printer.

While in our experience live printing in the classroom highly increases the students' motivation, it is still practicable only in the first lessons: Printing flat objects is done within minutes using standard 3D printers, but actual three-dimensional objects take a lot longer.

4.2 Long project phase allowing for creative results

In the 3rd and 4th lesson students were introduced to creating 3D objects by layering up base areas and varying their sizes using variables. Furthermore, they could explore the tricks from section 3.2.1, how formerly leaning towers can be turned into symmetrical vases. In the last three hours of the teaching series students could work on their own individual projects. Advanced students were given the opportunity to discover our stack register algorithm.

In general, the students created a broad range of different 3D artifacts: Especially decorative objects, such as vases, jewelry and small architectural prototypes were very popular. In order to get an impression, of what actual students' projects looked like, see figures 20 and 21.

Figure 20. The "Burj Khalifa" model created by a group of students in front of its archetype photo.

Figure 21. Vase constructed using the stack register in front of a mirror.

5. Conclusion

In this paper, we identified the underlying conceptual difference between 2D and 3D turtle geometry – the existence of *two* different coordinate systems – and explained its effects on how 3D objects can be created. Based on this, the big challenge of previous teaching attempts – the use of complicated advanced mathematical methods for creating base areas around their center point – is overcome with the concepts explored in section 3: We presented strategies, how some traditionally constructed base areas can be enhanced for making symmetric 3D objects possible, and even provided a general solution for all regular polygonal base areas. Furthermore, we showed that two-dimensional turtle art can be successfully recreated in 3D – completing our collection of fundamental concepts in 3D turtle geometry. Following these concepts, we designed a teaching series for programming novices *without* the mathematical challenges mentioned. Our experiences from the classroom show the potential of 3D turtle geometry for fostering the students' creativity.

Even though well-proven teaching concepts for the introduction to algorithms already exist, like creative teaching with Scratch (Romeike, 2008), established approaches regarding turtle geometry, in combination with modern 3D printing technologies offer a special connection to the real world. Never has it been so easy to transform abstract ideas into physical objects – inspiring even more programming novices and their teachers. In conclusion, 3D turtle geometry provides another encouraging setting for students.

Overall, our work can be seen as a first attempt to systematize the fundamental concepts of 3D turtle geometry for computer science classes. More advanced and promising concepts, such as recursion in fractals, are yet to be explored.

References

Buehler, E., Kane, S. K., & Hurst, A. (2014). Abc and 3D: Opportunities and Obstacles to 3D Printing in Special Education Environments. In S. Kurniawan (Ed.), *Proceedings of the 16th International*

SIGACCESS Conference on Computers & Accessability [i.E. Accessibilty], October 20 - 22, 2014, Rochester, New York, USA (pp. 107–114). ACM. https://doi.org/10.1145/2661334.2661365

Kastl, P., Krisch, O., & Romeike, R. (2017). 3D Printing as Medium for Motivation and Creativity in Computer Science Lessons. In V. Dagienė & A. Hellas (Eds.), *Informatics in Schools: Focus on Learning Programming* (pp. 27–36). Springer International Publishing.

Lutz, R. (2013). Enhancing information technology education (ITE) with the use of 3D printer technology. In W. D. Armitage, R. Friedman, & K. Baker (Eds.), *Sigite'13: Proceedings of the 2013 ACM SIGITE Annual Conference on Information Technology Education : October 10-12, 2013, Orlando, Florida, USA* (p. 157). ACM Association for Computing Machinery. https://doi.org/10.1145/2512276.2512327

Papert, S. (1972). On making a theorem for a child. In J. J. Donovan & R. Shields (Eds.), *Proceedings of the ACM annual conference on - ACM'72* (p. 345). ACM Press. https://doi.org/10.1145/800193.569942

Romagosa, B., Rosenbaum, E., & Koschitz, D. (2016). *From the Turtle to the Beetle: The Beetle Blocks programming environment*. http://goo.gl/QKpu8H

Romeike, R. (2008). Workshop: A Creative Introduction to Programming with Scratch. In M. Kendall & B. Samways (Eds.), *Learning to Live in the Knowledge Society* (pp. 341–344). Springer US.

Schwill, A. (1997). Computer science education based on fundamental ideas. *IFIP*. https://link.springer.com/content/pdf/10.1007%2F978-0-387-35081-3_36.pdf

Solomon, C. J., & Papert, S. (1976). A case study of a young child doing turtle graphics in LOGO. In Unknown (Ed.), *Proceedings of the June 7-10, 1976, national computer conference and exposition on - AFIPS '76* (p. 1049). ACM Press. https://doi.org/10.1145/1499799.1499945

How programming can serve young children as a language for expressing and exploring mathematics in their classes

E. Paul Goldenberg, *pgoldenberg@edc.org*
Education Development Center (EDC), Waltham, MA, USA

Cynthia J. Carter, *ccarter@rashi.org*
The Rashi School, Dedham, MA, USA

Abstract

Ideas can come from any kind of experience, but language is a key part of learning. We commonly rely on two languages for mathematics: the one we speak (and its written form) serves our needs for expressing the context of our problems and the nature of our thinking; and conventional mathematical notation (CMN) lets us record mathematical expressions, equations and formulas. Each is essential. But each also has limitations, especially for young learners.

With support from the National Science Foundation, EDC is researching the use of programming as a third language for children to use in expressing, exploring, and making mathematics. We study children's use/misuse of the first two languages (spoken natural language and CMN) to inform the design of a set of mathematical "maker spaces," microworlds in which students in grades 2–6 use programming in their regular lessons *as a language* for learning mathematics.

Our microworlds' goals are driven by math, not coding, but preserve conceptual coherence and developmental appropriateness in the programming and computer-science/computational-thinking (CS/CT). Topic-centered approaches in mathematics classrooms hide interconnections within mathematics, removing opportunities for surprise and delight and rendering the subject as non-creative. The microworlds we've built and those we're proposing aim to satisfy schools' topical orientation but without siloes, creatively using and foreshadowing other mathematical learning. They are designed and tested to fit easily within regular mathematics classroom study.

In this paper we show how these microworlds are inspired and informed by the way children, especially young children, use linguistic pattern in *spoken language* to build mathematical ideas. We also show how our design aims to circumvent the challenges beginners face with CMN, mathematics' *written language*, which provides the precision and concision natural language lacks and that mathematical exploration requires but which sometimes diminishes clarity for young children rather than enhancing it. We give brief descriptions of two (of four) microworlds that we designed and already used with over 200 second graders, and brief descriptions of how we are extending these to support later-grade learning. These illustrate how programming can be a valuable third language for mathematics, supporting creative exploration, key content, and essential mathematical habits of mind. CS/CT develops in tandem, integrated into the mathematical thinking. In another paper (Goldenberg & Carter, submitted), we describe wholly new microworlds under co-development with 6th graders.

The work described above uses careful observation to understand children's use of language in mathematics and to guide the iterative redesign of our microworlds. That part of our work is an exploratory study, using no formal methodology. We are *also* conducting a more formal study of the effects of the microworlds on children's learning—using cognitive interviews, video, and control classrooms—but that study is still in process and not reported here.

Keywords

mathematics; elementary school; microworlds; programming; mathematical maker space

Introduction

At EDC, we are designing *Math + C* (thinkmath.edc.org), a set of mathematical "maker spaces" in which students use programming in their regular lessons in order to express, explore, and make the mathematics they are learning. These microworlds are driven by the math, not by coding. During AY 2018-2019, with support from the National Science Foundation, we developed and piloted four microworlds (and accompanying materials) with >200 7-year-olds in their in-school mathematics classes (Goldenberg, 2019; Goldenberg et al., submitted). We are continuing that research with >200 more 7-year-olds and are designing/testing similar materials for grades 3-5, again four microworlds each. Microworlds are tested both below and above the target grade, because early studies showed that this alternative experience changed the learning trajectories.[21] Learning by making is different from learning on paper.

New experimental microworlds for 6th graders are growing out of a collaboration of students, teachers, and a curriculum designer at EDC.

Mathematical maker spaces for schools? Maker spaces[22] are environments in which people are free to create with the available materials and tools. These are not just rooms with stuff; stuff alone doesn't foster creativity. Maker spaces have domains: a "typical" maker space (is there such a thing?) might help one explore ideas for robots or for wood or plastic puzzles but not so much for poetry or mathematics. How might one *design* an elementary school mathematics maker space?

The ability to *use* a maker space—to *make* in it—taps one's prior experiences with tools and materials. We don't even *think* of using a scissor—let alone a less ubiquitous tool—if we don't know such things exist. Beyond that, we need *some* experience or help. Maker environments also have a culture. And rules (at the minimum, for safety), as all cultures do. They can be non-coercive, non-didactic precisely *because* of a culture in which *any* person present is seen as a potential source of help, information or ideas.

Our mathematical maker spaces are intended to live in school classrooms, where the cultures (due to constraints both "good" and "bad") are quite different. Designing *mathematical* maker spaces so that teachers can integrate making into classwork at grade level imposes even more constraints.

Our microworlds start with few tools and are limited to *particular* mathematical constructions, satisfying schools' typically topical orientation but without siloes, connecting mathematical learning, and preserving opportunities for surprise and creativity. We've honed the process for getting 2nd graders familiar with the tools to take only 10 minutes gathered on the rug, mostly doing the demonstrating themselves. Then they can do puzzles we pose or explore freely and invent their own ideas and puzzles. Because the tools are programming blocks, students can also create new tools.

Our programming microworlds are designed to support the mathematical content that teachers and schools deem most critical. We want children to experience material that is new to *them* just as mathematicians approach what's new to *them*: experiencing surprise, becoming curious, playing, tinkering and researching, puzzling through, and extending rather than just memorizing.

Our microworlds use programming as a language for mathematics, inspired by the way children, especially young ones, use linguistic pattern in *spoken language* to build mathematical ideas. They also aim to circumvent the challenges beginners face with conventional mathematical notation

[21] Even *physical* (motoric) milestones vary widely (including 4-month-olds standing erect independently without holding on!) depending on experience. "In contrast to the consistent and orderly progression implied by the [familiar] milestone chart," the authors report, "the skills infants acquire, the ages they first appear, and their subsequent developmental trajectories are highly responsive to cultural and historical differences in childrearing practices and infants' everyday experiences" (Rachwani, et al. in press). The order of acquisition of intellectual skills is also not immutable; experience matters.

[22] See, e.g., https://www.makerspaces.com/what-is-a-makerspace/

(CMN), mathematics' *written language*, which provides the precision and concision natural language lacks and that mathematical exploration requires. Microworlds we designed and used with children illustrate how programming as a third language for mathematics supports both creative exploration and essential content.

Spoken language and mathematics

It's fascinating how children use their adept language-brains to build or make sense of early mathematical ideas. Natural language conveys the semantic/contextual settings for mathematical problems. It's also the common way children share and refine their thinking. But, beyond classroom discourse, there's yet another way—one that gets unfortunately little attention in curricula or pedagogy—that children, especially young children, use their language-brains to process mathematical ideas: linguistic pattern.

We asked kindergarteners what they thought "five eighths plus five eighths" might be. One confidently chirped "Ten ayfs!! [pause] What's an ayf?" (Goldenberg, et al. 2017). That's correct, but linguistic, not "mathematical." (More precisely, the boundary between mathematical and linguistic thinking may be blurry at that age.)

We also asked kindergarteners to name the biggest numbers they could think of. Children offered "a hundred" or "infinity" or "a million," but not "two-hundred" or "three-million" (although one said "infinity-googolplex"). When asked what two-hundred plus three-hundred might be, most confidently answered five-hundred. We interpret that as a *linguistic* rather than mathematical strategy because the same children responded to "a hundred plus a hundred" by saying "a hundred," "a million" or "I have no idea." "Hundred" simply meant "big number." To some, a big number plus a big number is a big number. To others, big plus big is bigger. And to the sweet child who turned up her hands, smiled, and said "I have no idea," a big number plus a big number could be anything. These children weren't using the numberness of *hundred*; it was as if we asked what's three goats plus two goats. But "what's a goat plus a goat?" sounds unclear. Is it a trick? What's being asked?

Here's another way children use linguistic strategies. In a game with second graders, they call out half of whatever number we state. We start with familiar facts—the idea is to teach, not stump. If they can halve six or eight they have no trouble with sixty, eighty or eight-thousand; and they feel proud to use such big numbers. For just a few minutes, so the activity doesn't grow old, we mix these no-brainer numbers with others the children know, like halving ten, twenty, forty, four-hundred, twelve-million, repeating some for fun. Then we throw in eight-hundred-and-six (still deliberately chosen to be clear) and many can halve that. We sometimes say, as if revealing a great secret, that "some people don't realize that half of forty-eight is really half of forty and half of eight, but it is." When we then ask a class what half of forty-eight is, roughly half answer twenty-four; most others answer twenty-eight. That non-random, common error identifies the problem: executive functioning, not math. The children are juggling five numbers—forty-eight, forty, eight, twenty, and four—and that's hard for 7-year-olds doing this for a first time. Impressively, if we say only "Great! I heard twenty-four and twenty-eight. One of those is correct!" without saying *which* is correct, children improve at halving 64, 86, 42, and six-million-and-six. Why? And why does half of twenty-two remain a mystery for these same children?

Second-graders love repeating this exercise for a minute or so a couple of times a day for several days. To "compute" half of forty-eight, all they need do is *recite* the numbers (half-of-forty, half-of-eight) that they hear in their head. But half of twenty-two is not ten-one. Nothing about "eleven" *sounds* like half of twenty-two; less than ten minutes into their first try at halving wild numbers, their "mathematical" prowess remains mostly linguistic.

Halving twenty-two requires a calculation—a simple one, but not just a linguistic act. After a few days with numbers that keep the steps feeling intuitive (but with no "instruction"), children are ready for a new idea, *not* purely linguistic. They now understand twenty-two and can now halve fifty-four as twenty-five *plus* two. Halving *seventy*-four requires another step, halving seventy, so it waits for another day. Goldenberg, et al. (2015) report on high-schoolers for whom these child-strategies had been drummed out. One girl for whom halving forty was insultingly easy responded "how should I know?!" when asked to halve forty-eight; prior math experience taught her that what isn't memorized must be computed on paper. But the logic was easy to regain; she rose quickly to the top of where we want high-schoolers to be, and where too many are not.

Why didn't the second graders need to be told *which* answer was correct, as long as they knew *one* was? Young children appear to have a built-in cognitive (intuitive) analogue to the distributive property. Children aren't, of course, factoring out goats (or hundreds or "ayfs") to know that two goats plus three goats are five goats. Likewise, when asked to double , they draw  but aren't thinking "distributive property." Children don't start out knowing that half of forty-eight is half-forty and half-eight, but that logic feels so familiar that even when they lose track and fumble the numbers, confirmation that *one* of 24 and 28 is correct lets them intuit *which* one. Seven-year-olds need to build skill at holding five or so numbers mentally and tracking manipulations of them (executive function) but they soon develop that ability and get quite good at halving and doubling, starting with easy numbers.

Mathematics builds new knowledge from old. In elementary school arithmetic, we solve 26×4 not by memorizing *that* fact, but by having *some* starting places (perhaps the standard set of facts, or perhaps 25×4) and applying mathematical logic (e.g., a standard school algorithm, or an ad hoc procedure) to derive the new result. The more mental facts we easily recall, the less work is needed for deriving new results, but acquiring and maintaining mental facts takes work. So, we strike a balance, memorizing *some* easy starting places.

Memorized facts is one way. Linguistic strategies provide another. Laura, a 2nd grader learned to think of "twenty-eight" as a first and last name, like her own, and easily understood what remains if the last name (or first name) is removed. That's not a mathematical process, but it's also not a linguistic trick. Numbers aren't born with names; we name them to make such computations easy. Understanding $28 - 8$ that way, Laura readily computed $28 - 9$, following the linguistic step with a mathematical one. Similarly, after out-loud counting 10 from twenty-three to thirty-three and iterating to get forty-three, fifty-three, sixty-three, a child *hears* the linguistic pattern and anticipates seventy-three without counting. Adding *nine* can then use the linguistic pattern and adjust with an arithmetic step.

These oral/aural linguistic patterns that children are so competent (and inclined) to find powerfully support some concepts and ideas of mathematics that written techniques and manipulative methods don't make apparent or even obscure.

Written language of mathematics

Reading and writing CMN is not like reading and writing English.

First, reading *any* text is different from nearly every other kind of visual skill. The ability to recognize familiar objects in unfamiliar positions isn't born in, but infant brains quickly develop such essential geometric transformations, letting babies recognize a bottle even when the nipple doesn't face them directly. This visual processing takes learning but becomes automatic. For reading and writing, children must suppress that hard-won, now-automatic skill; otherwise p, d, b, and q are all the same. Writing ε and 3 interchangeably is normal, totally expectable at first. This is a reading-writing-only exception, following different rules from any other seeing. Attention to order also changes. We rarely care about order when listing three objects we see but understanding print requires it: *was* and *saw* are different. *Mathematical* print changes the rules again. While 315 and 513 are unequal, but 3×5 and 5×3 are equal. And worse, unlike prose text, mathematical expressions are not read strictly left to right.[23] Convention requires reading/processing both $5 + 4 \times 2$ and $5 \times (4 + 2)$ right to left, a strain to teach and to learn.

Worse yet, unlike prose text, mathematical notations are two-dimensional, requiring attention to both vertical and horizontal position. Students often encounter that first with graphs and charts.

[23] Well, to be precise, neither may be our visual processing of prose text!

Later they see it even in CMN. The typically uneven size and level of children's scrawls are just esthetic concerns; meaning isn't affected. But in mathematics, 315, 31^5 and 31_5 mean different things. Young children don't use those notations but do see $\frac{1}{2}$ early. By middle school all these distinctions matter.

A final example. Spoken (or printed, but spelled out as we do here), "five-eighths plus five-eighths" generally evokes the correct "ten-eighths" response. But written $\frac{5}{8} + \frac{5}{8}$ often evokes the canonical add-everything-in-sight wrong answer, even from bright eighth-graders. Similarly, some mathematical properties (like distributivity) seem essentially built into early cognition. Yet when that idea and its name are "introduced" in third grade (as commonly mandated in the U.S.), it is typically *taught* through a written string like $8 \times 7 = 8 \times (5 + 2) = (8 \times 5) + (8 \times 2) = 40 + 16 = 56$. Far from adding precision or a new idea, this obfuscates what children already know. Processing such a string takes focus and effort, so it isn't the optimal way to *introduce* ideas to an eight-year-old. In fact, despite your own mathematical fluency and adult cognition, probably even you zipped through it without reading closely enough to see if we typed it correctly. For a beginner, the cognitive load of decoding the notation obscures the idea.

CMN, which we now take for granted, was a relatively recent invention; mathematical progress absolutely depended on it. The features that make it so valuable to mathematics are its concision and precision. But, for learners, this comes at a price. Concision removes redundancy. In speech and prose text, some redundancy *helps* understanding. Context helps a beginning reader recognize a word or infer its meaning. And in speech and prose text, minor imprecisions often don't affect comprehension at all, which is on reason it is so difficult to proof-read a document and catch all the errors.[24] In CMN, one missing or misplaced character changes the meaning.

Primary teachers understand that invented spelling, technically "incorrect," is evidence of the emerging (correct!) sense that letters encode specific speech *sounds*, progress from the pre-schooler's letter-salads that show that the child recognizes only that letters tell stories. Conventional spelling comes later. Children learn to read and write text by applying a reservoir of world experience to the thought, content, and meaning of spoken language that the text records. We similarly accept "firemans" without instant correction; the meaning is clear, cosmetic details will come later.

In mathematics, too, thought, content and meaning must come first, conventions later. The difficulty in *acquiring* CMN is that, for young students, world experience and natural language don't sufficiently supplement CMN's concision to support that learning; the difficulty of *using* CMN makes it not ideal for building a basis of thought, content, and meaning for mathematics. When CMN does get taught, teachers must allow the same level of "algebraic babbling" (Cusi, Malara, Navarra, 2011) that is natural and necessary in developing spoken and written natural language.

Here's the problem. On the one hand, in order to express and explore mathematical ideas, even very young children need greater precision than natural language provides. Imaginably, that could be the role of CMN. But learning *any* subject—especially mathematics, which requires focused control of attention—via a language (e.g., CMN) in which one is not fluent raises the cognitive load. Children need another language.

Programming as a language for mathematics

This idea isn't new but the educational contexts in some of the seminal works (e.g., Papert, 1972, 1980) were informal ones, quite different from school settings, which are increasingly regulated in content. "Coding" is the new sexy study, with new initiatives, standards and Olympiad-like challenges in many countries[25], but many of these still imply independent efforts in school (i.e., robotics or coding classes) or informal out-of-school settings. Our own thinking was particularly influenced by efforts in Bulgaria (Sendova & Sendov 1994; Sendova 2013) and ScratchMaths

[24] Did you catch it?
[25] See, for example, https://www.codingforall.org/, https://www.csforall.org/, and http://www.bebraschallenge.org/

(Noss & Hoyles, 2018, Benton, et al., 2016, 2017) both of which aimed at functioning within the constraints of expectable school settings. The ScratchMaths project explicitly refers to programming's role *as a language* for maths. Our understanding of children's language development and use, particularly in mathematics, led us to focus on that role.

Expanding on a linguistic strategy: Building a microworld using the idea of "ayfs." One 2nd grade microworld presents a number line with only 0 labeled, not leftmost, a small number of programming blocks—for moves of -5, -3, +3 and +5 and for specifying a starting place START AT 0 (default 0, changeable by child)—and some suggestions for pure explorations or puzzles (Figure 1). Later puzzles offer new blocks, -500, -300, +300 and, +500 and a "zoomed out" number line to pose "mathematically new" puzzles (Figure 2).

Figure 1: Puzzles with smaller numbers

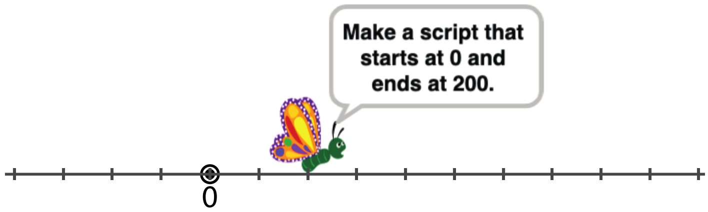


Figure 2: Puzzles that use what the child already knows linguistically but feel mathematically advanced and exciting

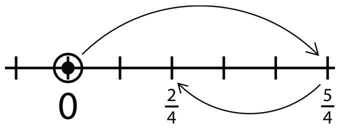
Children can create new blocks, if they like—e.g., a +1 block—using their puzzle solutions. Programs' outcomes are not "checked" as a tutorial app might do. Children see the effects they produce and can change them if they like. No puzzle aims to the left of 0 but children often arrive there by accident, and with delight. *Most* children recognized these and felt proud at finding them; *all* of them (*all*!) knew exactly what block to click to get back to "ordinary numbers."[26] Children rarely asked anything more about negative numbers—this *is*, after all, second grade!—but loved them and moved on. They don't need details or follow-up now, but their pleasure with them and experience moving to them and back "out" form a strong basis for future learning. Experience before formality.

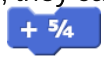
Having very few tools and strong contextual support—familiar symbols on blocks and a familiar number line image—meant that introducing this environment to 7-year-olds who had *no* prior programming experience took only 10 minutes, with them doing most of the demonstrating, before they raced to their desks to work the puzzles independently. Learning by doing in context.

With small changes and no number line, 2nd graders express and explore place-value ideas (Figure 3).

[26] On the distinction between "positive numbers" and just "ordinary numbers," see https://blogs.ams.org/matheducation/2018/09/02/ideas-under-construction-children-saying-what-they-know/

Figure 3: Combining two steps (from ±1, ±10, ±100) to make a +9 block.



Because environments like these "have legs" mathematically, they can grow with children. For example using analogous blocks 

, 

, 

, 

and puzzles on a zoom-*in* number line serves 3rd grade. That these puzzles feel so familiar as to be "trivial" is the point; fractions are just numbers; $^3/_4 + {}^3/_4$ gives $^6/_4$, not the canonically wrong $^6/_8$.

Algorithmic and functional thought and description. Mathematically comfortable adults might describe $7 \times (5+3)$ structurally as function composition, "seven times the quantity five plus three," passing the result of one operation $(+)$ to another $(\times)$. They describe the object not actions. Students likely describe actions "first add 5 and 3, then multiply by 7." The action steps *respect* the composition (add, then multiply), but express it differently. With more elaborate compositions like $\frac{500}{(x+1)^2+14} = 10$, older students' failure to see the composition $\left(\frac{500}{(something)} = 10\right.$, so the "something" must be 50) leaves them relying mechanically on expansion.

Because CMN admits to both descriptions, both ways of thinking, it doesn't help learners better articulate their current thinking and doesn't help them acquire the thinking they haven't developed.

Functional CMN brings its own mysteries. Students may see $f(x+1) = x^2 + 2x + 1$ and even $f(x) = x^2$ as if *f* is being multiplied by something. Beginners are also unsure what the variable is when they explore $f(x) = mx + b$ on graphing software (Goldenberg, 1991): after all, what students vary are *m* and *b*, not *x*.

By contrast, the *idea* of function—giving a specific response to a specific input/question—develops *very* early, but that intuitive idea doesn't seem to be readily codified into a way of thinking and describing, perhaps for lack of transparent exemplifying experiences. Certainly, the *notation* doesn't provide that easy experience.

A computer language could. Suitably designed and used, it could help students *recognize* the two ways of thinking. Because it is a *precise* language, it might even help students add precision to their own language as they articulate their thinking.

or

Figure 4. Algorithmic description (left) of the steps involved in computing $7 \times (5+3)$ *vs. functional description (right) of the nature of the computation, itself.*

In first and second grade grades, our observations seem to show that the experience of constructing the algorithmic description (figure 4, left) with blocks does help them articulate that process more clearly in words. We don't yet have evidence to show how, or whether, the experience of assembling functional constructs (figure 4, right) affects students' thinking or verbal descriptions; that's part of our current research.

Language in context and live notation. Programming is a "live" language, a notation that can be *run* to give direct, clear feedback on what it says, what effect it has on its "listener" (the computer). This is how children learn natural language and learn through discourse: we say things, others react, and we see what effect we've created. Language in context.

By contrast, a string of symbols that sits on paper (correct or incorrect) gives no feedback without the reader (re)reading and (re)processing it (or relying on outside authority to validate it). Devoid of potentially helpful context, that takes hard work, more than most students are naturally inclined to do.

And part of learning to reason logically involves focusing on the steps. Neither natural language nor CMN express process or algorithm well enough. A good programming language can provide that.

Language for thinking. Programming is also ideally suited to foster not only the mental practices typically associated with CS and sometimes labeled, in aggregate, "computational thinking" (Wing, 2006; DESE, 2016; CSTA, 2019; ISTE, 2019; K12CS, 2019) but also the mathematical habits of mind (Cuoco, et al., 2012; Mark, et al., 2012; Goldenberg, et al., 2012) now codified in the U.S. as the Mathematical Practice Standards (NGA/CCSSO, 2010), including these: (1) Programming provides a language with which students can "construct viable arguments and critique the reasoning of others,"[27] logically, providing a platform for constructing, experimenting with, testing, and thinking about ideas; (2) it eases the process of "beginning with concrete examples and abstracting regularity" or "look[ing] for and express[ing] regularity in repeated reasoning"); (3) used well, it can help students develop the disposition to "make sense of problems and persevere in solving them," and "expecting mathematics to make sense, to feel coherent and not arbitrary, and to have understandable reasons behind facts and methods"; (4) it is a perfect medium for helping students learn to "attend to precision", as computers do what we tell them to do and when that's not what we expect, we can dissect our instructions to find the ambiguity or misstep; and (5), of course, "[using] appropriate tools strategically," the tools being not just the computer and programming but also tools like a number line used *strategically*: *not* just to get answers, but to organize thought. Finally, perhaps best connected with the maker idea, programming can support mathematical creativity—to many people, an oxymoron—posing new puzzles and inventing new ideas.

Conclusion

The challenge is to design microworlds with tasks that are *core* to school at the target age, have mathematical integrity and high cognitive demand, but require little to get started and provide the experience that the harder puzzles depend on. Microworld maker spaces using a programming language that is faithful to mathematics and suitable for learners can do that. Understanding how children naturally use linguistic (and other "non-mathematical" cognitive strategies) can guide the work.

Acknowledgment

Funding for doing and reporting the work described in this paper was provided in part by the National Science Foundation, grants 1441075, 1543136 and 1741792. Views expressed here are those of the author and do not necessarily reflect the views of the Foundation.

[27] All quotations here are from NGA/CCSSO, 2010, pp. 6–8 or Goldenberg, et al., 2015, pp. 6 and 14.

References

Benton, L., Hoyles, C., Kalas, I., and Noss, R. (2016) Building mathematical knowledge with programming: Insights from the ScratchMaths project. In Sipitakiat, A., and Tutiyaphuengprasert, N. Constructionism in *action 2016. Conference proceedings*. Retrieved from https://drive.google.com/file/d/0BwnqbVelN16LbXY3NHJZaldQY3c/view

Benton, L., Hoyles, C., Kalas, I., and Noss, R. (2017) Bridging primary programming and mathematics: Some findings of design research in England. *Digital Experiences in Mathematics Education*, 1–24. https://doi.org/10.1007/s40751-017-0028-x

CSTA. 2019. CS Standards. https://www.csteachers.org/

Cuoco, A., Goldenberg, E.P., and Mark, J. (2012) Organizing a curriculum around mathematical habits of mind. In Hirsch, C., Reys, B. and Lappan, G., eds, *Curriculum Issues in an Era of Common Core State Standards for Mathematics*, Reston: NCTM. (First appeared in *Mathematics Teacher* 103(9) pp. 682-688, May 2010. Reston, VA: NCTM. 2010.)

Cusi, A., Malara, N., and Navarra, G. (2011) Theoretical Issues and Educational Strategies for Encouraging Teachers to Promote a Linguistic and Metacognitive Approach to Early Algebra. In Cai, J. and Knuth, E., eds. *Early algebraization: A global dialogue*. Pgs. 483–510.

DESE. (2016) Massachusetts Department of Elementary and Secondary Education. *2016 Massachusetts Digital Literacy and Computer Science (DLCS) Curriculum Framework*. Accessed at http://www.doe.mass.edu/frameworks/dlcs.pdf, 16 September 2016.

Goldenberg, E.P. (2019) Problem posing and creativity in elementary-school mathematics. *Constructivist Foundations* 14(3), 601–613. http://constructivist.info/14/3/501.goldenberg

Goldenberg, E.P. (1991) The difference between graphing software and *educational* graphing software. In Demana, F., and B. Waits, (eds.), *Proceedings of the Second Annual Conference on Technology in Collegiate Mathematics*. Addison-Wesley, 1991; co-published in Zimmerman, W., and S. Cunningham, (eds.) *Visualization in Mathematics*, Math. Assoc. of America.

Goldenberg, E.P. and Carter, C. (submitted, expected date 2020) Co-designing mathematical microworlds with 6th graders: mathematical maker spaces for grades 2 through 6.

Goldenberg, E.P, Carter, C., Mark, J., Reed, K., and Spencer, D. (submitted, expected 2020) Programming as language and manipulative for second grade mathematics.

Goldenberg, E.P., Mark, J., and Cuoco, A. (2012) An algebraic-habits-of-mind perspective on elementary school. In Hirsch, C., Reys, B. and Lappan, G., eds, *Curriculum Issues in an Era of Common Core State Standards for Mathematics*, Reston: NCTM. (First appeared in *Teaching Children Mathematics* 16(9) pp. 548-556, May 2010. Reston, VA: NCTM. 2010.)

Goldenberg, E.P., Mark, J., Kang, J., Fries, M., Carter, C. and Cordner, T. (2015) *Making Sense of Algebra: Developing Students' Mathematical Habits of Mind.* Heinemann: Portsmouth, NH, USA.

Goldenberg, E.P., Miller, S., Carter, C., and Reed, K. (2017) Mathematical Structure and Error in Kindergarten. *Young Children*, 72(3):38-44. Washington, DC: NAEYC.

ISTE. (2019) Computational Thinking Competencies, https://www.iste.org/standards/computational-thinking

K12CS (K12 Computer Science). (2019) K12 Computer Science Framework. https://k12cs.org/

Mark, J., Cuoco, A., Goldenberg, E.P., and Sword, S. (2012) Developing mathematical habits of mind. In Hirsch, C., Reys, B. and Lappan, G., eds, *Curriculum Issues in an Era of Common Core State Standards for Mathematics*, Reston: NCTM. (First appeared in *Mathematics Teaching in the Middle School* 15(9) pp. 505-509, May 2010. Reston, VA: NCTM. 2010.)

NGA/CCSSO (2010) Common core state standards for mathematics. National Governors Association Center for Best Practices, Council of Chief State School Officers. Washington, DC.

Noss, R. and Hoyles, C. (2018) The ScratchMaths project is directed by Richard Noss and Celia Hoyles, with Ivan Kalaš, Laura Benton, Alison Clark Wilson, and Piers Saunders. See http://www.ucl.ac.uk/ioe/research/projects/scratchmaths. Accessed March 29, 2018.

Papert, S. (1972) Teaching Children to be Mathematicians Versus Teaching About Mathematics, *Int. J. Math Ed in Science and Tech.*, Vol. 3, No. 3, pp. 249–262.

Papert, S. A. (1980) *Mindstorms: Children, computers, and powerful ideas*. New York City, NY: Basic Books.

Rachwani, J., Hoch, J. E., and Adolph, K. E. (in press). Action in development: Variability, flexibility, and plasticity. In C. S. Tamis-LeMonda and J. J. Lockman (Eds.). *Handbook of infant development.* Cambridge University Press. Preprint formatted by authors. Retrieved from http://www.psych.nyu.edu/adolph/publications/RachwaniHochAdolph-inpress-ActionInDevelopment.pdf, October 28, 2019.

Sendova, E. (2013) Assisting the art of discovery at school age: The Bulgarian experience. In *Talent Development Around the World* (pp. 39–98). Mérida, Yucatán.

Sendova, E., and Sendov, B. (1994) Using computers in school to provide linguistic approaches to mathematics: A Bulgarian example. *Machine-Mediated Learning*, *4*(1), 27–65.

Wing, J. (2006) Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Ideas, Technology and Skills: A taxonomy for digital projects

Oliver Quinlan, *oliver@raspberrypi.org*
Raspberry Pi Foundation, Cambridge, United Kingdom

Sue Sentance, *sue@raspberrypi.org*
Raspberry Pi Foundation, Cambridge, United Kingdom

Abstract

This paper seeks to develop the understanding of how young people engage with digital making projects. It proposes a simple taxonomy for thinking about the factors that are required or must be developed in order for young people to successfully complete a digital making project.

Open-ended projects addressing themes that young people care about are a compelling way of engaging with digital technologies and learning new skills. These projects can support young people's learning through a constructionist approach, where it is important that the supporting adults can facilitate exploratory and self-motivated learning without constraining the direction this takes. This can be challenging for adults as such projects inherently do not follow a set path, and areas in which support is needed are not always obvious, particularly to non-experts.

We visited two large showcasing events for digital making projects and interviewed young people who attended the events to share the projects they had made with others. The interviews were focused on young people's narratives of the process of working on the projects and what they had to learn in order to successfully complete them. Analysis of the interviews identified a number of key themes in these narratives. Although these themes were common, it was notable that different groups of young people had different starting points for their projects, with some being engaged initially by the technologies they wanted to use, others by the ideas or problems they wanted to explore, and some by the skills or capacities they wanted to learn more about.

For children working in an open-ended way on digital making projects there is a balance to be struck between three key factors. Projects are shaped by the ideas the children have, the technology they have available, and the skills that they have or have the capacity to learn in the course of the project. In a series of interviews at events for children to showcase their projects we found projects had a different emphasis on these factors depending on the starting point of the project and the context the children were working in.

The simple taxonomy developed in this work is designed to help adults supporting young people to create digital projects. It can be used to identify their starting points and the key factors they may need support with in order to develop their capacity to realise a finished project. We found that children's stories of the process often focused on one particular aspect of this taxonomy as a starting point, but that to develop a complete project they needed to consider all three. Support from adults may be best focused on identifying the area that is already a strong starting point, and facilitating children to further develop the other areas of the project.

We hope that this taxonomy will allow adults to more clearly identify the needs of young people, and support them to navigate open-ended digital projects successfully while learning new skills and capabilities. .

Keywords

projects, digital making, creativity, ideas

Introduction

Non-formal digital making and programming clubs such as those in the CoderDojo movement create opportunities for young people to work on open-ended digital projects, creating things they care about and learning skills as they do so. The open-ended nature of this process is key to the high levels of engagement, and the opportunity for constructionist learning as young people create their own projects and in turn construct new understandings of the tools and the contexts they are working in.

Existing research around the design and creation of projects stems back decades. Early research in project-based learning showed that projects were very motivating for young people (Blumenfeld et al,1991), embodying constructivist principles, and building bridges between principles and real-life experiences (Krajcik et al, 1994). More recently there has been a surge of interest in digital making projects, to a certain extent associated with the increase in programming education at K-12. The emergence of new, affordable, low entry physical computing devices has made digital making projects very accessible, and these have been shown to both engender creativity and provide challenge (Sentance and Schwiderski-Grosche, 2012, Sentance et al, 2017). Block-based programming languages such as Scratch (Resnick et al, 2009), with a huge community, and millions of uploaded projects, enable young people to generate ideas and develop them into an actual artefact, game or animation. In addition, the new landscape of clubs, hackathons, jams and other non-formal learning environments provide young people with places to go to participate and learn with others.

This kind of learning also takes place in what some researchers identify as non-formal learning environments, defined as taking place "in a planned but highly adaptable manner in institutions,organizations and situations beyond the spheres of formal or informal education" (Eshach, 2007, p.173). Non-formal learning is intentional although it takes place outside of the classroom, and adults are often not professional educators with access to formal teaching experience or training.

Facilitating and supporting open-ended projects is challenging, as is being able to understand the kinds of support young people need, the contexts that help them and the challenges they may face. To provide support for young people to effectively realise their ambitions, supporting adults have to be able to guide them to reach technical solutions they themselves may not have undertaken before, in novel contexts and with limited access to technologies. As these adults are often not formally trained educators, there is a need for support in terms of understanding how they can best facilitate learning while maintaining the interest led approach that non-formal learning contexts can provide.

The Study

In this project we set out to better understand the process that young people go through to create successful projects. We hoped that a clearer understanding of the processes involved with these projects could help adults to identify potential barriers to success in these kinds of projects, and support young people to navigate the process of creating a successfully finished project. Our research question was itself open-ended and exploratory. We aimed to collect narratives from young people showcasing their projects at a public event and explore the stories they told about their experience of making. Our research question is "*What starting point and initiators do young people use to stimulate their digital making projects?"*

Context

Coolest Projects is a series of showcasing events linked to the CoderDojo and Code Club networks of computing clubs. National events take place annually in the UK and the USA, with an international event taking place in Dublin every year. This research took place at the inaugural UK event in London in 2018 which was attended by around 40 teams or individuals and 65 children, and the international event in Dublin in 2018 which was attended by around 700 teams or individuals and 1000 children from across the world.

At Coolest Projects events, young makers bring their projects to share with others in a 'science fair'–style exhibition. The emphasis is on sharing and learning from others, and gaining inspiration from the wide range of projects shown. Coolest Projects started as part of the CoderDojo movement, which has since become part of the Raspberry Pi Foundation. It is open to any young person across the world with a digital project.

The ethos of Coolest Projects is that young people create open-ended projects about problems or themes they care about. The problems or ideas the projects address are chosen by the young people, and they submit their projects in one of a series of categories that are largely related to the type of technology they use. The categories of projects at the events used in this research were:

- Games and Web Games
- Hardware
- Mobile Apps
- Websites
- Scratch
- Evolution (projects that had been shown at previous events but significantly developed since then, usually presented by older and more experienced participants).

During the event a panel of judges for each category circulate and discuss every project with the team who created it. Towards the end of the event there is an awards ceremony where prizes are given to the top projects in each category, as well as some other awards for achievements such as the 'best Dojo' represented at the event. Despite this competitive element, the event is designed to have a collegiate ethos, and participants are encouraged to look at and discuss other's projects in order to learn from and inspire each other.

Data collection and analysis

At the Coolest Project events in London (UK) and Dublin (International), we interviewed teams and individual presenters to find out about their projects, what they learnt, and their participation in the event. We interviewed 9 teams or individuals at each event.

Interviews were semi-structured and focused on the following themes:

- Context: Where they worked on their project such as a club or at home.
- Narrative: The story of their project, how they began, developed, and completed the project.
- Learning: What challenges they faced, how they overcame them and what they needed to learn to complete the project.

A shortlist of participating teams and individuals were selected before each event using the data collected as part of the application process. Participants were selected to give a representative sample of each of the available categories, and where possible including a range of geographical areas, genders of participants, and group and individual projects. On the day of the event researchers conducted interviews with nine of those who were present at the event. Due to the free flowing nature of the event some participants on our shortlist could not be located, and some did not attend, so our sampling was to some extent opportunistic.

Interviews with participants lasted for 20–30 minutes, using a semi-structured interview schedule, using a template for responses during the interview, with more detailed notes taken immediately afterwards.

Children and young people taking part in this event had already completed a consent form to take part in the public event which required permission from their parents. We explained to participants the nature of our research, and provided them with an information sheet with this information and how to opt out in case they or their parents were unhappy with taking part or changed their mind. All participants had a parent or guardian present, who gave their permission to speak to their children.

Interviews were analysed using a thematic analysis, creating a series of codes (Kuckartz, 2014). The notes were analysed inductively, identifying themes that emerged from the stories of the creation of projects the children described. A number of key themes were identified from this inductive coding which recurred in many of the children's narratives of their project creation.

The analysis of interviews was viewed through a socio-cultural lens (Lave & Wenger, 1991). When identifying codes the existing knowledge and motivations that children were bringing to their activity was considered. The children's activities towards developing their project was mediated by the social environment of non formal learning clubs, spaces that can be construed as communities of practice. In identifying themes we considered how children interacted with these environments, and how this influenced their acquisition of skills or competencies.

Theme	Description
Long, evolving engagement	Engagement often long and consisting of many episodes not easily quantify how long they had spent on their project.
Goals	Finishing a project to showcase at the events as a goal in itself.
Learning	Learning new skills and technologies. A wide range of skills identified depending on the context of the project.
Confidence	Confidence developed through being able to talk about projects at the events.
Team roles	Individuals took on different roles depending on their skills, experience and their interests.
Legitimate peripheral participation	Some children joined teams in a less involved role than others, but had clearly benefited from being part of the team. The effect on them is an area for further research.
Influence of adults	The skills adults were able to teach, or ways they were able to facilitate learning.
Interdisciplinarity	Projects brought together different areas of learning, particularly the interplay between the ideas or problems being addressed and the technology skills used to address them.
Technologies	The technologies available determining the direction of projects.
Primary school programming	Learning programming early in their education as an influence.

Challenges getting help	The challenges of finding age and skill level appropriate help and learning resources. The highly contextual nature of projects can make generic resources hard to locate and apply to challenges they are experiencing. Adult facilitation addressed this for some participants.
Power of seeing others' projects	Inspiration drawn from seeing other children like them sharing projects at the events.

Case studies were created for each project setting out the narratives of the projects, the key themes in terms of types of project, make-up of the teams, and the problems they were addressing. These have subsequently been shared with the community surrounding the events to help with understanding the engagement with the events and the impact of the activity they generate (Quinlan & Flóriánová, 2018).

The taxonomy

The themes identified from coding and the case studies were analysed to identify the key factors of: technology, idea/problem and skills that were used to create our model for understanding the narratives around the creation of the projects. Figure 1 shows the taxonomy developed to analyse the case studies.

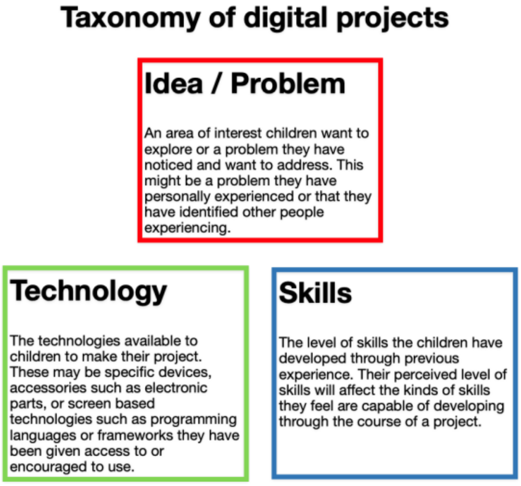


Figure 1. A taxonomy for supporting young people creating digital projects

The stories that children told us showed that there are different directions taken when choosing and developing a project. Some children start with an idea or a problem to be solved, and their project is driven by trying to address this. Others start with the technology they have at their disposal or that they want to explore. Still others start by looking at and reflecting on their own skills, and explore how to use these skills in interesting ways. Children have to balance the relationships between their ideas, the technology available, and their skills, in order to create a successful project.

If they are less focused on a particular area then this could make the development of a finished project difficult. The most obvious example, and something is often considered around digital learning, is the lack of access to appropriate technologies. Children may have ambitious ideas and the capability to develop skills, but there are some projects which will be very difficult or even impossible without access to certain technologies.

A lack of balance in the other areas can present challenges too. Access to an exciting piece of technology can be very motivating, but without a clear idea or problem to solve children may be less likely to formulate their activity into a cohesive project, with the learning benefits that a project based approach can bring.

A clear idea or problem and the appropriate technology can be held back by a lack of skills. Skills can be developed, and this may be a desirable area to be lacking in to begin with as it can encourage the learning of new skills.

Results

Technology as an entry point

Of the participants we interviewed, Linear Equation, Security System, and Zombie Defence System were among those who were initially driven by a particular available technology, and explored possibilities to define an idea for a project. These tended to follow existing examples of what a project could look like, although with their own modifications.

The young person who created Linear Equation had the **technology** of line following robots as their starting point, as shown in Figure 2. They were clearly intrigued by this technology and had access to the component parts needed to realise it. They developed their skills in order to implement this technology and create their own robot. The problem or idea that the project addressed had clearly come later, although their suggested use of the robot to assist people in carrying heavy objects showed an intention that the project had a purpose beyond their own process of creation and learning. This project started in the area of technology, and as it progressed it moved into the area of skills development, with a consideration of the problem it could be used to address coming later. All three areas were addressed to different degrees to constitute a project that the young person was proud to present at the event.

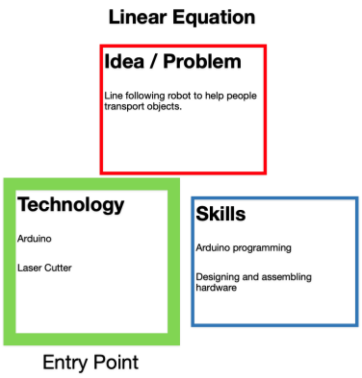


Figure 2: Linear Equation – showing Technology as an entry point.

Ideas as an entry point

Dragon King, Healthy and IOT Project to Track Pets were driven by strong **ideas** first. In some cases these were very ambitious, and the participants had designed and researched their idea but not yet fully implemented it.

The creator of 'Healthy' was drawn to creating a project around the theme of healthy eating. The children had identified that this was a social issue to be addressed, and that the solution could be to provide personalised information on how to eat healthily. They had explored the problem they wanted to address comprehensively, however their technical skills were not yet sophisticated enough to fully realise a working prototype of this idea. Therefore, they found an app design and prototyping platform that allowed them to demonstrate this ambitious idea without having to develop the considerable programming skills to create a working app. Programming a full app would have been a considerable undertaking in terms of skills development, and their zone of proximal development (Vygotsky, 1978) at that time was more suited to learning the design tools to create a demonstration of their ideas. Figure 3 shows the importance of ideas in the Healthy project.

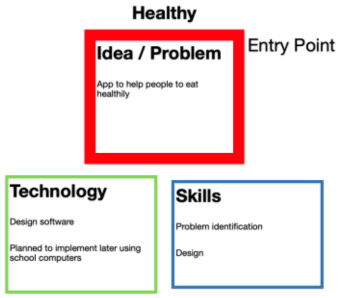


Figure 3: The Healthy project with the idea as an entry point

Skills as an entry point

Fewer projects were driven by the primary consideration of **skills**. This was perhaps the case with some of the simpler Scratch projects, but also one of the most sophisticated projects, Intelligentia, depicted in Figure 4. This maker had spent some time developing skills in facial recognition and machine learning, and decided to deploy them to solve the problem of locating missing persons. Their project used a mobile app to implement the skills they had learned in facial recognition to scan photos on a users phone and match faces against a photo of a missing personal that could be officially released by the police. They also had other applications of these skills in mind for the future, showing a 'skills first' approach to projects.

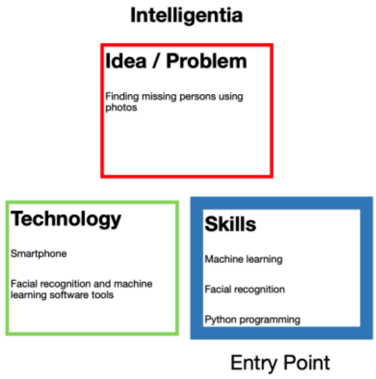


Figure 4: Intelligentia project shows skills as an entry point

Discussion

There is often a complex relationship between the idea or problem, the technology, and the children's skills, which children and their mentors must negotiate. Children who start entirely focused on a problem that interests them have the challenge of understanding what skills they need to address it and acquiring the appropriate technologies to do so. Indeed, most children interviewed said that they improved their skills to realise their project. To do this successfully they need to take account of the level of challenge and realistic skills development within the timeframe of the project. Where children start with a focus on doing something interesting with a given technology, they have to consider what problems this might be applicable to.

Negotiating between the idea or problem, the technology to use, and the skills required is key to a successful project. Different participants have different entry points for their projects, usually focusing mainly on one of these aspects to begin with. To create a finished project they have to find a balance of all three, although this balance is not always equal and often has a stronger focus on the area they started with than the others.

A balance has to be found between a compelling idea, the technology that is to hand or can be acquired, and the skills that exist or can be developed. As mentors and supporting adults are often responsible for facilitating these kinds of processes in a non-formal learning context, support for them to negotiate this territory and guide children through it would be of value.

Conclusion and Implications

Through interviews with children presenting their project at these events we identified a number of key starting points and initiators young people use to stimulate their digital projects. These were combined into a taxonomy highlighting the areas of ideas or problems, technology and skills.

For practice in the area of non-formal clubs such as CoderDojo, this model of the key factors needed for a successful project can be used to help adults to tailor their support for young people to create such a project. This taxonomy can enable adults to identify where young people are starting from, the challenges they are likely to face and the decisions they will need to make in order to develop their starting point and area of interest into something that is complete.

Every year hundreds of young people feel confident enough in their projects to submit them to the Coolest Projects events, but given the numbers involved in the club networks there must be many more who begin working on ideas but find it difficult to address all of the aspects that are needed to develop these into a complete project. With guidance from adults with a clear understanding of the factors that make a completed project, many more may be able to reach the stage where they feel happy to exhibit their projects.

The proposed taxonomy provides a map of the narratives of what makes a successful project, but does not specify how this map should be followed or what order young people should consider the development of their project. It is carefully designed as a tool which adults and young people themselves can use as a guide to open-ended activity. More research is needed to expand and elaborate on the taxonomy, which currently only provides an outline structure for entry points.

We anticipate adults could use this taxonomy to inform their support of young people by discussing projects with them at an early stage and identifying the entry point that the young people are taking and how much they have engaged with the other areas of the taxonomy. As access to technology is the most immediate and tangible aspect, it is likely that adults already often consider this area and guide children towards making use of particular technologies that are available. However, this is just one aspect of a successful project. Children may need guidance to frame their interests into a cohesive idea or clearly identify a problem they want to address. They may need support to reflect on the skills they have already and those that would need to be developed in order to successfully make their ideas.

As a high level overview, this taxonomy is designed to encourage adults and children to reflect on the starting points for projects and the likely areas where support would be needed. The intention is that it can provide the space for children to develop their ideas in an open-ended way, but encourage support to be put into place to enable them to develop all aspects of a successful project. In order to best support volunteers and educators without experience in technical skills or in constructionist approaches to learning, it could be supplemented with more detailed resources and activities to address the areas identified. There is a comprehensive literature on design thinking approaches for educators (IDEO, 2013), which could be drawn on to facilitate the development of project ideas and addressable problems. In formal education there are many traditions of reflective approaches to learning and formative assessment that can be drawn on to identify current skills and the next steps for skills development (Wiliam and Black, 2006). There is also potential in developing it into materials that young people can use themselves to support their thinking. Care must be taken to develop resources that are appropriate to the non formal learning environments in which the subjects of this research were working, as over-formalising the ways of working could detract from many of the positive aspects of these environments. With the right balance, and feedback from adults and children participating in these environments, developing such resources is a promising area for future research.

The taxonomy of digital products presented here provides a structure to make visible the aspects needed for a successful project. We anticipate it can provide a framework for thinking about these projects andhelps adult facilitators and child learners as they plan and navigate the exploration of the ideas and problems that they care about when making digital projects.

References

Black, P. and Wiliam, D. (2006). *Inside the Black Box: v.1: Raising Standards Through Classroom Assessment*. GL assessment Limited.

Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational psychologist*, *26*(3-4), 369-398.

Eshach, H.. (2007). Bridging in-school and out-of-school learning: Formal, non formal, and informal education. In *Journal of science education and technology 16*, 2, 171–190.

IDEO (2013), *Design Thinking for Educators Toolkit.* Available at: https://designthinkingforeducators.com/toolkit/.

Kafai, Y. B., & Burke, Q. (2013). The social turn in K-12 programming: moving from computational thinking to computational participation. In *Proceeding of the 44th ACM technical symposium on computer science education* (pp. 603-608). ACM.

Krajcik, J. S. (1994). A Collaborative Model for Helping Middle Grade Science Teachers Learn Project-Based Instruction. *Elementary School Journal*, *94*(5), 483-97.

Kuckartz, U. (2014). *Qualitative text analysis: A guide to methods, practice and using software*. Sage.

Lave, J. and Wenger, E. (1991) *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press

Quinlan, O. and Flóriánová, L. (2018) *How Children Make Digital Projects: Research from Coolest Projects 2018*. Available at: https://www.raspberrypi.org/app/uploads/2018/12/How-children-make-digital-projects-2018.pdf

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. B. (2009). Scratch: Programming for all. *Commun. Acm*, *52*(11), 60-67.

Sentance, S., & Schwiderski-Grosche, S. (2012). Challenge and creativity: using. NET gadgeteer in schools. In *Proceedings of the 7th Workshop in Primary and Secondary Computing Education* (pp. 90-100). ACM.

Sentance, S., Waite, J., Hodges, S., MacLeod, E., & Yeomans, L. (2017). Creating Cool Stuff: Pupils' Experience of the BBC micro: bit. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 531-536). ACM.

Vygotsky, L. (1978), *Mind in society: The development of higher psychological processes.* Harvard University Press..

Insights on Online Mobile Maker Portfolios for Process Documentation in K-5 Constructionist Learning Environments

Monica M. Chan, *monica.chan@tc.columbia.edu*
Teachers College, Columbia University, New York, USA

Nathan Holbert, *holbert@tc.columbia.edu*
Teachers College, Columbia University, New York, USA

Abstract

As there is no standardized method or technique of assessing student learning in non-traditional environments such as makerspaces, the advent of using portfolios as an alternative form of assessment and progress-tracking has emerged. This paper presents a qualitative research study that explores the interaction dynamics of elementary school-aged students in a North American urban private school using online portfolios on a mobile application as a medium to share and document their maker projects. The students generate posts on their portfolio tool (the mobile application Seesaw) regarding maker activities and projects they work on in school. All members of the school community (teachers, parents and other students) are able to view and comment on the posts that the students create. Main methodologies used in this study include semi-structured cognitive clinical interviews with the students and analysis of students' artifacts (Seesaw posts students generated). The findings of this study have been categorized into three main themes - "Big deal" projects, portfolio as a teaching tool, and demonstrating iteration. We elucidate each theme by illustrating an anecdotal example from observation, paired with findings from the interviews and artifact analysis. These examples illustrate the affordances of the portfolio's mobile application features in aiding students' communication and documentation about their interdisciplinary projects. However, these examples also demonstrate the tension between showcasing works-in-progress or failed attempts and glamorous final products. This study on mobile portfolios contributes to the advancement and experimentation of various methods of assessment in Constructionist learning environments. Overall, this study gives us an insight to how K-5 students perceive and document their maker-oriented projects and activities, in the context of constructing an online portfolio. The conclusion calls for future work that addresses how formative assessment tools could work symbiotically with the development of an ideal maker culture of sharing and documenting.

Keywords

Maker Education; Project-Based Learning; Documentation; Portfolios; Elementary Education; STEAM; STEM

Problem Statement

Making and makerspaces have been on the rise in the past decade, due to the widespread popularity of the Maker Movement and more affordable and accessible fabrication technologies that can be used to create personally meaningful products. Making constitutes experiential play and creative tinkering, which are key elements of a child's path to lifelong learning. The ideas of digital fabrication, creativity and innovation in learning that drive the development of makerspaces in schools has become more widespread in the United States (Gershenfeld, 2008) and internationally (Eriksson, Heath, Ljungstrand & Parnes, 2018). The recent expansion of makerspaces has been described as a phenomenon that builds on an individual's identity to create products and "be a maker" (Papavlasopoulou, Giannakos & Jaccheri, 2017). Schools too have begun integrating makerspaces into their curricula and environment to provide students with direct access to these emerging technologies and hands-on, project-based learning.

However, currently, there is little agreement on the methods or best practices for assessing and evaluating student learning in non-traditional environments such as makerspaces. As schools and informal learning spaces (such as libraries and museums) are promoting maker-oriented learning environments, there is a need to evaluate students' progress amongst educators and administrators (Sommerville, 2013; Merritt & Gangopadhyay, 2014). Various stakeholders have been creating personalized methods of assessment and evaluation, from rubrics to portfolio systems to quantitative analytical data. Due to the complexity of the makerspace learning environment in many settings, there is no consensus towards what works best across all sites at the moment, but there is room to generate insights into different scenarios and contexts.

For this exploratory research study, we investigate how elementary school-aged students in a maker-oriented project-based learning environment (their school's makerspace) use a social online portfolio tool to document, track and showcase their learning progress. The portfolio tool that the students use is Seesaw, a mobile (tablet) and web-based application for students to post their work where it is accessible to teachers, peers and parents. Students' interaction with and use of this portfolio tool could elucidate certain behavior and mindsets they have towards sharing and reviewing project-based work, and may potentially point us towards future directions to study portfolios as formative assessment for makerspaces.

This leads us to the following research questions: *How do K-5 students approach and engage in process documentation using an online maker portfolio? What affordances and limitations exist in the various modalities of a mobile portfolio application?*

Theoretical Framework

Maker-Oriented Learning & Constructionist Pedagogy

Notable scholars such as John Dewey, Jean Piaget, John Friedrich Froebel, and Maria Montessori have underscored making as key to the process of learning. Seymour Papert, who coined the theory of Constructionism, added that learners' journey of physically constructing an object would be an effective method for them to develop and demonstrate understanding (1980; Bevan, Petrich & Wilkinson, 2014). Constructionist, maker-oriented pedagogy associates itself with the theory that learning is embodied in tool and social interaction, whereby knowledge is constructed and reconstructed through interactions (Papert, 1980). Thus, we need process-oriented documentation tools to formatively assess maker-oriented learning, by observing, responding to and iteratively assessing non-linear making processes.

Use of Portfolios as Process-Oriented Assessment

As the use of portfolios expanded over the past years and permeated into makerspaces, scholars have now begun to question how portfolios are a viable means to evaluate student learning, in conjunction with how students develop maker identities and practices. Keune and Peppler discussed their observations on how two youth-serving makerspaces used portfolio assessment as an avenue to compile students' individual work and shared projects over a year (2017). Keune and Peppler discovered that compared to portfolios that focused on individual work alone, portfolios that included shared projects and documentation presented richer showcases, showed technical and social engagement with the community, and were assessed by a more diverse audience (2017).

An ongoing example is MIT and Maker Ed's Beyond Rubrics project which focuses on embedded assessment in school-based making activities (Murai et al., 2019). Murai and colleagues define embedded assessment as assessment interwoven in the students' learning environment and activities, so that students' flow of learning would be uninterrupted (Wilson & Sloane, 2000). They designed an assessment toolkit for educators following three main steps they had defined in the embedded assessment process - context setting, evidence collection, and meaning making (Murai et al., 2019). The findings from this study showed that most educators found the evidence-centered approach, coupled with tangible evidence (in the form of photographs, illustrations, videos, writing etc.) collected throughout the process, overall helpful. This project is currently in progress since it involves an iterative co-design process with educators, but this has built consensus regarding the usefulness of artifacts as evidence in formative assessment in maker-oriented learning.

Another ongoing research study on the same strand is MakEval at Indiana University. MakEval combines tool sets for formative assessment in makerspaces, namely in the key target areas of agency, STEM practices, creativity, STEM interest and identity, and critical thinking (Maltese, 2019). The suite of instruments for each of the five areas include portfolios, surveys, interviews and post-project reflections and a "tweet" wall.

The aforementioned studies have demonstrated the significance of evidence-centered formative assessment and shown that documentation via reflective portfolios can help capture collaborative constructions and allow for detailed documentation using a variety of media. The study that this paper illustrates provides a smaller-scale investigation into how K-5 students use online portfolios in their maker learning journeys.

Methodology

Data Collection

Research Site & Research Participants

The research site is a small private school in a dense urban area in a large city in the United States. The school currently serves the K-5 range, follows a project-based learning mantra, and has loose grade bands to accommodate mixed age groups. The teacher to student ratio is approximately 1 : 5 to 7, depending on the subject or class activity.

There are 15 students as research participants. The students are aged five to eleven years old; 11 are male and 4 are female. Students are separated into three learning groups based on their personal development levels - students in Group 1 are generally pre-reading, while students in Groups 2 and 3 have higher reading, mathematics, and technical abilities. Four students are in Group 3, nine students are in Group 2, and two students are in Group 1. As part of the school's project-based curriculum, students participate in interdisciplinary units that their teachers plan in advance. These units may span between one to four months long, and there are smaller shorter-term activities interspersed during these units. These groups are mixed-age and mixed-gender, although students in Group 1 tend to be younger than those in Groups 2 and 3. The students are generally from wealthy backgrounds, and reside in the city where the school is. The students' parents are highly involved and updated about their children's learning progress, through parent-teacher communication via multiple online platforms such as email and Seesaw.

Interviews

This is a primarily qualitative study. We conducted one-on-one semi-structured interviews with each of the 15 students to discover how they decide to and then generate posts on Seesaw. Interview questions aimed to elucidate the students' workflow when using Seesaw. At the start of the interview, we introduced a maker activity to the students to get them adjusted to our interview about making and documenting using Seesaw. Students were invited to describe how using a Gogo Board microcontroller might work and its associated sensors by following prompts on a Gogo Board worksheet (Thanapornsangsuth, 2016). An iPad with the Seesaw application was made available for the students to use during the activity. Students were instructed to complete an Inventor's Plan worksheet (Thanapornsangsuth, 2016) to scaffold their brainstorming process on what to create with a Gogo Board and its sensors, and we collected these worksheets as artifacts. After the maker activity, we asked students about how they made decisions on what, when, and which feature to use to post; how often they reflected or perused other students' portfolios; and how they receive and give feedback, and sometimes referring to instances during the maker activity, or examples from their Seesaw portfolio account.

Data Analysis

We conducted data analysis by creating contextual summaries of each student's interviews, and developed a coding scheme to delineate themes and patterns that emerged from the qualitative data (Miles, Huberman & Saldaña, 2014). An example of a code is "feedback", where we would highlight instances when a student mentions providing or receiving feedback from teachers, peers and parents. Although this code was applied literally in the beginning, upon going through the data in a second and third round, a more interpretative sense emerged from the code "feedback", whereby "feedback" was often associated with brief comments or likes on Seesaw and lengthier constructive feedback was given verbally in the classroom, rather than written on Seesaw. This culminated in a deeper analysis of the broader classroom culture that was built around providing and receiving feedback, and ponderance into how Seesaw worked for and against building the culture that the school had envisioned. This is elaborated later in the Results & Discussion section.

Apart from analyzing interviews, we analyzed Seesaw posts that students generated over 4 semesters, Fall 2017, Spring 2018, Fall 2018, and Spring 2019. I placed greater emphasis on Seesaw posts that students generated over the past year (i.e. Fall 2018 and Spring 2019), since these were posts that most students recalled more easily when they gave examples during the interviews, and since some students were new to the school and hence only had posts from Fall 2018 and Spring 2019. We analyzed these posts mostly for content, based on themes and patterns that had emerged from the interviews, thus following a coding scheme similar to that of the interviews.

Results & Discussion

"Big Deal" Projects

Anjana is a female student in Group 3, aged 10 years old, one of the older and more advanced students in the school. An example of Anjana's interview snippet is provided below, where she mentioned that her largest (and likely most important) audience is her parents:

> A: Usually after like Demo Night or exhibition, I post the finished... I usually like to keep finished products a surprise until I show my parents. But well if it's not like a big thing and if it's finished, I would take a picture of it too. So and also like when it's just like a small thing, it's like a sheet of paper or something, I usually do like work in progress. For example and I was nine years old I uhm drew a cell so and it was supposed to be a DNA poster instead of a cell poster, so it was just one part of the DNA so I just took a picture but I still haven't finished it.
> M: Okay so you usually do like to do finished products if you want to keep it as a surprise for your parents
> A: Yeah.
> M: So you don't want them to see it on there?
> A: But this one isn't like that much of a big deal, so I feel so I did that one and like this was after demo night and stuff so yeah.

M: So which ones are like… big deal products? Like, was the Design Thinking one a big deal one?

A: No. So this one was a big deal one.

M: What was that one?

A: It was a double helix that I made.

In this interview excerpt, Anjana revealed that she is very cognizant that her parents are the main audience that her Seesaw posts cater to. She posts on Seesaw based on how she thinks her parents will react to her work, and Seesaw serves as a communication platform for work that she is proud of and willing to share with her parents. However, an interesting point she made was that for work that she is exceptionally proud of, she would choose not to post on Seesaw, including most in-progress components (she only posted one origami DNA helix prototype a month before posting this final product), so that she can give her parents a surprise when they see her finished work in real life. She also makes a mental category of projects that she considers "big deal". The double helix that she mentioned (which she considered a "big deal" project) is pictured here:

Figure 1. Anjana's DNA Helix physical structure and voice recording

From her voice recording that described what she made, we infer that Anjana spent a lot of time and effort putting together the pieces of the DNA Helix, a project that not only required basic scientific understanding of how the DNA is shaped in a helical structure and of how its nitrogenous bonds react to join the DNA bases, but also planning and organization when building this structure in 3D. Anjana elaborated a bit about the science behind the helical structure in her voice recording (she referenced weak hydrogen bonds and the sugar backbone structure), but she recognized the importance of demonstrating the helical structure:

A: In the inside are the base pairs that are the 4-letter code to the DNA. The DNA is the instruction manual to life, so without the 4-letter code it wouldn't be working. They're in different orders so you can become any person. And so things around it are the white sugars. The sugars protect the DNA and the base pairs aren't actually in puzzle pieces; they are actually attached by weak hydrogen bonds, which is why the sugar needs to protect it.

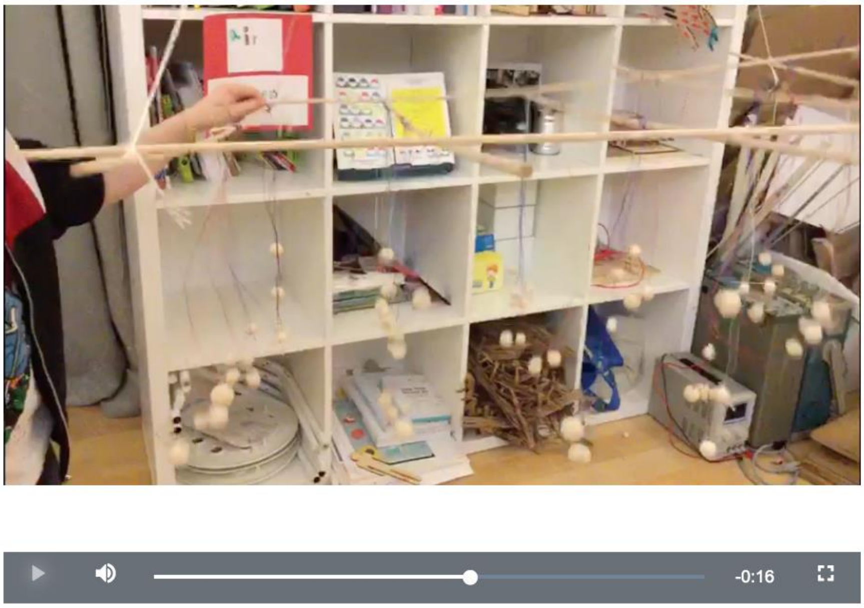
As to why Anjana considers some projects, and not others to be "big deal" projects, she makes this distinction based on the time and amount of effort she put into learning and understanding the science behind how DNA is formed, building the structure, choosing the right materials and differentiating colors to put forth scientific messages visually etc. These were many steps Anjana had to take before arriving at a polished, scientifically accurate finished product. In the end, she chose to portray only the final product on Seesaw, in an effort to portray the best of herself and her highest standard of learning towards her parents.

Anjana is not alone - upon closer artifact analysis, many students in Group 3 only post final products that they are proud of. This example encapsulates the dilemma of process-oriented documentation versus sharing final products. Constructionism recognizes that both process and product are key to the maker's journey of learning, but without sharing the process and only sharing the product does not holistically inform audiences about what the maker has learnt. The portfolio, although put forth as a documentation tool, has become a showcase tool that undermines its original purpose as a mode for formative assessment.

Portfolio as a Teaching Tool

Jim is a male student in Group 2, aged 8 years old, at the median age of students in this school. He created a model of how air molecules work, using wooden dowels attached to wooden balls suspended with string. Below is a transcript of his video recording and a screenshot of his video that shows his demonstration of how his model depicts air molecules:

> J: Hello everybody! Today I'll be speaking about air molecules. You see, these are air molecules (holding his wooden model), and they're always around you. And when you speak, they bump into each other to make vibrations, just like this (shakes his model to make the dangling wooden balls bounce lightly). And then… it turns into sound waves and travels to your ear.



air molecules model

Figure 2. Jim's Air Molecules Model and video recording and short caption

Through Jim's video demonstration and audio description of his air molecules model, one can grasp that Jim had to understand how air molecules worked scientifically in order to "teach" it via his video demonstration. The mode of offline teaching via video demonstration and presentation reinforces concepts that Jim learnt, and can also be viewed as a passive interaction he has with his audience. Although brief, Jim's description inserted the use of scientific terminology such as "vibration" and "sound waves", which he made an effort to show by shaking his molecule model to make the wooden balls move and make bumping noises. Through this artifact analysis, Jim's documentation of his work demonstrates a degree of understanding about scientific phenomena related to air molecules, and the features he chose to use (video and audio) accentuated his ability to communicate and record his understanding, as video showed movement (simulating vibrations) while audio displayed noise (simulating air molecules bumping into one another). However, this artifact resembles a final product, and his portfolio unfortunately minimally showcases his maker journey that led him to building this wooden air molecule model. Similar to the discussion above, both Jim and Anjana show some important understanding about relevant scientific phenomena in their completed construction and their portfolio posts, but both neglect to document and track how their process that lead to these discoveries.

Demonstrating Iteration

Zach is a male student in Group 1, aged 5 years old, one of the youngest students in the school. As part of a larger group activity to create representations of icebergs out of recycled and/or prototyping materials students were expected to illustrate how icebergs float in water. Zach chose to create his iceberg out of Legos, and placed them in a basin of water to demonstrate how icebergs float in the sea. Other students made their icebergs out of styrofoam, toilet paper rolls, egg cartons etc. and placed theirs in basins of water too. The interesting factor about Zach's iceberg was that he repeated this activity twice (on two separate days) because his first iceberg "didn't work". His description via voice recording and the photograph he took is shown below:

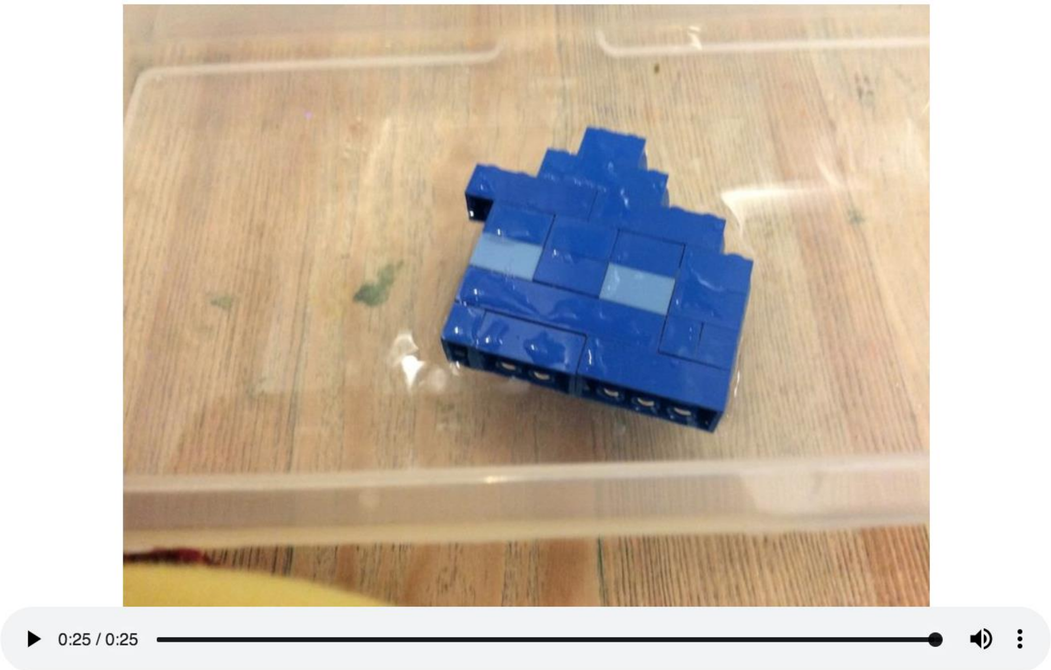


Figure 3. Zach's failed Lego Iceberg and voice recording

Z: This is my iceberg. It's made out of Legos. I built it and I put it in water, but it didn't work. So I tried to make it get stable, but none of those stuff worked. I built it higher but it didn't work.

In this brief description, Zach shared that his iceberg "didn't work", which we inferred that he meant that his iceberg could not stand upright in the basin of water, as depicted in the photograph where it is lying on its side. The next photograph and voice recording (from Zach's post the next day) depicted a successful attempt at building his Lego iceberg:

Figure 4. Zach's successful Lego Iceberg and voice recording

> Z: This is my iceberg. It's made out of Legos. I built it and I made it more big like a pyramid... and then when I put it to the water, it works because I tested it out and I built it in more Legos.

During Zach's interview, he stated that he typically uses voice recording to describe the content of the activity he had just done:

> Z: I would talk about it… about my work. … and we usually record it and we usually say what it is and what it does.

The fact that Zach generated a post about a failed attempt at creating his Lego iceberg speaks volumes about his willingness to show instances of both successful *and* unsuccessful attempts in his learning journey. Zach mentioned in his interview that he usually only describes what he did at school, which led to him describing a failed and successful attempt. Unfortunately, because this instance occurred the previous year and Zach did not recall the details of his workflow for this particular example, it is unclear whether he shared these examples on his own, or whether a teacher prompted him to photograph the failed attempt. However, as one of the youngest students, he has shown the school community that he tried again and in doing so, modified his existing design to create something that would work.

Implications & Conclusion

In sum, these are three examples of students using their portfolios to showcase projects they did in their school's makerspace, but each theme - "Big Deal" Projects, Portfolio as a Teaching Tool, and Demonstrating Iteration - include many more examples and is representative of the observations made at this research site. According to how this school site runs their workflow, students' use of their portfolio tool seems highly dependent on the classroom culture and teachers' expectations, where teachers would constantly remind students to document their work. If we refer back to learning theories associated with Constructionist pedagogy where social and embodied interaction play a pivotal role, our data shows that students seem somewhat actively engaged in sharing their work through documentation on their portfolios, but there is a lack of interactive sharing via comments and discussion on the portfolio platform. Perhaps the largest cause for concern depicted in the three themes stems from students' attitudes that the project they showcase should be considered "big deal". Although sharing is still part of the process and part of practical Constructionist pedagogy, the notion that only important phases should be shown is a culture that has modified the original Constructionist mindset with the introduction of the portfolio tool. Nevertheless, process documentation is vital for students to share via teaching and share via iterating, which are demonstrated in the other two themes.

Through this study, further questions have been generated: 1) how do we nurture a culture of sharing works-in-progress and failed attempts among K-5 students? and 2) how do we design features in process-oriented technological tools that can empower young learners to be proud of their works-in-progress to share confidently?

Future research should push towards discovering how the assessment tool could be designed to cultivate an ideal maker classroom culture, ultimately burgeoning a symbiotic relationship between an assessment tool and Constructionist classroom culture of sharing and documenting.

References

Bevan, B., Petrich, M., & Wilkinson, K. (2014). Tinkering is serious play. *Educational Leadership*, *72*(4), 28-33.

Eriksson, E., Heath, C., Ljungstrand, P., & Parnes, P. (2018). Makerspace in school—Considerations from a large-scale national testbed. *International journal of child-computer interaction, 16*, 9-15.

Gershenfeld, N. (2008). *Fab: the coming revolution on your desktop--from personal computers to personal fabrication.* Basic Books.

Keune, A., & Peppler, K. (2017). Maker portfolios as learning and community-building tools inside and outside makerspaces. Philadelphia, PA: International Society of the Learning Sciences.

Maltese, A. (2019, Sep 23). MakEval. MakEval. Retrieved from http://www.adammaltese.com/content/makeval/

Merritt, E., & Gangopadhyay, P. (2014). Building the Future of Education: Museums and the Learning Ecosystem. Building the Future of Education: Museums and the Learning Ecosystem. American Alliance of Museums. Retrieved from https://www.aam-us.org/wp-content/uploads/2017/12/Building-the-Future-of-Education.pdf

Miles, M. B. Huberman, A. M., & Saldana, J. (2014). Qualitative Data Analysis: A Methods Sourcebook. *3rd. Ed: Thousand Oaks, CA*: Sage.

Murai, Y., Kim, Y. J., Martin, E., Kirschmann, P., Rosenheck, L., & Reich, J. (2019, March). Embedding Assessment in School-Based Making: Preliminary Exploration of Principles for Embedded Assessment in Maker Learning. In *Proceedings of FabLearn 2019* (pp. 180-183). ACM.

Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2017). Empirical studies on the Maker Movement, a promising approach to learning: A literature review. *Entertainment Computing, 18*, 57-78.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* Basic Books, Inc.

Somerville, K. (2013). Museum and P-12 School Collaborations and the Role of a Third-Party Facilitator.

Thanapornsangsuth, S. (2016). *GOGO CARD* [PDF].

Thanapornsangsuth, S. (2016). *4. Blank Book* [PDF].

Learning by making to solve real problems in a constructionist environment

María Florencia Morado, *florencia.morado@una.ac.cr*
Coordinator, Masters in Technology and Education Innovation, National University of Costa Rica; Director of Creative Garage

Ayelén Eva Melo, *eva.melo.psm@gmail.com*
Masters' Student in Technology and Education Innovation, National University of Costa Rica; Teacher at Creative Garage; Vision Leader at Tree of Life International School

Angela Jarman, *angela.jarman@treeoflifelearning.com*
Academic Director, Tree of Life International School

Abstract

This paper looks at how a group of five children, aged 7 to 11, found solutions to real problems while programming with Micro:bit micro-processors in a constructionist learning environment. The children developed skills in *mathetics* and computational thinking as they tinkered in microworlds of their own specific interests, creating "objects-to-think-with". Several investigative questions guided educators in this learning experience: How can we motivate children to create their own microworlds where they can resolve real problems using technology? How do learner-driven projects permit exploration and deep thinking that goes beyond the technology itself? When children engage in projects that have personal significance, how can we best support them in their search for solutions to the problems they encounter along the way? How can we help children develop computational thinking? We present two projects where children constructed objects while tinkering with technology, resolving personally significant "real-life" problems and developing demonstrable skills in *mathetics*, computational thinking and problem resolution. Further, we describe the role of the educator as observer and guide working inside the constructionist paradigm, documenting the learners' progress towards greater independence and autonomy throughout the learning experience.

Keywords (style: Keywords)

Microworld, objects to think with, coding, micro:bit, computational thinking.

Introduction

The Creative Garage is an educational makerspace in Costa Rica which aims to create the necessary conditions for children to learn while constructing with technology. The center does not teach children how to use technology as an aim in itself; rather, children are challenged to learn while constructing personally-significant projects with technology. Along the way, educators accompany learners in the recursive process of exploring the limits of the technology, discovering what can be done and building "objects-to-think-with" in projects that are seeds of change, conveying powerful ideas that permit students to make personal connections to the newly-acquired knowledge (Papert, 1980).

Didactics is the art of teaching, but the art of learning, or *mathetics*, is short-changed in education (Papert, 1993). In school, children are not taught to learn. This is why the Creative Garage guides learners in *mathetics*, helping them learn to resolve problems, to analyze real-life situations, to identify prior knowledge that could be part of the solution, and to break problems into parts in order to deal with them one by one. This paper shows how children between 7 and 11 years of age solved real-life problems in learner-directed projects involving Micro:bit microprocessors. We describe two projects where students learned how to learn, by constructing what can be recognized as objects-to-think-with.

The first step is to recognize that real-life problems are complex, with contingencies and unforeseen difficulties that can emerge along the way; it is not possible to think in terms of set instructions or defined steps. This makes it necessary to analyze the context, to see how it can steer learners to possible solutions. Part of the context includes the different technologies and materials provided by Creative Garage, which students can use as they work towards a solution. However, educators also make available ideas, thinking strategies, concepts - the fruit of past learning. We want students to become tinkerers in the sense of developing the ability to progressively create solutions with the materials and concepts at hand, to resolve immediate challenges while advancing the project overall.

> The basic tenets of *bricolage* as a methodology for intellectual activity are: Use what you've got, improvise, make do. And for the true *bricoleur* the tools in the bag will have been selected over a long time by a process determined by more than pragmatic utility. (Papert, 1993, p. 144).

However, when problem-solving involves coding, learners must put more complex problem-solving strategies into action, strategies they may not yet possess, and educators must decide how to involve learners in "reflecting on more complex aspects of their own thinking" (Papert, 1980, p. 28). This is achieved by placing children in the programmer's position, permitting them to connect with ideas from science, mathematics and the art of construction. It is in the nature of children to learn from their surroundings. Without a doubt, learning how to communicate takes up the most time in a child's life, and it is something they learn to do quickly, and well. Programming is communicating with other entities, with computers and microprocessors, in a language common to both humans and machines. As a child learns to program, the learning process itself is transformed, becoming more active, autonomous and self-directed.

We believe that if the appropriate conditions are created for children to develop projects as they are developed in the Creative Garage - where, among other skills, children are permitted to program - the learner also develops a new way of thinking. Children learn to create their own video games, animations, characters and more, all the while increasing the complexity of their thinking inside the ever-widening spiral of learning, where children "**imagine** what they want to do, **create** a project based on their ideas, play with their creations, **share** their ideas and creations with others, **reflect** on their experiences – all of which leads them to **imagine** new ideas and new projects" (Resnick, 2007, p.1). We share two cases that demonstrate this learning path to greater complexity and authorship. Learners began programming games and playing with different technologies, then created solutions to serious problems, thus displaying two of the fundamental principles of *mathetics*: first, one must relate something that is new to something already known; second, one must take the new and make it one's own, by playing or constructing with it.

Microworlds (Papert, 1980) are spaces where learners can simulate scenarios in order to experience real-life situations. They are opportunities to play "make-believe" and develop powerful ideas and thought patterns. Microworlds become knowledge incubators, as they feature elements that are part of their real-world counterpart, on a more manageable scale. At Creative Garage, we provide technology and recyclables so that learners can create their own microworlds.

Context

Creative Garage is an educational makerspace that holds after-school workshops for children 3 years of age and older. Children voluntarily participate in what we call "creative technology." In weekly one-hour workshops, learners develop projects which may involve coding, robotics and 3D printing with materials like cardboard. Markers, paints and recycled plastic containers are just some of the many items available to learners who create projects which are fun because they are significant in terms of their personal interests.

As soon as they enter the makerspace, learners explore the different materials available -- the technological as well as the non-technological -- to see what they can make with them. We propose project ideas and accompany learners as they develop their chosen project, acquainting them along the way with the potential uses of the different technology. Learners will use this knowledge later, with more complex projects.

At present, 60 children participate regularly in the Creative Garage workshops, which permits us to accompany their exploration and learning long-term.

Methodology

This research is positioned inside an interpretative paradigm (Lorenzo, 2006; Ponterotto, 2005) where the results we observed are analyzed in light of the central concepts of constructionism, such as *mathetics*, tinkering, microworlds, objects-to-think-with and computational thinking. The projects are described in depth in their context, and provide evidence of their effectiveness for learning. Data is collected by an interactive researcher–participant in an actual educational context; the educators writing this paper were also the teachers facilitating the workshops, and seek to relate, understand and analyze the results of how the children developed skills fundamental to constructionism by creating projects using the microprocessor Micro:bit.

To meet the objectives of this investigation, the participants and their projects were chosen intentionally and deliberately as a purposive descriptive sample, using the *homogeneous sampling method* (Etikan, 2016). The target population meets practical criteria, with girls and boys aged 7 to 11 years. All are students of Creative Garage, which has developed complex projects that are significant to children, using Micro:bit. As well, they meet knowledge and experience criteria, having at least 15 hours experience in camps or workshops.

The sample features two projects, in order to emphasize the depth of learning instead of generalizing based on a larger sample. These projects were the work of three boys and two girls. One of these projects, the "Cubick Shaker", was developed in four hours, over four weeks; the "Thermometer with alarm" was created in four hours, over two days.

Technology in service of projects

At Creative Garage, we believe that technology should serve projects, not the other way around. Teaching technology for its own sake makes no sense as technology has most value meeting a specific need. For this reason, we propose different projects to students, immersing them in narratives that give context and purpose, where the technology is at the service of learning (Brennan, 2015).

We explore the learners' interests, proposing projects that relate to them. For some children, it may be the graphic, collaborative and multiuser video game Fortnite[28]; for others, it may be pop stars, crocodiles or pets. Dialogue and active listening help us learn what they like, or what concerns them.

Work begins with guided exploration of the technology, with the intention of discovering its potential uses in the learners' specific interests. Later, we offer possible projects which they can transform according to their needs. In this process, we guide and support learners as they develop their projects.

Programming microprocessors

For previous generations of learners, programming languages like basic or C++ were not easy to understand or to learn. They didn't have any pedagogical intention. Logo, created in 1967 at MIT, was the first coding language with the goal of making programming easy and learnable and in 2007, Scratch became another tool for learning computational thinking. The launch of Scratch in 2007 changed the direction of programming dramatically, making it easier and more accessible, thanks to the idea of blocks. Designed by Mitchel Resnick and Lifelong Kindergarten at MIT, blocks made learning about algorithms, programming logic and computational thinking possible for the many, not the few. Soon other possibilities opened up, based on the same logic of blocks of code: for example, Microsoft's open-source platform, Makecode (https://makecode.microbit.org/), which facilitates the programming of microprocessors like Micro:bit.

A microprocessor is the most complex integrated circuit in a computer system. One could call it the system's brain. It is where logic and arithmetic operations take place and where instructions and programs are executed. It receives external information and interacts with it, and controls internal and external devices. Micro:bit, as in Figure 1, is a small, robust microprocessor that permits the creation of projects with motors, servos, lights and speakers, using information

[28] https://www.epicgames.com/fortnite/en-US/ **Fortnite Battle Royale** is a third person shooter survival game where the player has to survive against 99 other players. The player can build forts and collect weapons on the way. The main game also has a mode called Fortnite Save The World, which is also available as a separate game. The game can be played on iOS, Android, Nintendo Switch, Windows, PS4 and Xbox One. (Source Wikipedia)

obtained from accelerometers and sensors for light, proximity, sound and temperature. It is a small but powerful cybernetic machine with individually programmable LEDs which can be used to display text, numbers and images. Two buttons can be programmed, as well as motors, LEDs or any other component or external sensors that can be connected to the edge connector. The light sensor can act as an input, allowing the detection of ambient light or ambient temperature, while movement sensors, like the accelerometer, are activated when the plate is moving, and can also detect other motions, like shaking, rotations and even free fall. It also has connectors to external batteries, which allows the Micro:bit to function as programmed when disconnected from the computer.

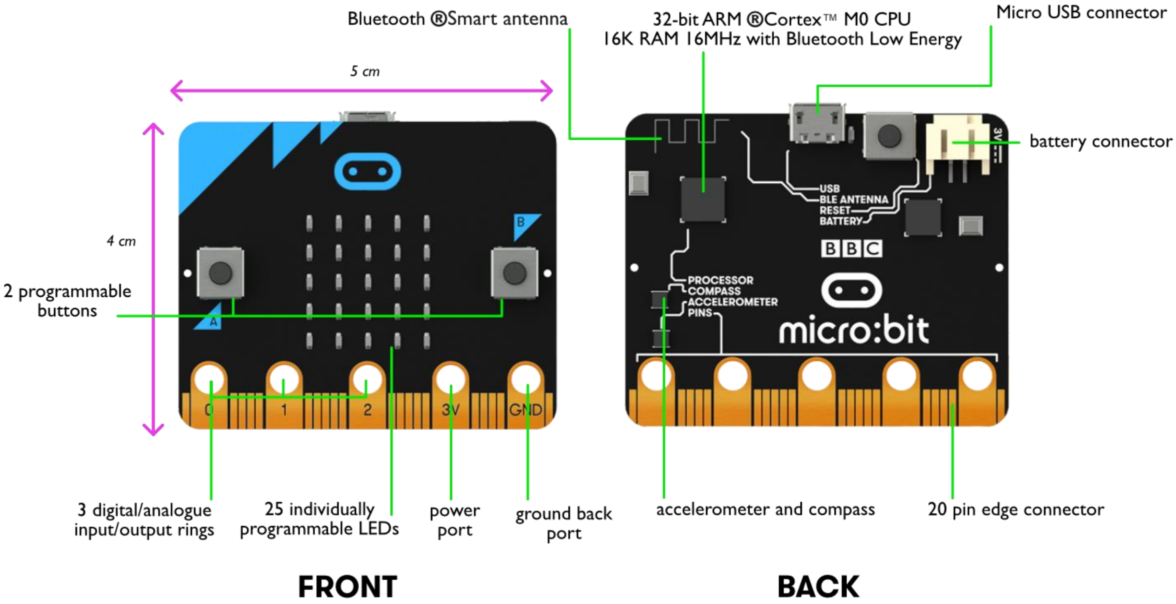


Figure1. Source: http://www.servicioti.com.uy/microbit/caracteristicas-microbit

What is interesting about Micro:bit is that it can be programmed with a language that uses blocks, providing a natural introduction to logical thinking. Programming with blocks "could contribute to mental processes not only instrumentally but in more essential, conceptual ways, influencing how people think" (Papert, 1980, p. 16), which is what we see as children extrapolate and problem-solve with logical thinking during different stages of their project.

It is simple to program the Micro:bit, by placing color-coded blocks in order. Children soon try new combinations with familiar blocks, experimenting with different functionalities. An onscreen simulator allows easy debugging and testing. Once the program is downloaded, external devices can be connected, making the project tangible, which generates ideas for improvements.

As the project advances, children grow familiar with programming concepts such as sequences, cycles, parallelism, events, conditions, operators, variables and extensions. Not only do they gain vocabulary, they adopt new approaches to programming. As children developed skills in interactive media, Brennan and Resnick observed that they adopted "four main sets of practices: *being incremental and iterative, testing and debugging, reusing and remixing*, and *abstracting and modularizing*". (Brennan & Resnick, 2012, p. 7)

Papert (1980) recounted how his childhood fascination with gears sparked an interest in mathematics. Gears were a natural part of his surroundings, and working with them allowed him to connect with others. He even saw analogies between the functioning of gears and the workings of his own body. To Papert, gears impacted his thinking; they served as “objects-to-think-with”. At Creative Garage, we want learners to interact with Micro:bits as Papert interacted with gears, in order to generate a natural opportunity for children to develop their thinking with others.

What are we going to do today?

Each time they walk into class, our students inquire curiously: “What are we going to do today?” They like to make things, and they get excited with the results: they learn as they *do*.

As children encounter the different technologies at Creative Garage, we can see the principles of *mathetics* at work: learners relate the new with what they already know, and make it theirs by making something with this new knowledge. Each week, our students enter class eagerly, intrigued by the possibilities open to them, ready for new challenges.

The first project emerged as part of the Micro:bit Challenge 2018, an international competition where young people were invited to resolve a real-world problem using a Micro:bit microprocessor. Two of our 11-year-olds decided they wanted to participate, and chose a theme related to health. They decided to develop a device for people who spent a lot of time playing video-games. Both boys are fans of Fortnite, and spend hours sitting in front of screens. Through investigation, they learned that sedentary people can develop heart problems and obesity. They proposed the “Cubic Shaker” as their solution; this device would be worn next to the body, in a pocket as shown in Figure 2. Sitting still for more than 20 minutes would trigger an alarm, making the Cubic Shaker vibrate, play an annoying tune and display the message “Move!” until the wearer moved energetically for a specific period of time. Any energetic movement would cause the Cubic Shaker to reset its timer.

Figure 2: The Micro:bit with its external devices in a t-shirt pocket

The first step was to break the project down into smaller challenges. They would need:

1. A regressive timer of 20 minutes;

2. If the counter reaches zero, activate three alarms:

a. Show text on screen;

b. Vibrate;

c. Play tune;

3. However, if the wearer moves during the 20-minute count-down, the timer restarts, without activating the alarms.

4. If activated, the three alarms should only stop if the person moves energetically.

5. The movement has to be more energetic than slight shifts in body angles and must activate the cardio-vascular system.

To meet some of these challenges, they re-mixed lines of code they had already used in previous exploratory projects; other challenges, however, represented learning opportunities, so they began searching online for similar solutions. They discovered no other project that was exactly like theirs, so they had to analyze the solutions most like their project and articulate how they worked, to draw parallels.

By addressing these smaller challenges, they resolved the "big problem" with the following code (Fig. 3):

1. Regressive Timer;

2. End of regressive timer and activation of alarms;

3. End of regressive timer in case of energetic movement;

4. Turning off alarms;

5. Detection of energetic movement;

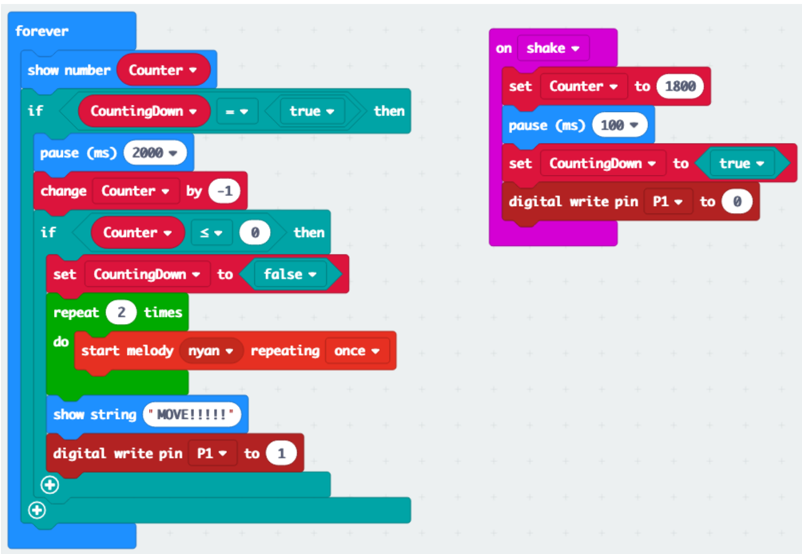


Figure 3: The code

Once they had their code ready and downloaded to the Micro:bit, they had to connect the speaker and the vibrator. The alligator clips they were using were too big if the Cubic Shaker was going to be pocket-sized, so they had to cut and adapt them. They weren't happy with a device dangling cables and drivers, so they decided to 3D design a cube-shaped container. They measured the microprocessor and its external devices, then designed openings for the screen, vibrator and speaker. This is how the cube-shaped model shown in Figure 4 got its name, "Cubic Shaker."

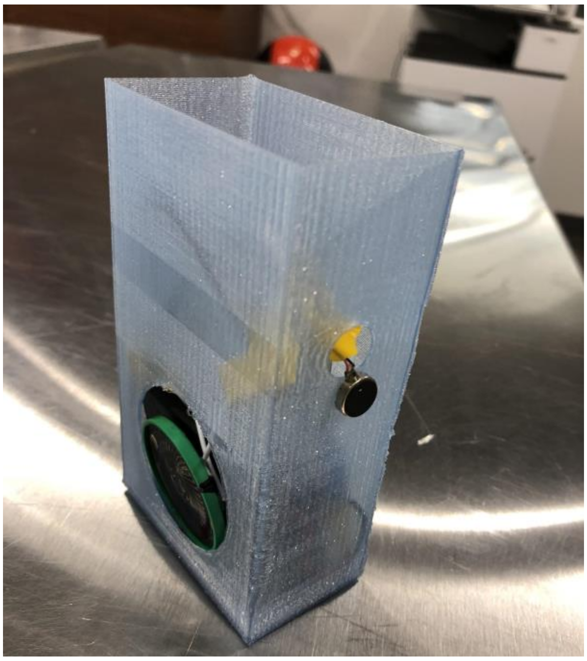


Figure 4: Cubic Shaker

The second project, “Thermometer with alarm,” emerged from a problem mentioned casually by a mother of one of our students. Her research involves biological samples that are stored in a freezer. The samples are ruined if the temperature doesn’t stay stable within a limited range. The freezer had a system that displayed the temperature and activated an alarm if it rose above the established limit, but on this particular day, the freezer’s display and alarm system had malfunctioned. The researchers did not know if the samples were still at the desired temperature. Three of our learners had already been working with Micro:bit, and we had already discussed its range of possibilities. We had already created projects with lights, thermometers and simple electronic circuits that displayed text and numbers on the Micro:bit screen.

We invited the mother to share about the problem she was having in her laboratory. When she finished explaining the issue, we asked the children if they thought they could create a solution. All three enthusiastically answered, “Yes!”

The project was divided into two days of activities. First, they decided to create a thermometer with a Micro:bit; they knew they could do this because it was something they had already done in previous projects**.** On day 1, after creating their thermometers, they recorded the ambient temperature. Next they held the Micro:bit in their hands, noting that as the temperature warmed up, their displays showed incrementally higher temperatures. Then they placed all of their Micro:bits in the refrigerator. They documented the changes in temperature, as shown in Figure 5.

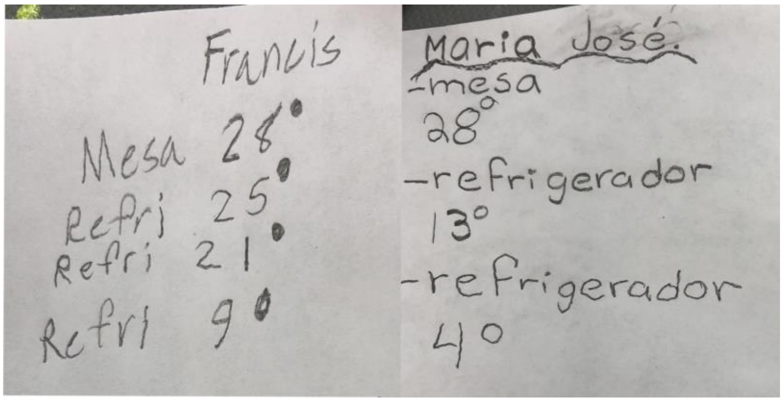


Figure 4: Temperature measurements

They were pleased with their achievements thus far. However, they knew they had not yet resolved the problem. On Day 2, they decided that they needed an alert message and a warning light if the temperature began to warm up.

Working with the temperatures measured the day before with the refrigerator tests, they set parameters: if the temperature was lower than 15º Celsius, everything was fine, but if it was higher, then there was a problem.

They programmed the Micro:bit based on the following requirements (Fig. 5):

1. It is necessary to measure the temperature.

2. If the temperature is greater than 15º C, show the text, “Danger,” and turn on the light.

3. If the temperature is less than 15º C, show the text “Ok” and turn off the light.

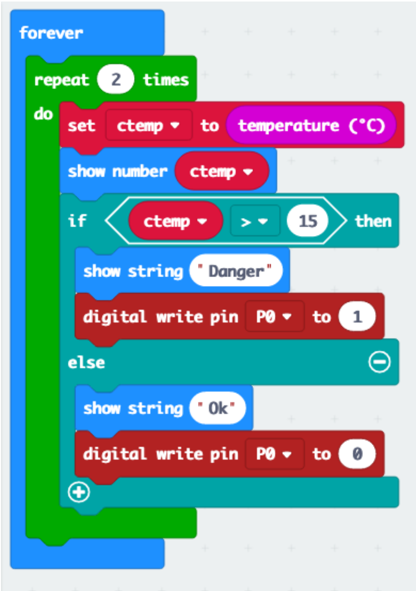


Figure 5: The code

Once again, the children reduced the big challenge into smaller steps.

1. Measure temperature;
2. Show temperature on screen;
3. Establish if temperature is above limit;
4. If above limit, execute alarms: show “Danger” text and switch light on;
5. If below limit, show “Ok” text and do not switch light on.

They made many tests, even putting the Micro:bit in plastic boxes in the refrigerator (Fig. 6), to test if a) the light would switch on as the temperature rose, and b) the warning text would appear on the screen.

Figure 6: The "Thermometers with Alarms" in the refrigerator

Discussion: Constructing constructionism

At Creative Garage, we have re-conceptualized education, aiming to offer learning experiences that are different from those typical of traditional education. We are not a space for the transmission of knowledge, but rather an environment that facilitates the construction of knowledge, through the creation of objects-to-think-with, taken from the learner's own particular context and interests.

The space itself forms a part of these learning experiences. It is designed and arranged in a way that provokes learning. It is what Assmann (2002) defines as a pleasure space, which is fundamental for ensuring that the learning process doesn't become an instruction process.

> The pedagogical environment needs to be an inventive place of fascination: rather than inhibiting, it provides the required dose of shared imagination and enthusiasm so that the learning process engages all of the senses. [...] because learning is, before anything, an integral physical process (p.28)

Recognizing that learning is a physical process helps us understand why pleasure and emotion must have a central role in education. The emotion our learners feel when they have made their project "work" is fundamental to their learning. Maturana (1990) states that "we live in a culture which has devalued the emotions in function of an over-evaluation of reason" (p. 65) and it is exactly for this reason that we recognize that emotion must guide their search — and their learning.

We guide our learners in an exploration of the full potential of the technology, so that they can attempt to face more complex challenges later. We invite them to play with reality, constructing objects in imagined scenarios, that could be solutions to real-life problems. In this way, our students learn from practice, and from reflecting over what they have made, they also learn how to make improvements, in a creative spiral (Resnick, 2007; Morado, 2018; Martinez & Stager, 2013).

Both the "Cubic Shaker" and the "Thermometer with Alarm" demonstrate how the strategy of being incremental and iterative works in practice. Incremental changes allow students to tweak their project, moving it towards the desired form. The work process itself is iterative, because students repeat a cycle of imagining and creating, producing multiple versions of the end result. They make, test and reflect, then repeat these steps as many times as necessary before ending up with the final project (Brennan & Resnick, 2012).

Projects like these become “objects-to-think-with” in that they allow learners to explore a broad variety of topics, from programming, electronic circuits, sensors and 3D printing, to health, social problems, the benefit of exercise and details on biological research like how temperature variations can affect samples.

As students test and debug programs, we discover alongside them that errors are not the end of a project; on the contrary, they are indispensable for improving the end results, and most importantly, for learning. As learners work through a project, we naturally establish a routine of identifying, analyzing and resolving problems as they appear.

This closer look at two projects brings us back to the principles of *mathetics*: that learners must relate the new to what is already known; and make what is new their own by doing something with it. Our students applied prior knowledge acquired from projects they had completed before, in new contexts. They found it easy to branch off from this know-how, re-mixing it, connecting it to new knowledge and using it to resolve new problems. In this way, they took a particular experience and made it abstract, modularizing it and applying it in a different context, for a different purpose, “building something large by putting together collections of smaller parts” (Brennan & Resnick, 2012, p. 9).

They adapted the materials they had at hand, constructing the parts they needed, becoming expert *bricoleurs*. They encountered and resolved challenges progressively and incrementally, guiding their improvements according to what actually worked.

They learned to program by programming. They learned to resolve problems by facing them, breaking them down into more manageable parts. When they encountered different results than expected or desired, they tested and tinkered until their goal was achieved. They applied strategies that are useful in the “real world,” as they explored a microworld. They thought about serious problems while playing with the materials they had at hand.

Conclusion

By challenging our students to solve problems with programming, we are permitting them not only to learn to code, we are also letting them learn by coding (Resnick, 2013). As they become proficient with programming strategies, they are also resolving problems, creating projects and sharing them with a wider community -- which is the heart of learning. Children should be challenged to create solutions to real problems because this helps them learn better. When their ideas serve their interests and meet others’ needs, children become enthusiastic creators. How the child sees the challenge relates to his or her level of engagement, and ultimately, to the degree of understanding that is achieved. To understand is to invent, to investigate, to test if things work, to make, to reflect, and to give value to what one has made. Understanding is at the heart of constructionist theory.

The significance of the challenge is crucial, but so is our practice in the classroom, and how we support learners as they work on their projects. We must guide the process actively and respectfully, giving attention to their needs in those moments when they get stuck, yet avoid solving their problems for them, so they learn how to find solutions for themselves.

In this way, we give life to the theories of constructionism, which affirm that the best learning is not derived from better forms of instruction, but from more opportunities to construct knowledge. Our role as educators consists in creating the necessary conditions for learning so that students are autonomous, curious, enthusiastic, challenged and supported.

References

Assmann, H. (2002). *Placer y ternura en la educación: hacia una sociedad aprendiente (Vol. 90)*. Narcea Ediciones.

Brennan, K. (2015). Beyond technocentrism. *Constructivist Foundations*, *10*(3), 289-296.

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada* (Vol. 1, p. 25).

Etikan, I., Musa, S. A., & Alkassim, R. S. (2016). Comparison of convenience sampling and purposive sampling. *American journal of theoretical and applied statistics*, *5*(1), 1-4.

Lorenzo, C. R. (2006). Contribución sobre los paradigmas de investigación. *Educação*, *31*(1), 11-22.

Martinez, S. L., & Stager, G. (2013). Invent to learn: Making, tinkering, and engineering in the classroom. Constructing modern knowledge press.

Maturana, H. R. (1990). *Emociones y lenguaje en educación y política.* JC Sáez Editor.

Morado, M. F. (2018). Entornos virtuales de aprendizaje complejos e innovadores: Una experiencia de creación participativa desde el paradigma emergente. *Revista Electrónica Educare*, *22*(1), 1.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc..

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. BasicBooks, 10 East 53rd St., New York, NY 10022-5299.

Ponterotto, J. G. (2005). Qualitative research in counseling psychology: A primer on research paradigms and philosophy of science. *Journal of counseling psychology*, *52*(2), 126.

Resnick, M. (2007). *All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten*. In Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition. pp. 1 - 6.

Resnick, M. (2013). Learn to Code, Code to Learn. EdSurge, May 2013. Recovered from https://www.edsurge.com/news/2013-05-08-learn-to-code-code-to-learn.

Lessons from a Teacher: Culturally Relevant Teaching in a Constructionist Classroom

Sawaros Thanapornsangsuth, *st2839@tc.columbia.edu*

Nathan Holbert, *holbert@tc.columbia.edu*
Department of Mathematics, Science and Technology, Teachers College, Columbia University, New York, USA

Abstract

This study investigates the shift of perspective of a Thai veteran teacher, Kru Ann, who facilitated her students in a constructionist learning experience that emphasized making for others by following the footsteps of the late Thai King. Through the process of making, the teacher developed new-found perspectives of her students while reflecting upon her own pedagogy. We analyze moments of the teacher facilitating her students through a constructionist learning environment. We observe how she learned from her students and held higher expectations of them. Drawing on literature from constructionist design paradigms and culturally relevant pedagogy, data from the constructionist class describes an emergent shift in the teacher's practices toward her students. This study addresses a gap in constructionist research as it spotlights on a teacher from a low-income community who practices constructionist pedagogy in a formal public school setting.

Keywords (style: Keywords)

Constructionism; Culturally Relevant Pedagogy; Thailand; Teacher

Introduction

The influence of constructionism can be seen across a broad range of spaces in education and school settings. It has gained public attention in schools, libraries, and public spaces across the United States and internationally. Nevertheless, many teachers who support the ideas of Constructionism find it complicated to navigate in traditional classroom structures. There are multiple rigid barriers such as curriculum constraints, lack of resources, and few allies (Suziedelis, 2018). There is a paucity of research examining teachers who practice constructionist pedagogy in a formal school setting. Likewise, little work has been done to explore the perception of a teacher from an underfunded community. In the Little Builders project, we engage teachers and young learners from a low-income community, in using tools and technologies to build projects for people in their own community. In this paper, we present early findings of interviews with a champion teacher and field notes observation, participating in the Little Builders project.

This paper examines how Kru Ann ("kru" means "teacher" in Thai) navigated, described and reflected upon her first co-designed constructionist classroom practices with the researchers through the lens of culturally relevant pedagogy. In examining Kru Ann's experience of shifting toward constructionist pedagogy, our paper aims to capture how her perception towards herself and her students change after assisting them with their making projects. We hope to inspire discussion particularly in the importance of teacher engagement and teacher-student relationships in a constructionist learning environment.

Literature Review

Papert and Harel (1991) called "instructionism," as a pedagogical paradigm that holds a sharp difference to constructionism. Instructionism focuses on the "nature of knowledge" rather than the "nature of knowing." It focuses on a technical matter that is commonly found in traditional practices in school. Instructionist mode of learning aligns with Freire's (1996) view of the banking concept in education. Freire deemed the concept as an oppressive classroom dynamic that solely gives the power to the teacher. He wrote, "Teacher teaches and the students are taught; teachers know everything and the students know nothing; teacher talks and students listen; and teacher chooses the program content and the students adopt it" (p.73) The banking concept in education doesn't only put students in a passive role of learning as they don't have the freedom to think for themselves or choose the path they are interested in. It also emphasizes the distance between a teacher and his or her students through its top-down structure.

Contrastingly, constructionism liberates learners from being taught when they are empowered to connect everything they wonder, know, and feel to expand themselves into learning new things (Blikstein, 2013). Constructionism shares Piaget roots that learning should be more experiential and connected to the real world. It extends Piaget's constructivism by proposing that the construction of knowledge "in your head" happens best when constructing tangible and shareable artifacts "in the world" (Papert, 1993, p. 142). The creation of an artifact—which could be either physical, virtual, or mental—allows learners to externalize and iterate on their thinking throughout the making process (Papert & Harel, 1991). Thus, the role of teachers in a constructionist classroom is shifted from being the only central source of knowledge to a more emergent one. Teachers are "pulling knowledge out" of her students like mining, rather than "putting knowledge into" like banking (Freire, 1996 p. 75; Ladson-Billings, 1994, p.34).

Ladson-Billings' (1994) seminal work on culturally relevant pedagogy also supports the idea of "teaching as mining." She emphasizes the importance of teachers creating relevant cultural connections to students' learning experiences. The pedagogy is designed to encourage teachers to question the nature of the student-teacher relationship, the curriculum, schooling, and society

(Ladson-Billings, 1995). Culturally relevant teachers move away from seeing their students as an empty vessel needed to be filled with their instructions. They hold high expectations and affirming views for all students. They consider students' social identities as assets rather than as deficits.

Methods

The study investigated a science teacher and 32 Thai 4th-grade students from an urban public school in Bangkok, Thailand. The students took a maker class as a part of the school's mandatory "Life Skills" class 2-3 times a week for one semester. The class was co-taught and co-designed by Kru Ann (the science teacher) and researchers.

Using qualitative research methods, the data collection includes pre and post interviews with Kru Ann, field notes observation, photos, and videos of the sessions. The information obtained through participants' pre and post interviews will provide a basis for the overall findings of this article. All interviews were in Thai. The interviews were video recorded, transcribed, and then translated to English. A pseudonym will identify each participant.

Class Design

The model of Little Builders consisted of 23 sessions taught in a formal classroom setting with Kru Ann as the lead teacher. Each class lasted one hour and occurred twice a week throughout the semester. The class began with an introductory session that allowed the students to get to know the researchers and introduced the class's goals of following Late King Bhumibol's footsteps by designing and making social inventions for their community. This session also encouraged students to think of problems that they cared about. After these introductory activities, the students interviewed the community members asking: "What are the problems or challenges in the community that you wished to be solved?" Following the design process, students brainstorm possible solutions to meet those needs and create a prototype. Students then solicited feedback on the prototype from the community members before making their final project. Kru Ann and researchers assisted the students throughout the course of building projects for the community. Finally, the students showcased their projects and shared knowledge with their teachers, classmates, and community members. They also delivered their projects to the community members.

Findings

Kru Ann is a veteran science teacher with more than 15 years of teaching experience. She is also a hands-on learning enthusiast. She heard of the term "constructivism" as one of the many learning theories in her early years of undergraduate studies but admitted that it was complicated to practice that in Thai classrooms. It was also uncommon to teach in such manners. Kru Ann was also the homeroom teacher of the students under this study. She was known among the students and her colleagues as being fun and approachable. The students who were considered mischievous by most of the other teachers generally got along well with Kru Ann, as she put in the effort to understand them and their family backgrounds. For example, Kru Ann mentioned in her pre-interview that she went to her student's home when her student was absent for multiple days.

Kru Ann took time off her regular working hours to visit her student's home. She said "I saw the condition of the house and understood. The dad was sick, the mom was pregnant but needed to go to work. The student was in 4th grade and she had two younger siblings. She had to take care of her dad and can't go to school." Kru Ann sympathized with the students. She knew that oftentimes when the students finally come to school, they will be scolded by the teachers. She wanted to foster an amiable classroom environment that didn't push her students away from coming to school. She feared that they will leave the system. Even though teachers scolding and physically striking misbehaving students are common practices in Thai schools, Kru Ann refrained from these practices, believing they might negatively impact her relationship with her students.

Kru Ann stated that “scolding would create fear but it is just temporary. We need to find other alternatives like being quiet, the results would be a lot more positive.”

When asked about her expectations of her students for the upcoming Little Builders class, Kru Ann expressed uncertainty about her students’ capabilities. Her concern was that her students “were trained to think inside a box” and the design tasks could be challenging. Kru Ann estimated that 50% of her students would do fine but 50% would struggle because they were not used to creating objects. However, she wished to see improvement in her students and she was excited to see her students help their own communities with their projects.

In terms of Kru Ann’s teaching style, she often stood at the front of the classroom, ask the students a few assuring questions, lectured her lesson and assigned each group a worksheet to complete during class time. If the students needed help with the worksheet, they could come to her table. Consider the following interaction, which exemplifies how Kru Ann taught and asked her students questions with an expected set of answers in her mind.

Figure 1. Kru Ann taught in front of the classroom, holding the picture of the late King Bhumibol.

In the very first session of Little Builders class, Kru Ann was at the front of the classroom addressing the photo of the late King Bhumibol (Figure 1). With the class goals of “making for others,” Kru Ann made sure that the students understood the connections by asking the students to come up with various royal projects that the King did for the country. She emphasized the fact that the King did the development projects not for his own good, but for the Thai people. After that, she informed her students that they would be following the late King’s footsteps this semester by making inventions for other people in their own community.

Kru Ann:	These are some of the projects the King had established. For whom?
Class:	For us!
Kru Ann:	For his people. Did he think of himself?
Class:	No
Kru Ann:	Who did he think of?
Class:	The people
Kru Ann:	Because the King always thought that Thai people were his family. He always did something for others. And did you ever hear of the phrase "follow the Father's footsteps."
Class:	Yes
Kru Ann:	And who is the Father?

Class: The King
Kru Ann: Do you want to make things for other people? Do you want to make other people happy?
Class: Yes
Kru Ann: If you think about making for other people, that's a good thing. But if you actually make it, it would be even better. Then when you look at his photo and you will feel...
Class: Proud!
Kru Ann: Proud. Whose footsteps are we going to follow?
Class: The King!

As the Little Builders class progressed through the design process, the class focused less on teacher's instruction but more on hands-on making and design. Kru Ann shifted her role from teaching in front of the classroom to working alongside her students (Figure 2). In her post-interview, the first thing she mentioned was that by sitting on the floor helping her students, she was able to see them in a new light. She compared helping her students complete their project with her regular instruction saying, "when I teach, I teach a lesson and the overall picture, but when I work with the students like this, I would be closer to my students and make me see something in them that I've never seen before."

Figure 2. Kru Ann sat alongside her students while helping them with their projects.

Kru Ann talked about Chan, a boy who was not good at school but well-loved by his friends. She discussed how she overlooked his unique personality in a regular classroom setting, but it became apparent to her once she had spent more time working on a project with him. Kru Ann said, "in a regular classroom, I wasn't aware of a situation that could help me see this, but from working with him, I've seen how Chan is like. Chan is an optimistic person. [...] He is a bit mischievous but the way he thinks and his heart--it's full of positive energy." Kru Ann taught Chan before when he was in second grade but she always saw him as a student who lacked focus, and often wandered around the classroom talking to other students (and disturbing their work).

Learning through making allowed Kru Ann to see differences in her students outside of the traditional school expectations. "If we look at the students in the context of this [Little Builders] class, no one is bad. If we look at Chan in a regular classroom he will be perceived as bad because he doesn't do his homework. Bao (another student) can also be considered as bad just because he can't read well! So if we look at how the students don't finish their assignment, yes they will be bad." She realized that her students hold more nuance in their capabilities but we must see them beyond the school expectations.

Kru Ann also mentioned a quiet girl named Tila, “I never heard her voice!” When Kru Ann was assisting her in using a handsaw to cut a plastic container, Kru Ann could see that Tila was very focused and determined. Tila’s closest friend, Dear, often complained when she took turns in cutting a plastic container that her hands are sore. Kru Ann was able to guide her and support her throughout the process. Kru Ann said: “It's like a discovery. Discovering students' positive traits when I was helping them”.

By knowing more about her students, Kru Ann could better support and unleash her students’ hidden potential. She remarked, “when we know the students better, we can spark their potential. It might not be clear sometimes, but we know that they have it and we can pull that out. Once we pull that out, it can be sustainable. It's sustainable in the way that they are a better version of themselves."

Kru Ann reflected upon her teaching during the post-interview that what she appreciated most from the co-designing process was “to ask questions like a researcher.” She said, “I can tell that when I teach, the way I ask questions and the way [the researcher] asks questions was different. [The researcher’s] questions can pull out a lot of information from the students. But mine, it's like a pattern "why, how" it doesn't cover a broad picture.” She observed how the researcher asks questions to learn more about her students and tried to change her own way of asking students questions. Throughout the semester, Kru Ann gradually moved away from asking leading questions to the kind of questions that valued her students’ opinions and allow them to express their own ideas. For example, the following conversation between Kru Ann and her students portrays how Kru Ann still adhered to her original way of probing with leading questions. However, she soon followed up with questions that encouraged her students to reflect upon their designs.

Kru Ann: Are you tired?
Class: Not at all!
Kru Ann: How about us teachers?
Class: Tired!
Kru Ann: But within that tiresome, what is it?
Mee: Energy!
Kru Ann: Apart from energy, what else?
Onn: Strength
Chan: Capabilities
Mon: Fun
Kru Ann: fun, strength, energy, capabilities
Park: group work and unity
Kru Ann: An ability to work as a team. Of all the things that we came up with today, aren’t they the lesson from hands-on making?
Class: Yes!

Kru Ann continued with more questions, but this time she invited the students to think further on their making process and reflected on their works in relation to the late King.

Kru Ann: Remember at the beginning of the class that we’ve talked about the late King and how he solved the problems of others. That was the beginning of how we started to think about making for others by following the King’s footsteps. Have we followed his footsteps?
Gem: We did.
Kru Ann: Don’t say anything yet. Think for one minute. In which way did you

follow his footsteps? Ok, raise your hand if you think you've followed his footsteps. [the class raised their hand] Ok, I think most of you think so. But let's go into details, of how?

Gem: I got to help others like Auntie Pan when she didn't have a garage can.

Chan: I got to help society. Solved the community problems so that they don't have to face similar problems in the future

Park: Got to make others happy

Por: Solved the community problems

By framing her questions differently and providing the space for her students to reflect, Kru Ann was able to listen to the students' authentic voice on their learning experiences.

Discussion

The goal of this study is to take an initial step in exploring the shift of perspective of a Thai veteran teacher who facilitated her students in a constructionist learning experience for the first time. The result of this study showed the ways in which constructionist practices take shape as a way of establishing new-found connections and perspectives between the teacher and students.

Being a veteran teacher with 15 years of experience, Kru Ann was comfortable in her profession. Although she was a hands-on learning enthusiast, Kru Ann hadn't practiced the constructionist way of teaching in her classroom prior to this study. As shown in the first excerpt of interaction on 'following the late King's footsteps,' Kru Ann tended to lecture her class in a traditional instructionist manner. She had a fixed set of expected answers in her mind when she was asking her students questions. She would ask leading questions to get students to say those expected answers. When asked about her expectations toward her students in the pre-interview, she was uncertain about their success. She thought that "50% will do fine but 50% will struggle." This notion contrasts with the culturally relevant teacher's concept of self and others, which believes that "all" the students were capable of academic success (Ladson-Billings, 1995). Though Kru Ann was devoted to understanding her students and their family backgrounds, she still held a deficit lens that assumes a low sense of expectation for what students from lower-income schools can do.

Working alongside with her students in a constructionist learning environment, Kru Ann was impressed to see the new positive sides in her students as well as their improvement in both technical and interpersonal skills. Kru Ann commented on how she discovered her students' latent qualities she previously overlooked. By paying close attention to her students and working alongside them, Kru Ann demonstrated a connectedness with all of her students--not just with the brightest students who usually shines in a traditional classroom setting. Kru Ann was aware of Chan as a talkative and sometimes disruptive student, but she didn't know what Chan is like as a person or even as a friend. Kru Ann remarked that she rarely heard Tila during regular class and that Dear often complained in class. By working alongside them, she found their hardworking qualities and witnessed their success in completing the project. Kru Ann trusted her students in their tasks and supported them along the way. This action reflects how Kru Ann gradually shifts towards culturally relevant teaching by holding high expectations and affirming views for all students. She believed that her students can succeed rather than believing that failure is inevitable (Ladson-Billings, 1995, p.25). Kru Ann's goal was to pull out her students' potential and to get the students to see a better version of themselves.

Moreover, the changes in how Kru Ann asked questions has an important implication. It shows that Kru Ann was more open to her students' personal answers, rather than trying to lead them towards her own answers. This notion aligns with culturally relevant teaching that encourages teachers to question the nature of the student-teacher relationship, the curriculum, and schooling. Kru Ann observed how the researchers asked the students questions in order to learn more about her students. As a teacher, she started to position her teaching as “pulling knowledge out” of her students like mining (Ladson-Billings, 1994, p.34). The transition in how Kru Ann framed her questions exemplifies how students can be sources and resources of knowledge and skills. She acknowledged, valued, and incorporated what her students knew into the classroom (Ladson-Billings, 1994).

Conclusion

By looking into how a teacher creates a constructionist learning environment, we can better facilitate and cultivate a learning community where teacher and students help one another, share their knowledge, and create meaningful projects together. We hope to inspire discussion on the importance of teacher engagement and teacher-student relationships, especially in low-income public school settings.

Through our analysis of Kru Ann’s teaching journey in her first constructionist class, we provide insights into what learning looks like for a teacher in a school-based constructionist learning environment. By working alongside her students rather than lecturing them, Kru Ann got to know her students in a novel way. She discovered the latent personality of her students and connected with them on a deeper level. Additionally, by reflecting upon her own teaching practices, Kru Ann is moving towards being a culturally relevant teacher.

Future Work

This study is a part of the wider two-year Little Builders design-based research project. In our future work, we aim to continue understanding the teacher’s experience in a constructionist learning environment. We also aim to explore students' experiences in formulating their identity as creators—building solutions for community challenges and seeing themselves as active contributors to their community. We will inform our study by using theoretical and pedagogical traditions such as constructionism, sociocultural views of learning, identity development, community-centered making, and culturally relevant pedagogy.

References

Blikstein, P. (2013). Digital Fabrication and “Making” in Education: The Democratization of Invention. FabLabs: Of Machines, Makers and Inventors, 1–21. https://doi.org/10.1080/10749039.2014.939762

Freire, P. (1996). Pedagogy of the oppressed (revised). New York: Continuum.

Ladson-Billings, G. (1994). The Dreamkeepers: Successful Teachers of African American Children. John Wiley & Sons. https://doi.org/10.2307/2077209

Ladson-Billings, G. (1995). Toward a Theory of Culturally Relevant Pedagogy. American Educational Research Journal, 32(3), 465–491. https://doi.org/10.3102/00028312032003465

Papert, S. (1980). Mindstorms. Current Directions in Psychological Science (Vol. 15). https://doi.org/10.1111/j.1467-8721.2006.00431.x

Papert, S. (1993). Seymour Papert The Children’s Machine- Rethinking School In The Age Of The Computer.pdf. Basicbooks.

Papert, S., & Harel, I. (1991). Situating Constructionism. Constructionism, 1–11.

Suziedelis, C. (2018). The Evolution of a Constructionist Teacher (with Reminders from Seymour Papert). Proceedings of Constructionism 2018. Vilnius, Lithuania.

Looking Beyond Supervised Classification and Image Recognition – Unsupervised Learning with Snap*!*

Tilman Michaeli, *tilman.michaeli@fau.de*
Computing Education Research Group, FAU Erlangen-Nürnberg, Martensstr. 3, 91058 Erlangen

Stefan Seegerer, *stefan.seegerer@fau.de*
Computing Education Research Group, FAU Erlangen-Nürnberg, Martensstr. 3, 91058 Erlangen

Sven Jatzlau, *sven.jatzlau@fau.de*
Computing Education Research Group, FAU Erlangen-Nürnberg, Martensstr. 3, 91058 Erlangen

Ralf Romeike, *ralf.romeike@fu-berlin.de*
Computing Education Research Group, FU Berlin, Königin-Luise-Str. 24-26, 14195 Berlin

Abstract

The possibilities of machine learning and its influence on our everyday lives are expanding rapidly. Computer science education must react to these developments. To this end, many new tools and approaches have been developed recently, particularly for novices. Whether "unplugged" or "plugged", most of these approaches focus on supervised learning as one of the three paradigms of machine learning – while the potential of the other two paradigms remains untapped. Furthermore, these approaches focus on mere usage of machine learning models by using external libraries for the actual learning. While learners are able to use machine learning in existing artifacts, they are not able to create their own or gain a deeper understanding of them. As the learning process itself is not addressed or visible for the learners, machine learning remains a black box. To address this problem, we developed SnAIp, a project aimed at enabling machine learning within the block-based language Snap*!*. We outline design principles guided by constructionist learning theory with the goal to break open the black box. Building upon these principles, we present a framework and curriculum for unsupervised learning in the classroom enabling learners to create and understand machine learning artifacts.

SnAIp Part B – unsupervised learning in Snap!

Keywords

AI, artificial intelligence, computer science education, Snap*!*, block-based, programming, machine learning

Introduction

Artificial intelligence (AI) is increasingly becoming the subject of attention due to its societal relevance in many sectors. Primarily driven by advances in machine learning (ML), it is now also discussed in pedagogical contexts (e.g. Touretzky et al., 2019; Heinze, Haase, & Higgins, 2010). Consequently, AI is also featured in international curricula (e.g. CSTA, 2017; Yu & Chen, 2018) with machine learning being a prominent aspect.

Novices are the target audience for many existing pedagogical frameworks. Typically, these frameworks show simple and intuitive ways for learners to use ML concepts – without the hurdles introduced by programming (Kahn et al., 2019). Many of these approaches share a common feature: they focus on supervised learning. Often, this results in the focus being shifted toward pattern recognition and supervised learning being equated to ML, while the potential offered by often-overlooked ML algorithms remains untapped. We argue that the potential of unsupervised learning (UL) is that it highlights the simplicity of ML algorithms particularly clearly (as ultimately all ML algorithms aim to make a simple machine appear intelligent). Furthermore, it helps in providing a broader view on how machines actually can "learn".

Another common trait of existing approaches is that the underlying AI framework remains a black box. In many cases, this is due to the actual computing being done by external API calls or servers - which severely limits the ability of the user to "look inside". This limitation makes it difficult for learners to gain a deeper understanding of the actual machine learning process. According to the constructionist learning theory (Papert & Harel, 1991), knowledge is constructed through the interaction of creating an artifact and understanding it. To allow constructionist learning in the context of ML, students need to go beyond the mere usage of ML; they also need to understand what they build. By incorporating both the creation of ML artifacts (rather than simply using them) and supporting their understanding, learners construct knowledge about ML. This deep, fundamental understanding is necessary if learners are expected to analyze the effects of AI on our society in a competent way.

In this paper, instead of focusing on supervised learning, we present approaches to another area of ML suitable for classrooms. The *SnAIp framework* for unsupervised learning is based on an implementation entirely in the Snap*!*-environment to enable constructionist learning by "breaking open the black box". After outlining the framework and its design principles, we also present parts of a corresponding curriculum to incorporate these topics into teaching.

Related Work

Paradigms of Machine Learning

The subject area of machine learning deals with algorithms that improve through experience over time (Mitchell, 1997). Machine Learning can be divided into the three paradigms *supervised*, *unsupervised* and *reinforcement learning*. Some authors further diversify this categorization, e.g. by adding *semi-supervised learning*, a combination of supervised and unsupervised approaches (Ayodele, 2010), or *evolutionary learning* (Marsland, 2014).

In supervised learning, the algorithm is given a set of sample inputs and corresponding outputs. It learns of the connection between input and output data independently and can then generalize it to any input data (Marsland, 2014).

The situation is different with unsupervised learning: the algorithm is also given a number of inputs, but there is no corresponding output, yet. Hence, the algorithm needs to identify similarities in data inputs in order to classify the data and assign an output label.

Reinforcement learning is a paradigm of machine learning inspired by psychology: The agent learns through reward and punishment. It learns to master a certain task autonomously through interaction with its environment by attempting to maximize the total reward. As is the case with unsupervised learning, the learning process is not supervised.

Machine Learning in the Classroom

Recently, numerous tools have been introduced to allow learners to use machine learning in programming. A large number of these tools use block-based languages.

The hands-on toolkit PopBots focuses on Pre-K and Kindergarten children (Williams et al., 2019). The toolkit utilizes the same blocks as ScratchJr, providing teaching materials for knowledge-based systems, supervised learning, and generative music. Due to the target audience, the toolkit material does not explore concepts of ML in-depth, or the underlying algorithms.

Cognimates is a framework that assists with building games, programming robots, and training supervised learning models (Druga, 2018). The framework focuses on social robots. The learning process, however, is not visualized to the user, instead taking place on servers.

Machine Learning for Kids (Lane, 2018) is an online platform that supports supervised learning, allowing the Computer to recognize text, images, sets of numbers, or sounds within Scratch projects. Users provide training data, however, as before, the learning process is done on external servers.

Ecraft2learn is a framework for AI programming in Snap*!* (Kahn et al. 2018). It provides a library with several AI-related blocks. Using these blocks, which are merely an interface for the underlying API, the user can experiment with a broad range of AI concepts within the Snap*!*-environment. Internally, APIs or server-based services handle the actual computation involved.

AlpacaML (Zimmermann-Niefield et al., 2019) is an app that allows users to collect data, create and test models with supervised machine learning. AlpacaML is intended to be used in sports. It utilizes the sensor data of a wearable computing device. After data collection, the user labels the data, e.g. good or bad passes in soccer and applies the model to new data.

An analysis of these existing approaches shows that most of these educational tools and approaches focus on supervised learning as one of the three paradigms of machine learning algorithms. This is typically done due to the paradigm's widespread distribution and its accessibility with common examples, such as image recognition. However, this limits the perceived scope of ML to supervised learning, and therefore to a very narrow view on (machine) learning. Furthermore, many of these frameworks do not provide the opportunity to explore how the actual learning processes take place. The machine learning remains hidden to the user, e.g. with an API call or a JavaScript function. As a result, ML remains a *magical mystery*, limiting constructionist learning.

Design Principles for Teaching ML with Block-Based Languages

To help students explore the underlying principles of ML, we previously developed a collection of activities and an underlying framework called SnAIp for Reinforcement Learning (Jatzlau et al., 2019). These activities make up Part A of SnAIp. After building a foundation of knowledge through unplugged activities, we explore the concept of reinforcement learning through an algorithm named Q-learning in Snap*!*. In order to enable constructionist learning in which learners reach understanding of ML, we introduced the following underlying design principles:

Looking behind the scenes: Instead of only using pre-trained models, applying libraries or making API calls, users should be able to look behind the scenes and be able to improve or make adaptations to the ML algorithms.

Everything inside the tool: Other systems utilize extra services to train models, or rely on background API calls. This tends to reinforce the notion of block-based languages being “learning languages”, and if the goal was instead to do “real programming”, they are not sufficient. To combat this misconception, we emphasize that everything should work from within the framework.

Useful projects over learning projects: When introducing difficult topics such as ML, we often choose "learning projects", artificial problems and examples, as these support the visualization of certain concepts. In this framework, however, we want learners to be able to create their own, personally-meaningful projects according to the constructionist learning theory (Harel & Papert, 1991).

Models as artifacts: Artifacts or *objects-to-think with* are central in contructionist learning (Ackermann, 2004). With block-based environments that allow users to store and save variables locally, models can become these tangible artifacts. This enables several new and interesting activities in the classroom, i. e. a competition for students to compare their models, discuss, and learn from each other.

In this paper, we present the part B of the SnAIp framework, which deals with unsupervised learning and is developed according to the same design principles.

SnAIp: Unsupervised Learning in Snap*!*

Unsupervised Learning and Linear Vector Quantization

Unsupervised learning is a paradigm of machine learning heavily influenced by statistics. Often, the task is to group or categorize unlabeled data, e.g. segmenting customers into groups with similar purchasing habits. This can be done to tailor ads to specific subgroups. An illustration of the underlying principle can be found in fig. 1.

For SnAIp, we use linear vector quantization (LVQ), an algorithm that finds such clusters in data sets. *Learning* takes the form of a fixed number of prototypes, which move in the multidimensional feature space determined by the different attributes of the underlying data. For each data point, the closest prototype (according to a chosen distance measurement) is selected and its coordinates are updated.

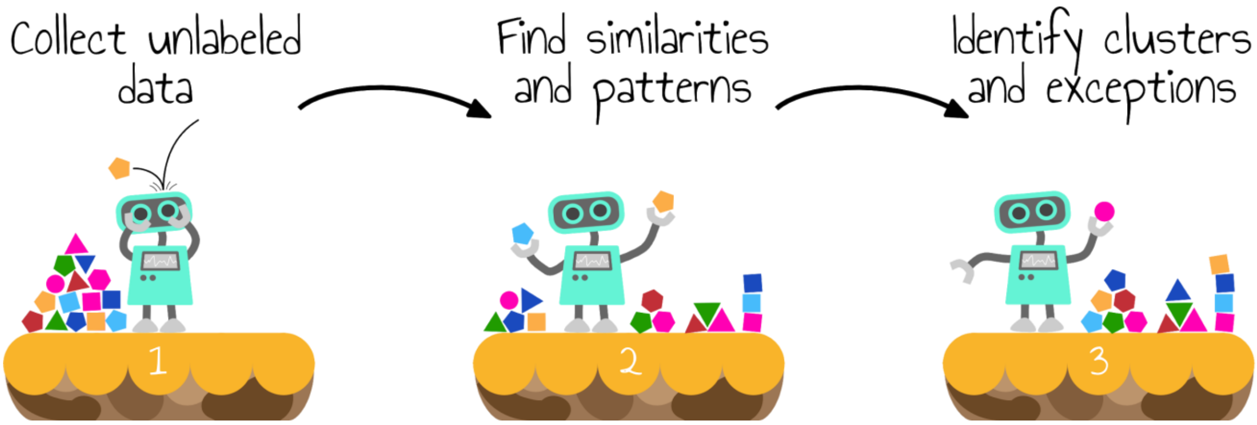


Figure 1. Unsupervised learning.

The SnAIp-framework

Using the visual programming language Snap*!*, SnAIp allows for the visualization of unlabeled data on a stage in two dimensions. Thus, students can visually explore the process of learning. Snap*!* as a tool and a programming environment provides a great deal of visualization and enables the implementation of complex projects. Compared to many similar programming environments and languages, it also supports efficient data processing. We chose Snap*!* for this framework in order to lower entry barriers to programming and to make ML graspable for students, following the ideals of the constructionist learning theory.

In the following section, we will illustrate how the design principles for SnAIp, which were laid out in the previous section, implement these goals for UL. With visualization being one of the primary factors, a number of blocks are required, which also aid the look behind the scenes.

This framework allows for experimenting with and applying unsupervised learning algorithms dealing with clustering in a two-dimensional space. To this end, we provide the following blocks that help with visualizing and processing two-dimensional data:

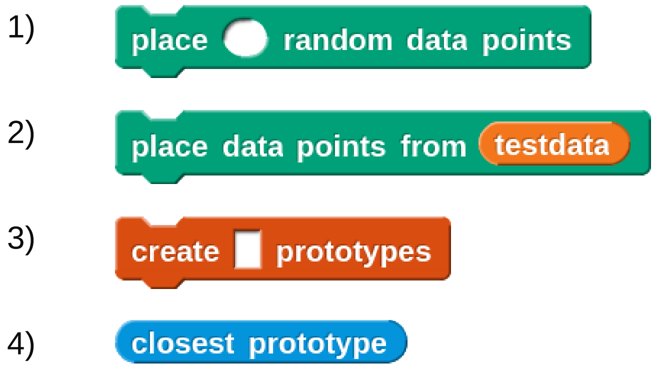


Figure 2. SnAIp framework for part B.

Initially, data points need to be created and, optionally, placed on the stage. For that, two blocks can be used (blocks 1 and 2 in figure 2). The first block creates a given number of **random** data points and clusters them on the stage. This is just a simple option to get started right away. For use cases that go beyond mere demonstration, the other block allows to use previously-imported data; After dragging-and-dropping data points from a **.csv file** into the Snap*!*-environment, this block normalizes out the **imported** data points (called "testdata" here) and spreads them out on the stage.

Both of the place-blocks are used for *visualization*. If data points do not need to be visualized on the stage, neither of the two blocks are necessary, as the algorithm relies exclusively on the prototype-blocks.

In the next step, prototypes need to be created. For this, block 3 in figure 2 is used. It creates a given number of prototypes and spreads them out randomly on a two-dimensional canvas (i. e. the stage in Snap*!* is used to visualize the data points, the prototypes, and the algorithm itself).

Finally, the closest prototype of each data point is determined and performs some type of action. The corresponding block (block 4 in figure 2) reports the closest prototype to a quantitative data point in an *n*-dimensional space and is used to determine which prototype needs to be moved.

SnAIp in the Classroom

In the following section, we will elaborate on our approach on how to introduce unsupervised learning in the classroom. The curriculum consists of three parts and is intended to be used in upper high school classrooms (ages 14 and older). Similar to part A, we start with an unplugged activity to introduce the concept of unsupervised learning in an active and motivating way. This is followed by an implementation in Snap*!*, in which the learners construct and optimize the unsupervised learning algorithm to solve problems from the ground up.

a) What is unsupervised learning

As unsupervised learning is mostly used for clustering, the introductory unplugged activity is about finding clusters in a set of two-dimensional data. In a wild west setting, the students' task is to determine the best possible digging site for three excavation teams. A good digging site is characterized as being as close as possible to many gold sources. After being introduced to the setting of the gold rush, students are given a map of the wild west (see fig. 3). They are also given 25 data points in the form of playing cards, consisting of *x* and *y* coordinates of a gold source.

Furthermore, they are provided 3 coins that serve as *prototypes*, visualizing the excavation teams. The students continuously draw data point cards, consider how to process the information they contain, and put them face-down on the discard pile. Therefore, they only see one datapoint at a time, the same way a computer would, and never "see all the data at once". As they have no way of marking the different data points, students have to use and move the coins in reaction to each new datapoint. They will come up with an algorithm conceptually similar to the following LVQ-variation:

1. Place the prototypes (coins) on the map (randomly).
2. For each datapoint:
    a. Identify the prototype closest to the data point
    b. Update the prototype's position by moving it half the distance toward the coordinates of the data point

Figure 3. Unplugged activity used for the introduction of unsupervised learning.

After processing all the data, students are provided with a projector slide containing all the data points, which they can lay over their map to evaluate their final results. With the help of this slide, they can then improve their algorithm. Based on the experience the students have made in this activity, principles and characteristics (see figure 1) of unsupervised learning can be discussed and compared to other machine learning paradigms.

b) Unsupervised learning in Snap!

In the next step, the previously-developed LVQ-algorithm is implemented in Snap*!*. Students receive a template that provides prototypes, and a dynamic number of randomly-created data points, both of which are already visualized on the stage. The task is to complete the actual LVQ-algorithm (see fig. 4). In doing so, students can experiment with the various parameters, such as number of prototypes, data points, etc.

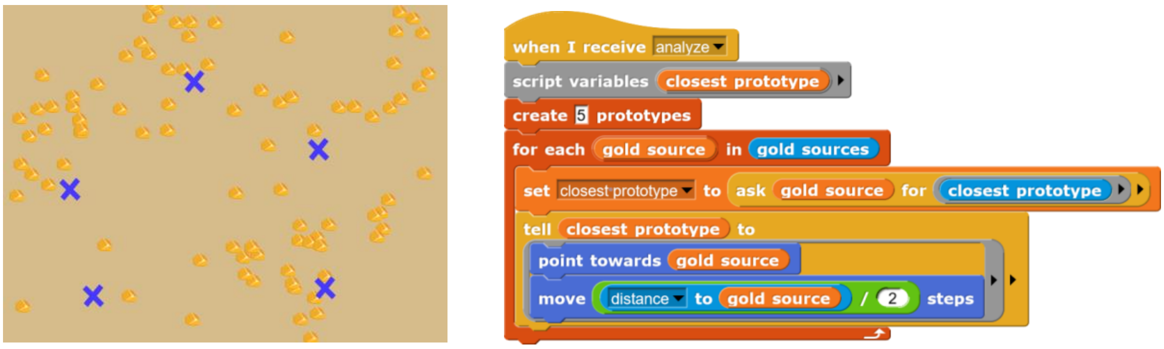


Figure 4. Gold rush clustering with unsupervised learning in Snap!

Rather than being limited to working with random data, the framework can also be applied to real-world data (which could even be gathered beforehand by/with the students), in the sense of constructionist learning theory. The framework enables the processing and clustering of real-world data by working with csv imports. One possible application of this might be to cluster residential locations to determine ideal sites for bus stops (the two dimensions in this case being longitude and latitude). In general, this algorithm can be applied to any data set. Categorical data, however, must be transformed into numerical inputs before.

c) Competition

Afterward, students are asked with optimizing the algorithm in a competition. As the algorithm presented here is only a very simplified approach, there is room for improvement and optimization. There are multiple possible approaches toward this, such as:

- Using multiple iterations to refine the final prototype locations
- Decreasing the movement range for each prototype gradually
- Increasing the priority of prototypes that have not been moved
- Optimizing the initial position of the prototypes

Similar optimisations are typically applied to unsupervised learning algorithms as well. By providing data sets that can be imported via Snap*!* and visualized with the frameworks' respective block, students' solutions can be compared and discussed in the classroom.

Discussion and Conclusion

An increased interest in machine learning creates the need for concepts, curricula and teaching material that make it possible to incorporate the different paradigms of ML into the classroom. Our approach incorporates both the creation of ML artifacts and supporting their understanding. Existing approaches fall short on providing a way to create a fundamental understanding of ML in order to construct their own knowledge, or focus on a single paradigm, supervised learning. Although many real-world applications of ML refer to supervised learning, there is a growing number of applications of unsupervised and reinforcement learning. Thus, students will interact with or experience these paradigms in apps or services they use.

The framework we outlined in this paper utilizes the potential provided by UL and fulfills our four design principles:

1. **Looking behind the scenes**: Learners are tasked with implementing the entire learning algorithm directly in Snap! itself with very little guidance. Constructing the algorithm from the ground up enables a unique look behind the scenes for learners.

2. **Everything inside the tool**: All the algorithms are implemented directly in Snap!, making it easy for learners to make adaptations and optimizations to the algorithm.

3. **Useful projects over learning projects**: According to the constructionist learning theory, learners are able to create their own, personally-meaningful projects by using and adapting the algorithms to fit their own projects. As an example, when working on an image compression project, users can identify patterns in their own csv-data and apply the concepts they learned.

4. **Models as artifacts**: With the ability to export data structures as files, learners create their own artifacts and use them for various activities, such as the competition outlined above.

In summary, in this paper we present how to look beyond supervised classification and image recognition with students in secondary education. To this end, we introduce a framework to explore machine learning in a constructionist and transparent way. Furthermore, we outline a corresponding curriculum to address unsupervised learning in the classroom. This enables learners to create and understand ML artifacts.

Acknowledgements

We want to thank Urs Lautebach for contributing the idea for the chosen unsupervised learning algorithm and unplugged activity and the Königstein group for developing the idea further.

References

Ackermann, E. (2004) Constructing Knowledge And Transforming The World, A learning zone of one's own: Sharing representations and flow in collaborative learning environments [M. Tokoro and L.Steels (Eds.). Amsterdam, Berlin, Oxford, Tokyo, Washington, DC. IOS Press. 1, 2 (pp. 15–37).

Ayodele, T. O. (2010). Types of machine learning algorithms. In: Y. Zhang. New advances in machine learning. IntechOpen.

CSTA (2017). About the CSTA K-12 computer science standards, https://www.csteachers.org/page/standards, last accessed 02/06/2020.

Druga, S. (2018). Growing up with AI: Cognimates: from coding to teaching machines (PhD thesis), Massachusetts Institute of Technology, Cambridge, USA.

Heinze, C., Haase, J. & Higgins, H. (2010). An action research report from a multi-year approach to teaching artificial intelligence at the k-6 level. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10). Atlanta, GA USA: AAAI Press (pp. 1890–1895).

Jatzlau, S., Michaeli, T., Seegerer, S., & Romeike, R. (2019). It's not Magic After All-Machine Learning in Snap! using Reinforcement Learning. In 2019 IEEE Blocks and Beyond Workshop (B&B). Memphis, USA TN: IEEE (pp. 37–41).

Kahn, K. Megasari, R., Piantari, E. & Junaeti, E. (2018). AI Programming by Children using Snap! Block Programming in a Developing Country. In EC-TEL Practitioner Proceedings 2018: 13th European Conference On Technology Enhanced Learning.

Lane, D. (2018). Explaining artificial intelligence. Hello World, 4, https://helloworld.raspberrypi.org/issues/4, last accessed 02/06/2020.

Marsland, S. (2015). Machine learning: an algorithmic perspective. Boca Raton, USA FL : CRC Press Taylor & Francis Group.

Mitchell, T. (1997) Machine Learning, New York, NY USA: McGraw-Hill Education Ltd.

Papert, S., & Harel, I. (1991). Situating constructionism. In: I. Harel, & S. Papert. Constructionism (pp. 1–13). Norwood, NJ USA: Ablex Pub. Corp.

Touretzky, D., Gardner-McCune, C., Martin, F. & Seehorn, D. (2019). Envisioning AI for K-12: What should every child know about AI? In "Blue sky talk" at the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19).

Watkins, C. J., & Dayan, P. (1992). Q-learning. Machine learning, 8(3-4). Boston, MA USA: Kluwer Academic Publishers (pp. 279–292).

Williams, R., Park, H. W., Oh, L., & Breazeal, C. (2019). Popbots: Designing an artificial intelligence curriculum for early childhood education. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, pp. 9729-9736).

Yu, Y., & Chen, Y. (2018). Design and development of high school artificial intelligence textbook based on computational thinking. Open Access Library Journal, 5(09):1.

Zimmermann-Niefield, A., Turner, M., Murphy, B., Kane, S. K., & Shapiro, R. B. (2019). Youth Learning Machine Learning through Building Models of Athletic Moves. In Proceedings of the 18th ACM International Conference on Interaction Design and Children, ACM (pp. 121-132).

Microworld for learning programming friendly to blind students

Lucia Budinská, Ľudmila Jašková, Michal Kováč, Karolína Miková, *,*
lucia.budinska@fmph.uniba.sk, ludmila.jaskova@fmph.uniba.sk, kovac254@uniba.sk, karolina.mikova@uniba.sk
Dept of Education, Comenius University, Bratislava, Slovak Republic

Mária Karasová, *maria.karasova@bee.sk*
Elementary School for Visually Impaired, Bratislava, Slovak Republic

Abstract

In this paper we present a part of our research aimed at development of algorithmic thinking and programming skills of blind students attending lower secondary schools. We have introduced a programming environment Coshi for blind students where they can control a virtual robot in a square grid with assigned sounds (Figure 1). It is possible to change an audio layout and thus create a new microworld with different motivation

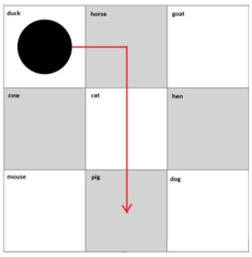


Figure 1. The virtual robot (black circle) moves in the grid with the animal sounds assigned to the fields

During verification of this environment, an educational scenario was created for six lessons. The aim of these lessons was to make students gradually acquainted with the programming environment and basic programming concepts (sequence of commands, counting loop, variable, conditional branching, conditional loop). The testing took place in special school for students with visual impairment with four practically blind students in eighth grade.

Throughout the testing, we also examined how students construct their knowledge of programming concepts. We posed two research questions, Q1: *How do observed blind pupils proceed in solving our research scenarios in Coshi environment?* and Q2: *How can observed blind pupils use conditional and counting loops to solve more complex tasks in Coshi environment?* We had used case study research strategy, we observed students during lessons, interviewed them and their teacher too, and gained a lot of field notes.

The students reacted positively to the Coshi environment, they used all functionalities in it and gradually used more difficult programming concepts. We observed that students solved the most difficult problems in four phases: (1) Task solving in the direct mode, (2) Creating a case-specific solution using basic commands, (3) Replacing sequences of repeating commands with loops, (4) Solving the problem in general using nested loops.

Keywords

programming, blind students, lower secondary education, case study research, programming concepts

Introduction

In recent years, many countries have been directing their efforts towards introducing computer science to pupils at all levels of education in the appropriate volume and form. Hromkovič et al. (2016) consider "*algorithmic thinking as one of the core concepts of computer science. It has proven a versatile and indispensable tool for problem solving and found applications far beyond science. Hence, sustainable computer science education should be built upon algorithmic thinking as its primary objective, thus unfolding benefits for a broad and general education*".

We consider developing algorithmic thinking to be very important for blind students as well. People with vision loss[29] have much less of a chance in finding a suitable job because of their impairment. However, if they have computer skills, they have a better chance of being employed. It is a great advantage if besides using applications, they have programming skills. We know several blind programmers who work successfully in software companies. We are convinced that the sooner the interest of blind learners in this profession is appropriately captured, the easier it will be for them to obtain the necessary qualifications.

The development of algorithmic thinking is also important for blind students who choose to pursue a career outside the IT field. The ability to follow the instructions, or to break down the problem into subproblems and to describe the procedure for solving them, are important skills necessary for learning, working and living in a society.

According to Papert, (1999) "*students learn more effectively if they can solve their own tasks, building their knowledge themselves and expressing ideas through a medium that allows direct experience*". The most popular and widespread programming environments Scratch and Logo allow students to learn programming in a constructivist way in the sense of Papert, but they are not accessible to blind students (Chapter 2).

We have developed a programming environment for blind children to control a virtual robot on the computer screen (Chapter 3). During its verification, an educational scenario was created for six lessons (Chapter 4.2). The aim of these lessons was to make students gradually acquainted with the programming environment and basic programming concepts. In addition to testing the environment and educational scenario, we examined how students construct their knowledge of new programming concept – conditional loop and how they can use it to solve more complex tasks (Chapter 4).

Related work

Teaching algorithms and problem solving in a standard class with sighted students is usually done via software that heavily leans on visual representations – Logo (Blaho, Kalaš, 2004), Scratch (Resnick, 2009) and others. These environments allow the manipulation of virtual objects with a wide variety of commands. But these educational software products are not usable if we teach blind students – who work with computers using a keyboard for input, a screen reader for output and the only information they can work with is text and sound.

Block-based languages like Scratch are very popular, because these environments remove the syntax complexities and they can be a good choice for an introduction to programming. Milne (2018) documented the accessibility challenges of block-based languages and developed *Blocks4All* prototype environment as a more inclusive alternative for Apple iPad environment. Unfortunately, the iPad is not widely used in Slovak schools yet because of the high price.

We considered the use of audio programming language (APL) for the blind developed in Chile (Sánchéz, 2005, 2006). The pilot testing (conducted by the Sánchez and his colleagues) showed that programming in this language was too abstract even for older learners (aged 17 to 20) and

[29] https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

therefore unsuitable for the younger learners in our group. Nevertheless, it inspired us to create a simpler audio programming language *Alan* (Kováč, 2016) for younger students (see Figure 2).

```
Say(What animal makes this sound?)
Repeat 3 times
  Play(donkey)
End of Repeat
Question(Is that a donkey?) Answer: Yes
  Say(Excellent!)
Else
  Say(No, it is a donkey)
End of Question
```

Figure 2. Program in Alan environment (translated into English)

Stefik with his team (2011) developed an auditory programming environment called *Sod-beans* with a programming language called *Hop*. They also developed and verified multi-sensory (sound and touch) curriculum to make text-based programming easier and more accessible for people with visual impairments. But it was designed for older students as well and it is not usable for young children who may still be developing literacy and typing skills.

Horn and his colleagues (2007) describe two tangible programming languages *Tern* and *Quetzal*. Tern language allows children to create a program to control a virtual robot Karel on a computer screen and the language Quetzal to control a LEGO Mindstorms™ robot. The authors created these tangible languages for sighted children to facilitate their collaborative development of the programs in the group. For blind children it is problematic to program the robot Karel because they cannot follow his movement on the screen. We found the Quetzal language to be suitable for our blind and partially sighted students. They could create a program by hand and check the robot executing the program by touch. Unfortunately, both languages are not commercially available yet, because their use has been restricted almost entirely to laboratory and research settings. A more suitable solution for blind children is the physical programming language *Torino* (Morrison, 2018), (see Figure 3).

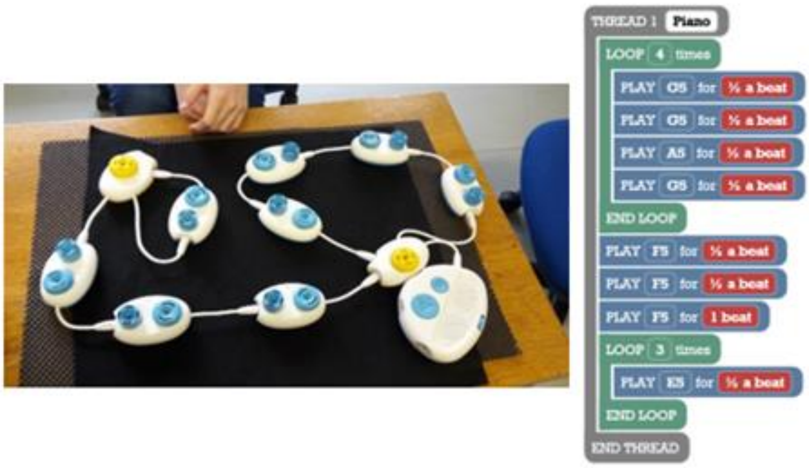


Figure 3. Program with two loops in Torino language (physical and text version of program)

Bee-Bot toy is by default very well designed and has many features that enable the toy to be used by totally blind. They were successfully used to teach basic algorithms to blind beginners that have no previous experience with programming (Kabátová, 2012), (Anonymous, 2013). We believe that robotic toys BlueBot[30] and Kibo[31] could also be adapted for blind students.

We have used text-based audio programming language Alan and physical programming language Torino (Anonymous, 2019). They both enable blind young students – novices in programming, to program simple melodies, stories or rhymes. We have found that they are suitable and extremely engaging for these students. The environments contribute to the development of children's creativity. Students can learn all the basic programming concepts without having to imagine moving an object on the screen. However, it is important for the profession of programmer to be familiar with the computer screen. Therefore, the Coshi environment (Kováč, 2019) was created, which we will describe in the next chapter.

During the verification of this environment with blind students, we wondered how our students construct their understanding of new programming concept – conditional loop. We were also wondering if they could abstract from a assignment of a task and create a solution that would fit a wider set of principally similar tasks.

Programming environment Coshi

Programming environment Coshi enables user to program robot motion in an on-screen square grid with assigned sounds. Sounds, their placement and also the size of the grid can be adjusted for each task. The programming language used in Coshi is easy with basic absolute direction motion commands – Up, Down, Right, Left. These commands move the robot in a chosen direction. There are three commands aimed at sound in Coshi – Play, Silent, Loud. The command Play plays the sound assigned to square robot is on. The command Silent turns on the silent mode, in which robot moves without playing sounds. The command Loud turns off the silent mode, so robot moves are audible. In Coshi, there are also some basic programming concepts – namely a counting cycle (FOR cycle), a conditional cycle (WHILE cycle), full conditional branching (IF-THEN-ELSE), a procedure without or with one parameter, and a number variable.

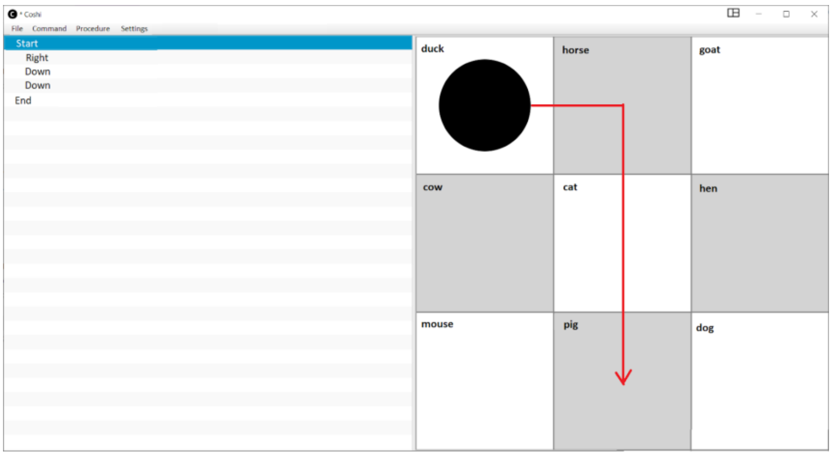


Figure 4. Application Coshi, playing area with sound layout Animals

The application works with a screen reader and there's an audio response for every action. Screen reader is able to interpret not only all menu items, but also written program. Each square in play grid has sound assigned to it and this sound is played when robot enters the square. The feedback for blind student, when they run their program, is a sequence of sounds that plays as robot moves through the grid. The names of the sounds can be visible in grid, as it helps teachers and sighted students. Before writing a program, blind students can explore the grid using arrows on their

[30] https://www.terrapinlogo.com/bluebot.html
[31] https://kinderlabrobotics.com/

keyboard – robot is moving, and the sounds are played (so called direct mode). Another help is that students can choose to have the coordinates of each square read out loud too (numbers of row and column).

Programming environment was built for blind novice programming students (lower secondary level), therefore it has a block-based features – the user does not have to type the commands but can select them from a list. An advanced user can use keyboard shortcuts to insert commands.

In the (Figure 4) there is a window of Coshi environment with 3x3 grid. Each square of the grid is assigned a sound of an animal. In the left side of the window there is a program. When running the program the robot is shown as a black circle which moves through squares and plays sounds of a horse, a cat and a pig in order.

There are several audio layouts built into the environment, but the editor for creating additional layouts is available as a stand-alone application. Using this editor, teacher can associate different sounds with grid, in different words they can create microworlds in which students solve different tasks to reach their educational goal.

Students can also be a part of microworlds creation – they can decide which sounds to use or what types of tasks they like. Therefore, Coshi enables to reach different educational goals. The main goal is the **algorithmic thinking development** and learning and mastering basic programming concepts (such as sequences of commands, cycles, conditions, procedures and variables). This is interconnected with mastering debugging – looking for an error in programs and fixing them. Besides algorithmic and programming skills students can also master different skills. For example:

- **Musical creativity development:** If each square in the grid is assigned a tone of the scale, pupils can create different melodies by programming the robot movement.
- **Orientation skills development** important in working with tables or squares grids. For example, if each square represents one house and each has a sound of its address, pupils can program a robot-postman movement from house to house.
- **Logical thinking and problem solving skills development**. For example, if each square is assigned some letter sound, pupils can program the robot movement in an alphabetical order (see Figure 5).

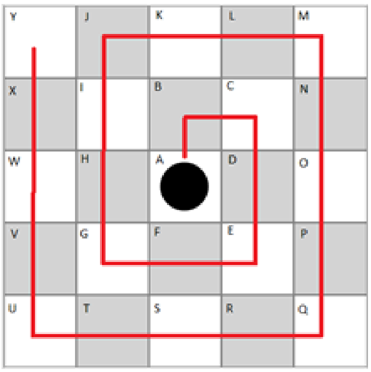


Figure 5. Grid with sound layout – the task is to go through the grid in alphabetical order of sounds

Methodology

Aims, methods and research sample

In our research, we have explored how blind pupils construct their knowledge of programming concepts. In this paper we try to find an answer to the research questions Q1 and Q2.

Q1: How do observed blind pupils proceed in solving our research scenarios in Coshi environment?

Q2: How can observed blind pupils use conditional and counting loops to solve more complex tasks in Coshi environment?

We used a case study qualitative research strategy. The observed cases were tasks solved by blind pupils. We had chosen the following research methods to collect and analyse data (Hendl, 2005), (Cresswell, 2008).

- Observation of students solving the tasks at computer science lessons.
- Semi-structured interviews with students and their informatics teacher.
- Analysis of our text notes done during computer science lessons.

Our research sample involved four visually impaired students – eighth graders. These students were aged 14-16 (two boys and two girls). They all attended the same class, they know each other well and they were used to working in the group. The observed students used a screen reader to work with the computer. One student with severe low vision also used magnification software. Students' computer skills were slightly above average, as we will explain below. All involved students had already used a text editor, a sound editor, a spreadsheet editor, an email client and a web browser. They have already learned programming in Alan programming environment and with Torino physical programming language. In these environments they created programs consisting of sequences of commands and counting loops. They also solved unplugged activities aimed at programming the movement of the object over a square grid. They have not used the conditional loop in any programming language yet.

During research each student worked on an educational scenario on a computer with headphones, a screen reader and Coshi environment with prepared tasks.

Educational scenario

We developed an educational scenario for six lessons, each consisting of a set of tasks with graded cognitive difficulty. Each lesson had its own motivation and used mostly the same sounds. Pupils solved tasks individually, but we discussed the solutions with them either in the group or individually. The aim of the discussions was not to disclose the solution to the students, but to bring them to the solution by asking questions.

The aim of the **first lesson** was to introduce the Coshi programming environment and the basic language commands. We used the Animals audio layout (Figure 4). We told students a story about a robot Coshi feeding animals on a farm. They were supposed to program the movement of the robot between the animals, using the basic commands left, right, up, down. During the **second lesson**, the students solved tasks aimed at using the counting loop. They should program the robot to move several times in the same direction. The **third lesson** was focused on tasks aimed at using the conditional loop. Students should program the robot to move to the end of a line or a column. The program was supposed to work for different square grid dimensions. The **fourth lesson** was focused on tasks aimed at using variables for counting sounds (e.g. dog barking) while the robot was moving. The **fifth lesson** was focused on tasks aimed at using the conditional command. Students should program the robot to count the number of free fields around him or they program the robot to decide if he can move (the next field is free). At **sixth lesson** students solved more complex tasks aimed at using nested programming structures.

Observations

Based on our observation, we can say that students worked in Coshi environment with enthusiasm. They liked the given tasks and invented some of their modifications. They were impatiently looking for new commands in the menu and asked us about their meaning and usage. They have also been curious about the audio layouts in menu. In the group discussion, students told us that the program was like an adventure game of the type - we move around different rooms and in each of them we deal with different, graded challenges. However, they would expect that their score would increase after each task is resolved, and they can continue to the next room. Teachers of various subjects informed us that our blind students were enthusiastically talking about the Coshi program outside of computer science lessons. This indicates a positive acceptance of the program. Regarding the user interface, students actively used keyboard shortcuts implemented in Coshi and editing the program did not cause them any major problems.

The students had no problems in solving tasks where it was necessary to create sequences of commands without loops (lesson 1). When students were creating a program containing a sequence of the same commands (lesson 2), they didn't use a counting loop, although they had already used this concept in Alan and Torino environments. They all started using it only when one student accidentally discovered this command in the menu. They recalled that replacing a repetitive sequence of commands with a counting loop is a good idea and then they used this construction without problems.

We noticed problems only when students were solving tasks aimed at using the conditional loop (lesson 3). The task was to program the robot to come to the end of a row or column and then stop. Students first counted the squares that the robot was supposed to go through and used a counting loop. Then we asked the students if it would work for a larger playing area. Students thought about how to change the program to make it work with an area of any size. In the menu, they looked for a command they could use and immediately noticed the While command. They created the program listed in Figure 6 on the left.

Infinite loop	**Correct program**

`while it is free to the right`
 	
`end of while`
	`while it is free to the right`
	`    right`
	`end of while`

Figure 6. Student programs containing a conditional loop

They were surprised that nothing happened on the screen, and they had to terminate the program with a special keyboard shortcut. We explained to students that they created an endless loop because the body of the cycle is empty, the robot does not move, so it never gets to the end of line. Students then added the command right to the loop body (Table 2 on the right). During the subsequent discussion with the students, we discovered that they understood the phrase **to the right** in the condition as a command to move to the right and not as verifying that robot could move to the right. Other principally similar tasks were solved by students without problems.

Lesson 4 aimed at using variables was not problematic for students. In this lesson students learned to define and set the variable in the beginning of the program and to add to it after something happened – the robot moved, a dog barked etc. To know what is in the variable, students need to display it.

The conditional branching (lesson 5) was new to the students. In the Coshi environment the conditional branching is implemented as full branching, therefore it has to have both branches - IF and ELSE even if ELSE branch is not needed. In that case one branch is empty. This implementation has shown as problematic because students could not easily orientate in conditional command structure and they were not sure where they should put their commands.

The task **Route programming** (lesson 6) was a real challenge for students. Fields of the playing area have been assigned words corresponding to the commands to be executed by the robot. This has determined the route for the robot to follow in order to get from the top left corner to the target field with the **Yes!** sound assigned (Figure 7). Students firstly used arrows keys to move robot in grid based on sound commands and they created program as the next step.

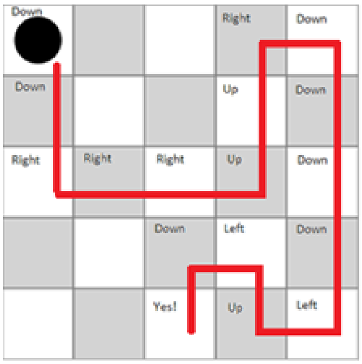


Figure 7. Playing area for the Route programming task

Figure 8 contains three different solutions to this task. All students first created a program without loops (Figure 8, first column). Reminding students that they can use the loop command, they replaced the sequences of the same commands with loops (Figure 8, second column). Some students used the counting loop and some the conditional loop. Then we changed the sounds on the playing area to create a different route. We asked the students whether the program would work in this case as well. They knew immediately that it would not work. After a brief discussion, students acknowledged that it would be useful to create a program that would work for any route. But they did not know what the program would look like. We asked one of them to play the robot and others to control him by commands. We asked students the following questions: How does he find out that he is not in the target field yet? What does he need to find out if he is not in the target field? What does he do when he finds out what direction to move? The students gradually described verbally how the program would work in natural language. Then they wrote it in the robot language – the program listed in the third column of Figure 8. We had created this task with conditional branching in mind – we had thought that students would use conditions and not nested conditional loops. But students did not come up with this solution, maybe due to complicated syntax of the conditional branch in Coshi.

Without loop	**With sequences of loops**	**With nested loops**

`down`
`down`
`right`
`right`
`right`
`up`
`up`
`right`
`down`
`down`
`down`
`down`
…
	`while the sound is down`
	`  down`
	`end of while`
	`while the sound is right`
	`  right`
	`end of while`
	…
		`while the sound is not yes`
		`  while the sound is right`
		`    right`
		`  end of while`
		`  while the sound is down`
		`    down`
		`  end of while`
		`  while the sound is left`
		`    left`
		`  end of while`
		`  while the sound is up`
		`    up`
		`  end of while`
		`end of while`

Figure 8. Three solutions for the Route programming task

We observed that students solved the problem in four phases:

1. Task solving in the direct mode (using keyboard to move the robot).
2. Creating a case-specific solution using basic commands.
3. Replacing sequences of repeating commands with loops.
4. Solving the problem in general using nested loops.

Conclusions and discussion

Firstly, we would like to state that we are aware of the biggest issue of our research and that is a small research sample. There are not so many blind students of this age in Slovakia. Although we work with a special school for visually impaired students, there could be some more blind students integrated to special or normal schools around the country. But there are no public reports where we could learn more about them or get in touch with them. There is, obviously, no other way to enlarge Slovak community of blind students and with it our research sample. Therefore, we are trying to create some collaboration with Czech Republic.

The drawback of our research is not only a small sample but also the fact that we have used only one educational scenario, therefore we cannot generalize our findings. We see this as an opportunity to continue in this research with new students (who are currently in lower grades) and with edited tasks – to more accurately reflect the cognitive load of programming tasks to blind students.

Despite of this, there stem some findings important for future research. Students in our research sample used a sequence of commands more than a cycle, which is caused by the fact that they solve each task as a one partial problem. Thinking about general solution is very abstract for them. To help them find the general solution, the problem needs to be divided into several subproblems, as it was also seen in (Anonymous, 2019). In our edited tasks we would like to find a good motivation to help students see problems more generally.

Students also used conditional cycles more than conditional branching. This could relate to the problematic syntax of conditional branching in the Coshi, but it could also be caused by some misconception. In (Mühling et al., 2015), while measuring the basic programming abilities of students in 7th to 10th grade, they found out that the students often thought that “the conditional statement is actually a repetition that they are executing until the condition is false”. This was probably caused by the fact that students were not familiar with a stand-alone conditional branching, but they were used to it in a cycle. Our students used correct structure for this interpretation, but it is possible that they did not see a difference and so they chose the easier one regarding the syntax.

Generally, we can say that the Coshi environment has been proven as appropriate for blind students. It has a good sound response and the creation of a program using commands from a menu or using the keyboard shortcuts was suitable for students. The commands were understandable, but it is important to implement a new command – conditional statement without the ELSE branch. The possibility to create own sound layouts is a good way to adjust motivation for each student. However, the audio layout editor is not yet fully accessible to blind users. We plan to make it accessible in the future. There are more educational activities we are planning to test, some of which are already prepared – these are aimed at learning procedures without or with parameter.

Acknowledgment

This paper was created as part of a project funded by KEGA grant (Kega 018UK-4/2019). Special thanks to the teachers and students of the elementary school for students with visual impairment where we conducted our research.

References

Hromkovič, J. et al. (2016) *Examples of algorithmic thinking in programming education.* Olympiads in Informatics, *10* (1-2), 111-124.

Bell, E. C. et al. (2015) *Employment Outcomes for Blind and Visually Impaired Adults*. In Journal of Blindness Innovation and Research, Vol 5, No 2, National Federation of the Blind.

Papert, S. (1999) *What is Logo? And who needs it?* Logo philosophy and ımplementation (pp. iv–xvi). Logo Computer Systems.

Blaho, A. and Kalaš, I. (2004) *Imagine Logo Primary workbook*. Logotron, Cambridge.

Resnick, M. et. al. (2009) *Scratch: Programming for all.* Commun. Acm, 52(11), 60-67.

Milne, L. R. et al. (2018, April) *Blocks4All: overcoming accessibility barriers to blocks programming for children with visual impairments.* In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (p. 69). ACM.

Sánchez, J. et al. (2006, July) *APL: audio programming language for blind learners*. In International Conference on Computers for Handicapped Persons (pp. 1334-1341). Springer, Berlin, Heidelberg.

Sánchez, J. et al. (2005, April) *Blind learners programming through audio.* In CHI'05 extended abstracts on Human factors in computing systems (pp. 1769-1772). ACM.

Kováč, M. (2016) *Programming environment for visually impaired pupils* (in Slovak), bachelor thesis, Comenius University, Bratislava.

Stefik, A. M. et al. (2011) *On the design of an educational infrastructure for the blind and visually impaired in computer science.* In Proceedings of the 42nd ACM technical symposium on Computer science education (pp. 571-576). ACM.

Horn, M. S. et al. (2007) *Designing tangible programming languages for classroom use.* In Proceedings of the 1st international conference on Tangible and embedded interaction (pp. 159-162). ACM.

Morrison, C. et al. (2018) Torino: *A tangible programming language inclusive of children with visual disabilities.* Human–Computer Interaction, (pp. 1-49).

Kabátová, M. et al. (2012) *Robotic activities for visually impaired secondary school children.* In 3rd International Workshop, Teaching Robotics, Teaching with Robotics (pp. 22-31).

Jašková, Ľ. (2013) *Blind Pupils Begin to Solve Algorithmic Problems*. In Proceedings of the 6th International Conference on Informatics in Schools - Situation, Evaluation and Perspectives, LNCS 7780, (pp. 68—79). Springer, Heidelberg.

Kováč, M. (2019) *Programming environment accessible to blind secondary school students for programming the movement of virtual object* (in Slovak), master thesis, Comenius University, Bratislava.

Jašková, Ľ. et al. (2019) *Two programming languages friendly to lower secondary blind students.* In Proceedings of the 12th International Conference on Informatics in Schools - Situation, Evaluation and Perspectives (pp. 13-21). University of Cyprus, Larnaca.

Hendl, J. (2005) *Qualitative Research. Basic Methods and Applications*. (translated into Slovak) Portál, Prague.

Creswell, J. (2008) *Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research*. Pearson Education, New Jersey.

Mühling, A. et al. (2015) *Design and First Results of a Psychometric Test for Measuring Basic Programming Abilities.* In Proceedings of the Workshop in Primary and Secondary Computing Education (pp. 2-10). ACM.

MoKraRoSA: A Constructionist Platform for All Ages and Talents

Pavel Petrovič, *pavel.petrovic@fmph.uniba.sk*
Dept of Applied Informatics, Comenius University, Bratislava, Slovakia

Jozef Vaško, *jozef.vasko@fablab.sk*
Fablab Bratislava, Slovak Centre for Scientific and Technical Information, Bratislava, Slovakia

Abstract

In the summer 2019, we organized three summer camps: a two weeks long summer camp of IT Akademia national project at Krahule in Slovakia, and two weeklong daycamps in FabLab Bratislava. For this purpose, we have designed an original versatile modular educational robotic platform MoKraRoSa. Children have built 80 such robots in the camps, participated in the developments of software, parts, attached various electronics components, designed their own choreographies, and later used the robot in other projects. We introduce the platform, share our goals, results and experiences from the summer camps.

Figure 1. Participants of summer camp of IT Akademia at Krahule building their robots

MoKraRoSA consists of 3D printed modules that could be connected in different ways to form various robot morphologies. It is controlled by a simple Arduino, communicates wirelessly over BlueTooth connection, and it is accessible to learners with any skill levels. The robots built by participants can walk, roll, and dance. In addition to robot building and programming, the participants learned about 3D modelling, Linux, and film making. The platform is completely open-source and available at Github.

Keywords

educational robotics, robotics platform, Arduino, legged robot, summer camp

Introduction

According to the register at fablabs.io, the renowned network of Fablabs currently consists of almost 5000 labs around the world. One of them, FabLab Bratislava, is open for general public daily. It has been providing its equipment, space, and expertise to pupils, students, groups, and others for more than 5 years. Thousands of projects of various proportions have been realized. Three informatics-related faculties located in its vicinity have been offering creative digital technology courses to their students in cooperation with Fablab for at least the past two years. An organic group of volunteers and supporters have formed around this space. Fablab regularly organizes excursions and workshops for elementary and secondary schools. Some students from a local technical high school carry out their obligatory practice there. A regular robotics seminar invites experts to give talks and presentations in Fablab every month.

IT Akademia is an (almost) nation-wide primarily EU-sponsored 20 million Eur project initiated by the Pavol Jozef Šafárik University in Košice shaping educational models to meet the demands of knowledge society and thus fostering the interest in informatics and ICT. In addition to providing methodologies, school subjects content definition and augmentation, it has also been providing courses, workshops, seminars, competitions, and summer camps. Due to its large scale and impact, informatics teachers have during the past three years of the project become familiar with the project calls. As a result, they now provide a very valuable and active communication channel allowing the organizers to reach the target audience of the project events.

LSTME is a traditional electronics summer camp whose origins can be traced back to early 1980s. Tens of children aged 12-18 years spend 14 days in the summer in the middle of nature with about 15 experienced adults to learn about technology in a series of workshops, talks and other activities. LSTME, due to its successful record, has been included into IT Akademia project. However, the cooperation only lasted for a single year. The project requirement not to support the same child in the same activity more than once was in a strong conflict with the importance of community building and participation recurrence for LSTME. A new empty space in the project had to be filled and the authors of this paper were called to action. This paper explains how we have approached this challenge and presents the constructionist platform that we have developed for the purpose of the IT Akademia summer camp in 2019. We believe the platform has a wider potential and therefore chose to share it with the constructionist community. We also report on the evaluation of the camp, and a couple of follow-up events.

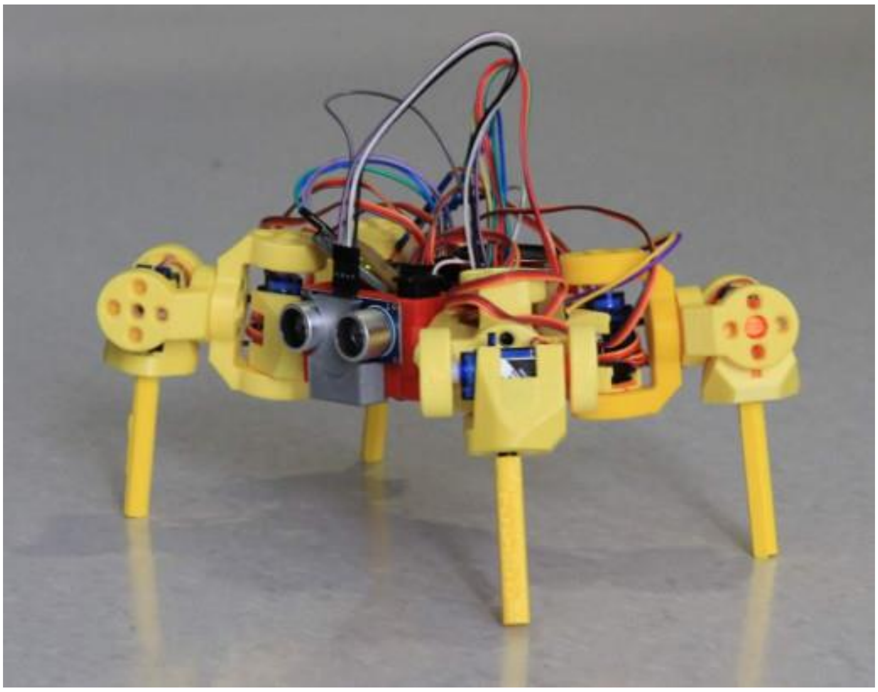


Figure 2. Robot MoKraRoSA: Modular Robot with Arduino built at Krahule in 2019

Previous Work

LSTME

The first week of the summer camp for talented youth in electronics consists of a standard set of workshops. Every child: 1) completely designs and builds a PCB-mounted flashing device, 2) gets an introduction to programming (of a chosen platform of the year, for instance JavaScript, C#, Python, PHP, Arduino, and Micro:bit, have been used previously), 3) learns basics of 3D modelling, and 4) learns basics of film editing and clip making. 3D modelling was added only after FabLab started to get involved. Our vision was to provide a common theme for several workshops, and even though it has not been shared with all the organizers, in two consecutive years, children have built a tiny mobile wheel-robot with line-following, obstacle-avoidance, remote-control, and sound capabilities, all at the cost of less than 20 Eur. The chassis was a parametric design in OpenSCAD and the participants have customized their own robot before they got it 3D-printed. Seeing various designs, they could compare their physics properties and their influence on the robot behavior and performance. We have repeated the Multimouse project also at LSTME 2017, but we ran into a couple of difficulties: 1) there were many other activities in the camp, and thus we have not dedicated sufficient time for this project and 2) the electronics based on Arduino Nano with a single 5V signal and only two GND signals required soldering wires for all the connected devices, a time-consuming task and an organizational challenge.

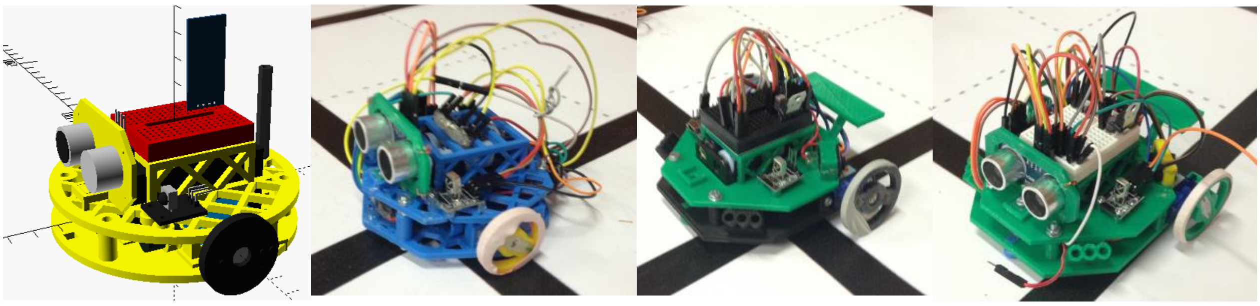


Figure 3. Multimouse – a mobile robot built at LSTME 2016, OpenSCAD model (left) and various modifications made by camp participants [1]

Figure 4. From a final presentation and playing with Otto robots in Fablab day camp 2018

(DT)^2 2018

Inspired by the outcome of the LSTME Multimouse project, we decided to organize a series of day camps in FabLab Bratislava and harness the previous experience for an even better outcome. This event was more focused, children coming only for the workshops and staying at home overnight. The camp accommodated only two simultaneous workshops – 3D modelling, and Arduino programming with robot building, children alternating after 4 hours of continuous work in each. In this case, children have built a modification of the famous robot Otto [2] – either a 3D printed version or a laser-cut plywood.

Again, children could take the robot that they have built home with them and continue the learning process at home further on. We have documented our work in [3]. One of the advantages of the Otto robot as compared to Multimouse was the Arduino Nano Expansion Board that contains multiple VCC and GND pins for the connected devices.

Related Work

Modularity

Before we proceed with the details on the 2019 camps, let us stop here at the concept of modularity that we consider having a much higher importance than is recognized. An ample example of modularity can be seen in solutions of the over-performing teams in FLL competitions. Their robots started from the base enter the field and solve a couple of missions before they return to the base again so that a new attachment could be installed. The children typically simply lift the previous attachment up and place a new one on top of the robot. The newly placed attachment automatically aligns with the robot-core and the gears „click-in“. No further action is needed, and the robot can immediately smoothly approach the new set of missions, see Figure 5.

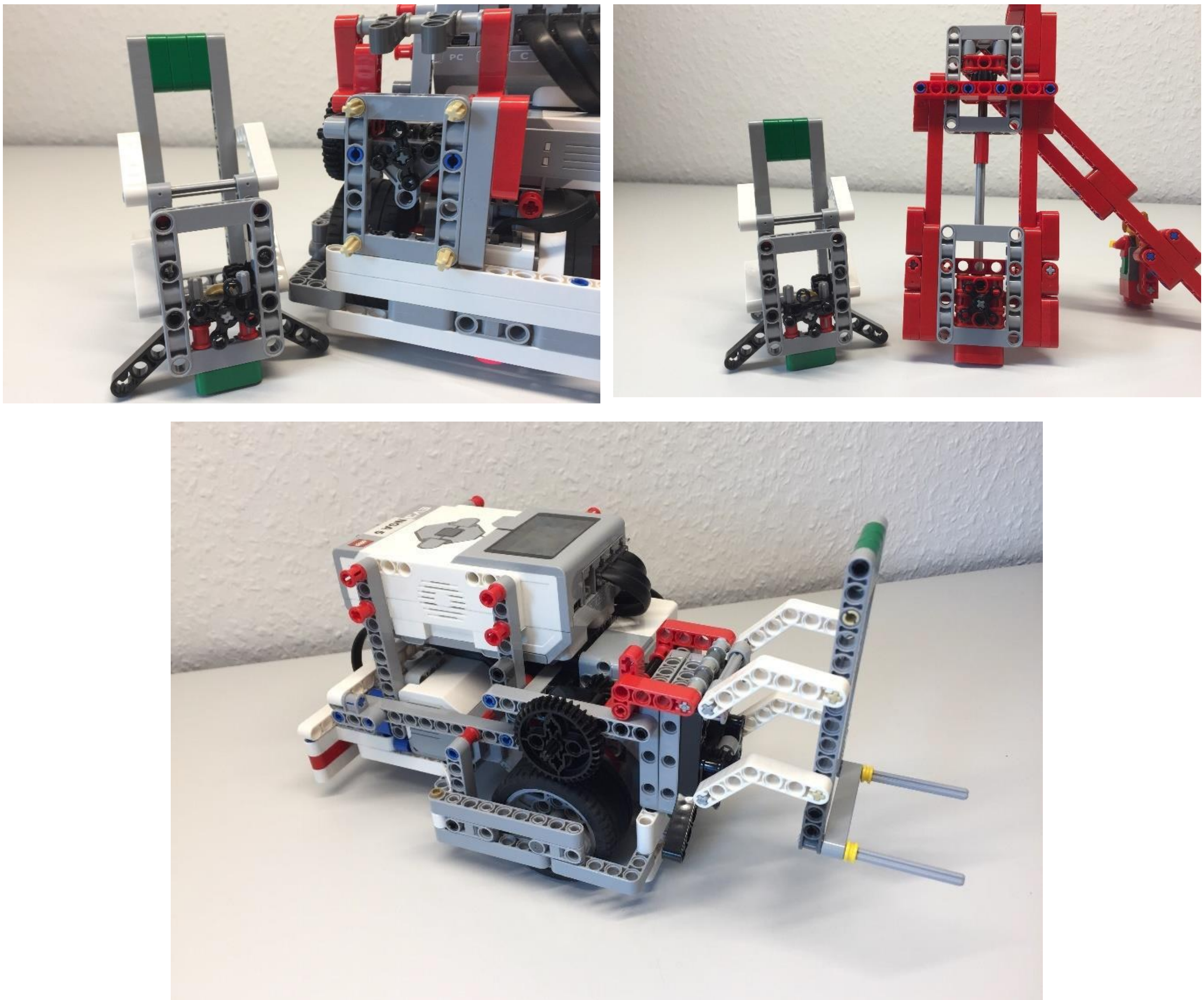


Figure 5. Modular robot attachments for FLL competition, from the Robot Design Hardware handbook provided by NanoGiants Academy e.V. [4]

In general, we consider a system to have a modular design, if it is configurable, consists of independent parts that can be developed separately and that can easily be added or removed from the system without any need for further modifications of other parts of the system, seamlessly adding or removing the respective functionality or feature. Designing a system in a modular way takes extra efforts, resources and time. The benefits are often invisible, especially in a single purpose system. Yet, the elegance of the modular design emerges as soon as the system needs to scale up to larger developer or user base, experiences multiple deployments with varying purposes, and when a repair is attempted. We believe that more efforts in the learning process are needed in order to allow the transition of our society to a real modularity, saving considerable resources, eliminating waste, improving product longevity as well as providing new employment opportunities.

Modular Robotics

Robots as devices that operate in real environments are especially prone to frequent repairs, the field is in a rapid development resulting in continuous partial upgrades, and obtaining the individual parts is (time or money) costly and thus their reuse is desirable. In addition, for certain tasks, specific robot morphologies are more fit than those with universal configurations. Modular robotics allows building complex robot systems from easily replaceable components. Truly modular robots are often called reconfigurable robots, while some of them can change their morphology autonomously, the self-reconfigurable robots. See [5] for an overview of such experimental systems. A direct inspiration for our work was the work of [6] on modular reconfigurable robots that the authors use in a student course to teach chaos and complex systems. Various morphologies can be constructed from the basic elements, see Figure 6. However, our goals were to build a fully autonomous robot without the many cables leading to control and power source. We needed to incorporate some control units. One idea we explored was to use a simple control unit in every component, incorporating them into a network of interconnected parts. This would allow a complete versatility and flexibility the modules would have both mechanical and electrical connections to convey power and information. A result could be similar to a wonderful modular robotics educational system Cubeletes [7]. We see some limitations though. The building blocks are somewhat bulky, the resulting constructions are inflexible and difficult to move around, designs are of a monotonous square-shape, and it is a commercial closed solution so also the cost is relatively high. In this occasion we aimed at designing something low cost, yet interesting, flexible, and moving in some interesting ways in its environment. We have therefore abandoned the idea of distributed control and accepted a solution with a single control unit.

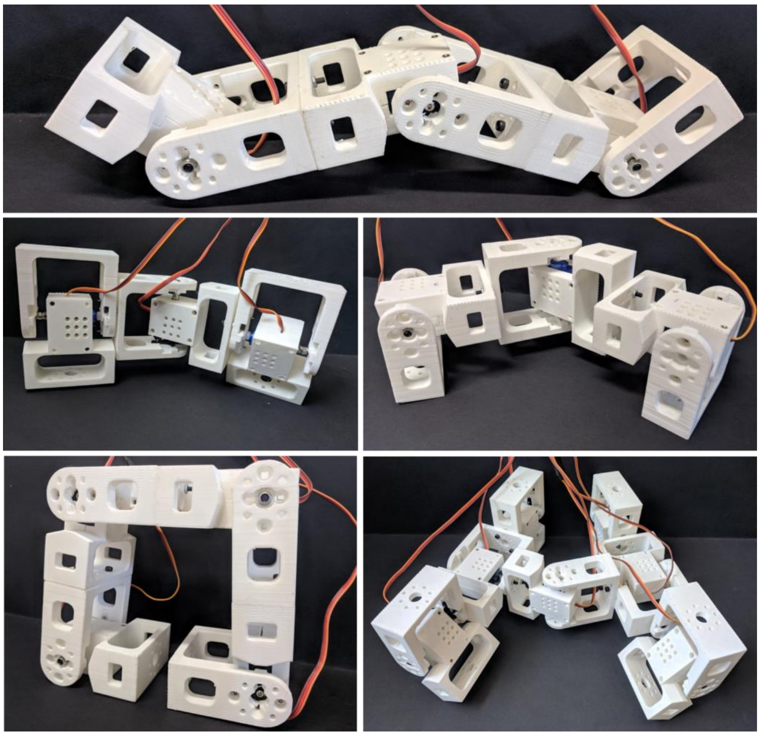


Figure 6. Modular EDMO robots used at Maastricht University in DKE Swarmlab, Laboratory for Cognitive Robotics and Complex Self-Organizing Systems [6]

MoKraRoSA platform

After the successful day-camp in Fablab in 2018, we immediately began thinking about the next season for the summer of 2019 and started to search for a new platform. Even though there still was a lot of potential to work with robot Otto, we had several reasons to change: some participants from 2018 were expected to visit the 2019 camp as well, then we wanted to get experience and study the suitability of different platforms, and our desire for creativity and the long term wish to learn about modular robotics also contributed. Inspired by EDMO described above and having experienced many LEGO robotics projects, we have soon arrived at an idea of creating a LEGO connector peg compatible mechanical parts that could hold motors, control units and any other parts. The first attempts were ours, but soon Martin Slimák, a student taking a practical semester project course and member of FabLab Bratislava has volunteered to create a complete design in a student version of Fusion 360 modelling software. Parts of the result after several iterations are shown in Figure 7.

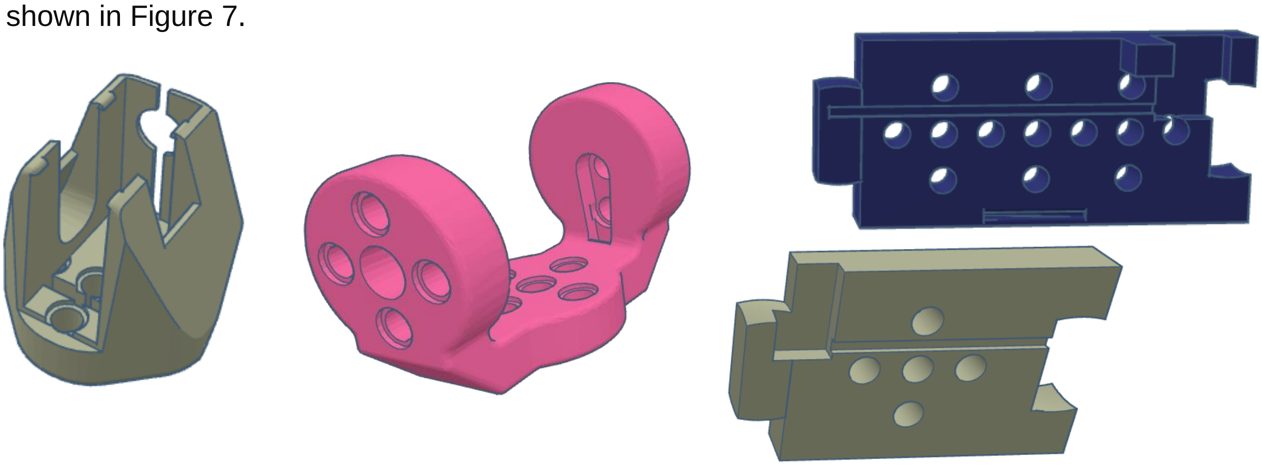


Figure 7. Some of the mechanical parts that formed the base of the modular robot morphology, version 1

The first robots that we have built were simple worms with four degrees of freedom, see figure 8. We have tried adding some components for building a robot arm, such as gripper, stand and length extensions, and a small humanoid, we have later focused on a 4-legged creature (version 1 shown in Figure 7) in order to provide a well-tuned starting point for experimentation.

Goals

Our goals were to prepare a modular 3D printed platform that is easy to assemble and disassemble and holds stable together without any screws, or other fixing except for LEGO connector pegs. Only the servo motors should be secured with the tiny screws they are accompanied with for better stability. We wanted to create a platform that could be both programmed and remotely controlled, with a software allowing easy design of choreographies without the need of programming (coding in a programming language). The resulting set should be easily adopted by the large number of schools that have recently purchased a 3D printer and are looking for its interesting educational applications.

Evolution of the Morphology

To save time and material, most pieces were printed with a very low infill percentage. The consecutive updates intended to accommodate more robustness. We started with the square-shaped designs similar to EDMO but improved on the ergonomics in the later stages. This has been a constructionist experience on its own. Student was learning about the Fusion 360 capabilities, interacting with us in an iterative manner, and discovering the principles of a stable construction throughout the course of several weeks. The four walls of the control unit box have carefully designed hubs curved in two directions that allow the parts to fit and hold together. The Arduino control unit is inserted into grooves in the walls. It is mounted together with the power supply, battery holder and switch as one unit. We have prepared these units beforehand so that the children did not have to deal with the soldering, and wire cutting (about 12 connections per piece). This was the most time-consuming preparational manual work that cost us about two days of work before each camp.

Electronics

Based on our previous positive experience with Arduino Nano in the Multimouse and Otto projects, we planned to remain with the same platform. The disadvantage of the plain Arduino Nano, the lack of the GND and VCC connections, was solved in the Otto project with the Nano Expansion Board. This time, we have found an even better solution: Arduino Nano Strong, a combination of the Expansion Board with Nano in single, smaller-factor board. Powering 8 motors that can move simultaneously is quite demanding. The 4-pack of AA-size batteries as used in Otto has two main disadvantages: it provides unstable voltage since the difference between fresh and almost used-up batteries is more than 1V, and it takes a lot of space and weight. In case of MoKraRoSA, we have used a pair of 14500 Li-ion batteries with the benefit of 66% reduction in weight and 50% reduction in space. Two such batteries provide the voltage between 7 and 8.4 V, which is further transformed to a stable 5V with a tiny power supply with up to 97% efficiency providing 4A output, which is sufficient to power the electronics and servomotors. A new challenge is that these batteries should never be fully discharged and thus require a voltage divider (we used 220K and 22K resistors) to scale the voltage down for the 1.1V ADC reference value. The most unreliable part was the MPU6050 gyroscope/accelerometer. It is a very useful part that can integrate all 6DOF directly in the sensor allowing a specialized firmware to be uploaded in its control chip, but both MPU6050 and HC-05 Bluetooth module are originally designed for 3.3V electronics. Fortunately, both work with 5V Arduino even without level shifter, but the sensitivity of the fragile I^2C bus protocol is high and sometimes leads to communication loss. Full list of parts is available at the platform website [12]. The overall cost of MoKraRoSA robot (from the cheapest suppliers) including the PLA material for 3D printing sums to about 35 Eur including the batteries and the charger.

Control Software

We started with a small program to test the motors and sensors and as the hardware capabilities were developing, we have been improving it incrementally. Soon we found that creating and storing choreographies would be useful, however, in a little bit different way than in Otto. Instead of specifying the choreography as a sequence of single motor moves at specified moments, here, it is a sequence of positions of all 8 motors. The software allows to manually control each degree of freedom and to observe the result and store the reached configuration as the next point of the choreography. Replaying the dance means interpolating all motors in the interval formed by the consecutive positions in 100 steps – in a specified speed. This allows the choreography designer to replay and observe the choreography step-by-step in a debugging mode, and adjust each individual position if needed, insert or delete the points in the choreography. It can easily be transferred to a PC or back, and it can be saved to the EEPROM memory. The robot can be configured to start playing the choreography automatically after it is powered. Alternately, it enters the control mode over the USB cable or Bluetooth connection, it lists the help for all commands when requested and contains a set of predefined moves – such as walking in all directions in a normal or upside-down orientation. Since the robot is almost symmetrical, it can walk and operate in both orientations. After the children have built their robots, we have provided them only with an early version that allowed creating the choreographies so that they have something to start with at all skill levels. At that time, we were only hoping that these creatures will be capable of stable walking one day. What followed, surprised us very much. While we were serving and helping all out of the 40 children in the camp to get started, some of them – the more experienced one asked for more parts, such as the Bluetooth module and Gyroscope, and for the details how to operate them, and soon we could observe pairs of the “spiders” walking and fighting in the corridors while controlled from their mobile phones. Several independent versions of walking emerged, and soon the children have copied and uploaded the best program and worked on improving it and integrating their ideas in it. We merged their and our program together. The software is open-source, it is being still improved and currently its version 9 is available at Github [8].

Figure 8. Worm with 4-degrees of freedom, the first robot we have built from the modular components

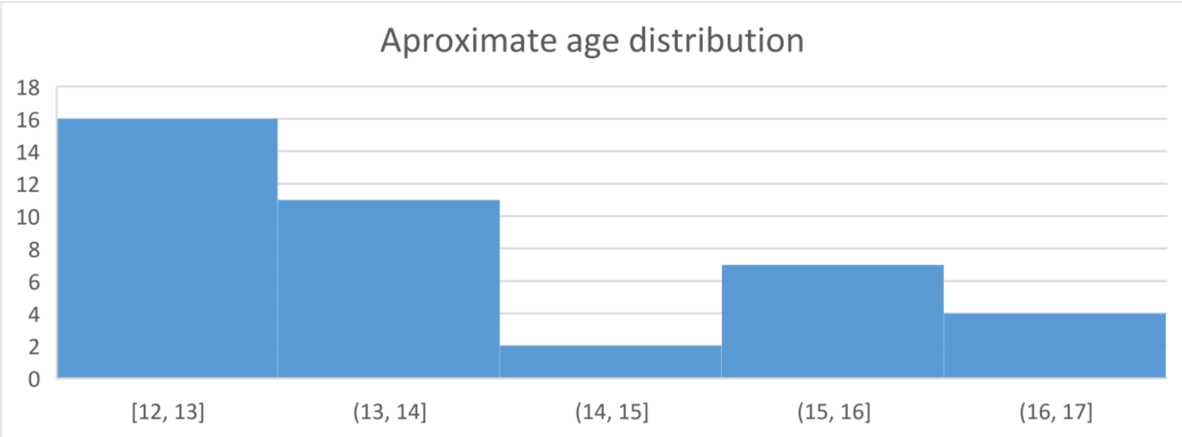


Figure 9. The age distribution in the IT Akademia camp

Summer Camp of IT Akademia

Selecting the Participants

IT Akademia as an EU sponsored project has strict rules and requires formally specified criteria with point system in advance. Points were assigned mostly for participation in various scientific, technological and robotics competitions. We received about 120 applications, and the system generated 40 best students to be invited. However, since the gender-equality values were included in the IT Akademia principles, we could use the option to raise a quote for female participants. After a discussion we have agreed on 50% girls participation. We remained somewhat frightened as we have never experienced a technical camp with such a composition. In the previous, we have both participated in multiple girls-only workshops organized by the organization *Aj Ty v IT* whose purpose is to promote girls' interest in informatics and technology. In our experience, girls are often very active and well-performing when in girls-only activity, while in mixed groups the prejudices originating from social and cultural stereotypes sometimes result in girls taking a passive role and lacking a healthy self-confidence. To our surprise none of these effects could be observed, and the group cooperated in a very productive atmosphere of joyful learning where everybody was equally actively involved. We were thus very happy about the choice of the 50% composition. It created a very nice and motivational social environment with successful outcomes. The age requirement was set to 12-17 years, Figure 9 shows the detailed age distribution.

Activities

The 40 participants of the camp were divided into 4 equal-sized groups at random. A typical day consisted of two about 4 -hour workshops and some more activities in the evening. In addition to sport, games, and social time together, the workgroups alternated between the following workshops.

Film Making

The goal of this workshop was to learn about film making and editing. The participants learned about making animated clips, and every group has produced one. The leader of the workshop was a graduate of the Film and TV Faculty at Academy of Performing Arts and she shared her expert knowledge in the field. All groups have produced several interesting artistic clips and they were filming workshops of other groups and documenting the camp.

Figure 10. From the film making workshop

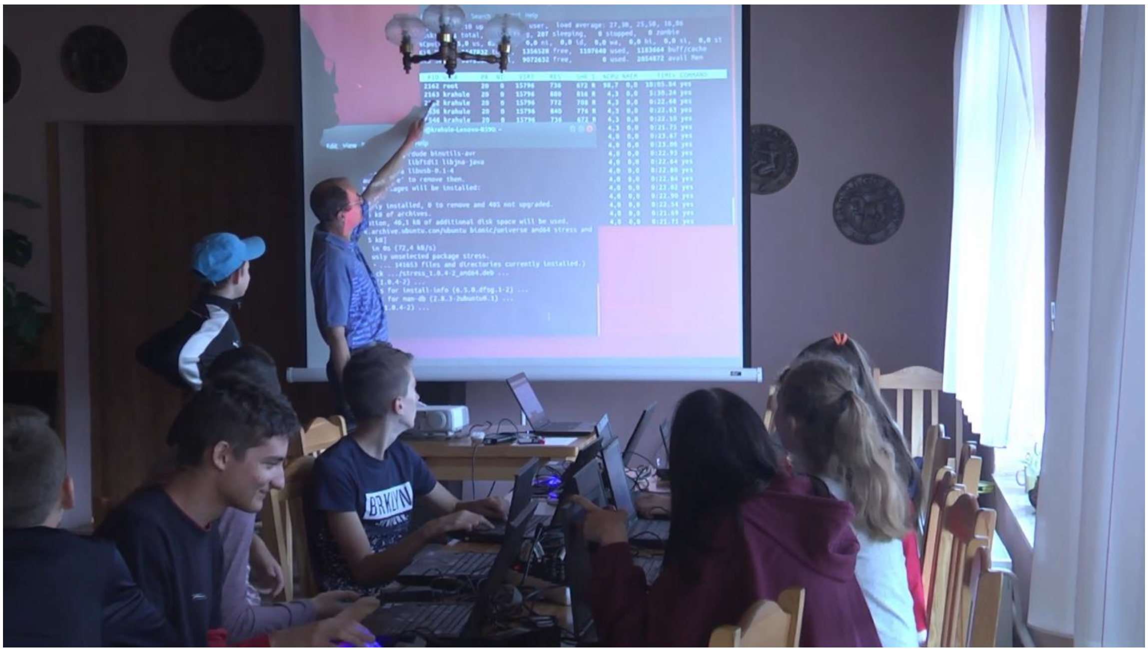


Figure 11. From the Linux workshop

Linux

In the various educational opportunities where we meet young people, we have observed that the awareness of Linux operating system, and the Linux user and programming skills are usually quite poor in Slovakia. Yet, we believe Linux has a very large potential, not only as a most typical server OS, but also as a base platform for educational applications, and a reasonable Desktop system. Especially after the release of Windows 10 that rides roughshod over the computer resources and changed many powerful PCs into a useless brick. This workshop was targeted on learning basic usage, scripting, and automation. Every participant learned how to install and setup Linux, and thanks to a generous sponsor contribution received a fast 32GB external drive where they have installed Linux and took home for further learning, using and exploring. They have also learned about the Raspberry Pi platform. One of the leaders was a practitioner from an IT company designing computer networks, and the other one was a university teacher with experience in teaching Linux system programming course among others.

Arduino Programming

Since the MoKraRoSA is based on Arduino platform, the choice for the programming language and environment to learn was C-programming and Arduino. The workshop was a project-based work. The participants learned about all the basic I/O, serial communication, introduction to C-programming. For this purpose, we used the Acrob mobile robots [9] in a playful way, and various sensors and other parts connected to a breadboard, such as potentiometer, microphone, ultrasonic sensor, gyroscope, passive buzzer, IR distance sensor, LEDs, etc.

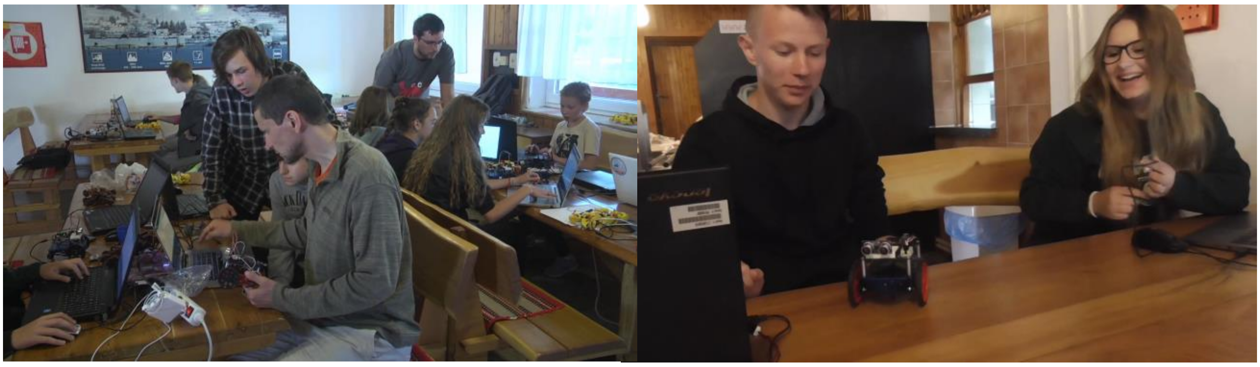


Figure 12. From the Arduino workshop

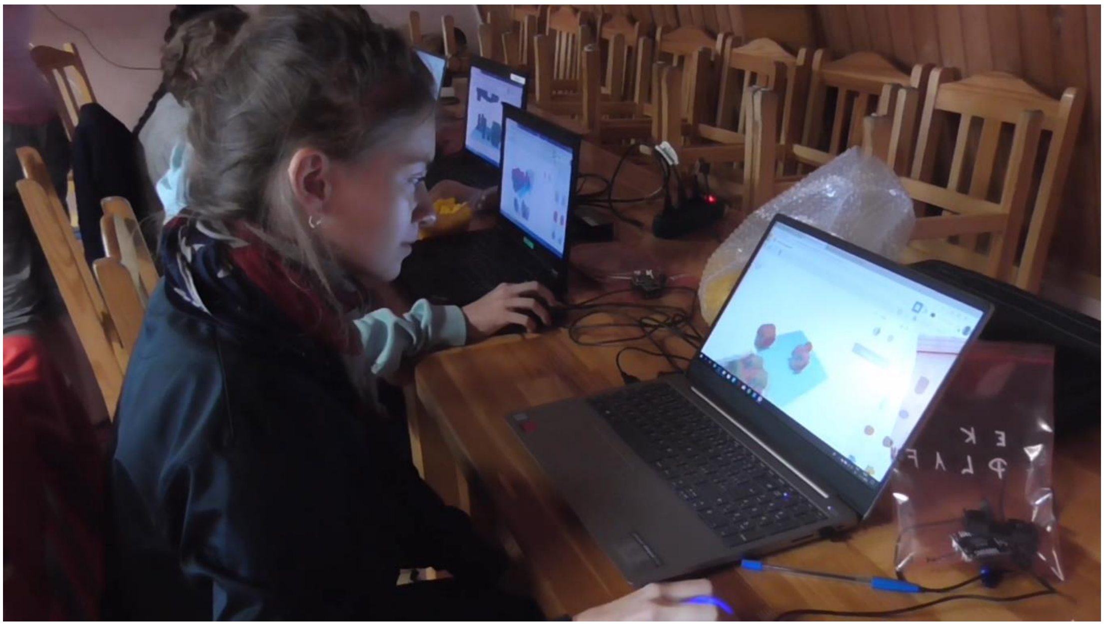


Figure 13. From the 3D modeling workshop

3D modelling

In 3D modelling workshop, the participants have learned basics of the solid geometry modeling using the TinkerCAD system. They have received the designs of the MoKraRoSA robot and all of them have designed their own legs for the robot. A graduate student who has a bachelor's degree in the field related to 3D printing and who is also a Fablab volunteer was servicing almost ten 3D printers that were in a permanent operation during the camp. After the first week, we have merged the Arduino and 3D modeling workshops into a single workshop focused on building and programming MoKraRoSA, which had a double frequency. Learning design enables constructionism at new level. With design, the subject gains the skills that open possibilities for personal and group discovery in an iterative experimentation, observation and hands-on learning.

Building the Robot

Actual robot assembly took about 2 hours, however a special care had to be taken when inserting the servomotors to have them properly pre-calibrated and mounted at a proper position. All the electronic parts were mounted around the robot at various places either using a separate 3D printed parts (such as the ultrasonic sensor, mp3 player, passive buzzer) or simply attached using a thick double-sided tape. The wires of the servo-motors were folded and the folds inserted in separate envelopes formed in the servomotor holders. Based on the reports from the participants, this building part was one of the most interesting activity for them. 30 out of 40 participants chose to take the robot with them home, we have provided it for the raw cost of the parts.

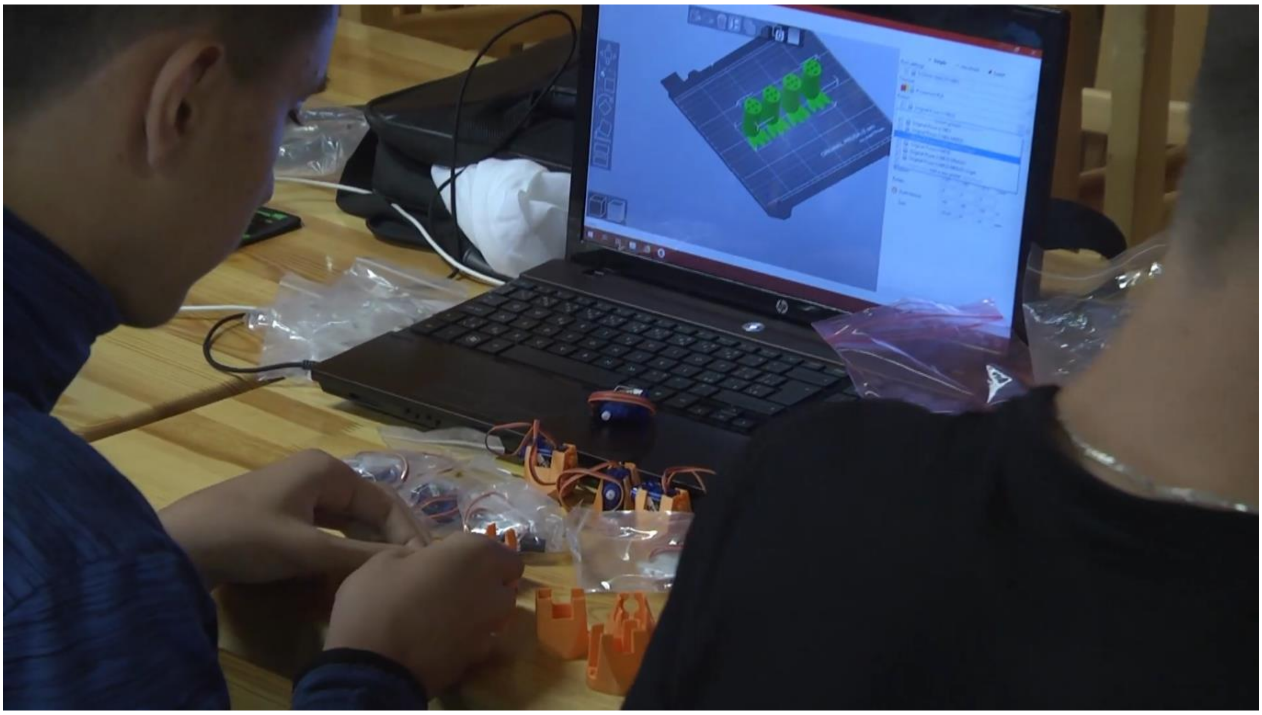


Figure 14. From the robot building workshop. The legs designed by the participant are on the screen

Figure 15. From the robot fighting tournament

What has happened

After the participants had built their robots, we expected them to work on their own a choreography so that we could see some ideas for creative movements, and many of them did. However, some of them have asked for parts that we hoped to start using later in the program and worked evenings on their own. We have planned to focus on walking gaits only after everyone would be comfortable preparing the choreographies. However, these talented students soon came up with an efficient solution for walking and programmed remote control using the Bluetooth module and their mobile phones, implemented various gestures and arranged fighting tournaments, and they have shared the best programs among themselves on their own initiative. This has been an unexpected development, but it has shown that MoKraRoSA is a functioning constructionist platform that motivated learning and sharing in a natural way.

Other Activities

As in our other camps, we have used some of the activities from The Systems Thinking Playbook [10] to help open minds of the participants. Every morning started with a sport warm-up activity, for the evenings, we have prepared quiz, the participants prepared dances, and songs, we have played various social and sport games, and the final half-day consisted of a rallye, where the participants used the learned knowledge. Almost all the time schedule was covered by organized activities.

DT^2 camp

After a break one week long, we have continued with two day-camps directly in the space of Fablab Bratislava. About 30% of the participants joined us already for the second year. We have used the experience on MoKraRoSA platform that we collected in the two-week camp so that the building and programming would be smoother. This camp was targeted at the age of 11-15 years. 20 participants in each camp were split into two groups based on their skill level. The groups alternated between two 4-hour workshops. In the first workshop, the participants enjoyed the usual Fablab activities – such as making T-shirt and textile bag decorations, producing small 3D printed and laser-cut jewelry and items and learning about 2D and 3D modelling, while in the second workshop they first learned about Arduino programming in hands-on projects and then built their robots and worked on their choreographies. All 40 children have taken their robot home. At the end of all three camps, the participants have demonstrated the results of their work to their peers in a constructionist spirit.

Example choreographies

When connected over the USB cable or Bluetooth connection, the robot can be controlled directly in control mode or switched to edit mode for defining a choreography. Pressing a certain key in edit mode (16 keys altogether) moves one of the 8 DOF in one or the other direction instantly. As soon as the user is satisfied with the next position, he or she hits the ENTER key and the position is stored as the next point in a choreography. The user provides the speed of movement for each such transition. At any time, the user can replay the current sequence once, or in a repetitive loop. After entering the debug mode, the user can perform the steps one at a time, insert, delete or modify the respective steps at any position. Preparing a choreography is a kind of a programming activity, but it is available to anyone without any programming experience.

Greeting

The simplest of the three example choreographies is the greeting. The robot first moves 3 of its legs in a position where they provide a stable tripod. The fourth remaining leg is free and lifted and in a vertical or horizontal movement greets the observer.

Swimming

This is the easiest choreography that allows the robot to travel. We call it swimming as it resembles the movements of a swimmer. It can be performed in as few as four steps. The robot first lays down by lifting all its legs simultaneously. Then it moves all the legs forward, while laying on its belly. After putting down and standing on its legs again, it moves the body forward simply by moving the legs backward.

Acrobat

The robot is designed to be almost symmetrical and it can flip upside-down on its own. Among the popular choreographies designed by several participants were various forms of roll overs. All three choreographies can be seen in a gallery on the platform website [12].

Feedback and evaluation

It was difficult to leave the first camp for most of us as the two weeks period is long enough to form friendships and bonds. We have sent a questionnaire to all the participants to receive a feedback on all the activities in the camp. 80% of them responded, and Figure 16 contains the results. We also asked for verbal comments, which were very encouraging and indulging the authors appetite for a perfectly cooperative group. Overall, not only the goals of the camp were fulfilled, but we are confident that the platform we have prepared serves as a useful constructionist case study.

After the camp

We tried to stay in contact with the participants sending news, and updates, met several of them in person to help them correct any issues that they have experienced after returning home. Two girls from a secondary technical school participated in Amavet Science Festival, where they have prepared and presented a popularization project for the MoKraRoSA platform. After advancing from the regional round, they have been awarded one of the prizes at the national level.

We have continued the software development after the camp for a while to improve on some of the features. However, we have also developed a new version of the robot body – so that all of the electronic parts are seated in their little containers, and the box for the control unit is covered from both sides – thus storing and hiding all of the wires except of those leading to servo motors. A comparison of the two versions in shown at figure 18.

On November 15th, the first Mini Maker Faire festival took place in Bratislava. Fablab has prepared a workshop where 10 registered visitors have built their own MoKraRoSA robot. Ten other visitors have built a Makey version of robot Otto. This turned out to be more challenging in a noisy and busy place as compared to a camp with much controlled conditions and more time, but the enthusiastic visitors have managed to overcome all the obstacles. We have also demonstrated the platform to public in a similar event 3D Expo in Bratislava in November (Figure 18), where one of the authors held a talk about this robot story for public.

Some of the participants have already spent efforts to upgrade their robots to a new version. We have prepared a public website about the robot with example choreographies, links to the open source code, all STL files for the 3D printed parts, and more.

Conclusions and Further Plans

We have described our experiences with the modular mobile robot platform MoKraRoSA. It has served as a constructionist platform for all ages: participants in the camps aged from 11 to 17 have been building it, programming it and designing choreographies. A student of Applied Informatics in his 2nd year of study has worked on designing the robot body, and the two authors of the article were designing and implementing its control software, and managing the overall design process, and the system philosophy, and tinkered about the applications, and the way of introducing the platform to the participants. In total, four generations of constructionists worked in a concert towards a common goal, they all learned together in groups, shared and presented their knowledge and experience to each other. All parts of the system are open-source and published on Github [8].

Open Challenges

Our aim is to increase the user base of the platform, primarily in the community of primary and secondary schools in Slovakia. We have already negotiated with a private company that delivers 3D printers to schools and we have agreed on a cooperation. The idea is to prepare sets of parts, documentation, manuals, and tutorials, and the schools will print and build and further develop their robots. We plan building new complete systems from our modular platform in addition to the 4-legged creature. Inspired by another similar activity Robot League [11] we are considering an online contest – instead of regular problems published every two weeks, we provide several open challenges that will start after the participants will try to solve them.

Figure 16. Evaluation of workshops (1 means more excellent, 7 means less excellent)

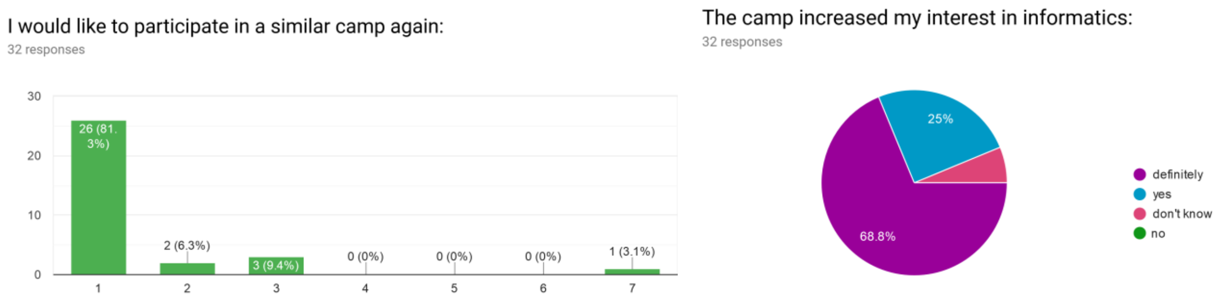


Figure 17. Evaluation: general questions (1 means more excellent, 7 means less excellent)

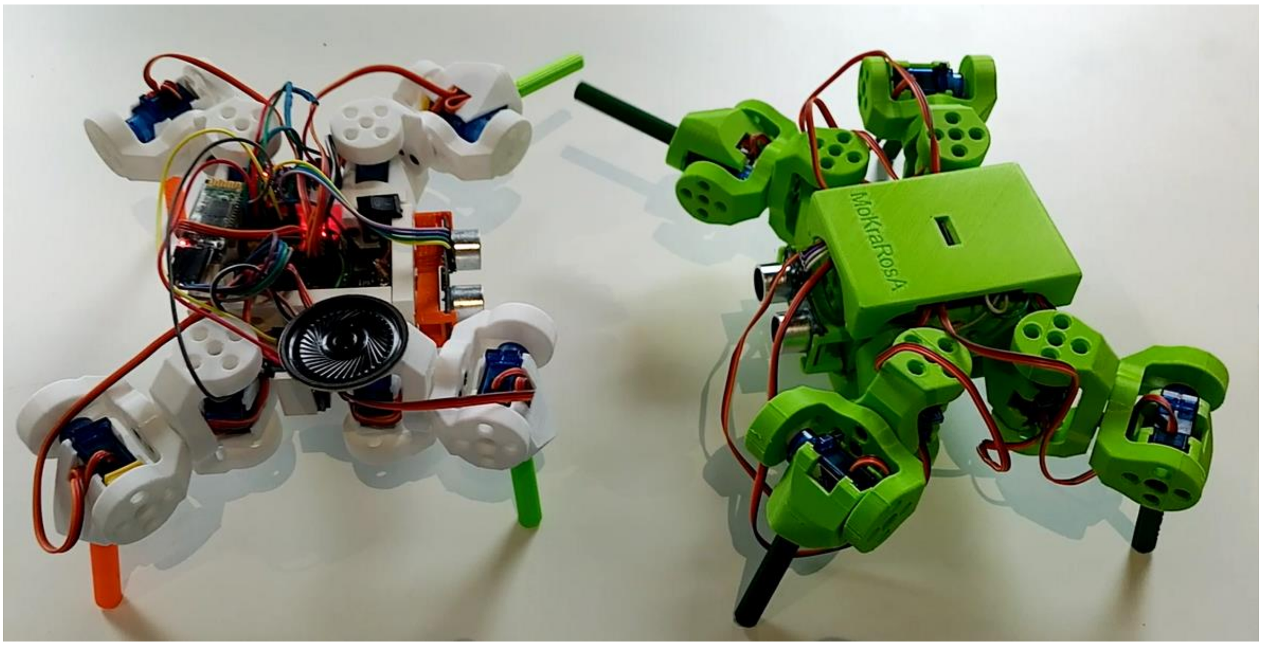


Figure 18. Comparison of the version of the robot used in camps and a new version developed in autumn

References

[1] Multimouse, Fablab Bratislava project, 2006, online: dev2.fablab.sk/en/multimouse

[2] Otto DIY, Otto DIY Workshop s.r.o. online: www.ottodiy.com

[3] Pavel Petrovič and Jozef Vaško: An Open Solution for a Low-Cost Educational Toy, Robotics in Education, pp.196-208, 2019, doi: 10.1007/978-3-030-26945-6_18.

[4] Thomas Madeya: FIRST LEGO League Robot Design Hardware, NanoGiants Academy e.V., 2017, online: nano-giants.net/en/first-lego-league-robot-design-hardware

[5] S. Sankhar Reddy Chennareddy, Anita Agrawal, and Anupama Karuppiah, "Modular Self-Reconfigurable Robotic Systems: A Survey on Hardware Architectures," Journal of Robotics, vol. 2017, Article ID 5013532, 19 pages, 2017. https://doi.org/10.1155/2017/5013532.

[6] Rico Mockel, Lucas Dahl and Seethu Mariyam Christopher: Interdisciplinary Teaching with the Versatile Low-Cost Modular Robotic Platform EDMO, International Conference Educational Robotics 2018, in: Michele Moro, Dimitris Alimisis, Luca Iocchi (ed.): Educational Robotics in the Context of the Maker Movement, Springer 2020, doi:10.1007/978-3-030-18141-3.

[7] Eric Schweikardt: Modular Robotics Studio, in: Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction (TEI'11). pp. 353-356, 2011. doi: 10.1145/1935701.1935784

[8] MoKraRoSA Github Repository. online: github.com/Robotics-DAI-FMFI-UK/mokrarosa

[9] Richard Balogh: Educational robotic platform based on Arduino, Proceedings of the 1st international conference on Robotics in Education, RiE2010, pp. 119-122, 2010.

[10] Linda Booth Sweeney Ed.D., Dennis Meadows: The Systems Thinking Playbook: Exercises to Stretch and Build Learning and Systems Thinking Capabilities, Chelsea Green Publishing, 2010.

[11] Pavel Petrovič and Richard Balogh: Robotics League - a Unique On-line Robotics Competition, in. Robotics, in Robotics in Education – Current Research and Innovations, pp. 344 – 355, 2019.

[12] MoKraRoSa Website. online: mokrarosa.ninja.

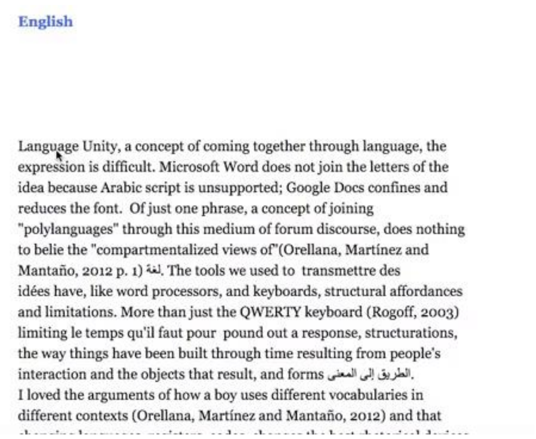
Movable Type Translingual Composition: "Being Multilingual is Like Having Many Friends"

Kit Martin, *kitmartin@u.northwerstern.edu*
Dept of Learning Sciences, Northwestern University, Evanston, U.S.

Eva Lam, *evalam@northwestern.edu*
Dept of Learning Sciences, Northwestern University, Evanston, U.S.

Abstract

Transnational youth use digital media to affiliate with diverse cultural and linguistic practices, as demonstrated through the use of multiple languages and hybrid linguistic codes, media genres and multimodal expressions in the youths' online communication and writing (Black, 2009; Domingo, 2014; Kim, 2016). This study introduces a learning activity, the multimodal writing design *Movable Type*. With *Movable Type*, we investigate how multi-lingual college students use different languages in their composition in the context of 7-week creative writing curriculum, to construct multilingual narratives. The workshop was a constructionist learning activity. We find youth engage with the tool to communicate across multiple audiences.



English

Language Unity, a concept of coming together through language, the expression is difficult. Microsoft Word does not join the letters of the idea because Arabic script is unsupported; Google Docs confines and reduces the font. Of just one phrase, a concept of joining "polylanguages" through this medium of forum discourse, does nothing to belie the "compartmentalized views of"(Orellana, Martínez and Mantaño, 2012 p. 1) لغة. The tools we used to transmettre des idées have, like word processors, and keyboards, structural affordances and limitations. More than just the QWERTY keyboard (Rogoff, 2003) limiting le temps qu'il faut pour pound out a response, structurations, the way things have been built through time resulting from people's interaction and the objects that result, and forms الطريق إلى المعنى.
I loved the arguments of how a boy uses different vocabularies in different contexts (Orellana, Martínez and Mantaño, 2012) and that

Figure 1. Visit the online demo and click on *Language Unity* to see the words turn over.

Imagine every word in this sentence as a wooden block. On the front side of one block is the word *wooden.* The block's other sides are different languages, such as *ahşap, de madera,* or *en bois,* in Turkish, Spanish, and French, respectively. The reader can turn the block in the sentence, and see the word "wooden" in more than one language. Each word in the sentence could be organized like this. Thereby the sentence's author could write a dynamic multilingual text. *Movable Type* takes this design idea for multilingual writing and digitizes it to meet the needs of translingual youth on digital platforms to express themselves in more than one register. The approach implies that we can conduct multilingual writing exercises, even in predominately monolingual education environments. The approach contributes to our understanding of constructionism by providing examples of multilingual text construction through open-ended, multilingual construction, tinkering with lexicons, and lowering the floor to transnational youth practices in classrooms.

Keywords

Translanguaging, Literacy Studies, Creative Writing, Constructionism, Design Study

Introduction

Constructionism is a pedagogical approach to learning that prioritizes designing the ideal environment for learners to optimize their construction of mental models in interaction with mediating objects, such as learning technologies. Constructionist thinking influences the learning science and educational research, particularly when addressing learning technologies, mathematics, and science education reform. Papert coined the name Constructionism, as a mnemonic (Papert, 1986), to describe a species of constructivist thought. It focuses on the benefits of learning from the external construction of an artifact beside the internal construction of a mental model, or framework. It focuses on scaffolding through clever design of technology (Wilensky, and Papert, 2010), learner freedom (Feriere, 1987), and less regimented time where people can explore ideas (i.e., microworlds, Edwards, 1995). In this intervention, we took the perspective that creative writing is an excellent way to enable multilingual students the ability to construct a mental model of multilingual composition in formal writing settings. We had learners explore multilingual composition of a creative writing project in a constructionist learning environment. *Movable Type* was built to promote this multilingual writing in a constructionist writing course, mixing computer coding and composition.

In this paper, we will introduce a multilingual composition approach, using HTML, JavaScript, and a program called Twine. This project came out of the idea from where the first author grew up: in Nairobi, Kenya and Khartoum Sudan, both in East Africa. When he was growing up, language plurality was the norm, kids would play in German, Swahili, Arabic, Dinka, Nuer, and English and mixing what we said, with polyglot slang. This is polyglot communication, or as a researcher would revoice it as translanguaging (Garcia & Wei, 2014). Moreover, he learned to read late at 11 years old. When writing, it confused him that only one language was used. Most people would speak in more than one language, but when it came to writing they would write in one dominate language which did not make sense. In this paper we introduce a design, and then demonstrate user testing of an approach to change this situation. We introduce a constructionist writing app, *Movable Type*. This desire fits into a need emerging from online youth culture.

Significant research has shown how multilingual youth use digital media to affiliate with diverse cultural and linguistic practices, as demonstrated through the use of multiple languages, hybrid linguistic codes, media genres, and multimodal expressions in youths' online communication and writing (Black, 2009; Domingo, 2014; Kim, 2016). To bring this reality into the classroom, we have designed a constructionist multilingual creative writing environment, *Movable Type*.[32] This design is inspired by multilingual affinity space research (Lam, 2006; Thorne, Black & Sykes 2009; Gee & Hayes, 2011) and recognizes that in online affinity spaces, participation is often multi-modal (Lammers, Curwood, Magnifico, 2012). New synthesizing presentation technologies have made creating, capturing, editing, uploading, and sharing significantly easier, which increases the applicability of multimodal contributions in schools (Curwood & Gibbons, 2009). Furthermore, people do not work in a vacuum, instead, we are surrounded by a socio technical system and embedded in it (Hutchins, 1995). This embedding influences how we do our activities, for instance, when youth post on Facebook or Twitter their posts are influenced by the system they employ to make that post (Curwood, Magnifico and Lammers, 2013). As a result, when a student writes in English only classrooms, they are likewise influenced.

Design

Our design specifically targets expanding opportunity for multilingual students through such multimodal presentation that emerging technology affords to express themselves with dignity and fluidity across linguistic domains, using multi-voiced (Wertsch, 1991) expression and registers, widening the opportunities for multilingual students to construct and share multilingual texts. We intend that this product can contribute to the democratization of new media access and enjoyment as youth draw from their diverse linguistic repertoire to express themselves and redefine their

[32] (For the demo see: http://philome.la/Annonym37753843/movable-type/play).

relationship to English and other dominant codes in society. This design has the promise to open up new opportunities for constructionist learning in a code first environment, that we hope this pilot can point towards.

Movable Type

In our design, imagine each word is a wood block. The author can select words or phrases and have them turn over by click. We put different words on different sides of the block, so readers can click to have an alternative word or phrase appear. This allows the writer to author more than one voice, for multiple audiences. This allows dynamic moving of text. For instance, a Mandarin only audience, or a polyglot online fan fiction forum can be authored for more than one audience.

These alternatives, like the original single composition, are author defined. As such, the author can speak to multiple registers within one composition. Like a reader referring to a citation or dictionary, the alternative terms bridge the gap between the author and readers' linguistic registers. This design affords multi-lingual, multi-register composing. Unlike alternatives, the author is in charge of choosing all alternative expression types. Whole sentences, or single terms can be changed, thus avoiding the issue of word order needing to be changed between languages.

Bringing together lean product design and constructionist learning we facilitate the expression of students from diverse language backgrounds. *Movable Type,* is a means to bring multi-lingual composition into educational environments, using off-the-shelf technology that provides creative writing expression without requiring monolingual demonstration.

Methods

We conducted a two-part study to demonstrate the uses of multilingual writing. We employed iterative design, and qualitative methods. In the first part, we iteratively designed the environment, where users can use inline programming in an HTML environment to author multilingual text.

In the second part, we had six users test the program over a 7-week multilingual creative writing workshop. We collected pre-post interviews, screen captures, class collected video and audio, as well as collected student artifacts. We used these data sources to analyze the intervention to generate case studies of digitally-mediated, multilingual composition to demonstrate our design's affordances and constraints.

The Curriculum

We organized a 7-part creative writing course for multilingual college students.

Table 1: The 7 week curriculum for movable type

Session 1: Making Games with Twine *Goal***:** Introduce participants to the basics of Twine and begin creating. We introduced them to some programming in Twine, and talked about the basics of making a character. Then we spent 20 minutes writing our first Twine Game.
Session 2: Multilingual Composition Techniques *Goal***:** Write First Multilingual Composition using Movable Type.
Session 3: Examining Multilingual Texts: *Drown* by Junto Díaz *Goal***:** Write compelling stories using our new tools, based on *Drown*, a multi-lingual text.
Sessions 4-6: Writing Workshops *Goal***:** Provide time to write then complete a short story to present
Session 7: Creating a public object to share *Goal***:** Make a public artifact to share.

The purpose of this class was to 1) document and describe a possible curriculum model for a digitally-integrated multilingual composition in a Language Arts environment, and 2) analyze and articulate the learning mechanisms of multilingual composition, with the goal of ultimately developing a meaningful multilingual authoring curriculum and environment to be used in other learning environments. This second goal will not be discussed in this paper, but will be in future work.

Participants were college students recruited at a large Midwestern university through email and fliers. They ranged in ages from 18 to 26 years old. All participants spoke and wrote multilingually. Participants were six youth, three Asian Americans, who spoke and wrote English and Mandarin. Two of these also had learned to speak and write German. One was of Middle Eastern heritage, and spoke and wrote English and Arabic. Two were Latinx, and spoke and wrote English and Spanish. Five participants were female, one was male.

Research Questions

This study aimed to investigate how multi-lingual college students use multi-lingual and multi-modal elements in their composition in the context of a creative writing curriculum, and how digital tools affect this sense making process. To increase our knowledge about use of multimodal contributions in formal spaces, like writing in schools (Curwood & Gibbons, 2009), we sought to answer the following questions:

- How do students use multi-lingual elements through *Movable Type* to engage their audience?
- What practices do students employ during their use?
  For instance, do students engage with generative grammar?
- What are the constraints and affordances of *Movable Type*'s design to facilitate multi-lingual creative writing and authoring?

Findings

In this section we will report the results from three of our participants, Ju, Angela, and Cecilia (pseudonyms). We will focus on our three research questions: how students used the multi- lingual elements through *Movable Type*, how students engage and used globalized social practices in a creative writing environment, and what constraints and affordances of the *Movable Type*'s design facilitate multi-lingual creative writing.

Multi-Lingual Authoring for Multiple Audiences

To answer research question 1 and 2, how users engage in *Movable Type* and with what practices they employ, we used an interview protocol. In this section we will discuss results from one user, Ju. Her family was originally from Mexico, but she grew up in the Midwestern United States, and entered as a first year at a local University. Using *Movable Type*, Ju crafted two narratives that used translanguaging (Garcia & Wei, 2014) to various extent, where she drafted texts in more than one language. When we asked her why she normally chose to speak monolingually among friends she argued the addressivity of her language:

> Interviewer: Why do you mostly choose to speak in one language?
>
> Ju: I think it's because of the audience. So like if I were to speak in both, people would constantly be asking me what it means.
>
> Interviewer: And that would make you?...
>
> Ju: Then I would have to teach them. [Laugh] That sounds very very lazy but like when I do [speak in more than one language], and people make fun of me. The [words] I am using [multilingually] already have like an expressive quality to it. Like I could say *Ay, por Dios [0:48]* and no one has to know exactly what it means because they are gonna guess that I am frustrated. I roll my eyes. I wave my hand like that [raising and flipping her hand]. Also, I was always taught not to say "Oh My God", because I'm not suppose to take the lord's

name in vain, but for some reason, I don't think the same way about *Ay, por Dios*. I was never taught that.

The same phrase literally translated had different meanings between the two languages. She has a rule when she says the phrase *oh my God*, that does not apply when she says *ay, por Dios* even though they literally mean the same thing. When I asked her if the word *dios* is the same as the word *god*, she said: "They are not. While the literal translation is the same, the way that it's said is different. I've never heard someone say *ay, por Dios* because they are excited. It's always like, are you kidding me?!" The same idea has different meanings in different languages, and being able to use the right one in a text has potential benefits. Using both meanings allows Ju to author the version she means to portray. Ju found that teaching others, unless she was using the expressive quality of an interjection, did not work in everyday dialogue. So she chose who to speak to multilingually based her impression of the audience's language competence.

She argued this multi-languaging allowed her to "communicate with more people." When she volunteers at wellness centres, and soup kitchens, she can actually talk to the people at the shelters. "Whereas other people can't." Languages allow for human connection with the globalized needs caused by immigration to a Midwestern town. Additionally, within her own family, her younger cousins never got close to her grandfather, because they did not know Spanish. Language, both monolingual and polylingual, brings people together.

Translanguaging

She found that *Movable Type* encouraged her to translanguage. She found that writing in multiple languages was like "having many friends." Because she had more words to lean on when she needed to.

> Ju: When you are multilingual you have so many more opportunities. Like for friendships, or for new ideas, or to connect with other cultures. There is so much, because languages are not just words. There are reason why certain words exist. The things that are said in Spanish are adaptable. So having many friends, you have more resources. A different quality of experiences. Or like, every word is based upon their history. Being multilingual you have a back way into all of these languages.

Bringing all of her friends into the formal discourse of creative writing seemed to embolden her. While, bringing translanguage into the formal context of creative writing at first was difficult for her, on practicing it week after week she grew to interject multilingual thoughts into writing which she usually would perform monolingually. She wrote multilingually in a variety of contexts:

> Ju: In a context where it incorporates all of the languages. Or if I am writing a letter, but really I mean a message, to someone who speaks both languages, or if I am talking to someone in one language and I am like I don't know exactly how to say a certain word, I will just say the other and hope for the best.

She also said that during her writing in *Movable Type*, when she wrote about her grandmother, she wrote multilingually, "Because a lot of my experiences with her have been in Spanish, so like Spanish thoughts just appear." She argues that *Movable Type* affords her view of interweaving language:

> Ju: Because you can like fix your audience. I wouldn't have to worry about using certain expressions that come to my mind. Like, you know how I say *ay por Dios,* those things [interjections] are expressive on their own. But depending who is in my audience, I will choose certain words because I am not going to say something someone doesn't understand. But with the clicking [through words] function in *Movable Type* I can say something I'd usually say in another language because it makes more sense to the audience. [Movable Type] is helpful because I can express myself in whatever way I think is best, and my audience will always get the message.

This practice brought together her interest in globalized identities, bonding with her family, and communicating to multiple audiences the way she imagined they wanted to be talked to while filling a personal desire to bring out the flair, that she would use in speech, in her writing.

Generative Grammar

Ju found that *Movable Type* poses some problems to grammar. For example, in English you say *to walk, to eat*, but in Spanish, you do not put a preposition before the infinitive. So when she used the infinitive she kept the preposition *to* in English, and then used the Spanish verb, forming: *to caminar.* She kept the *to* because: "In my mind, it's the infinitive version and meant *to do that*. I didn't translate it as *to to walk*, but instead, *to the 'movement of walking*. In this process, Ju generated grammar in time, as she needed it. Generative grammar did not stress her out. She constructed it as she went. Ju found that the *Movable Type* design afforded her the ability to integrate her multilingual speech into her creative writing without leaving out audiences who do not themselves speak multilingually.

Coding While Writing is Powerful, but Confounding

Ju found the use of the code snippet in the HTML at times overwhelming. She said:

> Ju: I found the coding cool, cause I don't do that. If coding was part of the intention of it, if you wanted them to learn about coding, then keep it, because it makes total sense and expand on that portion of it. But because it would become so bulky and stuff [in the writing HTML] if its more so just for the language component, then put the coding behind the scenes. Just type this word and highlight it, and it will go to another word.

She felt the design should focus more on either the coding, or the writing. She argued she did not really code, and switching back and forth was distracting her from writing. This constraint of coding while writing made her feel less creative than she would have.

These constraints and affordances of *Movable Type's* design facilitated multi-lingual creative writing, but they exposed that the coding aspect detracted from her flow of creativity. As we continue to iterate the design we will decide if the coding is part of the design and integrate it more fully, or if writing is the focus. If writing, we will need to provide a graphical user interface (GUI) to go over the multilingual writing. In Ju's words, "Do one or the other."

In the process users generate grammar to meet their new expressions like they would in a text message to a multilingual addressee. They mix and match words and because they can explain unknown words through Movable Type, they can express themselves more fluidly. This gave Ju a sense that she had more resources to express herself because she had many friends. Thus Ju, and in our observation the rest of the students, use multi-lingual elements through Movable Type to engage their words to talk to their audience so that they understand.

Ju employed translanguaging, and was comfortable with the multiple sources of language. The ability to mix and match in formal writing, like she already does in speech, supported her effort to construct multilingual texts through code and creative writing.

Satisfaction of Multilingual Communication

Angela was originally from the US, but spoke and wrote Mandarin, English, and German. She saw the communication of language as being like having multiple selves and the process of multilingualism deeply satisfying:

> Interviewer: What is satisfying about communication in more than one language?
>
> Angela: For me part of it is technical. Like I love it's something I can learn quickly, and apply that knowledge, to interactions. When you learn math, I guess it is exciting but, it's like 'YAY' now I can interact with this test. Maybe that's just me. But language, when you learn language it means you can interact with another person. It means you can be less dependent when you are in a place of that context. It's a bridge to another culture. The

> famous example with the Germans is *Schuld*, which means both debt and guilt. And it's true. Germans don't like debt.

The process of communicating in more than one language excited Angela. Because it allowed her to understand the underlying meanings and connection hidden within the words. It also excited her because language lets her connect to others. She found multilingualism was like having more than one self:

> Interviewer: Finnish this sentence, Being Multilingual is like...
>
> Angela: It's actually like have multiple personalities. It does kind of change the brain a little bit from language to language. My voice changes when I speak German. Maybe it's the movies influencing me. It is kind of like having multiple selves. It is not even just the language, but when you go to another country, it's easier to integrate into that life style if you can speak and act the language as well.

Like Ju found that having more words was like having more friends, Angela found having more languages is like having more selves. With these selves, she found even humor changes. In German, she authors a more sarcastic self. She complains more. She was not sure how she could justify what she said.

She felt the process of coding words which can change on click, *Movable Type*, is a more realistic expression of multilingualism than a monolingual classroom. She especially thought when we brought *Drown*, a multilingual story by Junot Díaz (1997), into class it helped to put the design, *Movable Type*, into some context. It helped her realize that multilingual communication, "is something we do all the time. This is what a lot of Americans have experienced in their lives, and it's like, I like the idea of Movable Type… I wish it was easier to code." She felt the idea behind *Movable Type* reflects something real about language; something about how we use it. She equated it to the communicative method of language learning, where it does not matter if you learn the grammar or the language perfectly. What matters is that you are supposed to be able to communicate. But sometimes, that means you switch into other languages to communicate.

She felt *Movable Type* is a better reflection of the notion of what communication looks like. Even when we all speak the same language we are not all speaking the same language.

> Angela: When you think about jargon and things like that, if a friend of mine came into some of my classes, they probably wouldn't understand half the things that were going on. And that sort of mix thing happens in every conversation.

She felt the process happens, even within the same language, much less without. She argued that the idea that anyone can ever be completely monolingual is not true.

Constraints and Affordances of Movable Type's Design in Multi-Lingual Creative Writing

The constraints of the design existed around its ease of use, and on the cognitive load (Sweller, 2011) that system entailed. Cecilia wrote a story, *Wizard of the Sea*. In it, she mixed German and English. Cecilia copies the

```
    <body >
   <span> <button{
    display:inline;}
class='pushme'>متحدة اللغات</button><script>
        $(function(){
   $(".pushme").click(function () {
      $(this).text(function(i, text){
          return text === "Language Unity" ? "متحدة اللغات" :
"Language Unity" ;
      })
   });
})
    </script></span><span>, a concept of coming together through
language, the expression is difficult. Microsoft Word does not
```

Figure 4: Multilingual composition that can be changed by user interaction with text.

source code for a multilingual phrase into the editor to begin the process of writing a multilingual component. As shown in Figure 4, the code chunk is long. This rests inside of the writing environment. We employed this method to ‘white box’ the code, allowing full remixing of the code by the user. This afforded Cecilia the opportunity to appropriate the material to her own needs. Once the chunk is pasted in, Cecilia rewrote the parts of the code to change the representation on click. Defining first what the chunk will say to start, and then what it will say on click.

The interface afforded remixability and a dynamic authoring environment. The technical nature of that adaptability, however, means the writing process becomes a technical endeavor. Some users found this inhibited their creativity. In future iterations we plan to focus more on the design of the experience to either target the coding segment, or facilitate the creative writing by providing a GUI for authors. Walking the line between these users’ needs seems like a reality in delivering a tool into the underserved need of multilingual authors.

Conclusion

Participants in a 7-week creative writing workshop used *Movable Type*, an open source multilingual writing technology, to author multilingual stories. During the process they engaged in translingual construction of texts with multiple audiences in mind. This provided a way for all of the different linguistic resources translingual youth have to find a place in their writing. Unlike the monolingual environments that formal writing often engenders, *Movable Type* opened up space for these multiple voices within the constructionist learning environment. This affordance addresses one of the social challenges of multilingual authoring in monolingual majority classrooms. The technical system, however, posed some challenges. The major one was the cognitive load (Sweller, 2011) of both authoring text while editing code. In this pilot study, we included coding to meet the needs identified in the the design phase, but due to the high load and potential to push out certain voices who are the target users, we believe developing a GUI only version along with a code exposed version will meet the divergent needs of our intended design more fully. We intend to develop this in the next iteration. Further, it was difficult to be creative and technical at the same time. In future iterations we will build on these early promising results for multilingual authoring to address the challenge of constructing multilingual narratives in formal writing environments.

Multi-modal, multilingual expressions are meaningful for today’s youth, not just youth who have experienced migration (Kim, 2016). As a result, we have designed an authoring environment that educators can use in their classroom today, using free software to aid students in creating and expressing. This design, we hope brings the globalized social practices of today’s youth into the educational environment, so that education practices can better keep up with youths’ innovations.

References

Black, R. W. (2009). Online fanfiction, global identities, and imagination. *Research in the Teaching of English, 43*(4), 397-425.

Curwood, J. S., & Gibbons, D. (2009). “Just like I have felt”: Multimodal counternarratives in youth-produced digital media. *International Journal of Learning and Media, 1*(4), 59-77.

Curwood, J. S., Magnifico, A. M., & Lammers, J. C. (2013). Writing in the Wild: writers’ motivation in fan-based affinity spaces. Journal of Adolescent & Adult Literacy, 56(8), 677-685.

Degges, R. (2016). How I Learned to Program. Retrieved from:

https://www.rdegges.com/2012/how-i-learned-to-program/

Domingo, M. (2014). Transnational language flows in digital platforms: A study of urban youth and their multimodal text making. *Pedagogies: An International Journal*, *9*(1), 7-25.

Foucault, M. (1979). *Discipline and punish: The birth of the prison*. New York: Vintage Press.

Gee, J.P. (2007). *Social linguistics and literacies: Ideology in discourses*. London: Routledge.

Gee, J. P., & Hayes, E. R. (2011). *Language and learning in the digital age*. Routledge.

Holland, D., Lachicotte, JR., W., Skinner, D., & Cain, C. (1998). *Identity and agency in cultural worlds*. Cambridge, MA: Harvard University Press.

Hull, G., & Schultz, K. (2001). Literacy and learning out of school: A review of theory and research. *Review of Educational Research*, *71*(4), 575–611. doi:10.3102/00346543071004575

Hutchins, E. (1995). How a cockpit remembers its speeds. *Cognitive science*, *19*(3), 265-288.

Kim, G. M. (2016). Transcultural Digital Literacies: Cross-Border Connections and Self-Representations in an Online Forum. *Reading Research Quarterly, 51*(2), 199–219.

Lammers, J. C., Curwood, J. S., & Magnifico, A. M. (2012). Toward an affinity space methodology: Considerations for literacy research. *English Teaching: Practice and Critique,* 11(2), 44-58.

Lee, C.D. (2001). Is October Brown Chinese? A cultural modeling activity system for underachieving students. *American Educational Research Journal, 38*(1) 97-141.

Olsen, D. (2015). *The Lean Product Playbook: How to Innovate with Minimum Viable Products and Rapid Customer Feedback*. John Wiley & Sons.

Orellana, M.F., Martínez, D.C., Lee, C.H., Montano, E. (2012). Language as a tool in diverse forms of learning. *Linguistics and Education, 23,* 373-387.

Rogoff, B. (2003). *The cultural nature of human development*. Oxford: Oxford University Press.
Said, E. W. (1979). *Orientalism*. Vintage.

Sweller, J. (2011). Cognitive load theory. In *Psychology of learning and motivation* (Vol. 55, pp. 37-76). Academic Press.

Thorne, S., Black, R., & Sykes, J. (2009). Second Language Use, Socialization, and Learning in Internet Interest Communities and Online Gaming. *The Modern Language Journal, 93*, 802-821.

Wertsch, J. V. (1991). *Voices of the mind: A sociocultural approach to mediated action.* Cambridge, MA: Harvard University Press

Patches as an Expressive Medium for Agent-Based Modelling and Programming

Corey Brady, *corey.brady@vanderbilt.edu*
Department of Teaching and Learning, Vanderbilt University, Nashville, TN, USA

Lauren Vogelstein, *lauren.e.vogelstein@vanderbilt.edu*
Department of Teaching and Learning, Vanderbilt University, Nashville, TN, USA

Abstract

Agent-based modelling (ABM) is a powerful approach for simulating complexity and for understanding the emergent phenomena core to multiple disciplines across the physical and social sciences (Wilensky, 2001). ABM is thus often understood as an innovation in STEM education, providing a representational infrastructure for understanding complexity by "growing it" (Epstein & Axtell, 1996; Wilensky & Papert, 2010). While this is certainly true, we argue that expressive and artistic uses of "swarms" of computational agents can *also* provide accessible entry points for learners and can support them in developing a range of intuitions about the kinds of phenomena that they might simulated with ABM. This offers a "STEAM" oriented introduction to modelling, connecting *artistic* perspectives with scientific perspectives in fundamental ways.

In this paper we describe the iterative design and implementation of activities that highlight the expressive potential and social syntonicity (Brady et al, 2016) of one of the fundamental types of agent in the ABM toolkit (the "patches"). We describe a setting in which we have done design-based research over two years, in summer camps (entitled "Code Your Art") and school-year activities involving rising fifth through eighth grade students (participants aged from 10-15) attending school in a mid-sized urban district in the southeastern USA with a high proportion of traditionally underserved and minoritized youth. Our research questions were:

1. How can we cultivate mappings between *patches* and *pixels* as a provocation for young learners new to programming and ABM to create personally-meaningful visual effects?
2. How can we cultivate mappings between *people* and *patches* as a provocation for youth new to ABM to think ambitiously about designing and creating *dynamic* visual effects?

We present designs two iterations of this camp and in school-year implementations in the classrooms of partner teachers, where investigation of the research questions has continued.

Looking toward future work, we suggest that this approach to making ABM ideas accessible also has a "high ceiling," closing the paper with currently in-process work in the present school year at the schools of partner teachers, that aims to answer the following question:

3. How might the two above provocations to expressive work with patches combine to produce performance phenomena that could be put in conversation with topics in distributed computing normally considered advanced, including 2D cellular automata, fuzzy logic, and the emergent behaviour of computational systems?

This work contributes to the field as a proof of the feasibility of introducing the powerful infrastructure of ABM to young learners through artistic expression using the full text-based interface of NetLogo. Students drew on personal interests and social dynamics to make sense of and develop understandings of key ideas about agent-based representations. Future directions envision much larger groups of learners (stadium-sized collectives) to construct distributed computing environments that exhibit emergent properties in real dynamic displays.

Keywords

Agent-based modelling; Computational thinking; Group-based design; Digital expressivity

Introduction

Agent-based modeling (ABM) offers a powerful representational infrastructure (Hegedus & Moreno-Armella, 2009) for simulating and understanding emergent phenomena and complex systems (Wilensky, 2001; Wilensky & Rand, 2015). ABM *also* provides a medium that can be used to create computational art of various kinds. We argue that such expressive uses of the ABM toolkit can both be intrinsically valuable for learners and also serve as an entry point for learning about the agent-based approach. We present designs from the first two years of an ongoing project that engages young learners artistically with a grid of programmable *patches*, one of the representational building blocks of ABM.

Research Objectives

In motivating coding and mathematics through artistic expression, our goal is to identify entry points that might make some of the powerful ideas of ABM more accessible. We sought to stabilize these entry points in activities that could be reliably enacted with students in a variety of settings and with different facilitating teachers. We wanted these activities to place key ideas of ABM in a context where learners could build familiarity with them, guided by desires to create compelling visual effects. In taking a patch-centered view on this enterprise, we were guided initially by the following research questions:

1. How can we cultivate mappings between *patches* and *pixels* as a provocation for young learners new to programming and ABM to create personally-meaningful visual effects?
2. How can we cultivate mappings between *people* and *patches* as a provocation for youth new to ABM to think ambitiously about designing and creating *dynamic* visual effects?

As we have begun to see the independent expressive potential for collectives in playing the role of a patch grid, we have formulated an additional question for future work:

3. How might the two above provocations to expressive work with patches combine to produce performance phenomena that could be put in conversation with topics in distributed computing normally considered advanced, including 2D cellular automata, fuzzy logic, and the emergent behaviour of computational systems?

Literature Review

In this section we situate our work on the expressive potential of patches within the constructionist literature on agent-based modeling, in terms of three themes.

Our selection of NetLogo (Wilensky, 1999) was a pivotal design decision. We chose it as our construction environment from out of a wide range of alternatives that might be seen as better oriented to the age (middle school) and experience level (beginners) of our participants. We made this choice in part due to NetLogo's ability to produce images from the computational state of a grid of agents. This aligned with our aims to integrate graphical design with computational thinking and with reasoning about agent-aggregate relations and emergence. We also favored NetLogo in part due to its "high ceiling" nature, which ensured that expertise and agent intuitions developed by learners would provide them enduring connections to rich tools for thinking.

The two[33] principal types of agents in NetLogo are turtles and patches. The *turtle* is the more familiar type among constructionists, introduced as it was in the original LOGO. Our project's patch orientation set us apart from this line of work. We focused on the expressive visual potential of a grid of *immobile* agents that can be imagined in terms of the panels of a quilt or the pixels of a screen. Unlike turtles, these *patches* have *fixed* Cartesian coordinates and uniform shape (squares); but like turtles, they can change colour and hold variables. The turtle as a computational object has been extolled for various kinds of *syntonicity* (Papert, 1980)—attach points for learners

[33] Links are in fact a third type of agent, also treated as "first-class" in NetLogo. But a link can exist only as a connector between two other pre-existing agents.

to identify with the turtle and use its perspective as a lens on problems and phenomena. One of the challenges we set ourselves in this project was to identify ways that a learner or group of learners might identify with and think through the group of patches.

Below, we outline research statements that guide our enterprise of introducing young learners to ABM through expressive uses of NetLogo's patch grid, drawing sources from work on agent-based computing in particular and constructionist thought more broadly.

A characteristic perspective on entry points

Research within the Constructionist tradition (Papert & Harel, 1991), has been interested in engaging learners in computational thinking both (a) in focused, topic-specific explorations, using the design construct of microworlds (Papert, 1980), and (b) in broader, more discipline-general ways with open software environments and construction kits that help learners explore the generativity of systems of powerful ideas (Papert, 1980). Indeed, an important aspect of the power of constructionist environments is that they enable transitions back and forth between constructing *within* microworlds to the construction *of* microworlds, whether designing learning environments for others (Harel & Papert, 1990) or creating as objects-to-think-with (Papert, 1980) for oneself (or both of these at the same time).

This strategy—of establishing entry points that learners do not 'use up' but rather return to with new perspectives and purposes—can be seen as an instance of a general principle of constructionist design. Here, assuming the standing point of agent-based modeling, we outline several pairings of this kind that are relevant to our research questions and to the enterprise of connecting artistic expression with patches to broader adoption of an agent-based perspective.

Playful, artistic expressivity and scientific inquiry

Though the LOGO turtle was a deliberate *restructuration* (Wilensky & Papert, 2010) of Euclidean geometry and thus born as a citizen of MathLand (Papert, 1980), it was also immediately and enthusiastically offered to learners as an expressive partner with which to engineer beautiful and visually compelling creations. Turtles have thus been used to explore forms of artistic expressivity that make use of the capabilities of these agents, in environments such as TurtleArt (Papert, 1980; Bontá, Papert, & Silverman, 2010) and more recently, Scratch (Resnick et al, 2009). A fundamental idea behind this strand of the LOGO tradition is the notion that creating turtle graphics or producing media with turtles can introduce learners to the core principles of turtle geometry and/or programming in playful ways that foreground the construction of personally meaningful artifacts (Papert, 1980; Papert & Harel, 1991).

Playfulness is an enduring and generative feature of constructionist inquiry, and powerful ideas (Papert, 1980) are understood in the constructionist aesthetic to overlap strongly with *wonderful* ideas (cf, Duckworth, 2006). Moreover, in the specifically multi-agent setting of NetLogo and ABM, the legitimacy of exploring the expressive range of agents and their aggregate representations is captured in Wilensky & Rand's (2015) assertion of the value of "exploratory modeling" as a complement and partner to "phenomena-based modeling." We thus felt on solid ground in fostering explorations of the range of compelling and visually appealing effects that students might create with the patch grid.

Participatory (intrinsic and shared) and reified computational (externalized and shareable) representations

To help learners make sense of their ideas towards creating or using computational representations, constructionist researchers have often advocated that they enact their thinking and animate agent behaviours and interactions through role-play. In both single-turtle and multi-turtle programming, physical, embodied simulation has thus provided critical entry points for learner. For instance, the practice of *playing turtle* (Papert, 1980) supported LOGO learners in understanding the turtle's turning actions while drawing closed polygons. Imagining an agent's experience of its environment and its behavior in interactions led to *embodied modeling* of systems (Wilensky & Riesman 2006); and collective role-play enabled groups of learners gain conceptual traction in modeling groups of turtles in activities called *star people* (Resnick & Wilensky, 1998) and *participatory simulations* (Brady et al, 2017; Colella et al 1999; Wilensky & Stroup, 1999).

As with our other dialectical pairs, constructionists do not necessarily think of participatory models as being "used up" on the way to building a reified computational artifact. As important as an external, runnable, and shareable computational representation is, this is not necessarily the single *telos* of modeling work. The concept of *syntonicity*, mentioned earlier, provides an enduring connection between participatory and externalized representations, supporting learners in assigning meaning to computational results and maintaining the connections to agents that help them to reason about models' behavior under new conditions. And representations at these two extremes support the investigative moves that Edith Ackermann (1996; 1999) called "diving in" and "stepping back."

Moreover, we have been interested in the enduring value that enactive representations have of their own, supporting learners in inferential discussion, in running new "socially distributed" experiments, and in reasoning about possibilities for change in the represented systems (Hjorth, Brady, & Wilensky, 2018; Reimers & Brady, 2019; Vogelstein, Brady, & Hall, 2017; 2019). In the Code Your Art camp activities, we set ourselves the challenge of giving classroom-sized groups of students shared and meaningful experiences with the NetLogo patch grid. We have reported on the diversity of ways our teachers took up and facilitated such patch participatory simulations (Vogelstein & Brady, 2019), and the aim is to continue to explore this activity space as a realm for experiences that are worth returning to and reasoning about.

These three themes – creating compelling entry points that are rich enough to be worth returning to; seeing artistic expressivity and conceptual depth as compatible and mutually supportive; and making use of groups of learners to create shared experiences of producing phenomena relevant to shared inquiry – drove many of the designs we present in this paper.

Methods and Data Collection

The activities described here occurred within the Computational-Thinking And Mathematics Play Spaces project, or CAMPS, which uses design-based research (Cobb et al, 2003) to identify ways of exploring computational thinking and mathematics with middle school learners in an expressive, artistic environment. Our first design iteration consisted of a one-week (five-day) free summer camp for middle school students, held in a middle school in a southeastern U.S. city.

After building a proposed curriculum, we worked with four teachers from the public schools of the district during an intensive one-week professional development and co-design workshop. The teachers alternated between engaging with activities in the student role, suggesting and testing adaptations, and working through their facilitation plans. During the camp itself, the four teachers worked in two pairs, each with their own group of students. They facilitated the camp activities themselves, with technical support as needed from the research team. During the intervening academic year, two of the four teachers invited the research team into their classrooms, and one started an after-school coding club. In the second summer, all four of the teachers returned and recruited another four. The second year's professional development workshop thus paired a returning teacher with a newcomer, and tackled both minor and major revisions to the camp curriculum. In the minor-revision camp ("Image Camp"), we expanded on the Year 1 camp with iterative refinements based on student work and school-year findings. In the major-revision camp ("Action Camp") we introduced a new curriculum, created in partnership with professional dancers, and foregrounding NetLogo *turtles* as opposed to patches. In this paper we confine ourselves to the design trajectory that produced the Image Camp, though aspects of our future work are informed by the design work to place professional artists' perspectives and practices in conversation with agent-based representation.

The camp was titled "Code Your Art," and advertised as involving computer programming and art for rising 6^{th}, 7^{th}, and 8^{th} grade students. In camp sessions, multiple forms of data were collected, which support our design reflections here. Consenting students were interviewed at the start and end of the week and completed pre- and post-questionnaires. During activities, consenting students were captured in screen-recordings, and multiple cameras (fixed cameras and body-mounted GoPros) documented 'offline' activities in each classroom. Finally, students' projects were captured as digital files and through the shared practice of "publishing" one's work to a classroom-specific online "gallery." These data sources have provided invaluable records of the experience of our activities, informing ongoing design and analysis.

Results

In this section, we describe designs related to our first two research questions: first, emphasizing connections for learners between patches and pixels; and second, between patches and people.

Patches and pixels.

Designs that encouraged learners to connect patches and pixels focused on image construction and manipulation. In both of our camps, we began with constructions using Perler™ beads (heat-fusable coloured plastic beads that students can lay out on a grid and form into a solid, unified construction by applying a hot iron). Alongside these physical constructions, students also could work with a virtual Perler bead design NetLogo model, with the same grid size (29x29) as the Perler bead frame. This permitted students to create paired constructions, as well as to begin their coding-based image manipulation work from an initial design shared with their physical construction.

Figure 1. Sample student Perler Bead Design constructions from year 1, as posted to the class galleries

These activities gave students the basic notion of images as comprised of pixels, each in a particular location and, for a fixed image, each having a particular colour.

Over the week, students engaged with, adapted, and modified effects—many using patches alone; others using turtles and patches in coordination. Effects using patches alone included logic that changed patches' colours, based on colour their current colour. This category included, code such as:

```
ask patches [ set pcolor pcolor + 10 ]
```

or

```
ask patches with [ pcolor = blue ] [ set pcolor red ]
```

Students made their own adaptations to such code, motivated by the challenges of new images or desired visual effects. For instance, they changed constants (*e.g.,* set pcolor pcolor + 2), or, later, added logic to select patches whose pcolors were in a range (*e.g.,* ask patches with [pcolor > 100 and pcolor < 110] [set pcolor pcolor - 90]). Such changes were particularly important as participants shifted to images that they collected from the internet.

A second category of effect involved changing patches' colours based on their coordinate locations. And a third involved using patch-owned variables to store multiple colours in each patch, allowing them to participate as pixels in multiple images. This enabled students to create patch performances that involved stop-motion animation or composite images in which different regions showed corresponding pixels from different images (see Figures 2, 3).

Figure 2. Five stills from a Year 1 final project, beginning with a pair of images with similar shape, and using patch memory and turtles to create a performance that composed and decomposed the images.

Turtles entered the picture in part as messengers or transmitters of colour from one part of a composition to another. These ideas supported many students in creating dynamic effects and transitions.

Figure 3. Still from a Year 1 project, using a high-resolution image with turtle and patch effects to "dematerialize" part of the image.

As described below, we focused in the school year on foregrounding and exploring the power of mathematical operations in the patch context. Our intention was to support students in creating "interactive image performances" and making these interactions open-ended and contingent on user interaction or randomness. We also aimed supporting a trend we noted of taking "actions on images" again in a way that foregrounded mathematical transformations.

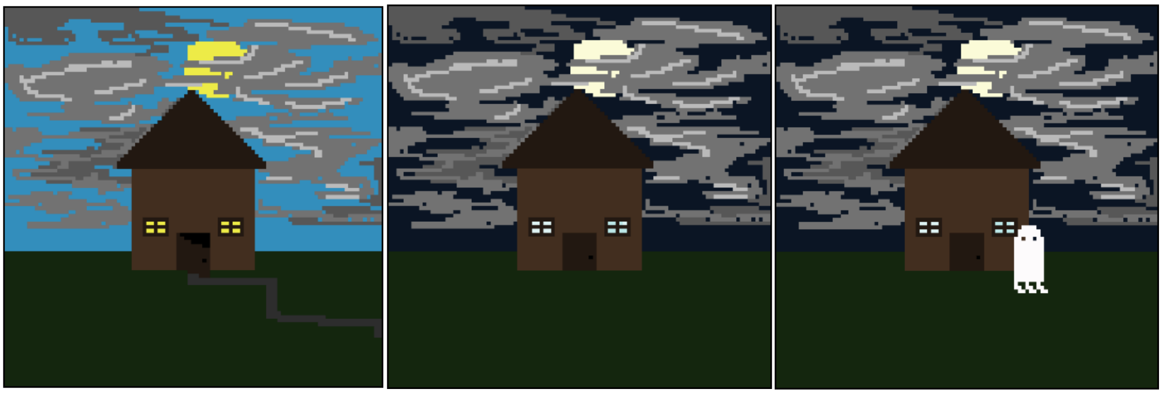


Figure 4. Three stills from a Year 2 final project, that began with a drawn image and used a variety of transformations to achieve effects such as the change from day to night, and haunting.

Patches and people.

The connections we encouraged between patches and *pixels* aimed to foster a view of the NetLogo environment as a 'canvas' for dynamic graphic design, visual effects, and interactive art. The connections we encouraged between patches and *people* aimed to support strategies for conceptualizing the patch agentset and orchestrating their collective action.

Year one experiences

One of the most dramatic examples of connecting patches with people was our use of "stadium cards" activities. Based on the "card stunt" events done in large sporting arenas (see Figure 5), stadium cards activities asked a classroom group of students to populate the patch grid. In the course of the five-day camp, we did stadium cards activities on the first three days. Through these activities, we introduced patches as agents, highlighting (a) that they could be spoken to as a whole group (agentset) or in sub-groups using a "with" selector; (b) that they had changeable *pcolor*, where the colour possibilities were indexed by a number; (c) that they had fixed location, indexed by *pxcor* and *pycor* coordinates similar to a Cartesian plane; (d) that while they could not get a view of the whole grid, they could see and communicate with their *neighbors*; and (e) that they could *sprout* turtles, who could "take on" their colour if asked.

Figure 5. Card stunts, shown at increasing scales.

Our two classrooms from the Year 1 camp implemented stadium cards activities in very different ways. Although both approaches were consistent with the goals expressed in the professional development sessions, each highlighted different aspects of the "patch experience." As described in Vogelstein & Brady (2019), one approach to the activities foregrounded the limited knowledge of the patches, placing students inside the grid, holding their colour-cards (Figure 6).

Figure 6. Students as patches in a Stadium Card activity. Left: presenting colours. Right: changing colours.

An alternative approach placed students on the “sidelines” of the patch grid, laying their colour-cards on the floor. This arrangement focused on the limited *agency* of the patches, while affording them a view of the whole construction (Figure 7).

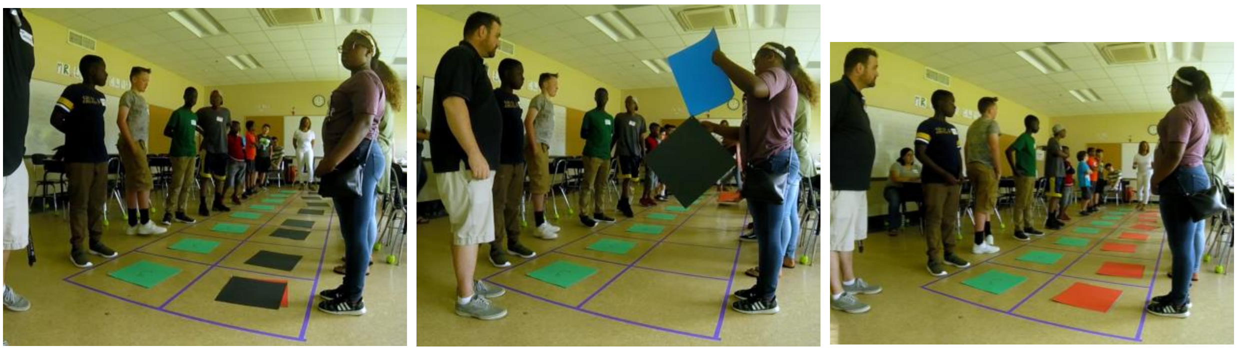


Figure 7. Stadium Card patch grid responding to: ask patches [if pcolor = black [set pcolor red]].

Year two experiments

In our school-year work and in the Year 2 camp, we aimed to solidify promising directions that had emerged in our first-year explorations. Both our successes and our failures in this regard were worthy of mention.

Foregrounding mathematics and generative syntax

In an effort to foreground mathematical manipulations and their power as tools for patch effects, we designed and tested an approach to introducing new NetLogo syntax that foregrounded what we referred to as “combinatorial play” and that facilitated the diffusion of ideas in the class. To do this, we used a collaborative version of NetLogo called GbCC (Brady et al, 2018), which offered a public interactive gallery for sharing and remixing work.

For example, early on in the first day of the school-year activities, we introduced the command to the agentset of patches:

patches> if pxcor = 2 [set pcolor red]

Students typed this code in to their version of the activity interface, to confirm that their screen looked like the instructor’s. They were asked to make a change to one part of that line of code, and to investigate the difference in the effect, and repeat. Whenever they created an interesting effect, they were told to Share their results, publishing their screen and the associated code. As students’ results appeared in the gallery, they “bootstrapped” each other in exploring variations such as:

patches> if pxcor = 2 [set pcolor yellow]

patches> if pxcor > 2 [set pcolor red]

patches> if pycor < 5 [set pcolor orange]

patches> if pxcor = pycor [set pcolor blue]

patches> if pcolor = orange [set pcolor yellow]

Because the effects of commands accumulated on the patch grid, the designs in the gallery became increasingly intricate as the activity progressed. (See Figures 8, 9.)

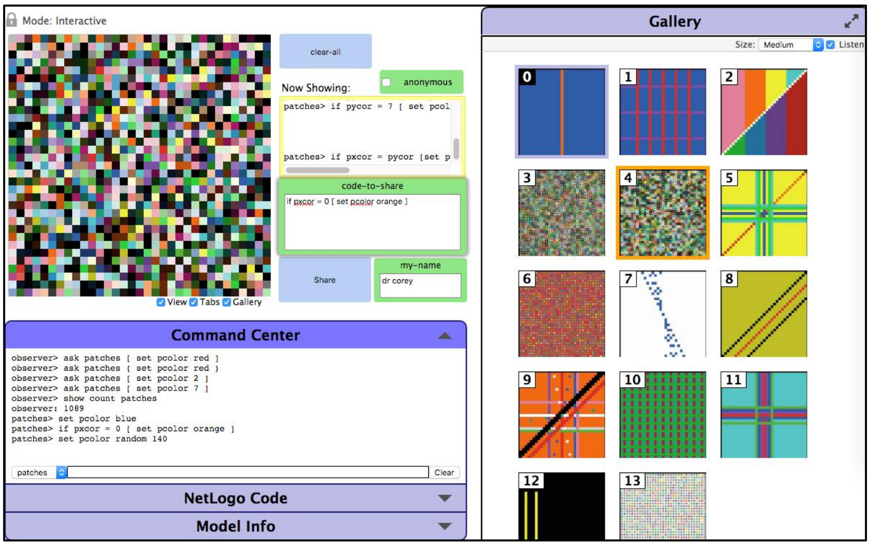


Figure 8. Gallery entries in the coding club implementation of the collaborative patch code introduction. Complexity accumulates.

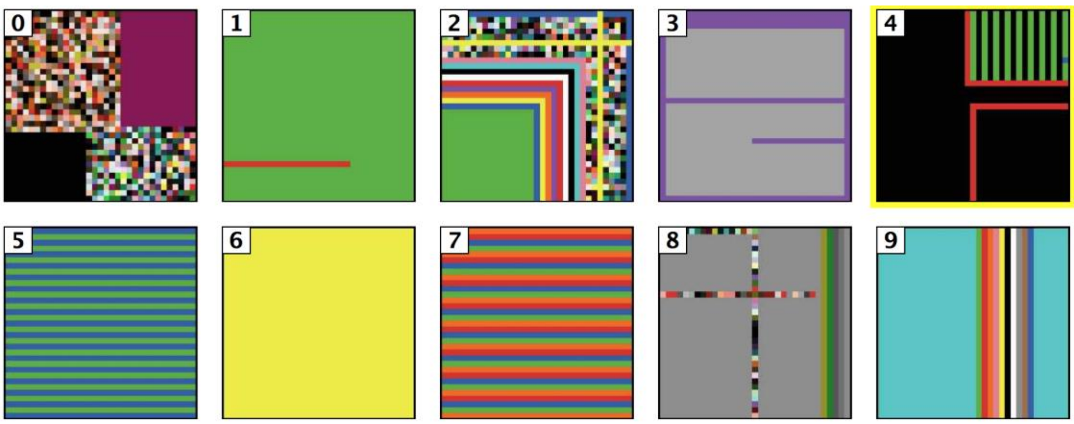


Figure 9. Ten gallery entries from late in the opening collaborative session with patches. By selecting Canvas #8, any student could discover that the top horizontal element was produced by the code: ***patches>*** *if pycor = 16 and pxcor < 0 [set pcolor random 130]*

We also designed collective activities in which the classroom group of students could move about virtually in the Cartesian space in real time. We saw these as virtual complements to the stadium card activities on one hand, and to the "combinatorial play" activities on the other, all intended to support early exposure to the patch grid. With each student controlling a point, the facilitator would propose coordinate-based rules. For instance, in Figure 10, the rule is "find patches whose pxcor equals -3." The students' points suggest the pattern that this rule will produce, which they can then explore in their own space using NetLogo syntax. For example, the middle screen shows a student asking patches, "if pxcor < -3 [set pcolor blue]" and then, in the right-hand screen "if pxcor = -3 [set pcolor lime]." We designed this collaborative activity as a way to promote students' conceptual understanding of coordinate-based patch commands.

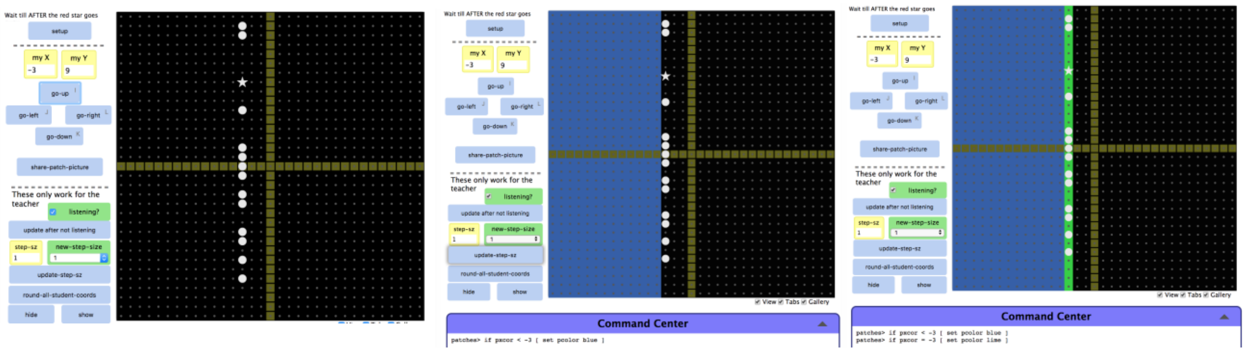


Figure 10. Collaborative Cartesian activities. All students see themselves as star-shaped icons, and their peers as circles. They move with arrow keys, and see the coordinates of their current point. In the background of this point-based exploration, they can issue commands to the patches.

Stadium cards in Year 2

Our adjustments to the stadium card activities were less uniformly successful. In Year 2 of the camp, our room layouts changed, with students working at large seminar-style tables and no space in the room for a permanent taped-out patch grid (Figure 11). Though the students did execute challenging prompts, the activities seemed less like physical performances, perhaps because the students remained in their work areas during this version of the stadium cards activities. This robbed the activities of some of their charm, which made us realize the value of the activities' performative side.

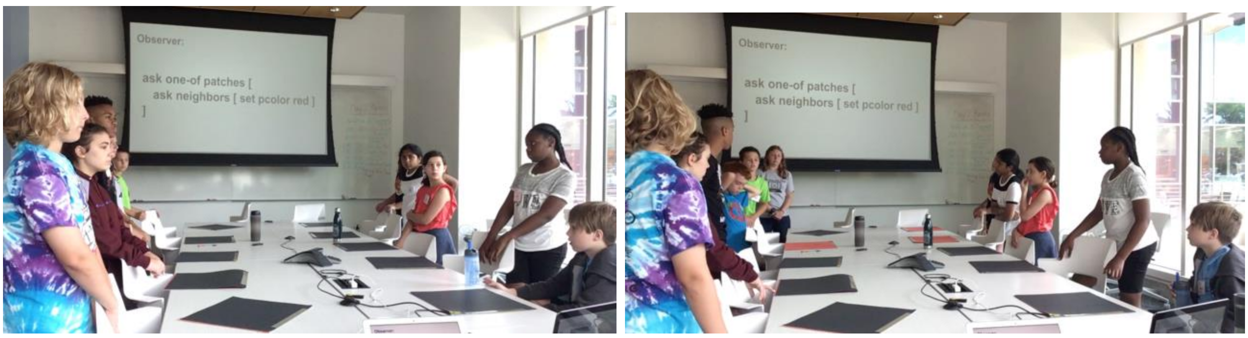


Figure 11. Year 2 stadium cards. Room space constraints led us to use the seminar tables as the surface of our patch grid.

Discussion

In exploring the expressive potential of NetLogo patches, we have been drawn to activities that have foregrounded performative aspects. This has been true for primary artifacts (increasingly encouraging students to think of computational logic as producing patch performances), and also for introductory activities (our collaborative gallery-based activities leveraged performative value, while our year-2 version of stadium cards lacked theatricality and suffered as a result).

In future work, we are interested in pushing this direction further with stadium cards. We began designing these activities, following a reference to a familiar social phenomenon (Card Stunts). However, taking the artistic potential of Card Stunts seriously as an *independent* activity may establish a new relation to the ABM infrastructure and suggests exciting future directions for STEAM work.

In particular, in NetLogo patches are (a) perfect rule-followers, (b) absolutely limited in their knowledge of other agents, (c) timely in their rule-execution (whether synchronized or in random order), and (d) indifferent to the emergent outcome. These assumptions are fundamental to the use of NetLogo to study computational phenomena such as 2D cellular automata (cf Conway, 1970), and changes in the assumptions cause significant changes in the emergent behavior on the patch grid.

However, when *human* agents comprise a patch-like grid, at least two key differences appear. On one hand, the assumptions above may need to be "softened" or even abandoned, as humans are unable to meet these conditions. On the other, human versions of such performances may be able tap into humans' unique capabilities, making the "patch" grid a medium for the human collective to express patterns of its own design or improvisation.

To investigate these two lines of work more fully, we are working with two schools to enact card-stunt inspired performances. Our first experiments will involve a 120-student seventh grade class, all with two-sided cards (blue and white) in a gymnasium with a 5-row set of bleachers (Figure 12). To explore the first line of inquiry, we will ask students to enact dynamic rules that are known to be sensitive to execution order on automated patch grids. For instance, "copy your neighbor to the right" or "make your colour the opposite of the majority of your neighbors." And to explore the second, we will present the students with a live video feed of their own collective state. We are curious about what behaviors and patterns may emerge, both spontaneously and in response to underdetermined or ambiguous prompts such as "make half of the grid blue" or "make the grid just a little bit blue."

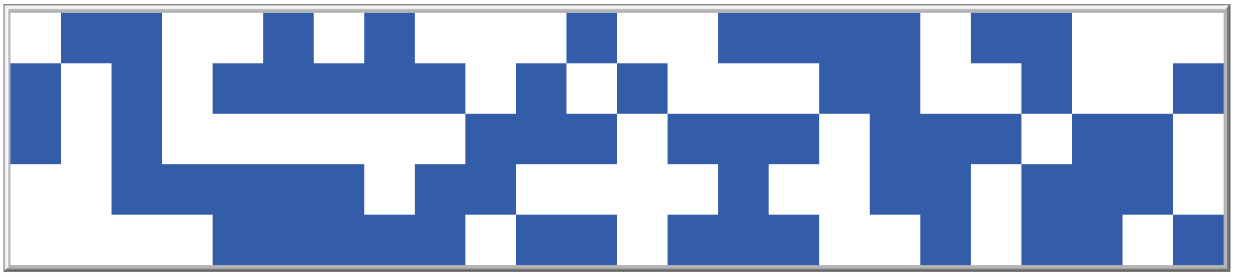


Figure 12. The size of the seventh-graders' patch grid (5x24).

Exploring what human patch grids can do, may provoke research in extended cellular automata: the behavior of the humans might then be placed in conversation with the behavior of automated agents that violate some of the assumptions above. For instance, fuzzy logic applied to cellular automata is an active area of research (cf Gerakakis et al 2019; Sisodia et al, 2018). From this perspective, comparisons of automated versus human grids of patches provide opportunities to inquire into mechanism as well as to explore the robustness of emergent behavior when assumptions are relaxed.

From the perspective of collective performances, card stunts with visual feedback allows us to explore the potential in turning the patch grid over to the patches. And participating in or witnessing such performances could motivate programmers to explore new visual effects.

Acknowledgments

This material is based in part upon work supported by the US National Science Foundation under Grant No. 1742257.

References

Ackermann, E. (1996) Perspective-Taking and Object Construction. In Y. Kafai, & M. Resnick (Eds.) *Constructionism in Practice* (pp.25-37). Mahwah, NJ: Lawrence Erlbaum.

Ackermann, E. (1999). Enactive representations in learning: pretense, models, and machines. *Learning Sites: Social and technological contexts for learning, Elsevier*, 144-154.

Bontá, P., Papert, A., & Silverman, B. (2010). Turtle, art, turtleart. In *Proc. of Constructionism 2010 Conference*.

Brady, C., Weintrop, D., Anton, G., & Wilensky, U. (2016). Constructionist learning at the group level with programmable badges. In *Proc. Constructionism Conf* (pp. 61-68).

Brady, C., Orton, K., Weintrop, D., Anton, G., Rodriguez, S., & Wilensky, U. (2016). All roads lead to computing: Making, participatory simulations, and social computing as pathways to computer science. *IEEE Transactions on Education*, *60*(1), 59-66

Brady, C., Stroup, W., Petrosino, A., & Wilensky, U. (2018). Group-based Simulation and Modelling: Technology Supports for Social Constructionism. Proceedings of Constructionism 2018, Vilnius, Lithuania (August 21-25).

Cobb, P., Confrey, J., DiSessa, A., Lehrer, R., & Schauble, L. (2003). Design experiments in educational research. *Educational researcher*, *32*(1), 9-13.

Colella, V. et al. (1998). Participatory simulations: using computational objects to learn about dynamic systems. *Proceedings of SIGCHI 98,* New York: NY, 9–10

Conway, J. (1970). The game of life. *Scientific American, 223*(4), 4.

Duckworth, E. (2006). *The having of wonderful ideas and other essays on teaching and learning*. Teachers College Press.

Eisenberg, M. (2003). Mindstuff: Educational technology beyond the computer. *Convergence*, *9*(2), 29-53.

Elumeze, N. & Eisenberg, M. (2005). SmartTiles: Designing Interactive “Room-Sized” Artifacts for Educational Computing. *Children Youth and Environments*, *15*(1), 54-66.

Epstein, J. & Axtell, R. (1996). Growing Artifical Societies: Social Science from the Bottom Up . Washington: Brookings Institution Press.

Gerakakis, I., Gavriilidis, P., Dourvas, N. I., Georgoudas, I. G., Trunfio, G. A., & Sirakoulis, G. C. (2019). Accelerating fuzzy cellular automata for modeling crowd dynamics. *Journal of Computational Science*, *32*, 125-140.

Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive learning environments*, *1*(1), 1-32.

Hegedus, S. J., & Moreno-Armella, L. (2009). Intersecting representation and communication infrastructures. *ZDM*, *41*(4), 399-412.

Hjorth, A., Brady, C., & Wilensky, U. Sharing is Caring in the Commons – Students’ Conceptions about Sharing and Sustainability in Social-Ecological Systems. In *Proceedings of Constructionism* (pp. 263-273). Vilnius, Lithuania.

Ntinas, V. G., Moutafis, B. E., Trunfio, G. A., & Sirakoulis, G. C. (2017). Parallel fuzzy cellular automata for data-driven simulation of wildfire spreading. *Journal of computational science*, *21*, 469-485.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc..

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, *36*(2), 1-11.

Reimers, J. & Brady, C. (2019). Maggie Mars: Theatrical Modeling and the Phenomenological Understanding of Solar Systems. In *Proceedings of CSCL* (pp. 704-707).

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, *52*(11), 60-67.

Resnick, M., & Wilensky, U. (1998). Diving into complexity: Developing probabilistic decentralized thinking through role-playing activities. *Journal of the Learning Sciences*, *7*(2), 153-172.

Sisodia, P. S., Shekhawat, R. S., & Tiwari, V. (2018). Modelling urban sprawl using fuzzy cellular automata model for the city of Jaipur. *International Journal of Society Systems Science*, *10*(4), 285-299.

Vogelstein, L., & Brady, C. (2019). Taking the Patch Perspective: A Comparative Analysis of a Patch Based Participatory Simulation. In *Proceedings of CSCL* (pp. 512-519).

Vogelstein, L., Brady, C., & Hall, R. (2019). Reenacting mathematical concepts found in large-scale dance performance can provide both material and method for ensemble learning. *ZDM*, *51*(2), 331-346.

Vogelstein, L., Brady, C., & Hall, R. (2017, June). Mathematical reflections: The design potential of ensemble performance. In *Proceedings of the 2017 Conference on Interaction Design and Children* (pp. 583-588).

Wilensky, U. (2001, August). Modeling nature's emergent patterns with multi-agent languages. In *Proceedings of EuroLogo* (pp. 1-6).

Wilensky, U., & Papert, S. (2010). Restructurations: Reformulations of knowledge disciplines through new representational forms. *Constructionism*.

Wilensky, U., & Rand, W. (2015). An introduction to agent-based modeling. *Modeling Natural, Social, and Engineered Complex Systems with NetLogo*, 504.

Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—an embodied modeling approach. *Cognition and instruction*, *24*(2), 171-209.

Wilensky, U. & Stroup, W. (1999). Learning through participatory simulations: network-based design for systems learning in classrooms. Proceedings of Computer Supported Collaborative Learning (CSCL'99). Stanford, CA.

Performances-to-think-with: Theatrical modeling of solar systems and celestial phenomena

Jackson Reimers, *jackson.e.reimers@vanderbilt.edu*
Department of Teaching & Learning, Vanderbilt University, Nashville, USA

Corey Brady, *corey.brady@vanderbilt.edu*
Department of Teaching & Learning, Vanderbilt University, Nashville, USA

Abstract

Solar systems present a unique opportunity to study student perspective-taking, a topic of importance in both school and non-school learning sites. We seek to understand how communities of learners might themselves constitute their own representational apparatus for this knowledge-seeking endeavor through techniques and frameworks of participatory theater and agent-based scientific modeling. Building on a view that participatory theater is a socially distributed way of engaging participant perspectives, we imagined a kind of participatory theater that groups might turn to when they had *scientific* questions requiring perspective-taking. We have called such activity "theatrical modeling."

We aim to support learners' engagement in the representational and perspective-taking practices we see as common to both theater and modeling, while acknowledging and leveraging the tension between them as potentially generative sources of change in students' lives, in formal science education, and in constructionism learning environments elsewhere. Through a sequence of design studies with participants in a public middle school STEM classroom in the mid-South USA, we explored these tensions and resonances. This paper gives an overview of the study, using two contrasting cases to illustrate how learners engaged in perspective-taking practices in theatrical modeling activities. Findings suggest that the theatrical frame provides a range of activity structures that students are both willing and able to engage with in generative ways.

Celestial bodies and their relations within solar systems seem to afford an imaginary space rich enough for students to explore in a variety of ways. Moreover, theatrical modeling activities seem to support multiply syntonic learning, enabling students to engage in diverse kinds of consequential perspective-taking. We close with considerations about the uses of CHAT as an analytic lens and design framework, paying specific attention to its affordances for designing and describing hybrid forms of activity with the potential to transform school science practices. This relatively new design work is inspired by constructionist learning activities, and we invite others to consider how generative parallels between participatory theater and agent-based modeling could re-shape practices around publicly shared artifacts.

Keywords

theatrical modeling; participatory theater; agent-based modeling; hybridity; design-based research; perspective-taking; syntonicity

Problem statement

Solar systems present a unique opportunity to study student perspective-taking, a topic of importance in both school and non-school learning sites. On one hand, solar systems are complex systems that give rise to phenomena that fundamentally depend on perspective. On the other hand, they can be captured using a relatively small number of interacting bodies, enabling small groups of learners to use theatrical approaches to enact and study the behavior they are working to understand. Another opportunity arises when we consider how forms of representational activity most commonly thought to belong to the humanities, namely theater, interact with modes of scientific inquiry that are typically brought to bear on complex systems: computational and embodied modeling. Focusing heavily on collaborative and performative forms of activity, we seek to understand how communities of learners might themselves constitute their own representational apparatus through techniques and frameworks of participatory theater, collaborative storytelling, and agent-based modeling. Moreover, we hope to leverage these insights to better understand what happens when typically distinct, but potentially complementary, socially situated genres of activity are combined in a third setting.

Research objectives

The study explores the potential of theatrical activity designs for modeling solar systems with an emphasis on phenomena that depend upon the adoption or coordination of perspectives. Building on a view that participatory theater is a socially distributed way of engaging participant perspectives, we imagined a kind of participatory theater that groups might turn to when they had *scientific* questions requiring perspective-taking. We have called such activity “theatrical modeling.” We did not begin this study with a pre-formed set of principles for doing theatrical modeling; rather, our intent was to understand the practical and theoretical virtues of such a hybrid, as well as the mechanics by which it could develop – both for our participants and for ourselves. To realize this intent, we delineated a design space that can be summed up in the following activity system diagram. We drew on this design space to establish conditions under which learners themselves would generate diverse representational forms in the course of ongoing classroom activity, each time inventing new forms of theatrical modeling.

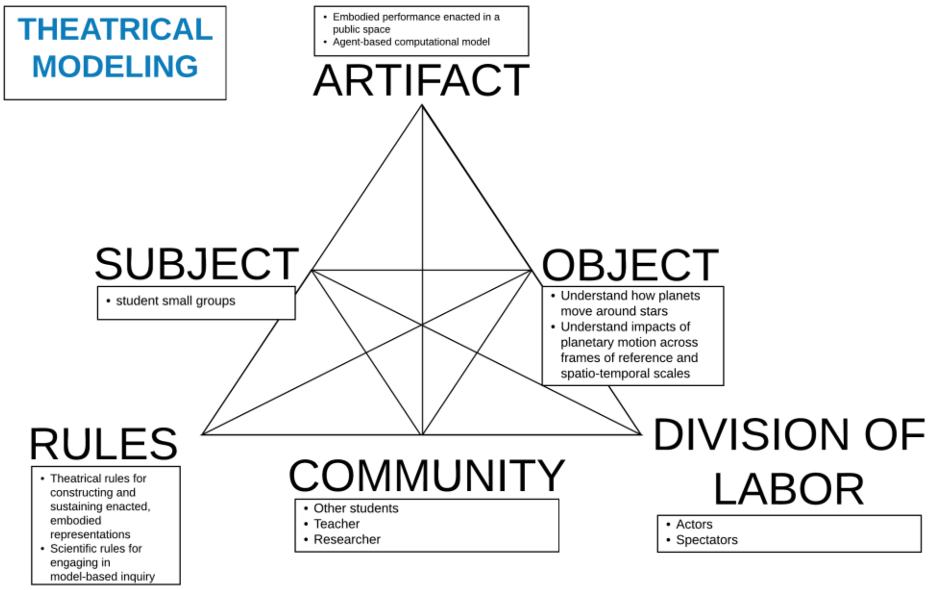


Figure 1. Activity system diagram for theatrical modeling.

Approaching the design of theatrical modeling activities in this way, we proposed several objectives:

1. Explore several varieties of theatrical modeling to better understand the potential affordances and limitations of the approach for supporting collaborative scientific inquiry
2. Place theatrical modeling in conversation with other kinds of theatrical expression and with scientific modeling to better understand its interactions with its “parent” genres of activity

Theoretical background

Modeling is “the driving practice of science” (Lehrer & Schauble, 2015, p. 677). In the Constructionist tradition (Papert and Harel, 1991), modeling has been employed as a process of both authentic disciplinary inquiry and personal, public expression. Many constructionist activity designs attempt to recruit computers as “thought partners” in the modeling enterprise, giving rise to distinct forms of productive interplay between modelers, modeling media, and the systems they model (cf. Pierson, Brady, & Clark 2020). A particularly powerful example of this is agent-based modeling, or ABM (Wilensky, 2001; Wilensky & Rand, 2015), in which natural systems are represented as collectives of computational agents that follow simple behavioral rules.

In investigating perspectival phenomena, learners benefit from projecting themselves into the representational infrastructure (in our case, theatrically; in others, through imaginatively assuming the role of modeled entities (Wilensky & Reisman, 2006)). Constructionist researchers have long argued that embodiment and agent-based modeling are mutually supportive, in what Papert called “syntonic learning” (Papert, 1980). In this regard, the representational agent is a critical pivot, allowing learners either to project themselves into its role or physically to play out its actions. For instance, a common strategy in agent-based modeling activities is to “play turtle,” a technique by which learners take advantage of the “body-syntonic” nature of the agent and leverage their own embodied sense of self to plan out solutions to thorny computational problems. At the systems level, this notion has been built upon to develop and study instances of collective embodiment in the form of participatory simulations and role-playing activities (Resnick & Wilensky, 1998; Wilensky & Stroup, 1999). Such activities draw on the notion of “social syntonicity” (Brady, Weintrop, Anton, & Wilensky, 2016) to leverage structural parallels between human collectives and multi-body systems and both enrich modeling expressivity and support student sense-making.

To capture the embedded features of our multiple activity designs, we draw on representations and concepts from cultural-historical activity theory (CHAT; Engeström, Miettenen, & Punamäki, 1999). CHAT is a way of thinking about the growth and change of social practices that happens concurrently with individual and collective learning. A central concept from CHAT is the notion of the activity system, which builds on Vygotsky’s (1978) notions of tool-mediated activity to highlight the interrelatedness of individuals, communities, and practices, and draw attention to how they are situated among broader cultural and historical narratives. We adapt the classic triangular activity system diagram to represent similarities and differences among designs for learning activities in our classroom site, attending to how our designs draw on existing culturally and historically situated social practices (e.g., participatory theater and scientific modeling).

Study overview

Larger project context

The context of this work is an intensive sequence of design studies (Cobb et al, 2003), set within a larger, trans-institutional overarching project called SAIL+CTM. SAIL+CTM is developing a year-long science curriculum that builds on and synergizes two ongoing projects: "Science And Integrated Language" (SAIL) and StarLogo Nova. A key feature of the SAIL curriculum is the scientific practice of modeling (Lehrer & Schauble, 2015). In each of the four, 9-week units that comprise the SAIL curriculum, students make sense of phenomena and represent their ideas using physical models in material environments and diagrammatic models in print environments. The SAIL+CTM project expands the SAIL curriculum to integrate computational modeling in online environments using StarLogo Nova, a blocks-based programming environment with an agent-based simulation engine for modeling complex systems. By cultivating and orchestrating a variety of modeling modalities, the project hopes to leverage linguistically diverse learners' repertoires of both verbal and non-verbal representational practices (Gutiérrez & Rogoff, 2003) for the sake of broadening and transforming all learners' participation in STEM practices.

School setting

The data reported on in this work were collected in a fifth-grade STEM class at a public middle school (grades 5-8; ages 10-14) in the Southeastern United States. Each year students are required to take a 9-week course called "STEM Related Arts," the goal of which is to engage students in hands-on projects that integrate STEM disciplinary content knowledge with 21st century skills such as computer programming, rapid prototyping, and design thinking. Enrolled students meet daily for a half-hour period. The course repeats on a quarterly basis; every 9 weeks, a new group of 25 students cycles through the STEM Related Arts curriculum. At the time of writing, a total of 105 students across 5 implementations have consented to participate in the study. In keeping with SAIL+CTM's focus on supporting linguistically diverse learners, students in the study come from a variety of linguistic, ethnic, and economic backgrounds.

Our partner in design has been the teacher for this course, Ms. S, a veteran middle school teacher with 26 years of experience teaching mathematics to eighth-grade students. She volunteered to become the school's STEM Related Arts teacher when the school launched the course three years ago. Ms. S has been an invaluable partner in this work as a design collaborator, meeting regularly with the research team to develop the learning activities in addition to being the primary instructor on the course.

Methods

Data collection

Our methods of data collection included documenting student-made artifacts (e.g., photos and copies of written scripts, hand-drawn diagrams, and GoPro video recordings), using screen-recording software to capture students' computer-based activity, and collecting multi-stream video and audio recordings of each class session. The first author attended each class session as a participant observer and co-facilitator; field notes from these visits are also included in the dataset. In total, the corpus for this study consists of over 200 hours of audio and video data and over 100 artifacts of student work distributed across 77 class sessions, beginning in August 2018 and running until October 2019. As data were collected, they were indexed via a daily content log in which moments of suspected importance were flagged for later follow-up.

Data analysis

The analysis for this paper draws on a subset of the overall data selected from the first three iterations of the study. First, we consulted the content log to look for notable examples of representational hybridity. After identifying several candidates, we re-viewed each episode and summarized it using both endogenous terms and concepts from literature. We then synthesized across these summaries with attention to obvious similarities and important distinctions. At various points throughout the multi-year course of this project, short segments of video data from these episodes have been brought before a group of independent researchers to conduct an Interaction Analysis (Jordan & Henderson, 1995). These sessions informed the authors' thinking and noticing about the data presented here, often pointing out new avenues for both theoretical interpretation and further empirical investigation.

Findings

We organize our findings as two contrastive cases selected from the first rounds of implementation. This juxtaposition highlights several potentials for generative activity in our design space of possible combinations of theater, narrative, and computational modeling.

Exploration 1: (Re-)enacting *Maggie Mars*

In our first implementation we asked students to work in groups of 3 to 5 to develop a "theatrical model" of an imagined solar system. Students wrote scripts for enacting narratives that would address two questions: 1) how do planets move through solar systems? and 2) what are some consequences of that motion? Students video-recorded these enactments using head-mounted cameras, and these recordings were shared with the class as modeling artifacts.

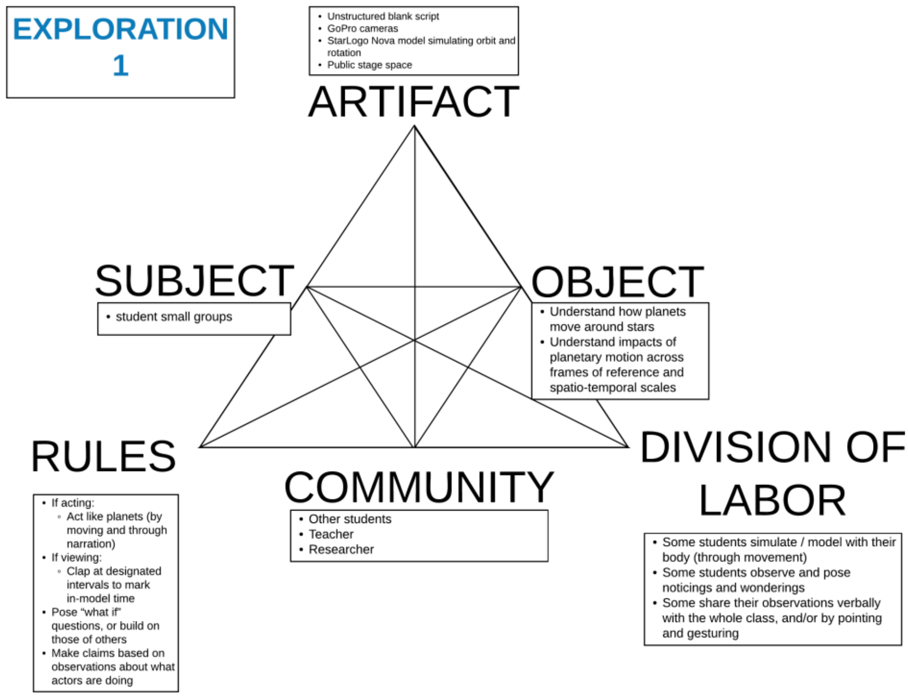


Figure 2. Activity system diagram showing Exploration 1 design features.

Students' constructions drew on representational forms from across the genres of dramatic, narrative theater and agent-based computational modeling. The case we have selected for analysis is paradigmatic but it does not necessarily represent the collection of student works. We chose it because it highlighted features that emerged as distinctive and characteristic of this implementation. The story elements of their performances blended celestial systems with their own social lives, explaining how Earth and the Sun were close friends while Mars looked on from a distance. They adapted scientific terms (like orbit and rotate), giving them local definitions derived from a series of embodied and computational activities in StarLogo Nova. In the example case below, "orbit" means *to walk a circular path [around the orbited body]*, while "rotate" means *to spin about one's own axis* – although this definition became troubled when the class explored issues of frames of reference for rotation.

A: [left to right] Amanda (dark shirt) orbits and rotates by walking around Britney.

B: Hi I'm Maggie mars. I have been watching the Amanda earth and how it orbits and rotates the Britney sun. The Amanda earth is a wonderful planet, very talkative and happy. The Amanda earth has told me that my days are longer than hers! I don't completely know how that is, but the Amanda earth has told me that this happens because I rotate slower than her, I can see why because when I watch her spin it takes 2 seconds! My day takes about 4 seconds. Amanda Earth is a very good friend of the Britney sun, but that might be because I have a longer way to walk so I am very tired and not as energetic as Amanda earth is. Britney stays still while Amanda rotates in a circle while orbiting the sun. That is what I, Maggie mars have to say about how Amanda earth rotates and orbits the Britney sun.

Figure 3. Script and still frames from Maggie Mars.

The case of Maggie Mars (Reimers & Brady, 2019; see Figure 3) exemplifies the versatility of theatrical representations produced by learners in this iteration (and across the study). Their initial production was a cinematic performance fusing elements of social life, solar systems phenomena, and observations rooted in the embodied representation and emerging from its enactment. Its meanings and uses were then elaborated when it entered into public discourse in the classroom community. With Ms. S's prompting, the other members of the class added a temporal "markup" to the representation by clapping when they thought a year or day had happened on Amanda or on Maggie. This added measured time and salient rhythm to what might otherwise have been seen as a primarily aesthetic performance. Next, a second group of students got up to re-enact their interpretation of *Maggie Mars* as live theater for the rest of the class. Creating this performance as a live, interactive, and manipulable enactment provided a more malleable model of the solar system depicted in the video. Performing Maggie Mars as participatory theater also enabled the class to "run" it in different modes (first continuously and then 'stepwise'); to highlight and attend to different perspectives (examining daylight first on Maggie Mars and then on Amanda Earth); and to modify the enactment to explore the effects of alternative planetary motion (asking Maggie Mars to orbit without rotating).

Maggie Mars began as a performance based on a prompt about personified planets; it was presented as a complete, cinematic unit. In this sense it was an artifact of theater, even traditional theater, albeit one grounded in a scientific phenomenon. Once the on-screen performance became interactive (when the class augmented its metric specificity by clapping), *Maggie Mars* also became a modeling artifact. Its performative qualities—the choreography of it—served not only as a recipe to coordinate multi-actor performance but also as a specification of key parameters of a planetary system that could be highlighted and measured. This choreography was remarkably effective in communicating these parameters, as evidenced by how readily it was re-enacted live by relative amateurs after a single viewing.

The classroom use of Maggie Mars illustrated a design commitment of the study: rather than using the performance as an occasion to create a conventional diagrammatic summary or re-representation, the class "took up" the performance by re-enacting it as live theater. They then re-ran it iteratively, augmenting it, extending it, and using it as a source of new knowledge and insights – i.e., as a model. Figure 4 illustrates how this occurred.

Macey and Kasey model a kind of non-rotating planetary system as Macey (the planet, in blue), orbits around Kasey (the sun, in green) while facing the same direction. As the model runs, Esteban (a spectator, in white) points in the rightmost frame to indicate the position at which dawn would occur on Macey's face.

The model pauses when a question is asked about where midnight would occur during Macey's orbit. At left, Esteban claims that it would be when Macey is behind Kasey and therefore out of his view. At right, Macey repositions herself and gestures to evoke an image of her field of view, refuting Esteban's claim and explaining that midnight occurs when she cannot see Kasey.

Figure 4. Images and descriptions of theatrical modeling in action.

Despite the relative instability and (scientifically) unconventional nature of this theatrical representation, it actually supported deeper scientific engagement in the form of both quantity and complexity of student claims about the model's subjects. We propose that the theatrical re-presentation of the model provoked richer scientific modeling *because* the phenomena were perspectival in nature. By animating their representation through theatrical enactment, the students in this class were better able to understand and reason about perspective across frames of reference.

Exploration 2: Scripting languages and *Jupiter-quakes*

Exploration 1 revealed that a student group could respond generatively to theatrical performances with modeling referents. In particular, they could view these performances as having the properties of models and "read," analyze, and revise them through re-enactment. This potential guided our next design iteration, when we were curious if we could support a modeling stance in the construction of the original performance. Theatrical modeling in the second iteration of the design study placed a stronger emphasis on using computational conventions to structure theatrical representation. Student small groups developed scripts for theatrical performances using a hybridized version of stage directions that drew on movement and location conventions from both traditional theater and Logo programming languages. Using this theatrical-programming language, students collaborated to develop a performance that would answer the same two questions that they addressed in Exploration 1, namely: 1) how do planets move through solar systems? and 2) what are some consequences of the way they move? Integrating computational modeling and theater more deeply in the initial productions, we anticipated that questions about lived experience of inhabitants of the planets might emerge more strongly. For example, what would it be like to live on that planet, and how would its motion shape life there? These questions prompted students to use their imaginations to simulate and represent fictional life on other planets.

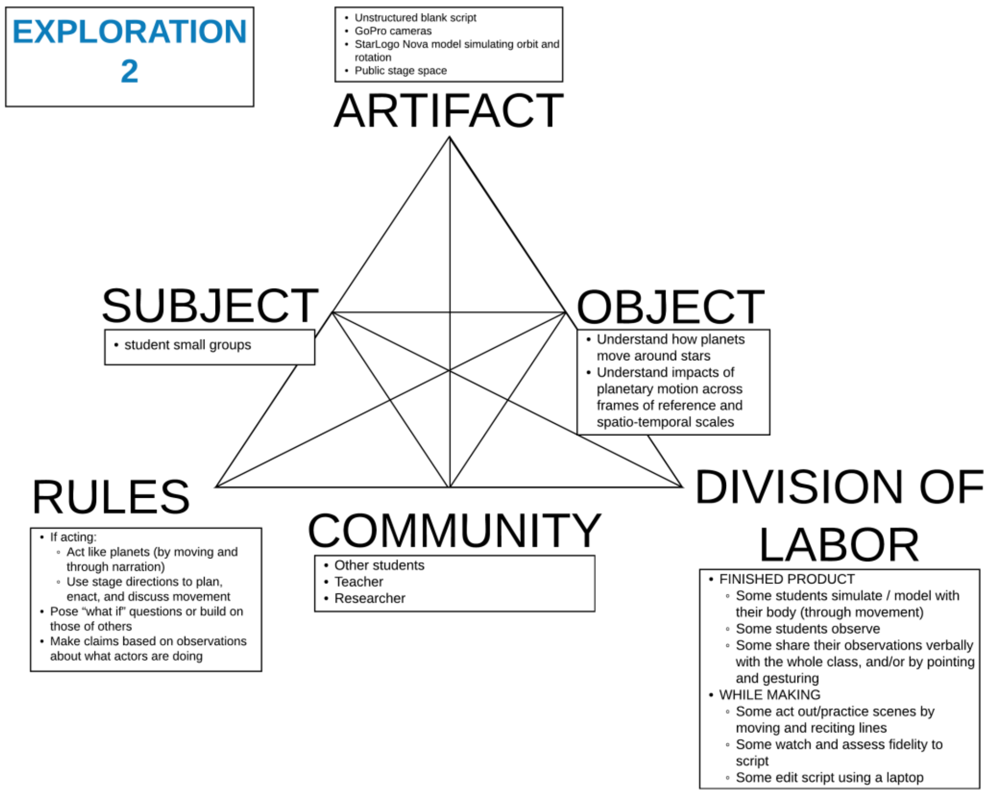


Figure 5. Activity system diagram showing Exploration 2 design features.

In this design, student groups used Logo conventions to direct actors' on-stage motion. Performances were invoked using declarative statements that assigned actors to specific positions (e.g., "stage right" or "stage left"), and actors' subsequent motions were governed by looped command statements (*e.g.*, "forward 10" or "right-turn 45"). While these stage directions afforded rich conversation across performances, they also constrained students' motion to what they could express in Logo code. This feature manifested in interesting ways, most obviously through polygonal orbits and stepwise approximations of simultaneous rotation and orbit.

In these implementations, students developed and performed models of imagined solar systems that investigated perspectival phenomena across scales. In the case below, participants investigated a fictitious Jupiter (at a system-scale perspective) and the impacts of its irregular motion on the life of an alien denizen (at a human-scale perspective). The interplay of computational and theatrical conventions provided a generative framework for the group's own scientific reasoning, as can be seen in the actor-author group's own iterative refinement of the performance as they developed it.

The case of *Jupiter-quakes* focuses on a group of three students, who together crafted a story based on a hypothetical version of the planet Jupiter that differed from the real Jupiter in student-chosen features of its motion. This Jupiter did not rotate (with respect to its sun), and its orbit was a rough polygonal approximation of an ellipse. Whenever Jupiter reached a turning point in its orbital path (denoted in the stage-space by a tape arrow on the ground), the alien living on Jupiter would experience a massive "Jupiter-quake" as a result of the sudden acceleration. This entailed comical portrayals of daily life for this hapless extraterrestrial in addition to a robot-like portrayal of Jupiter's pre-programmed motion.

While *Jupiter-quakes* presents an interesting case of computational-theatrical hybridity, it is instructive to examine the process by which this performance was developed and refined. The three student authors each adopted different roles: one (a script-writer) working at the laptop to plot new movements using NetLogo and record them in a script document; one (an actor) enacting these movements on stage as they entered the script; and one (a director) watching the actor and interpreting the motion to the script-writer using stage-direction terms. The students established these roles of their own accord. As the group explored and rehearsed their performance, several pathways developed by which computational, theatrical, narrative, and scientific elements of the performance interacted. For example, *the shape that Jupiter's orbit ought to be* originated in students' ideas about Jupiter's motion. However, this ideal orbit had to be adapted to fit the constraints of the Logo-style stage directions. The adapted motion then provoked new insights about how irregularities in the motion would impact conditions on the surface and interior of the planet, on the human scale, which were then incorporated into the narrative of the performance.

The students' work here contrasts with *Maggie Mars* in a number of ways. First, it involves perspectival claims differentiated at the level of scale instead of frame of reference. While it would not have been "off-limits" for students to ask what the living conditions were like on Maggie Mars, the central focus was not on such issues of scale but rather on distinguishing between Maggie's and Amanda's frames of reference with respect to cycles of day and night. Second, the representational form of the script, merging Logo commands and stage directions, seemed to provoke a multi-vocal conversation between students' ideas, their enactments, their capture in the script, and their imagined significance for life on the planet. Each of these modes of thinking "about Jupiter" influenced the others. The iterative construction played these forces out and the final performance reflected a kind of conceptual equilibrium between them. Finally, this rich modeling-like examination of the entailments of representations occurred during the author group's own construction work, while in Exploration 1, seeing performances as models unfolded more strongly in response to presenting them to the whole class.

Discussion, Implications, and Open Questions

Several themes have become apparent through examination of our two cases.

First, these data suggest a distinction among modes of creation, presentation, and reception within theatrical modeling activities. As a performative contribution, one of these hybrid theatrical models is:

a) Conceived with a certain blend of affiliations, a stance on the phenomenon, a set of representational aims, and a vision of enactment;
b) Actually enacted, either as the first of a series of enactments of the 'same' thing or as not-the-first. If it is the first, there may be a stronger sense of need for a 'director'-like figure
c) Received by an audience made up of people in some (potentially multiple) relation(s) to the enactors.

The theatrical frame thus provides a range of activity structures that students seem both willing and able to engage with in generative ways. Celestial bodies and their relations within solar systems seem to afford an imaginary space rich enough for students to explore in a variety of ways. (This appeared in both examples). As students conceive of, fashion, and iteratively refine an enactment, they collaborate to make meaning. (This was foregrounded in our example from Exploration 2.) Finally, as they present their performances, the classroom group has the opportunity to take up these representations as models; to augment them with audience participation (like clapping to highlight events); and to inhabit them by reenacting the performance themselves. (This was foregrounded in our example from Exploration 1.) Given the generativity of theater in this setting, we wonder how other constructionist learning designs might be impacted by framing public artifacts as performances and drawing on conventions from theater to shape community participation.

Second, students' perspective-taking was multi-faceted and complex. Each type of syntonicity outlined by Papert (1980) and Brady et al. (2016) entails a different sense in which learners take on agents' perspectives (e.g., where body-syntonicity connotes a more physical, objective use of perspective, ego-syntonicity reflects a narrative, identity-based perspective). Theatrical modeling activities provide opportunities for learners to leverage several of these syntonicities to accomplish collective epistemic goals by taking on perspectives across these various senses.

Third, all of our participants (who were novices in both theater and modeling) seemed to have a workable idea of how to do theatrical modeling (including conceiving, enacting, and receiving theatrical models). The seeming universality of these competencies suggests that learners at this level already have access to a rich latent pool of resources for engaging in theatrical representation and perspective-taking, presumably because these activities have analogs or precursors in everyday social life. This does not mean that theatrical modeling processes are mundane or uninteresting from a scientific knowledge-building perspective. On the contrary, it suggests that theatrical modeling is especially apt as a method when participants have scientific questions that necessitate perspective-taking. Theatrical modeling borrows the know-how of our shared social situation (which we have come to describe as *homo scenicus*), by building on the repertoire of interaction games that humans have built over historical time.

Finally, taking a CHAT perspective as an analytical lens has helped us to elaborate the productive resonances and tensions between iterations by investigating the interactions among the different nodes of the activity triangles. We are intrigued by the promise of activity systems as both analytic constructs and design frameworks (cf. Danish, 2014). Attending to change between explorations, we notice that divisions of labor, rules, and artifacts vary most starkly. For instance, Exploration 2 provides a compelling example of how computational rules for representation can shape students' activity across the performative modes outlined above. The fact that a single change in rules produced such a remarkable change in activity suggests that rules for talking about embodied action are particularly crucial for these representational endeavors. Within explorations, it is clear that students' perspectival reasoning can be mediated through many different artifacts: bodies, cameras, narratives, and scripts. Less clear are the pragmatics of these arrangements: when do students opt for one artifact over another, and why? This set of elements presents a rich field of possibilities for group learning activities with low technical overhead, and further design work should prioritize understanding how changing these dimensions of theatrical modeling can shape the conceptual, enactive, and interpretive activities of participants.

Acknowledgments

Funding from the National Science Foundation (Grant No. 1742138) is gratefully acknowledged.

References

Brady, C., Weintrop, D., Anton, G., & Wilensky, U. (2016). Constructionist learning at the group level with programmable badges. In *Proc. Constructionism Conf* (pp. 61-68).

Cobb, P., Confrey, J., DiSessa, A., Lehrer, R., & Schauble, L. (2003). Design experiments in educational research. *Educational researcher*, *32*(1), 9-13.

Danish, J. A. (2014). Applying an activity theory lens to designing instruction for learning about the structure, behavior, and function of a honeybee system. *Journal of the Learning Sciences*, *23*(2), 100-148.

Engeström, Y., Miettinen, R., & Punamäki, R. L. (Eds.). (1999). *Perspectives on activity theory*. Cambridge university press.

Gutiérrez, K. D., & Rogoff, B. (2003). Cultural Ways of Learning: Individual Traits or Repertoires of Practice. *Educational Researcher*, *32*(5), 19–25.

Jordan, B., & Henderson, A. (1995). Interaction Analysis: Foundations and Practice. *Journal of the Learning Sciences*, *4*(1), 39–103.

Lehrer, R., & Schauble, L. (2015). The Development of Scientific Thinking. *Handbook of Child Psychology and Developmental Science*, *2*(7), 671–714.

Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. Basic Books, Inc.

Papert, S., & Harel, I. (1991). Situating Constructionism. In *Constructionism*. Ablex Publishing.

Pierson, A. E., Brady, C. E., & Clark, D. B. (2020). Balancing the environment: Computational models as interactive participants in a STEM classroom. *Journal of Science Education and Technology*, *29*(1), 101-119.

Reimers, J., & Brady, C. (2019). Maggie Mars: Theatrical Modeling and the Phenomenological Understanding of Solar Systems. In Lund, K., Niccolai, G. P., Hmelo-Silver, C., Gweon, G., & Baker, M. (Eds.). *A Wide Lens: Combining Embodied, Enactive, Extended, and Embedded Learning in Collaborative Settings*: Proceedings of the 13th Edition of the International Conference on Computer Supported Collaborative Learning.

Resnick, M., & Wilensky, U. (1998). Diving into complexity: Developing probabilistic decentralized thinking through role-playing activities. *The Journal of the Learning Sciences*, *7*(2), 153-172.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Wilensky, U. (2001, August). Modeling nature’s emergent patterns with multi-agent languages. In *Proceedings of EuroLogo* (pp. 1-6).

Wilensky, U., & Rand, W. (2015). *An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo*. MIT Press.

Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—an embodied modeling approach. *Cognition and instruction*, *24*(2), 171-209.

Wilensky, U., & Stroup, W. (1999, December). Learning through participatory simulations: Network-based design for systems learning in classrooms. In *Proceedings of the 1999 conference on Computer support for collaborative learning* (p. 80). International Society of the Learning Sciences.

Phenomenology and Visual Modeling: Making the Object the Subject

James E. Clayson *james@clayson.org*
The American University of Paris, Paris, France.

Back to the things themselves!
Edmund Husserl (2001, p.168)

Phenomenology wants to slow the researcher down and hold his or her gaze on the phenomenon itself – the lived-experience of some activity – seeking not to locate it in an abstract matrix by saying how its abstracted structure might be similar to others, but rather to illumine its specific quality as an experience.
Peter Willis (2001, p.2)

Abstract

Both phenomenology and visual modeling encourage individuals to understand their own approach to sense-making through careful description of how they see, model and interact with physical objects. Both caution us that clear description is best achieved only by first attempting to move beyond past experiences and expectations. Moving beyond the past is hard, however, and we need tools to help us. Seymour Papert's gear story is one vivid example of the power of objects to promote such curiosity and exploration. But since Papert didn't explain how he got from gears to constructionist guru, my paper suggests some phenomenological tricks that could help to fill in the gaps. I also reference the literature on transformational objects to put this gear story into context.

Emboldened by Papert's tale, I present ten powerful ideas, some related to personally meaningful objects, that have shaped my work. I discuss the role that such objects, particularly the special class of objects labeled as transformative, play in fostering our meaning-making constructions. I suggest that constructionists should give more attention to both phenomenological methodology and the ongoing search for transformational objects.

The paper ends with a short demonstration of how I have used my own visual modeling approach, informed by phenomenology, to start thinking about, and recording my thoughts, for a proposed course called: "thinking computationally about color."

Keywords

phenomenology, visual modeling, Seymour Papert, constructionism, transformational objects, psychology of color, computational thinking.

Old story: new reading

A close look at the gear-story mentioned in the preface to Papert's most significant work, *Mindstorms* (Papert, 1993), reveals a narrative style very similar to that proposed by phenomenologists. In this paper I will explore how examining these similarities might help us rethink some of our constructionist goals and to become better teachers and researchers.

Papert's story is personal and poignant. He tells us that, as a young child, he loved automobiles and, perhaps not surprisingly, he loved playing with the tiny gear cluster inside a model car given him by his much-loved father. He imagined being inside the car, inside the gears, and remembered watching that happen and thinking about it. When he was older, he used his young person's gear intelligence as a tool for thinking about something new: algebra. He claims that it was his personal meaning-making in algebra using his gear thinking, and the thrill he got from that, which in turn stimulated him to study mathematics.

Reading this brief story in the preface to *Mindstorms*, we already sense the origins of Papert's future constructionist thinking, including several of his "powerful ideas": simulation, debugging, body syntonicity, emergence and restructuration. But we also realize it is first and foremost a love story, narrated later in life, about Papert as a young boy exploring physical objects and how that led him into new pursuits. He even confesses, *"I fell in love with my gears"* (Papert, 1993, p. xx). This is already a remarkable testament, but then we also discover his core message when we come the last sentence in the preface: *"This book is the result of my own attempts over the past decade to turn computers into instruments flexible enough so that many children can each create for themselves something like what the gears were for me."* (Papert, 1993, p. xxi).

How might children answer Papert's call to computationally search for transformational objects?

I think there are three directions that children – and adults – might take in using computational tools to seek out those strangely powerful objects that might give them the kind of pleasure and stimulation that Papert found from his gears:

(1) they could build, using languages like Logo, new virtual environments within which they might find new and exciting virtual transformational objects; (2) they might use computational modeling tools to look more closely at physical objects that they already find interesting; or (3) they might combine both of these methods to see the surprisingly weird relationships between the physical target and computational objects that emerge as we look computationally at solid world things.

Whatever path is taken to search for transformational objects, there are four useful suggestions for all of us from Papert's gear story:

- recall your own transformational experiences with objects like Papert's gears;
- remember and revel in the excitement you felt from interacting with your own personally meaningful objects;
- imagine ways to extend these past experiences and seek out new ones using simple exploratory modeling;
- try to maintain a childlike joy as you do these things.

Despite these lessons, Papert did not provide us any description of how he got from gears to computational thinking, nor any narrative to help guide others who might wish to follow in his footsteps. Without such a "journal" we can't see how his interactions with objects greatly affected his own sense-making process and how he might have modeled that. I have found that phenomenology suggests a methodology for exploring this type of investigation.

Phenomenology

Beginning with Edmund Husserl (Husserl,1900: Hopkins, 2011), phenomenologists have argued that subjectively exploring our very concrete “being in the world” offers a path that is complementary to that taken by positivists and natural scientists, for understanding our world. Phenomenologists expect to realize *“… a rigorous and unbiased study of things as they appear so that one might come to an essential [fundamental] understanding of … experience.”* (Valle and Halling, 1989, p.6).

Husserl’s basic premise about thinking, labeled *intentionality*, states that thinking always points eventually to objects; that thinking links the thinker-subject with the thought-object; that we are always engaged with the physical world; that it is narrative – “describing things” as Husserl puts it – that allows us to make these connections explicit.

The primal act for phenomenologists, the “phenomenological reduction”, is to find ways to study the interactions between one person and one object. This investigation, they argue, can provide a view into this person’s thinking process, but only if we can do the observations in a way free from previous and biased assumptions. Phenomenologists want us to put preconceptions aside. The phenomenological reduction is called “reduction” because it reduces our attention on what and how we have seen things in the past, thus liberating us to focus more innocently on what we are looking at now. Phenomenologists want us to look freshly, openly and childishly at lived phenomena: and to record in *words* the raw experience. They also recommend that we go back to these narratives later and look for salient themes. (Giorgi, 2009).

Transformational objects elicit the richest narratives

Papert’s gear story describes what psychologists label transformative events and the transformative objects that trigger them (Bollas, 1987, p. 13ff). The small child’s pacifier, stuffed animal or security blanket are proto examples, called transitional objects. (Winnicott, 1971) Later, when the child is older, these primal transitional objects may be replaced by transformational objects, like Papert’s gears. In some cases, interaction with transformative objects may produce an extraordinary aesthetic experience of “rapt, intransitive attention.” (Krieger, 1976, p. 11). Christopher Bollas, a British psychoanalyst and writer has demonstrated that, in fact, people need and seek out transformational objects over their entire life (Bollas, 1987 p. 17).

Another social scientist, MIT’s Sherry Turkle, has collected and published many people’s transformational event stories, showing us just how common these occurrences are (Turkle 2007, 2011). I think we constructionists can learn a lot about subject-object relationships by reading more widely in psychological disciplines and by seeing how it is narrative that gives these relationships coherence and meaning.

Visual modeling shares both Papert’s and basic phenomenological goals

My own approach to visual modeling (Clayson, 1985, 2007, 2008, 2015, 2018) builds on ideas from Seymour Papert, especially his belief that computational tools can help us find and benefit from transformational objects. I have selected the path that looks for these objects in the physical world rather than in the virtual one (p. 2). Phenomenology offers substantial help in doing this.

Phenomenologists warn us that this search cannot be done without first decluttering the mind. Any decluttering effort at reduction, that is reducing our attachment to previous ways of looking, is not easy. To achieve “reductionist innocence” when modeling objects, I resort to mixing many different media – ranging from such traditional forms as drawing, verbalizing and writing – to modern computational and quantitative techniques. Paradoxically, the mixing of media can force a fresher and more innocent view than that offered by a single medium alone.

Powerful ideas within constructionist activity

While Papert's view of powerful ideas (for example see Papert 1985, chapter 6) is central to his constructionist project, he never defined them. Rather, he gave examples of heuristic rules individuals created, debugged and altered as they tested them in their attempts at making sense of their world. This sounds very much like the constructivist psychologist George Kelly's claim that we are all personal scientists (Kelly, 1956). Both Papert and Kelly believed that ideas are powerful and useful only when their creators see that that they are powerful as they apply them.

The "construction" in "constructionism" is threefold: (1) the building and debugging of powerful heuristic ideas through model and object manipulation; (2) the materializing of sense, skill, knowledge and meaning that arises from this activity; (3) the emerging of personal narratives of meaning-making that comes from close reading of notebooks kept during this endeavor. The constructionist project is to create environments where this enormously complex activity is facilitated, documented and studied (Noss and Hoyles, 1996).

Powerful ideas, phenomenological reduction and visual modeling

Powerful ideas, especially imperative ones – that is a rule that says "do this" – suggest ways to interact with objects and to trace that activity – as it happens – in a journal/notebook. That trace is a narrative of our meaning-making in a specific context. In fact, one might consider the meaning itself. That text should be read, along with our other narratives, to see if a *theory* about our meaning-making might emerge.

In my teaching experience, powerful ideas are most potent only when they have been written out and shared with others who are doing the same. This is effectively done in a classroom when instructors share their learning heuristics in real contexts with students who are asked to do the same.

But there is a paradox here. The greatest limitation to our imagination is when we rely only on what has worked well in the past. Writing out our rules helps us apply them properly and consistently. But, as the phenomenologists tell us, we must then put our powerful ideas aside – along with the expectations they carry – in order to keep advancing our observational capacities: this is the phenomenological reduction.

Some of my powerful ideas

Here are ten of my powerful ideas for visual modeling. Each idea/rule is directed at uncluttering our minds, opening us up to surprise. Some rules tell us to carefully and clearly report what we did and what we built; and to reread our narratives and watch what happens to us as we do this.

1. Always visualize emotionally and aesthetically as well as intellectually. Mix that up.
2. Look at works of art as you work. Find visual keys for looking freshly at your objects: borrow views, motifs, colors and perspectives from them.
3. Keep your computational code simple and always use it for wide and wild simulation.
4. Find alternative views, different methods and media: the more the better
5. Keep a journal notebook in which to record your modeling activities.
6. Share your powerful ideas and journal notebooks with others; read theirs; record the resulting conversations.
7. Include words, sketches, photos and diagrams to complement generated images, math, computational code and tasks.
8. Read widely from psychology. Use their ideas to help combine the qualitative and quantitative approaches you use.
9. Reread your earlier observations and notebooks and compare these with current work.

10. Regard all powerful ideas as exploratory, tentative and artificial.

Case study: thinking about thinking about color

Modeling of all varieties is generally used to explore one idea, one physical object or to carry out a single given task. But modeling, especially visual modeling, can also be helpful in exploring fuzzy, non-structured thoughts and plans. Visual modeling is open-ended and intended to generate multiple outcomes ….

I am currently thinking about how to design an undergraduate workshop course that will focus primarily on color. I want to encourage participants to record in their journals how, why and what happens as they make color choices within design exercises. Then at the end I want them to analyze their comments and try to describe their own color model.

To support my course proposal, I needed illustrations of what we would "actually be doing" in the class. I thought this general task could offer a case study of visual modeling in a more open-ended role. In fact, the approach used to structure the course would be part of the substance of the course itself. My case study would also show students a picture of how I used my own powerful ideas as an encouragement for their own documentation of their rules.

My examples illustrate an exploratory approach to constructionist modeling, which is why my initial goals are left deliberately vague. These models are simple and open-ended. Their use – in this case by me alone – produces verbal description along with the computational play; each influences the other and they are difficult to separate. They are organized here into seven explorations and are illustrated in Figures 1-12.

Why so many examples? The narrative of our sense-making is enriched by looking at our world in different ways. In this exercise I want a rich chronicle to give my students as an introduction to "my way" so that they can bracket my narrative, put it aside, and move generously beyond that and me. Therefore, I offer a variety of examples. Each is structured around a task. The analysis of my narratives will be done in class.

Color as perceptual process

Mazaviita Chirimuuta (2015) a British philosopher of neuroscience, perception and psychology, sets the tone for my thought excursions.

"… colors are not properties of objects (like the U.N. flag) or atmospheres (like the sky) but of perceptual processes—interactions which involve psychological subjects and physical objects. In my view, colors are not properties of things, they are ways that objects appear to us, and at the same time, ways that we perceive certain kinds of objects. [Our] account of color opens up a perspective on the nature of consciousness itself." (Chirimuuta, 2015)

Some sense of the colors we actually "see" might emerge as we watch and record what we feel is happening as we manipulate *simple* color *objects* using simple rules. This exercise should strengthen our ability to see the color characteristics inside our viewing of real physical objects.

Simple color objects

All the simple color objects in this tiny series are virtual rectangles or circles filled with whatever is generated by standard Red Green Blue (RGB) values. Besides this "colored light", the additional characteristics of these objects will be shape, scale and orientation. Each rectangle will sit within a collection of other rectangles within a fixed-orientation rectangular "canvas" of variable shape, scale and background light.

For this proposed course I am using Python with several new modules and a modified turtle graphics package. I wrote these with help from two computer scientists, Peter Tomcsanyi and Andrej Blaho, at Comenius University in Slovakia. Later I added the ability to define and manipulate colors using Hue, Saturation and Brightness (HSB, 2019), in addition to the default RGB notation.

Figure 1 illustrates a color wheel of 24 rectangular color objects of different hues.

Figure 1. 24 rectangular color objects: different hues, equal saturation and brightness

Exploration 1: show that seen color is not an intrinsic quality of the object but emerges from context

Pick one color rectangle in figure 2 and watch how it seems to change when its background changes.

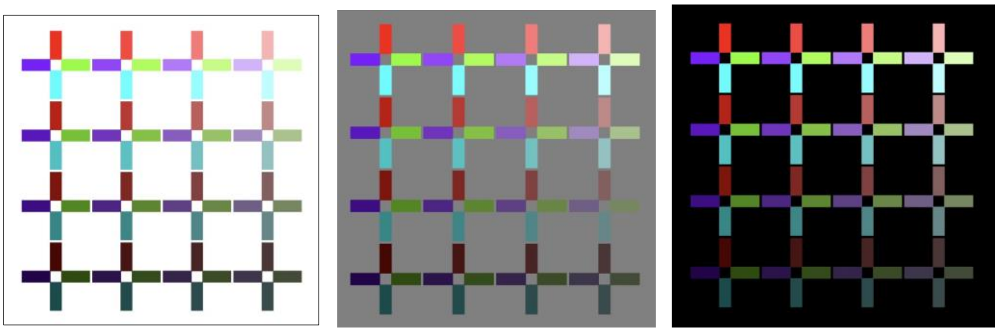


Figure 2. Four hues: different saturations (horizontal dimension), different brightness (vertical dimension). Three different backgrounds: white, grey, black.

Exploration 2: show a method for suggesting surprising color palettes

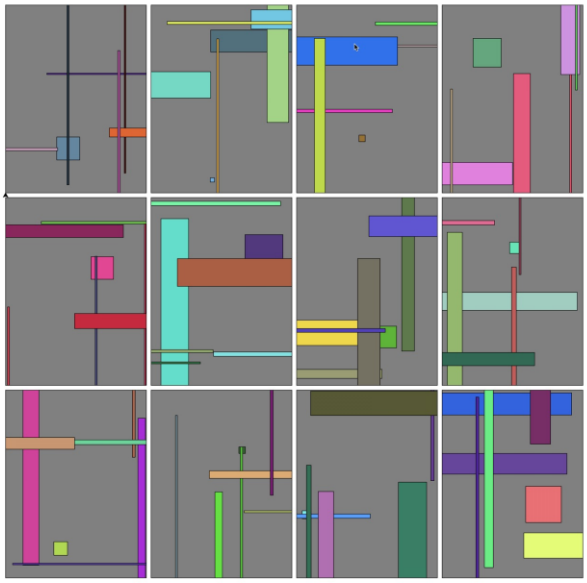


Figure 3. Floating objects. Looking for surprising color combinations.

Figure 3 is a good example of an interesting graphic that is easy to code for complete beginners: randomly float rectangles within a rectangular canvas: introduce randomness to rectangle placement, shape, size and color; attach some rectangles to edges, let others float freely; generate a series of images. The results are surprisingly aesthetic and encourage close looking. In this example my explorations led me to discover color combinations that I would never have imagined before.

Exploration 3: illustrate variations in saturation and brightness for one and two hues

I have been greatly influenced by the art and design pedagogy Josef Albers offered at Black Mountain College in North Carolina. Before fleeing to America in 1933, Albers taught at the German Bauhaus. Albers' book, *Interaction of Color* (Albers, 2013), is a classic. The cover, Figure 4a, shows eight slightly jiggled, fixed size horizonal color bars, each with a different saturation and brightness value of a one red hue. I wanted to honor Albers by using this design, but I needed to show more kinds of interaction than those viewable on his book cover.

Figure 4b illustrates another design, this time realized in black and white. I have taken Albers' lovely but rather tight graphic and opened it up, using randomly sized and randomly placed rectangles. Doing so allows us to see more and different interactions between the component parts, which we then can use to explore color as illustrated in Figure 5.

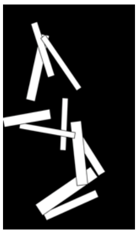


Figure 4a. Josef Albers' book cover. Figure 4b. White sticks in black box as a variation on Albers' book cover.

Figure 5 shows further variations, this time in color.

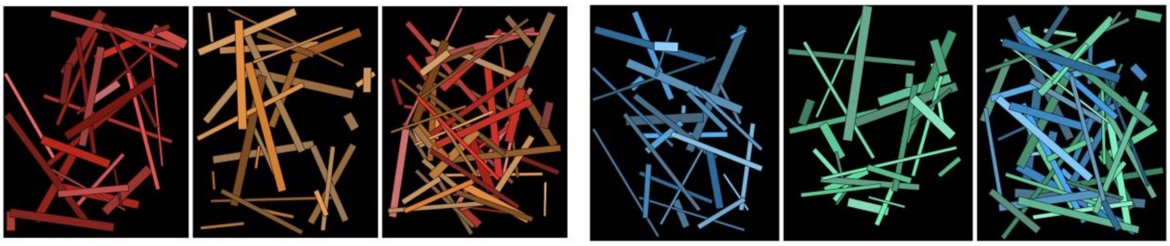


Figure 5. Colored sticks in boxes: Random saturation and brightness values for several hues that are close together.

Exploration 4: illustrate color interactions – not just overlaps

All of the illustration in Figure 5 show overlapping colors. These colors, however, do not interact with each other when and where they overlap. But what if they did?

The Russian abstractionist, Wassily Kandinsky, painted many studies of intersecting discs like those shown in Figure 6. His passion was to paint disc intersections in a color that is related to but different from the colors of the intersecting discs themselves, and to show what happens (Derouet & Boissel, 1985, pp. 293, 294). For me, Kandinsky's colored disk interactions on a flat dark picture plane gives the whole canvas a mysteriously glowing depth.

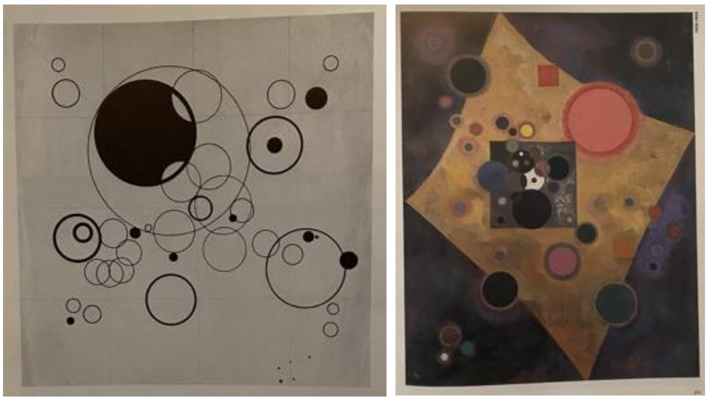


Figure 6. Kandinsky's intersecting discs: pen and ink study and painting,

Excited by Kandinsky's painterly investigations, I wanted to use computational tools to play with different ways of coloring the intersections. Would I simply blend the colors of intersecting shapes or do something else?

My results are shown in Figure 7. I used colors picked at random and applied a simple blending rule. What happened, though, was not so simple. Even random colors, manipulated according to naïve rules, can sometimes surprise and become meaningful to us.

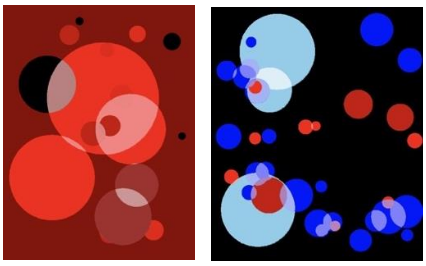


Figure 7. My simple intersecting discs

Josef Albers sometimes required students to work creatively with colors that they didn't like or that they alleged didn't work well together. In my case, I don't much like the combination of pink, orange and black. But recently, I saw a painting called "Desert Moon" that uses exactly these colors, and I liked it a lot. This painting is by Lee Krasner, an American abstract expressionist. See Figure 8.

Figure 8. Lee Krasner: Desert Moon, 1955, LACMA.

Intrigued by my positive response to this painting, I then decided to look more closely at those Krasner colors by mapping them into my Kandinsky-inspired overlapping disk world. See Figure 9. I found that looking at a color palette that I initially disliked, displayed in a design that I very much liked, not only changed my appreciation for pink/orange/black/red but made me aware of these colors, never noticed before, during my walks around my *quartier* in Paris. Clearly, as a result of these color experiments, the subject-object relationship has been altered.

Figure 9. My intersecting circles with colors from Krasner's Desert Moon

Exploration 5: illustrate "colors" you see in a well-known object

Only some of my exercises are inspired by artwork. Others come from more mundane subjects such as the fruit I buy from a tiny neighborhood shop. First, I photographed a number of apples that I had just brought home. Then I "extracted" six colors from one of them and produced a matrix of my selected colors. The juxtaposing of the two images shown in Figure 10 affects how we look from one view to the other, taking one color, or combinations of colors, from the right and looking for that in the apple, and vice-versa. The way we see the colors in one object can be altered by a display of those colors in another form or shape. The way we look can be slowed down and enriched by multiple viewing methods.

Figure 10. French apple: two views

Exploration 6: elicit words to describe and differentiate "colors"

Throughout my teaching career I have tried to apply ideas from the American constructivist psychologist George Kelly's personal construct psychology (PCP). I especially like his repertory grid technique that elicits from a participant a set of bipolar constructs that that person suggests might differentiate the members of collection of similar objects (Kelly, 1955). The opposing poles of each bipolar construct are given names and each item is then evaluated by placing it along this construct's scale. I generally use numbers from 1 (the white pole) to 5 (the grey pole). The result of evaluating all of the objects on all of the constructs is a matrix of numbers that can be studied using a variety of statistical techniques (Clayson, 2013). Here I have chosen to use principal component analysis (PCA).

How did I use Kelly's repertory grid in conjunction with these color explorations? To start, I picked twelve color names. My color selection is close to the eleven "colors" suggested by two American anthropologists, Brent Berlin and Paul Kay, in their *Basic Color Terms* (Berlin & Kay, 1969). These researchers studied many different languages to come up with a color list that they claim includes all of the colors used across their selection.

The major reason for using Kelly's grids is that they allow students to compare and contrast different constructs. This is where record keeping in a journal notebook becomes essential. The Figure 11 plot is based on my own bi-polar constructs and may reveal more about me than I would wish. There are six constructs and PCA effectively found two principal components that describe 80% of the original total variance across all constructs. The Figure 11 plot is two different spaces, one on top of another. The first is principal component (PC) space, and the second is the correlation space between principal component and the original constructs.

One can guess "my meanings" for the dimensions of Figure 11: to the left we have pessimistic, calming, male and organic; to the right we have aggressive, optimistic, female and chemical. Moving up we have open and life; and moving down we have closed and death. The red line is a minimal spanning tree.

Repertory grids are an exploratory technique. I use them to open up topics for discussion and reflection.

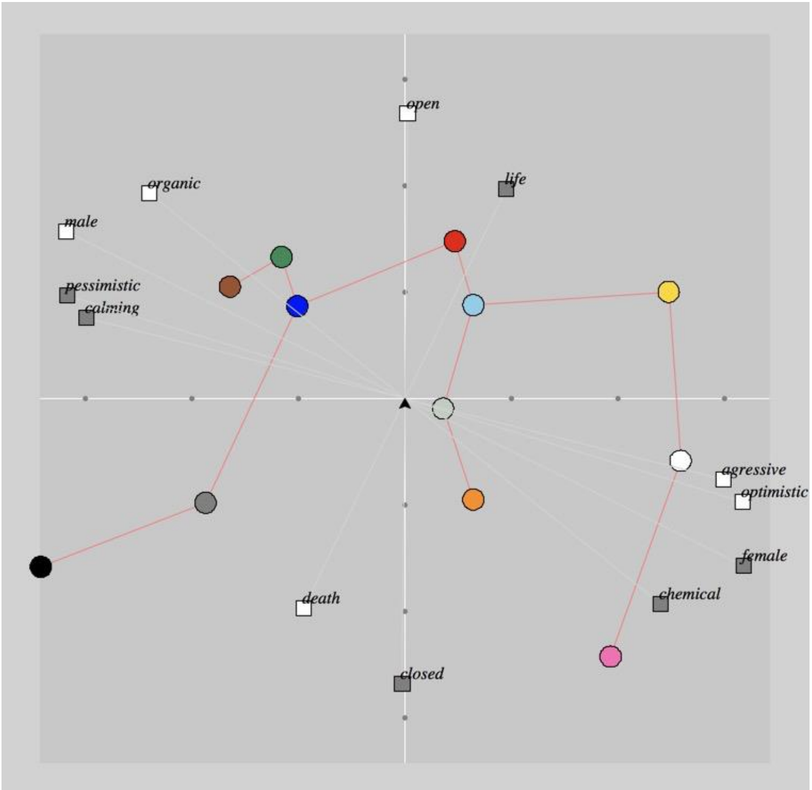


Figure 11. The first two principal components (pc) of my color space. Balls represent elements (pc values); boxes represent pc-construct correlations.

Exploration 7: Explore one traditional color relationship notion

"Complementary colors" is a traditional but poor descriptor of color interaction. What is generally suggested by this term is: how do we find colors that clash when used together, colors that shock or surprise? The rule that we all know from school is that these complementary colors are found directly across from each other in the color wheel, that is, 180 degrees apart. I wanted to computationally play with this idea as I looked at combinations of colors. I also wanted to try out different words to describe how I felt about different color combinations.

How did I proceed with this self-experiment? I used a 1-5 Likert-style color "shock/surprise" index. I wrote a procedure to generate random, pure-hue, color combinations; to display them and to ask me for my shock/surprise value. I did this for 100 color combinations. I then wrote procedures to plot the results as shown in Figure 12. The vertical scale is degrees of color separation from 0 to 180; the horizontal is my categorical surprise index, 1-5.

I connected the median values in red and the upper and lower quartiles in black. The larger plot shows all the color combination data as well as data variability across categories. Here outliers can be judged as well as the validity of my index.

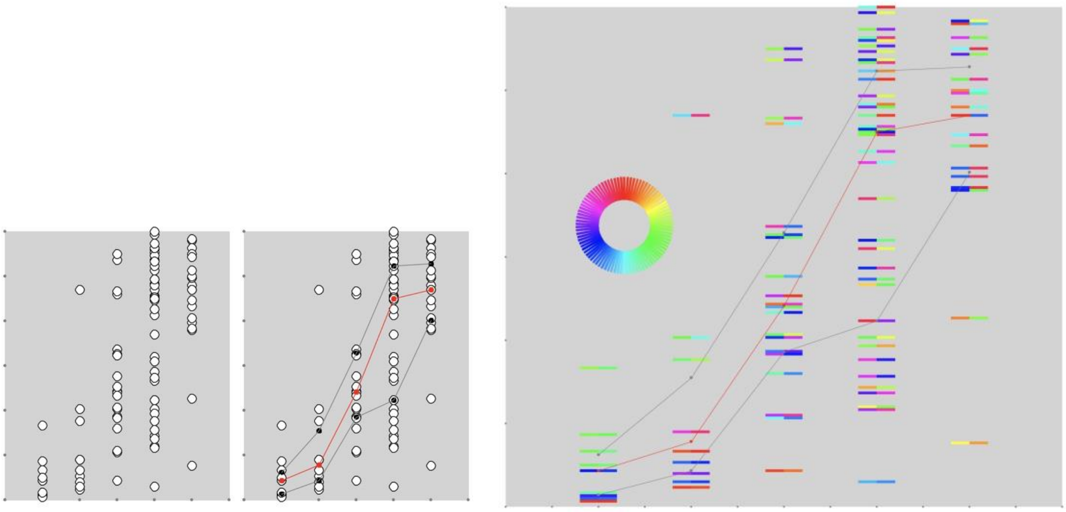


Figure 12. Three views: my feelings about 100 two-color combinations.

Visual exploratory data analysis, like this example, encourages a fresh look at what complementary color might mean to each of us personally and how that notion might be characterized, defined and measured.

Thinking about color

I have built this short color example around several ideas that intrigued me. I have described some differing approaches, all based on simple simulations, that could encourage others to play, think and talk about colors in new and original ways. Each student needs to build their own tools and methodologies.

Doing the phenomenological reduction does not mean forgetting or not using what we know; it means putting aside our expectations for where any approach might lead us and any hope for specific results. We must be childlike and open to surprise as we advance on many fronts. We know so little about how individual sense-making works; this is why it is so important for constructionists and phenomenologists to explore objects whose qualities already intrigue us in the hope that this activity will illustrate our own workings.

Conclusion

Visual modeling, like phenomenology, promotes two core principles:

1. The need to describe, model and narrate our complex interactions with personally meaningful objects.

2. The need to re-read the notes and observations of these interactions to write a personalized – yet more general – narrative account of how we as individuals go about making sense of the world.

Making the object the subject enhances our understanding of how our consciousness works.

References

Albers, J. (2013) Interaction of Color, 50th Anniversary Edition, New Haven & London: Yale U. Press.

Berlin, B., Kay, P. (1969) Basic Color Terms. Berkeley: University of California Press.

Bollas, C. (1987) The Shadow of the Object: Psychoanalysis of the Unthought Known. London: Free Association Books.

Chirimuutya, M. (2015) *The Reality of Color is Perception: an argument for a new definition of color*. Nautilus 026, July 2015. http://nautil.us/issue/26/color

Clayson, J. (1985) Visual Modeling with Logo: A Structured Approach to Seeing. Cambridge, MA: MIT Press.

Clayson, J. (2007) *Radical Bricolage: Making the Liberal Arts Coherent*, in Kalas, I (ed), Proceedings of EuroLogo 2007, Bratislava.

Clayson, J. (2008) *Radical Bricolage: Building Coherence in the Liberal Arts Using Art, Modeling and Language,* International Journal of Education through Art 4(2): 141-161

Clayson, J. (2013) *Talking Statistics/Talking Ourselves: Some Constructionist Lessons from the Work of George Kelly*, Technology, Knowledge and Learning 18: 181-199.

Clayson, J. (2015) A Computational Eye: Visual Modeling with Python, Bishop, CA: Deep Springs College.

Clayson, J. (2018) *Artifacts, Visual Modeling and Constructionism: To Look More Closely, To Watch What Happens*, Problemos, 2018 Supplement: 8-23.

Derouet, C., Boissel, J. (1985) Kandinsky : Œuvres de Vassily Kandinsky (1866-1944), Paris: Collections du Musée National d'Art Moderne

Giorgi, A. (2009). The Descriptive Phenomenological Method in Psychology. Pittsburgh, PA: Duquesne University Press.

Hopkins, B. (2011) The Philosophy of Husserl, Durham: Acumen.

HSB, (2019) https://en.wikipedia.org/wiki/HSL_and_HSV#Use_in_end-user_software

Husserl, E. (2001) [1900/1901] Logical Investigations, Dermot Moran (editor). 2nd ed. 2 vols. London: Routledge.

Kelly, G. (1955) The Psychology of Personal Constructs. Vols 1 & 2. New York: Norton.

Krieger, M. (1976) Theory of Criticism, Baltimore: Johns Hopkins University Press.

Noss, R., Hoyles, C. (1996) Windows on Mathematical Meanings: Dordreckt, Klewer Academic Publishers.

Papert, S. (1993) Mindstorms: Children, Computers and Powerful Ideas, Second edition. Jackson: Perseus Books.

Turkle, S. (2007) Evocative Objects: Things We Think With. London: MIT Press.

Turkle, S. (2011) Falling for Science: Objects in Mind. London: MIT Press.

Valle, R., Halling, S. (1989) *An introduction to existential-phenomenological thought in psychology*. In Valle, R., King, M. & Halling, S. (Eds) Existential-phenomenological Alternatives for Psychology. New York: Plenum Press

Wald, C. (2019) Why Red Means Red in Almost Every Language: The Confounding Consistency of Color Categories, Nautilus, 76.

Willis, P. *The "Things Themselves" in Phenomenology*, Indo-Pacific Journal of Phenomenology, 1:1, 1-12, DOI 10.1080/20797222.2001.11433860.

Winnicott, D. (1971) Playing and Reality. London: Tavistock Publications.

Recovering Constructionism in Computer Science: Design of a Ninth-grade Introductory Computer Science Course

Chris Proctor, *proctor5@buffalo.edu*
Graduate School of Education, University at Buffalo--SUNY, Buffalo, USA.

Jenny Han, *jlhan@stanford.edu*
Department of Computer Science, Stanford University, Stanford, USA.

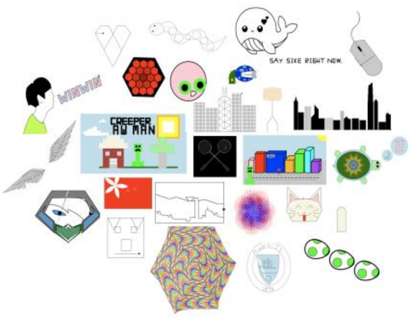
Jacob Wolf, *jwolf@isf.edu.hk*
International Schools Foundation Academy, Pokfulam, Hong Kong.

Krates Ng, *hnkng@isf.edu.hk*
International Schools Foundation Academy, Pokfulam, Hong Kong.

Paulo Blikstein, *paulob@tc.columbia.edu*
Teachers College, Columbia University, New York City, USA.

Abstract

Constructionism provided an early justification for children to study computers, but today's dominant approaches to K-12 computer science education are vulnerable to some of the same critiques Papert (1980) made of traditional schooling. In this paper, we identify three themes of Constructionism (computing cultures, material intelligence, and liberatory pedagogy) and use them to analyze existing approaches to K-12 computer science education. We then use these themes as design goals for a Constructionist ninth-grade introductory computer science course which is currently being implemented. This paper is part of a larger research project whose goal is to demonstrate the feasibility of a course focused on fully realizing the ambitious epistemological goals of Constructionism. As we contribute to a vision for K-12 computer science education, we hope to help recover the central role of Constructionism.



Keywords

Computer science education, Constructionism, computer cultures, material intelligence, liberatory pedagogy

Introduction

In most contemporary computer science classes the computer is used to put children through their paces, to provide exercises of an appropriate level of difficulty, and to dispense information. This sentence, which we believe accurately describes today's K-12 computer science landscape, also paraphrases the first sentence of *Mindstorms* (1980). Papert critiques the status quo of schooling and articulates a different vision of computer-supported education. Today, as computer science starts to be taken up broadly as a K-12 discipline, the leading implementations are vulnerable to the same critique which helped to justify teaching computer science in the first place.

This irony is the starting point for this study, in which we report on the design of a Constructionist computer science course which is currently being implemented in a high school in Hong Kong. This study is part of a larger research project whose goal is to demonstrate what might be achievable in one year of a Constructionist computer science course and to document how it comes about. As exploratory research, the goal is not to claim that the design of this course could be or should be implemented at other schools, particularly at schools with access to different resources. Nor is the goal to criticize projects like Hour of Code (Code.org), which prioritize expanded participation over any particular learning goals, and which are designed to scale up across existing conditions at schools around the world. However, we are concerned that in making compromises to scale up computer science education, current manifestations of K-12 computer science may have given up too quickly on the ambitious epistemological goals articulated by Constructionism.

In this paper, we surface three central themes of Constructionism, frame a design goal around each, and briefly consider several existing introductory computer science courses with respect to these goals. We then describe the context and design of an introductory ninth-grade computer science course, and close by grounding this design in an ongoing research project and considering how that project might help shape our vision of for K-12 computer science.

Background

In this section, we articulate three themes of Constructionism which frame our design goals.

Computer cultures

Constructionism is rooted in the belief that knowledge does not exist in a vacuum but rather lives and grows in situated context (Ackermann, 2001). Powerful thinking with computers requires a computer culture in which to participate. A computer culture provides ideas and media--tools to think with--as well as norms and practices to guide participation, define the community, and shore up the identities of participants. Just as Piaget's constructivism was rooted in the relationship between an organism and its environment (Fosnot & Perry, 1996), computer science is something people do within a computer culture.

Almost forty years ago, Papert imagined "the computer cultures that may develop everywhere in the next decades" (Papert, 1980, p. 20) Today, our environment is profoundly shaped by, made from, and mediated by, computers. We live in digital worlds which permeate, augment, and co-constitute what we perceive to be the real world. Youth growing up today almost universally participate in digital media, extensively and in important ways (Anderson & Jiang, 2018). These activities constitute rich and diverse computer cultures. They are to varying degrees emergent, viral, and engineered. Even though powerful ideas from computer science do not always flourish in these cultures, we propose that such cultures should be considered funds of knowledge (Moll, Amanti, Neff, & Gonzalez, 1992) which can be employed in the practice of constructing knowledge.

Material intelligence

Papert suggests that one reason computer cultures had not yet emerged in the late twentieth century was a relative "poverty in materials… from which intellectual structures can be built" (1980, p. 20). While our built environment and cultural practices offer innumerable opportunities to engage with and benefit from algebra and geometry, the claim was that fewer such learning opportunities existed for the powerful ideas of computing. This is certainly no longer the case. The computational infrastructure mediating our digital worlds is a leaky abstraction which constantly exposes the algorithms and computational properties with which it is designed, providing innumerable opportunities to encounter computational phenomena and to become powerful by making use of computational ideas. For example, the everyday practice of navigating social media involves informal social network analysis. Managing identities online requires constantly thinking in layers across interfaces. The essence of a meme relies on the practice of abstraction.

At the same time, many forms of work are now characterized by specialized computational media and practices in which computers are used intentionally and metacognitively as tools for thinking about thinking. diSessa refers to the ability to interface with a representational medium, for social practices or toward cognitive ends, as “material intelligence.” Using computers as tools for thinking--and for thinking about thinking-- was of central importance to Papert. Wilensky (2010) uses the term “restructurations” to describe how knowledge can be reformulated via new representational forms which make different properties available. The protean nature of computers is powerful not just because they can take on many different forms to meet our existing needs (as in an app store), but because they support richer understandings of the structure of problems.

Liberatory pedagogy

Finally, we see Constructionism through a critical pedagogy lens, where education is a political act and computers could function as agents of emancipation (Blikstein, 2008). Instead of accepting an education “where children are segregated from society and segregated among themselves by age and put through a curriculum,” Papert argued in agreement with Paulo Freire for a “problem-posing education” that encouraged students to approach important problems around themselves and in their worlds (Papert & Freire, n.d.). Each of the previous two themes contributes to a pedagogy of liberation, which we analyze in terms of Berlin’s (1969) positive (freedom to) and negative liberty (freedom from).

An education grounded in computer cultures could be particularly well-suited to address students’ agency of self-determination by connecting to their existing funds of knowledge (Rodriguez, 2013). Students (along with teachers, parents, and community members) must be treated as agents in the classroom with equal power to construct knowledge. To do this, teachers must understand students’ backgrounds and cultural practices as wealths of experience which can be employed in the practice of constructing knowledge. Such a pedagogy promotes positive liberty by developing a classroom space where students can develop their identities while forming relationships with powerful ideas. These relationships contribute to learners’ agency by expanding what they can do and understand, and by increasing their ability to contribute their own thoughts, ideas, or extensions of ideas to a conceptual domain (Boaler, 2003).

Developing material intelligence with computing could be particularly effective in supporting negative liberty, or freedom from oppression. Beyond defensively learning how to keep oneself safe online, learning how computing works could support youth in understanding how computation shapes our ideas about the world and our place in it, a computational analogy to Freire and Macedo's “reading the word and reading the world” (1998). Just as Freire connects a reading of words to a writing of words, we can connect understanding the impact of technology to creating technology that has an impact. As the negative social consequences of computing become more apparent in our world and joining the computing workforce perhaps loses some of its idealistic luster, a critical understanding of computer science may become an essential tool in utilizing technology to resist and deconstruct oppression.

Comparison with existing approaches

These three Constructionist themes can help structure the design space of possible ways of teaching introductory computer science courses (Nelson & Ko, 2018). In this section we briefly position several other projects with respect to the three themes.

First, Hour of Code’s goal is “broad participation across gender and ethnic and socioeconomic groups” (Code.org) Toward this end Hour of Code is designed to be self-contained, requiring no teacher or community and usable on a wide variety of devices (or even without a computer at all). The scripted, puzzle-like activities are embedded in cultural worlds which might be familiar and appealing to a wide variety of students (e.g. programming sprites to dance to a soundtrack), and guide users toward understanding how to control those worlds using block-based programming. If Hour of Code succeeds at connecting with youth cultures (including informal computer cultures), the cost is very little development of material intelligence. The tools provided obviously cannot be used to make anything real. (Even in open-ended environments such as Scratch, students have trouble viewing their programming as real or building on their experience for future learning.) If Hour of Code has liberatory potential, it is in changing attitudes toward future computer science learning opportunities.

If Hour of Code can be dropped into any classroom, Exploring Computer Science (ECS) is a full introductory computer science curriculum supported by professional development (Goode & Margolis, 2011). Like Hour of Code, ECS is focused on broadening participation and stresses "ease of implementation for teachers and maximal engagement for students," but ECS is also focused on developing computer science knowledge (About ECS). ECS prioritizes computational thinking over programming; much of the curriculum is focused on working with computational problems and ideas in a social context rather than on implementing working programs. Therefore, we view ECS as being highly committed to building computing cultures while not emphasizing as much material intelligence. ECS could be seen as liberatory both in its focus on providing students access into the world of computer science, and in cultivating agency within the computing cultures it supports.

Finally, Beauty and Joy of Computing (BJC) is an introductory computer science curriculum which contrasts with ECS in its heavy emphasis on programming (Harvey, 2012). BJC is taught in Snap!, a block-based variant of Scratch which supports more explicit restructuration of mathematical ideas. While BJC also provides pedagogical support for diverse learners, in comparison with ECS its emphasis is more on powerful mathematical and computational ideas than it is on connecting to students' existing cultures or using computing in those worlds. As its name suggests, BJC stresses the liberatory potential of the ideas themselves. Like Hour of Code and ECS, BJC is taught in a block-based language which prioritizes accessibility over the ability to create personally-meaningful projects in domains already important to students.

Design of the Course

These examples of existing computer science courses suggest a general tradeoff between providing broad access to computer cultures and engaging deeply with material intelligence. Our broader research goal is to question whether this tradeoff is necessary. By reconfiguring the relationship between computer cultures and material intelligence, we hope to show that these goals can be mutually supportive and can result in new liberatory possibilities.

Context

The research and development of this course takes place at a bilingual private school in Hong Kong during the 2019-2020 school year. A teaching team of three instructors, two of which are associated with the university-based research group, works with twenty-eight Grade 9 students, divided among 2 classes. For the majority of the students, this course is their first exposure to computer science. For the school, this is the first iteration of a Constructionist computer science course. Notably, the material and financial resources available in this setting made low student-to-teacher ratios, highly-qualified teachers, and one-to-one computing possible.

At the same time, the students, the course, and the school face pressure to succeed in the context of both the Chinese and the overseas educational systems (notably, U.S. and U.K.). They are under pressure to demonstrate success through grades, AP, IB scores, college admission, and have historically relied on extrinsic motivation to achieve this performance. Thus, while the context of this research provides atypical access to resources, it also provides stringent demands that a Constructionist computer science class be able to demonstrate success on traditional terms as well as its own.

Overview

In designing the course, the central goal is to create a rich, diverse community of people making things with code, through which they can develop personal relationships with powerful ideas. The course is designed to help all students learn to interact with code as an expressive, evocative medium, which helps to structure thought. At the same time, the course is designed to support computational literacy, connecting with students' existing ways of reading and writing.

The course is composed of five curriculum units intended to introduce students to various computing domains and programming paradigms, while supporting progressive growth in particular skills. Each unit introduces new skills and concepts, and then devotes significant time to an open-ended student project in a real-world computing domain through which they will encounter programming challenges in authentic contexts and, with teacher support, learn what they need to solve them.

Table 1. Summary of curriculum units.

Topic	Project	Paradigm
Turtle drawing	A drawing (optionally using makerspace to etch/cut)	Imperative
Data science	Data-based argument about our digital worlds	Functional
Games	Create a game	Object-oriented
Networking	Create a networked microservice	Reactive
Web applications	Create a web application for real-world users	Human-computer interaction, collaboration on larger system

Each unit consists of labs (for student-driven exploration), mini-lessons (for just-in-time teaching), assignments (for review and practice of concepts), and projects (for open-ended creation). Below, we outline three central design goals, each attached to the aforementioned themes of Constructionism, and provide examples of specific decisions in our course design that align with our goals. In practice, we cannot separate the themes of Constructionism into three distinct design decisions, but we structure below accordingly for clarity.

Let students drive liberatory pedagogy

We dedicate most of our class time to student-directed engagement with course materials (called labs). Not unlike science labs where researchers plan and conduct experiments, computer science labs introduce groups of students to concepts, practices, and tools in computer science by giving them project-based problems to solve. Such a problem-posing education necessitates learning by doing. Consider the following examples:

Navigating the Terminal. In their first lab, students learned to navigate the terminal by exploring a directory with text files, subdirectories, and python files that played like an adventure game when students used the terminal commands to run the programs, change directories, read the text files, and other basic command-line tasks. Importantly, students were not told what to learn or memorize. Instead, students were provided with simplified documentation of various commands, and they chose to learn the terminal commands that they felt were most important or necessary in the moment.

Introduction to Loops and Lists. In this lab, students used print statements, lists, and loops for the first time. Rather than read about the syntax of a for-loop or why it may be useful, students experienced the concept for themselves. They explored a list of subway stops in their district and wrote different loops to traverse through the stops. Notably, we never provide answer keys to the labs or assignments. This design decision shows students that there is no one correct approach to a computer science problem; instead, we encourage students to consider and share multiple solutions in their groups.

The teaching team relies on a collection of norms for interactions in a student-driven class (see Figure 1). As instructors, we minimize our time in front of the class. Sometimes, we offer guidance "mini-lessons" about concepts to the entire class before they begin the lab. More frequently, this manifests in holding "mini-lessons" for individuals or small groups of students in the moment, when they get stuck on a lab or assignment. When one student grasps a concept or resolves a bug, we invite them to act as "student experts" and teach their peers. In this way, students drive the knowledge creation in the classroom and become agents of meaning-making.

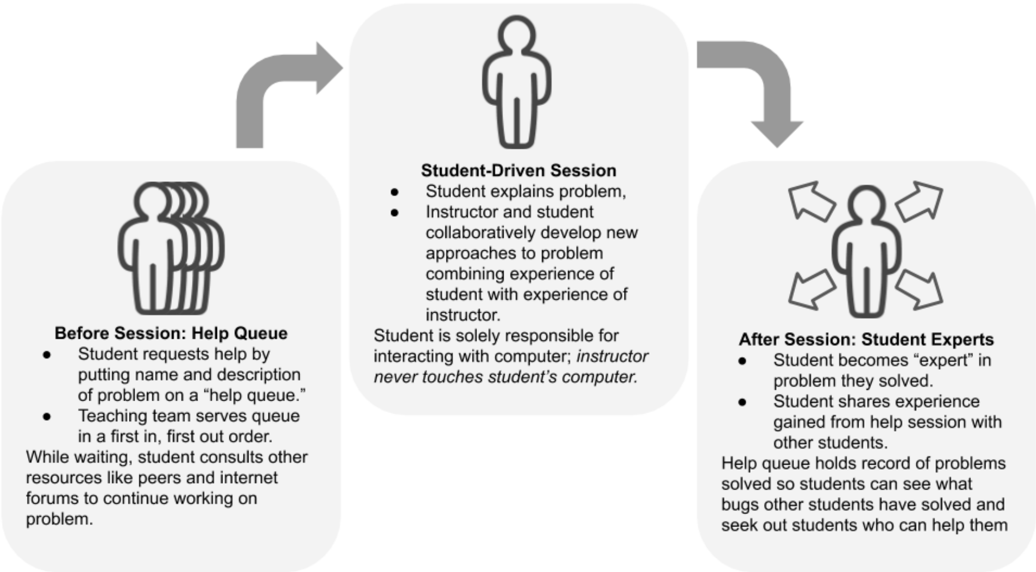


Figure 1. Process and norms of getting help including requesting a help session, doing a help session, and sharing knowledge after a help session.

Develop material intelligence using practical tools with high ceilings.

In designing the course, we paid particular attention to the tools we chose for our students to use to engage with computation. We were particularly interested in the Constructionist idea that powerful tools help students develop deep connections to a conceptual domain. We decided to use tools which we believed would increase students' material intelligence, their agency to engage with computation. We decided to use a UNIX/Linux terminal user interface, Atom, and Python as the development tools for our class (Figure 2). Importantly, each of these tools provides an open-ended development experience with very high ceilings on what they can be used to create. Further, these tools are all regularly used in computer science practices from personal to academic to industry, allowing our students' use of the tools to interface with broader practices of computing.

This approach is far from the sandboxed strategy adopted by many computer science curricula, and required significant up-front setup, downloading and configuring software and development environments. The tradeoff is that students are now able to powerfully use their computers as tools for general purpose computing in this class and beyond. In the first weeks of the course, students learn the basics of how to navigate the file system using a command line interface, how to create and edit Python files using the text editor, execute and debug them from the command line. Soon thereafter, students were using version control (git) to clone starter projects, track their own progress, reflect on their process (using a customized template for commit messages), and dialogue with teachers as they revise their work.

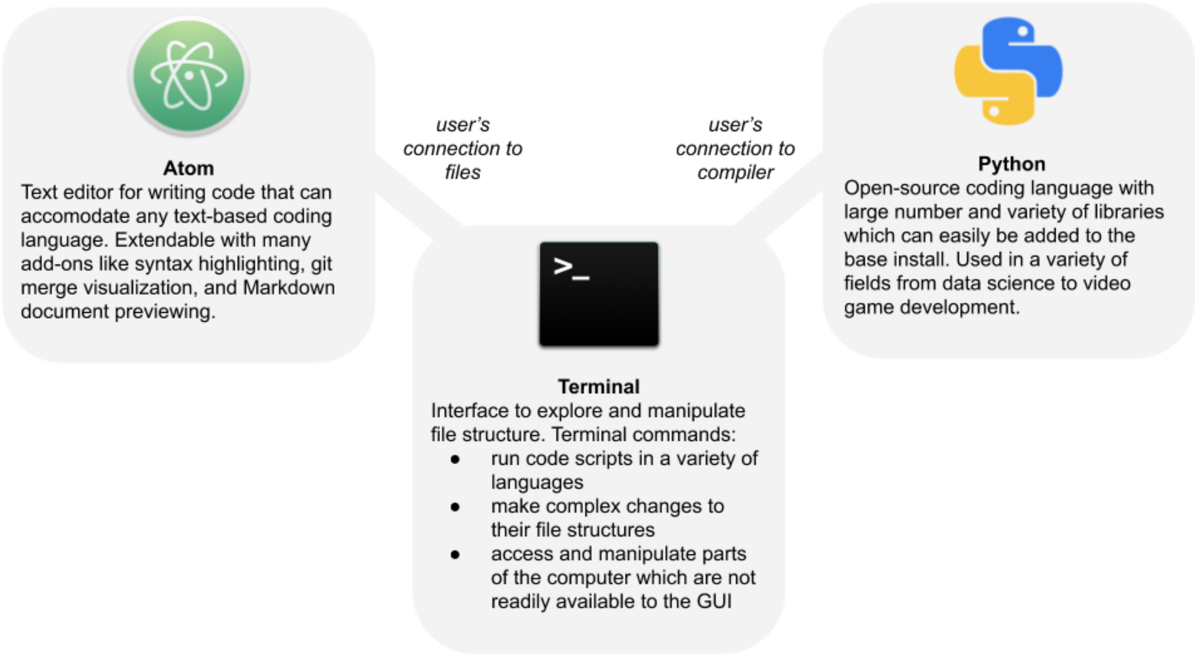


Figure 2. Visualization of tools and workflow for code development in our class: find/open/create files with Terminal, write Python scripts as files with Atom, use Terminal to run Python scripts.

Implement personally meaningful projects draw in students' computer cultures

Finally, we designed our class to rely on students' own experiences with computers while expanding their ability to understand and interact with their digital worlds. We feel that the most effective way to engage students with the "underlying representational form" (diSessa, 2001, pg. 24) of computation is to ask students to draw in elements of their own computer cultures. To do this, we spend a significant amount of time at the end of each unit supporting students as they develop a project which utilizes the tools and concepts explored in the unit. The goal of this project is always to create something personally meaningful: a piece of art, a data-informed answer to a question about their worlds, a game inspired by their own favorite game. Some of these projects are individual asking students to draw from their independent experiences while others are group projects asking students to mesh their computer cultures with the computer cultures of others. Further, these projects often expect students to build upon the work of others, hacking and remixing code for their own purposes.

This approach to projects has been particularly generative in creating space for students' computer cultures in our classroom. At the end of our Turtle drawing unit, many students' projects featured their own computer cultures. One student created her own version of a meme showcasing her efforts while another student invoked his computer culture by integrating Minecraft characters into his drawing (Figure 3).

Students' engagement with personally meaningful projects also allows students to connect other literacies to their computer cultures. Many students' projects connected outside identities, interests, and skills to the classroom computer culture, potentially enriching both (Figure 4).

Figure 3. Students using elements of their computer cultures, such as memes (left) and Minecraft (right), as inspiration for their projects.

Figure 4. Students using elements of their cultural identities, such as their Hong Kong heritage (left) and Korean pop music interests (right), as inspiration for their projects.

Future Considerations

As we finish the first semester of the course, we reach a point where we can share preliminary results, limitations, and next steps. We are currently conducting research to substantiate the initial positive results we see from our perspective as reflective practitioners. Our intention is to revise and then share the curriculum broadly and to learn about what kinds of support will be needed when it is taught at other schools. Because our choice of tools requires access to lower levels of the computer, it would not be possible on a phone, tablet, or browser-based laptop. The hardware required may make this curriculum less accessible and scalable for other schools. It also requires significant human resources as teachers will need to familiarize themselves with these tools, which are more complex than other beginner-friendly platforms. The teacher must also feel human and disciplinary agency when using the tools. For many schools, this may present a difficulty if the computer science teacher is not a computer scientist by training.

Even though we have passed up interfaces designed for accessibility such as web-based and block-based environments, initial results suggest that all of our students have been able to access and participate in working with computational ideas using a powerful representational medium. The fact that we are using tools common to real-world formal and informal computing contexts also provides numerous ancillary benefits. Students can also begin to develop practices of using real-world resources such as man pages, library documentation, and support forums. Student curiosity about how things work is often rewarded by excursions into important phenomena ranging from their operating systems to random number generation to the layers of abstraction supporting user interfaces.

Conclusion: What kind of CS for all?

As computer science has gained recognition as a mainstream K-12 subject in school, the question of what kind of computer science we want has also begun to gain belated traction. Several recent frameworks have articulated different motivations for teaching computer science (Blikstein, 2018), visions for the field (Santo, Vogel, & Ching, 2019), and theoretical framings (Kafai, Proctor, & Lui, 2019). However, the dominant approaches in practice make compromises between the goals of broadly accessible computer cultures, deep engagement with material intelligence, and liberatory pedagogy. This paper develops the design rationale for a course and a research project seeking to recover the original spirit of a Constructionist approach to computer science, and in doing so, to question whether these goals could be mutually supportive rather than at each others' expense.

References

About ECS. Retrieved from http://www.exploringcs.org/about/about-ecs.

Ackermann, E. (2001). Piaget's constructivism, Papert's Constructionism: What's the difference. *Future of learning group*, 5(3), 438.

Anderson, M., & Jiang, J. (2018). *Teens' Social Media Habits and Experiences*. Pew Research Center.

Berlin, I. (1969). Two Concepts of Liberty. In *Four Essays on Liberty* (pp. 118–172).

Blikstein, P. (2008). Travels in Troy with Freire: Technology as an Agent of Emancipation. In C. A. Torres & P. Noguera (Eds.), *Social Justice Education for Teachers* (pp. 205–235).

Blikstein, P. (2018). *Pre-College computer science education: A survey of the field*. Mountain View, CA: Google.

Boaler, J. (2003). Studying and Capturing the Complexity of Practice—The Case of the "Dance of Agency." *International Group for the Psychology of Mathematics Education*, 1, 3–16.

Code.org. *How to teach one Hour of Code with your class*. Retrieved from https://hourofcode.com/ac/how-to.

diSessa, A. A. (2001). *Changing minds: Computers, learning, and literacy*. Mit Press.

Fosnot, C. T., & Perry, R. S. (1996). Constructivism: A psychological theory of learning. *Constructivism: Theory, Perspectives, and Practice*, 2, 8-33.

Freire, P., & Macedo, D. (1988). *Reading the Word, Reading the World*.

Goode, J., & Margolis, J. (2011). Exploring Computer Science: A Case Study of School Reform. *ACM Transactions on Computing Education*, 11(2), 1–16.

Harvey, B. (2012). The beauty and joy of computing: Computer science for everyone. *Proceedings of Constructionism*, 33-39.

Kafai, Y., Proctor, C., & Lui, D. (2019). From Theory Bias to Theory Dialogue: Embracing Cognitive, Situated, and Critical Framings of Computational Thinking in K-12 CS Education. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (pp. 101-109). ACM.

Moll, L. C., Amanti, C., Neff, D., & Gonzalez, N. (1992). Funds of knowledge for teaching: Using a qualitative approach to connect homes and classrooms. *Theory into Practice*, 31(2), 132–141.

Nelson, G. L., & Ko, A. (2018). On Use of Theory in Computing Education Research. *Proceedings of the 2018 ACM Conference on International Computing Education Research - ICER '18*, 31–39.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.

Papert, S., & Freire, P. (n.d.). *The Future of School*. Retrieved from http://www.papert.org/articles/freire/freirePart1.html.

Rodriguez, G. M. (2013). Power and Agency in Education: Exploring the Pedagogical Dimensions of Funds of Knowledge. *Review of Research in Education*, 37(1), 87–120.

Santo, R., Vogel, S., & Ching, D. (2019). *CS for What? Diverse Visions of Computer Science Education in Practice*. New York, NY: CSforALL.

Wilensky, U., & Papert, S. (2010). Restructurations: Reformulations of knowledge disciplines through new representational forms. *Constructionism.*

Representation of constructions of memory palaces with Learner-Generated Augmentation

Kenneth Y T Lim, *voyager@mac.com*
Office of Education Research, National Institute of Education, Singapore

Jie Bin Lok, *h1510085@nushigh.edu.sg*
NUS High School of Math & Science, Singapore

Andrew Gunawan, *andrewgunawan222@gmail.com*
Anglo-Chinese Junior College, Singapore

Ryan Lim, *ryan_lim_xing_yan@hotmail.com*
National Institute of Education, Singapore

Abstract

This study describes an approach to History education which leverages Augmented Reality (AR). It supplements work reported in Lim & Lim (2020) and shares how the profiles of participants and their respective responses to the intervention might be represented in forms not previously reported, with a view to increasing intelligibility of the data. The intervention aimed to explore the affordances of AR in mediating learner-initiated construction of representations of their memory palaces / method of loci, using the Learner-Generated Augmentation approach described in Lim *et al*. (2018), as applied to the memorization of historical facts. In this activity, participants used a free AR mobile application - *Just a Line* - to sketch out memory palaces of key information from a prose passage. This activity was trialled on student-teachers who are majoring in History at the National Institute of Education, Singapore. The resulting dataset was relatively complex, not least because the constructions of the memory palaces were potentially in three dimensions and overlain on the local environments of each participant. A contribution of this paper is the proposal of a format for representing such datasets to facilitate analyses and subsequent scaling.

Keywords

Augmented Reality, Phenomenology, Memory Palace, Method of Loci, Assessment for Learning, Learner-Generated Augmentation, History Education

Introduction

This study describes a pedagogical approach which leverages the affordances of Augmented Reality (henceforth abbreviated as AR) for learning, in the context of History education. Most learning interventions using AR technology currently adopt the paradigm of an expert-led model of teaching, in which the content is created by the teacher or an expert. This approach limits the degree to which learners can engage actively with the design iterations of the AR artifacts, in effect relegating them to not much more than consumers of content.

In contrast, this project explores the possibility of using the affordances of AR to facilitate participant-led learning interventions, in which the learner constructs representations based on their emerging understanding of a given topic.

In doing so, this project adopts the Learner-Generated Augmentation approach proposed by Lim *et al* in 2018. This approach affords learners, rather than domain experts, the opportunities to create their own boundary objects with AR, in contexts that are authentic to the learners themselves. This design approach is enabled by the introduction of new AR technologies, in particular, the introduction of the freely downloadable AR app Just a Line by Google in 2018 (Stolyar, 2018). Such AR apps enable users to sketch virtually on their surrounding environments using only their fingers and their mobile devices (Looper, 2018). This is depicted in Figure 1.

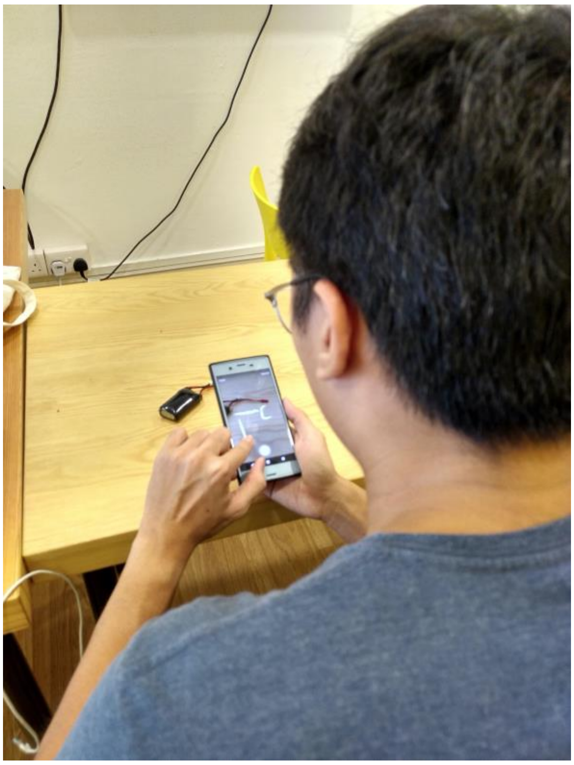


Figure 1. User sketching using Just a Line app

The learning activity combined the established Memory Palace (otherwise known as Method of Loci) strategy (Maguire, Valentine, Wilding, & Kapur, 2003) with the AR application Just a Line.

Students were asked to sketch out a memory palace using the Just a Line app and record this process.

This paper supplements work reported in Lim & Lim (2020) and shares how the profiles of participants and their respective responses to the intervention might be represented in forms not previously reported, with a view to increasing intelligibility of the data.

Review of literature

The study reported in this paper has roots in the work of Husserl and his writings on phenomenology. Husserl's earliest writings on phenomenology - *Reine Phänomenologie* - are now more than a century old. They remain extremely relevant to the issues facing the learning sciences today.

The memory palace / method of loci is a strategy that makes use of two important concepts, namely place, that is as familiar to the learner as possible (loci), and images representing the content that one wants to remember (imagines) (Yates, 1966). The learner begins by recalling a loci that is extremely familiar to themselves (Foer, 2011). The learner will then 'walk' through the loci in their minds and places images representing the set of information that he / she wants to remember at specific points. To recall this set of information, the learner passes through the specific points around these loci where the images of information were placed by him / her, triggering the learner's memory, thereby facilitating recall (Foer, 2011; Yates, 1966).

A study by Maguire *et al* (2003) comparing fMRI scans of participants of the World Memory Championships and a control group of participants who did not report exceptional memory capabilities found that when the method of loci was used by the former group of participants, regions of the brain important for memory, spatial memory and navigation were more active, suggesting a neurological basis for the effectiveness of this method.

However, a limitation of the memory palace technique is the fact that it is very dependent on the learner's sense of imagination. Some learners have less vivid imagination than others, making this method less effective for some. Another limitation would be that the memory palace is situated tacitly in the learner's imagination and as such, the teacher is unable to access the learner's memory palace to assess his / her naïve understanding of the topic.

In 2018, because of efforts of companies such as Facebook and Google in Augmented Reality, such intentional reifications of Husserl's 'objects of the mind' (Husserl, 1982) are now possible into the actual local environments of the learner.

By definition, with its emphasis on augmenting reality, AR is well placed to help learners connect the concrete to the abstract. Thus far, applications of AR technology are heavily reliant upon the expert to create the content for learners (Lim *et al*, 2018). This limits the potential scalability of AR technologies in education.

The Learner-Generated Augmentation approach addresses this limitation by inviting learners, rather than domain experts, to express their emergent understandings. Using the affordances of AR technology, learners can represent their naïve understandings, and their AR artifacts become boundary objects with which their peers and teachers can engage.

This affordance for learning – in which the learner is able to reify an intentional object in its 'proper presentation' (Brentano, 1973), in his / her local environment, from no pre-existing resources – has never been possible, till the present day.

The implications of this affordance as windows of intersubjectivity into the minds of learners are significant. The approach is therefore inherently constructionist in framing.

Objective

The objective of the study reported in this paper was to explore the affordances of AR with respect to the construction of memory palaces in the context of History education. The results of the study are described in detail in Lim & Lim (2020). The present paper focuses on the proposition of a format for representing the multi-dimensionality of the dataset which arose from the study.

Methodology

Participant profile

A pilot study was carried out with five student-teachers (three males and two females in their early twenties) from the National Institute of Education, Singapore. These student-teachers were all majoring in History and will become secondary school history teachers upon graduation. All of them had experience learning history at the secondary level, and all self-reported that they were comfortable with technology and used smartphone apps several times a day. Before they started participation in this study, participants were asked to rate their familiarity with the topic (the history of Malaya from the late 1930s to 1950) on of a scale of 1 to 5, where 1 represented no familiarity and 5 represented being very familiar with the topic. Participants self-reported a rating of 3 or more.

Only participant A reported prior experience with memory palace as a strategy. He elaborated that he did not like using the strategy, as he felt that it obligated him to memorise what the teacher wanted to him to remember. Other participants used methods such as writing out their own notes and repeating chunked information aloud to themselves to remember the huge amount of historical information accurately.

Description of learning activity

Each of the participants was invited to experience the intervention separately. Each run of the intervention was carried out across two separate sessions:

Session One – constructing the memory palace

- Participants were introduced to the memory palace technique and how it is generally used.
- Participants were introduced to the app *Just a Line* and practised sketching using this app.
- Participants were then given a passage from a historical text and asked to remember the information from this text by sketching a memory palace with Just a Line. Participants were encouraged to complete their sketches in an environment that was personally meaningful to themselves. Participants were asked to record their sketching process and were also told that during the subsequent session there would be a test of how much they could recall.

Session Two – revisiting the memory palace

- A week later, participants were administered a written test of recall. There was no time limit for participants to complete this written test of recall; participants were told to compete the test to the best of their ability.
- Each participant was asked to describe what they were attempting to sketch in their memory palace. Their description was recorded for subsequent analysis and for triangulation.

Results

Participants were given a passage from a history book – Owen, R. G., Chandler, D. P., & Roff, W. R. (2005) *The emergence of modern Southeast Asia: a new history*. Chapter 21 of the book documents the History of Malaya between the late 1930s to 1950. Participants were required to select from this text what they considered as key information that they should remember. Consequently, the information that participants represented on their memory palaces and wrote down in the test reflected what they deemed as important, thereby permitting possible insight into their respective understandings of the text.

Recording the construction process

The ability to record the screen as the learner sketches is one of the affordances of the app *Just a Line*. Using this function, each participant recorded a video of their memory palace. Participants tended to construct their memory palaces in their respective dormitory bedrooms. Each video recording was then analysed using a coding system with the following parameters, which reflected the major possible affordances of *Just a Line*:

- whether the participant made use of the ‘real-world’ background as anchor for the information given;
- whether the participant was physically stationary or moving around while completing the memory palace;
- whether the participant used the x, y and z axes to complete their memory palace; and
- whether the participant used words, numbers, and / or symbols, to represent information.

Figure 2 illustrates the use of some of these elements, as captured through a screenshot of single frame in one such video recording made by participants.

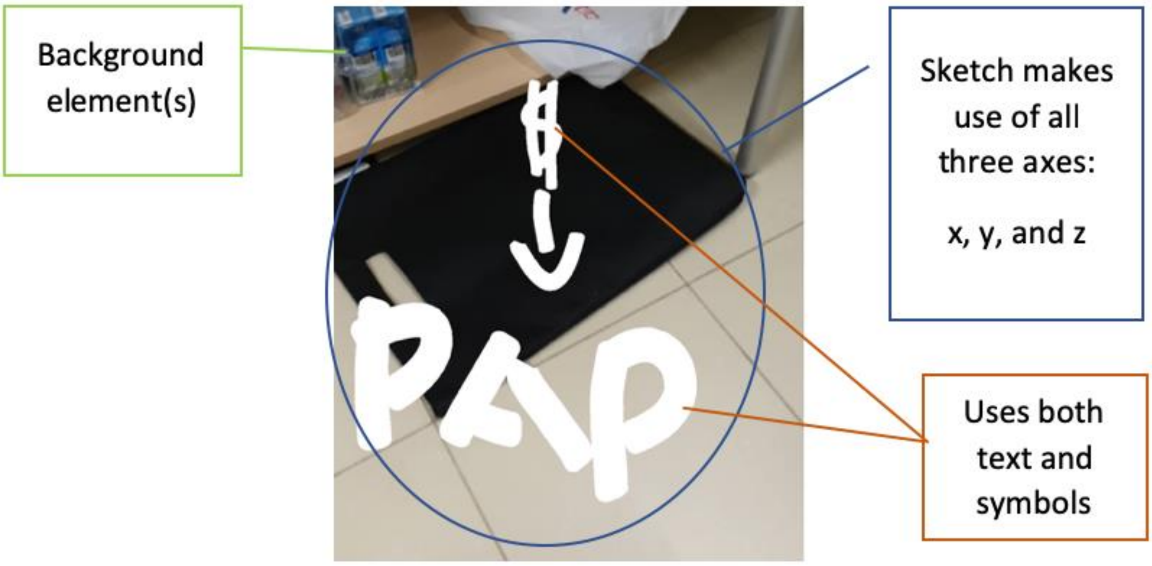


Figure 2. Annotated section of a memory palace as sketched by a participant

The results of this coding are reflected in Table 1.

Participant	Uses background elements	Moves around (not stationary on the spot)	Uses z axis	Uses symbols (in addition to text)
A	Y	N	N	Y
B	N	Y	Y	N
C	Y	Y	Y	Y
D	N	Y	Y	Y
E	Y	N	N	Y

Table 1. Coding of participants' Memory Palace sketches

Interview with participants

Interviews with participants are not reported in the present paper. The reader is invited to refer to Lim & Lim (2020) for details of the interviews.

Test of recall

The test of recall consisted of two main questions – one asking the participants to list events from the passage in a chronological order, while the other required participants to classify information according to the given categories. This test attempted to evaluate the effectiveness of this learning intervention in helping students remember information, rather than memorise it by rote (Mayer, 2002).

Participants' responses to the test were analysed by breaking them down into basic chunks of historical information. This method of using chunks to evaluate the accuracy of how much participants remember is consistent with similar studies, eg Landauer (1986) and Loftus (1985). The first question tested participants' skill of chronological thinking – a chunk was defined as a set of information that consisted of both an event and the time that the event occurred at. For the second question, which tested a student's ability to classify information given to them in the passage, a chunk was defined as one point of information written down by participants.

Given the importance of accuracy in the writing of historical facts when doing history (Carr, 1961), this study also attempted to evaluate the accuracy of the chunks remembered and written down by participants. Accuracy in this context was taken to be the degree to which information that was stated in the given text was correctly written. Using this working understanding of accuracy, even if participants wrote something that was historically accurate, but not stated in this given text, the chunk was considered invalid. If the information written by the participant omitted certain elements of time (e.g. if the passage gave the exact date, month and year of the event and the participant only wrote the month and year of the event), the chunk would be considered as partially valid.

The participants' responses in the test of recall were analysed based on the preceding concept of chunks. For a more detailed description of the coding scheme, the reader is invited to refer to Lim & Lim (2020).

The results of the analysis of all five participants' responses to Question One of this test are as follows:

Participant	No. of chunks given	No. of chunks correct	No. of chunks partially correct	No. of chunks not correct
A	3	1	0	2
B	9	8	1	0
C	8	7	1	0
D	7	3	2	2 (1 chunk not given in text)
E	8	7	1	0

Table 2. Coding of responses to Question One of the test of recall

The results of the analysis of all five participants' responses to Question Two of this test are as follows:

Participant	No. of chunks given	No. of chunks correct	Partially Accurate	No. of chunks not correct
A	6	4	0	2
B	12	12	0	0
C	8	6	1 (some parts of chunk not given in text)	2 (1 chunk not given in text)
D	7	5	0	2 (2 chunks not given in text)
E	11	6	0	5 (5 chunks not given in text)

Table 3. Coding of responses to Question Two of the test of recall

Analysis and Discussion

This paper supplements Lim's & Lim's (2020) preceding work in which data from the intervention was reported in an initial iteration. The coding of the data was represented in that paper in classical tabular formats.

Given the multi-dimensionality of the dataset, representing the data in tables potentially constrains its intelligibility, and – therefore – its subsequent use and translation to other contexts.

The Learner-Generated Augmentation approach was previously applied in the context of teaching Chemistry (as documented in Lim *et al*, 2018), in which students were asked to sketch molecular structures. However, differing ontologies of the disciplinary domains of Chemistry and History mean that the previous experiences of applying the Learner-Generated Augmentation approach in the teaching of Chemistry cannot be directly translated to the teaching of History, which is the intent of this study. In contrast with Chemistry, practitioners of History are required to form interpretations of historical events by interpreting 'facts' from different sources of information to form a historical narrative of past events. The sources one selects in this narrative will result in different people coming up with a diversity of historical narratives of the same historical event (Carr, 1961). Therefore, unlike the context of Chemistry, the sketches of history can vary significantly across various students.

Given the preceding argument, the authors of the present paper deliberated on the representation of the findings, such that they might be intelligible and facilitate analysis. A significant challenge was to afford a concise representation of each participant, as well as a panoptic overview across participants, in terms of:

- background / demographic profile;
- attributes of the memory palace / method of loci sketched; and
- performance in the assessment tasks of the intervention.

From the perspective the participant as the unit of analysis, data from Tables 1, 2 and 3 can be represented in the schematic form shown in Figure 3, in which:

- data in the innermost rings represents demographic profile
    - gender is in-most (blue = male, pink = female)
    - prior experience with memory palace is second in-most (green = yes, red = no)
- data in the bottom half of the diagram represents attributes of the memory palace / method of loci sketched (Table 1, green = yes, red = no), and
- data in the top half of the diagram represents performance in the two assessment tasks of the intervention (top left for Table 2 and top right for Table 3, going clockwise for each respective column in each of the two tables).

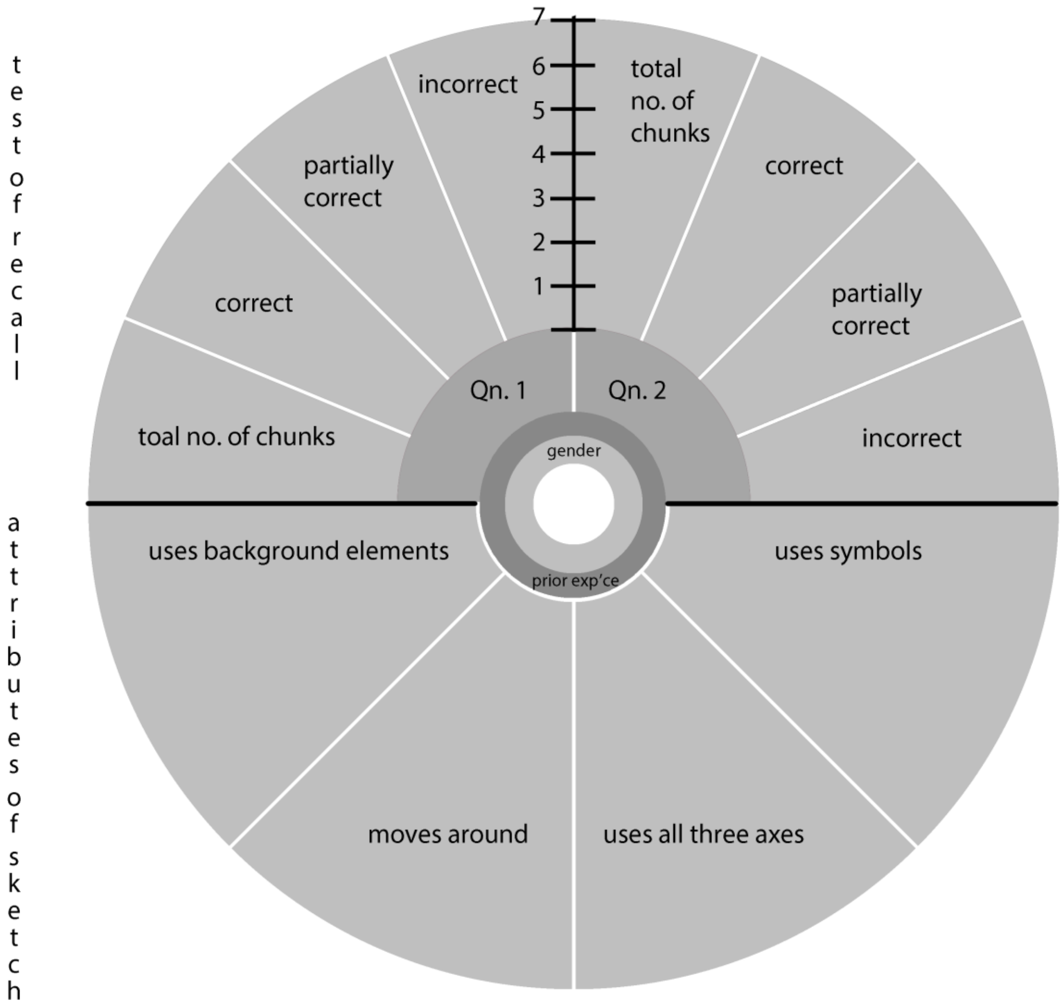


Figure 3. General schematic form

Such a representation offers the advantage of flexibility of scale to any similar subsequent studies. For example, Figure 4 depicts a panoptic overview of the major datasets arising from the present study, with the participant as the unit of analysis.

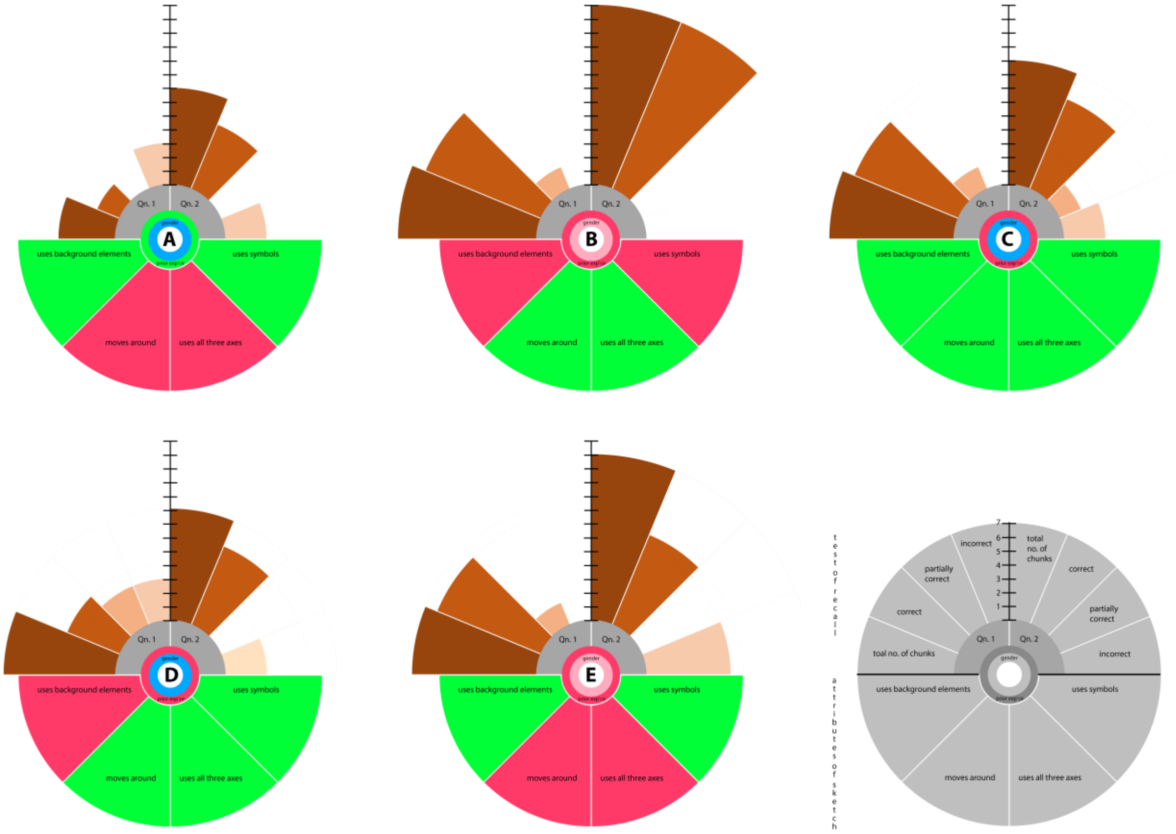


Figure 4. Graphical overview of all participants

Taking participant C as an example (Figure 5), he made use of all of the major affordances which sketching in AR afforded, namely, he used background elements and symbols, he moved around as he sketched, and he did not just sketch on a two-dimensional plane but made use of depth as well.

His description of the role of the context (the background elements of the sketch) and movements (such as looking up, down, and around) in enhancing their recall of different chunks of information is consistent with findings of Madan and Singhal (2012) with respect to embodied cognition. As Participant C articulated: “I realized that even say a day or two later, I’m able to remember the things that I compartmentalized. I can remember what labels I associate with the items that I tagged. I can remember what is the rationale of me selecting these particular facilities or function for a particular event”.

The deliberate efforts that participants invested in planning their respective memory palaces by thinking about how and where to represent information before actually sketching, meant that they were making explicit to their peers, what would otherwise remain as tacit organizational schema. In the words of Participant C: “How can I compartmentalize the information that I have, in to the facilities around me? I understand history as a form of narrative, so definitely there is a sequence of events, there’s a logical flow as to the narrative, so let’s see my environment, like how can it fit?”

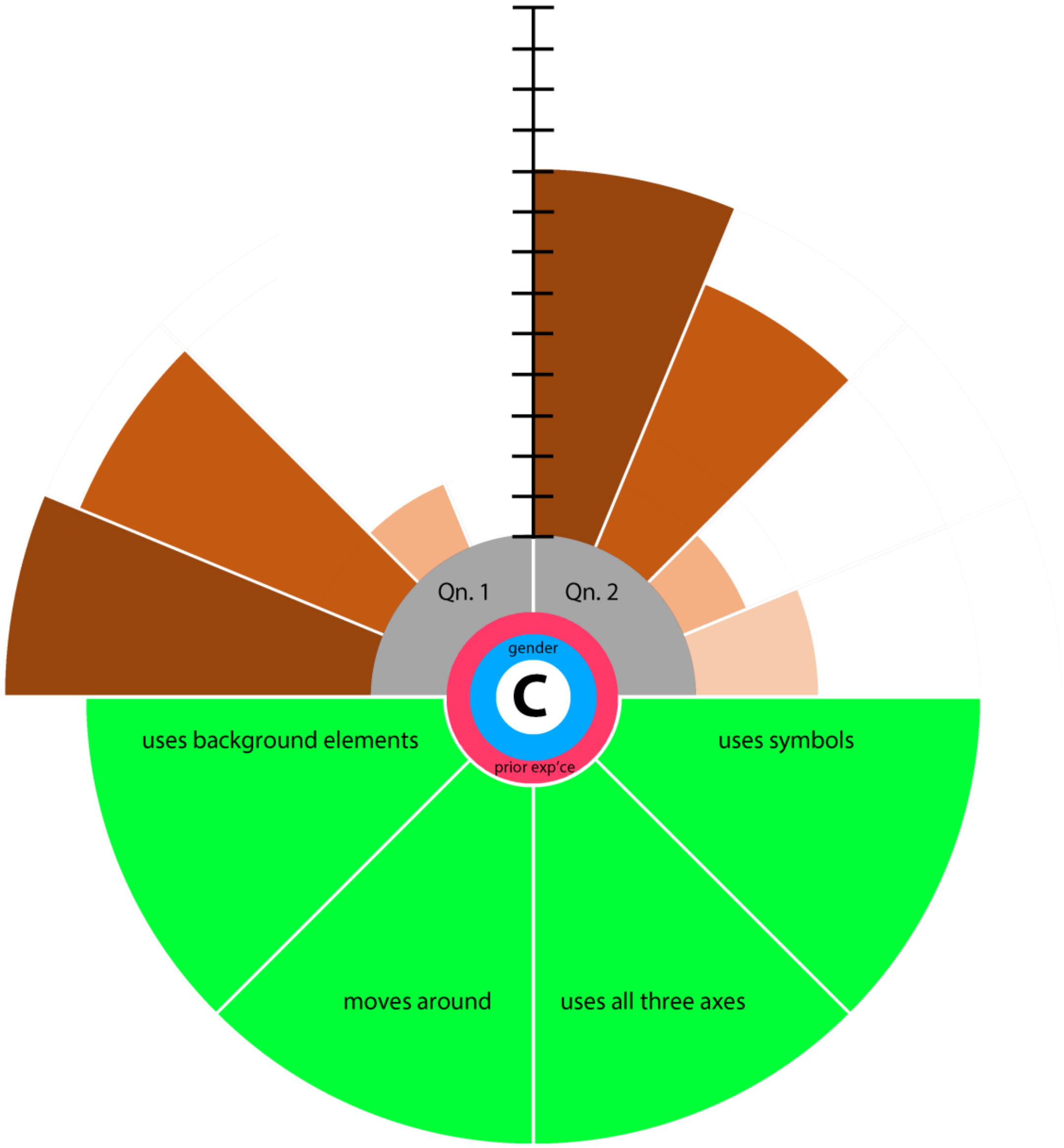


Figure 5. Participant C

Space permits a further discussion only of Participants B and E. As can be seen from the respective figures below, these two participants used very different affordances of sketching in AR. In terms of the test of recall, Participant B (Figure 6) gave the highest number of chunks of information amongst all participants; she also remembered these chunks largely accurately, making only one partial error out of the 21 chunks she remembered. Her sketch covered a greater spatial extent. From this, it was deduced that participant B's memory palace was more elaborate, enabling her to encode more chunks of important information that she wanted to remember into this memory palace.

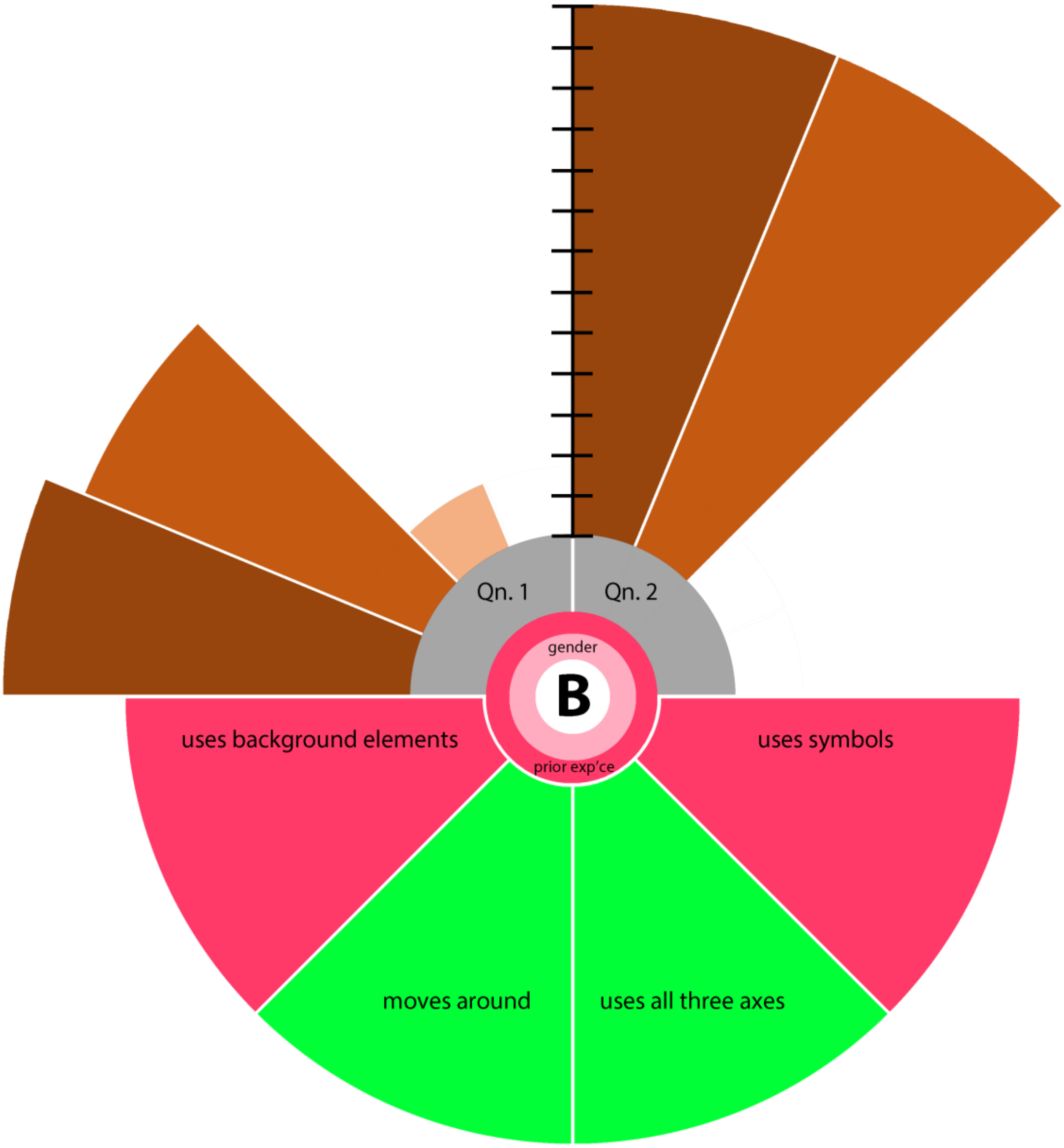


Figure 6. Participant B

As illustrated in Figure 7, one of the attributes contributing to the elaborate nature of Participant B's memory palace was that she (as did Participants C and D) sketched in three dimensions, and not simply on a two-dimensional plane.

Figure 7. Section of an example of a sketch in three dimensions – Participant B

This is in contrast to Participant E, the graphic of whom is depicted in Figure 8. A comparison of Figures 6 and 8 shows that the respective performances of Participants B and E in the test of classification of information was dissimilar.

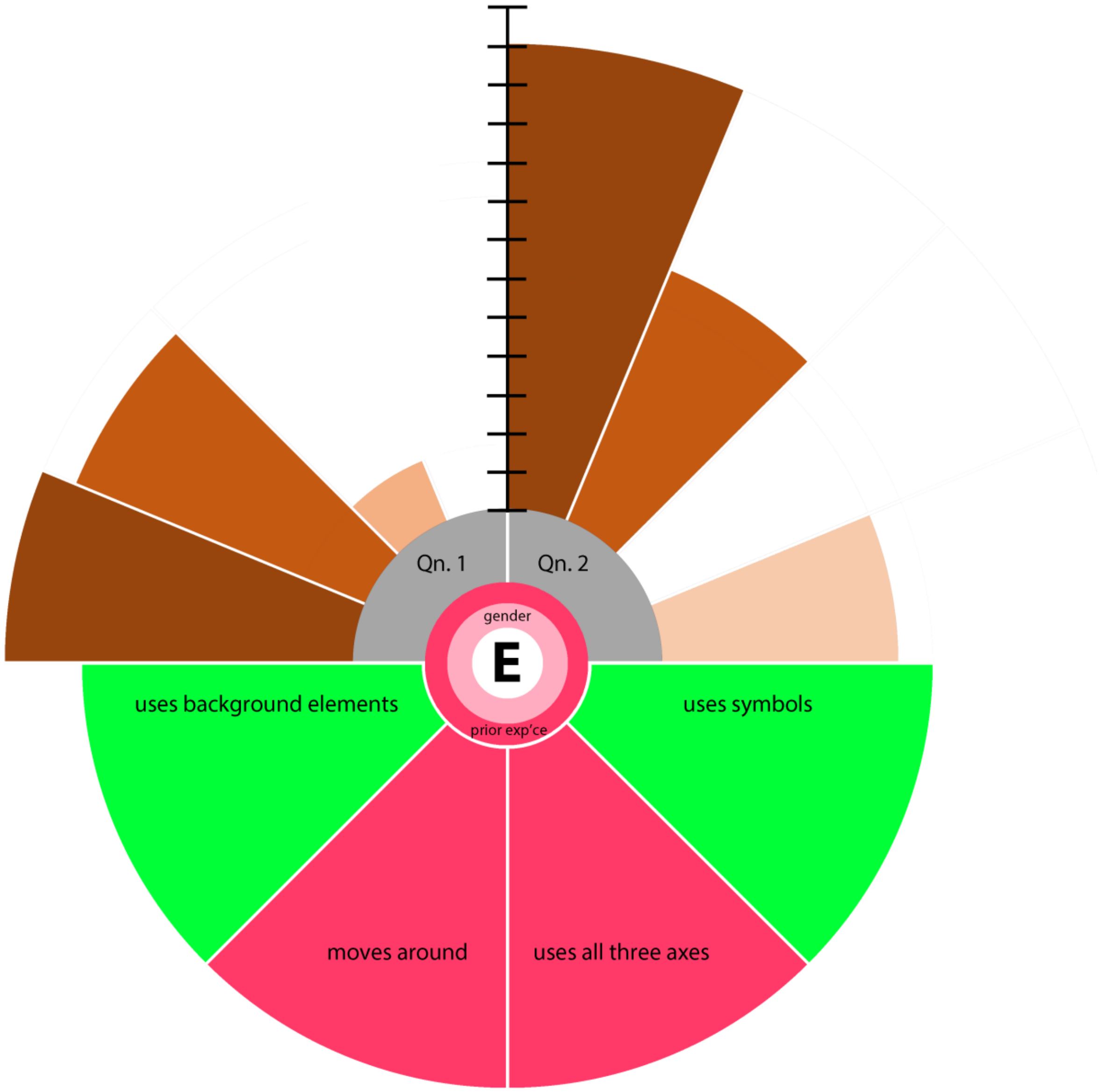


Figure 8. Participant E

The preceding figures depict of course only the bald facts of the relative similarities (or lack thereof) among the participants. No causal link is claimed between the bottom half of each of the figures with respect to the participants' respective performance as depicted in the top half. The figures are only meant as a shorthand to facilitate identification of interesting cases for deeper comparison and follow-up in any subsequent larger studies. As for the present study, the nuances which emerged from the interviews are elaborated upon in the earlier paper by Lim & Lim (2020).

Conclusion

The study described in this paper is an early attempt at understanding how AR might be used as an expressive medium for learners to surface what might otherwise have been tacit to their peers and teacher, through the construction of sketches in three dimensions.

AR is dissimilar to other mediums of sketching in the sense that canvases have the potential to be three-dimensional, as well as transparent (ie, background elements from the ‘real-world’ can potentially be anchors for elements of the sketch). These particular affordances of AR as an expressive medium pose challenges for the coding of the resulting sketches. In turn, these challenges are compounded when the object of the sketch has no received domain representation in and of itself (eg, a memory palace as opposed to a representation of molecular structure).

In the present paper, we have proposed a symbolic means of representing such data, as inspired by the geographic wind rose. Each ‘rose’ for Learner-Generated Augmentation takes a single participant as the unit of analysis, and attempts to convey the participant’s profile and responses to the intervention in terms of:

- background / demographic profile,
- attributes of the memory palace / method of loci sketched, and
- performance in the assessment tasks of the intervention.

In turn, it is our hope that this format for representing the multi-dimensionality of data which might potentially accrue from an AR-based learning intervention might perhaps inspire subsequent studies and analyses, not least because an advantage of this format is that it enables scholars to have a panoptic overview of the variations in the dataset, as expressed graphically.

In this way, we have attempted to suggest how what has long been seen as a virtual and transient medium of visualization – the authorship of which has thus far been confined to domain experts and technical experts, with concomitant implications to cost and ease of scalability – might be democratized as a canvas for construction as novices seek to represent their own naïve and evolving understandings, in disciplinary domains as ontologically dissimilar as Chemistry (Lim *et al*., 2018) and History (Lim & Lim, 2020).

Augmented Reality is presently a rapidly evolving field. Major stakeholders in the technology industry have been contributing new affordances and seeking to establish industry-wide standards in each of recent years. For example, in May 2018, Google introduced the affordance for learners to work collaboratively on the same sketch (Damiani, 2018). Likewise, on the horizon as a possible industry-wide standard are protocols to ensure a degree of permanence to AR artifacts anchored in local environments. These and other functionalities create possibilities for extensions of this activity beyond what is covered in the study described in this paper.

The state of the AR industry at present is not unlike where Massively Multi-player Online Role-Playing Games (MMORPGs) and virtual worlds were in the mid-2000s. At that time, the worlds such as *World of Warcraft*, *Second Life*, and *Minecraft* were portrayed in the mass media as little more than community spaces of social interaction and entertainment, but not as sandboxes for construction. It was not until the insights of Gee (2003), Shaffer (2007) and others encouraged scholars to explore these worlds through the lenses of epistemic frames and embodied cognition that their affordances for learning and social constructivism began to be understood. Their legacy can still be seen today in active *EVE Online*, *OpenSim* and *Minecraft* communities which have sustained.

These communities – of course – trace their roots to Papert’s earliest work on constructionism and his vision of *Mathland* and the intentional explorative movements of the turtle. The present authors see the work described in this paper as part of this palimpsest of constructed understandings, in how it seeks to afford learners with connections to their everyday environments by augmenting their phenomenological interactions.

References

Brentano, F. (1973). *Psychology from an empirical standpoint*. New York, NY, US: Routledge.

Carr, E. H. (1961). *What Is History?* Cambridge: University of Cambridge Press.

Damiani, J. (30 May, 2018). *Google Releases 'Just A Line' Drawing App For iOS, Furthering True Cross-Platform AR*. Retrieved from Forbes: https://www.forbes.com/sites/jessedamiani/2018/05/30/google-releases-just-a-line-drawing-app-for-ios-furthering-true-cross-platform-ar/#65f5f65173ca

Foer, J. (2011). *Moonwalking with Einstein: The Art and Science of Remembering Everything*. Penguin Books.

Gee, J. (2003). What video games have to teach us about learning and literacy. New York, NY: Palgrave.

Husserl, E. (1982). *Ideas pertaining to a pure phenomenology and to a phenomenological philosophy*. Dordrecht, The Netherlands: Springer.

Landauer, T. K. (1986). How Much Do People Remember? Some Estimates of the Quantity of Learned Information in Long-term Memory. *Cognitive Science*, 10, 477-493.

Lim, K. Y. T., Chen, K. H.-C., Lin, S.-W., Huang, J.-C., Ng, K. S.-E., Ng, J. J., . . . Woong, N. (December, 2018). Representations of Novice Conceptions with Learner-Generated Augmentation: A Framework for Curriculum Design with Augmented Reality. *Journal of Virtual Worlds Research*, 11(3), 1-18.

Lim, K. Y. T., & Lim, R. (2020). Semiotics, memory and Augmented Reality: History education with Learner-Generated Augmentation. *British Journal of Educational Technology*.

Loftus, G. R. (1985). Evaluating Forgetting Curves. *Journal of Experimental Psychology*, 11(2), 397-406.

Looper, C. d. (21 March, 2018). *Draw a line on the world around you with Google's 'Just a Line' ARCore app*. Retrieved from Digital Trends: https://www.digitaltrends.com/mobile/google-just-a-line-arcore/

Madan, C. R., & Singhal, A. (19 November, 2012). Using actions to enhance memory: effects of enactment, gestures, and exercise on human memory. *Frontiers in Psychology*, 3, pp. 1-4.

Maguire, E. A., Valentine, E. R., Wilding, J. M., & Kapur, N. (January, 2003). Routes to remembering: the brains behind superior memory. *Nature Neuroscience*, 6(1), 90-95.

Mayer, R. E. (Autumn, 2002). Rote Versus Meaningful Learning. *Theory into Practice*, 41(4), 226-232.

Shaffer, D. (2007). How computer games help children learn. Dordrecht, Netherlands: Springer.

Stolyar, B. (30 May, 2018). *Google's Just a Line AR app draws Android and iOS users together*. Retrieved from Digital Trends: https://www.digitaltrends.com/news/just-a-line-for-ios-and-new-collaborative-feature/

Yates, F. A. (1966). *The Art of Memory*. Chicago: The University of Chicago Press.

Seeds of (r)Evolution: Constructionist Co-Design with High School Science Teachers

Jacob Kelter[1,2,3], *jacobkelter@u.northwestern.edu*
Amanda Peel[1,3], *amanda.peel@northwestern.edu*
Connor Bain[1,2,3], *connorbain@u.northwestern.edu*
Gabriella Anton[1,3], *gabriella.anton@u.northwestern.edu*
Sugat Dabholkar[1,3], *sugat@u.northwestern.edu*
Ümit Aslan[1,3], *umitaslan@u.northwestern.edu*
Michael S. Horn[1,2,3], *michael-horn@northwestern.edu*
Uri Wilensky[1,2,3], *uri@northwestern.edu*
Learning Sciences[1], Computer Science[2] and Center for Connected Learning[3], Northwestern University, Evanston, IL, USA

Abstract

In the decades since Papert published *Mindstorms* (1980), computation has transformed nearly every branch of scientific practice. Accordingly, there is increasing recognition that computation and computational thinking (CT) must be a core part of STEM education in a broad range of subjects. Previous work has demonstrated the efficacy of incorporating computation into STEM courses and introduced a taxonomy of CT practices in STEM. However, this work rarely involved teachers as more than implementers of units designed by researchers.

In *The Children's Machine*, Papert asked "What can be done to mobilize the potential force for change inherent in the position of teachers?" (Papert, 1994, pg. 79). We argue that involving teachers as co-design partners supports them to be cultural change agents in education. We report here on the first phase of a research project in which we worked with STEM educators to co-design curricular science units that incorporate computational thinking and practices. Eight high school teachers and one university professor joined nine members of our research team for a month-long Computational Thinking Summer Institute (CTSI). The co-design process was a constructionist design and learning experience for both the teachers and researchers. We focus here on understanding the co-design process and its implications for teachers by asking: (1) How did teachers shift in their attitudes and confidence regarding CT? (2) What different co-design styles emerged and did any tensions arise?

Generally, we found that teachers gained confidence and skills in CT and computational tools over the course of the summer. Only one teacher reported a decrease in confidence in one aspect of CT (computational modeling), but this seemed to result from gaining a broader and more nuanced understanding of this rich area.

A range of co-design styles emerged over the summer. Some teachers chose to focus on designing the curriculum and advising on the computational tools to be used in it, while leaving the construction of those tools to their co-designers. Other teachers actively participated in constructing models and computational tools themselves. The pluralism of co-design styles allowed teachers of various comfort levels with computation to meaningfully contribute to a computationally enhanced constructionist curriculum. However, it also led to a tension for some teachers between working to finish their curriculum versus gaining experience with computational tools. In the time crunch to complete their unit during CTSI, some teachers chose to save time by working on the curriculum while their co-design partners (researchers) created the supporting computational tools. These teachers still grew in their computational sophistication, but they could not devote as much time as they wanted to their own computational learning.

Keywords

Co-design, teachers, high school, science, STEM, modeling, computational thinking

Introduction

In *Mindstorms* (1980), Papert foresaw the increasing role of computation in STEM and STEM education and began to describe how STEM content can be reformulated using computational representations. Wilensky & Papert (2010) named this kind of transformation a restructuration of knowledge. In the decades since *Mindstorms'* publication, computation has transformed nearly every branch of scientific practice. Accordingly, there is increasing recognition reflected in standards documents and reports that computation and computational thinking (CT) must be a core part of STEM education in a broad range of subjects. The approach of integrating CT into STEM courses has broad benefits including: (1) aligning science education with authentic scientific practices (2) introducing students to powerful computational ideas and tools that can be used to understand a broad range of scientific concepts and (3) increasing participation in computing by incorporating it into courses that every student takes (Wilensky et al., 2014).

Previous work developed a taxonomy of CT practices for STEM education (CT-STEM) consisting of four broad categories: data practices, modeling and simulation practices, computational problem-solving practices, and systems thinking practices (Weintrop et al., 2016). This taxonomy can serve as a framework to help design and analyze STEM curricula that incorporate CT. Several successful examples of such curricula have been designed and implemented in a range of scientific subjects including chemistry, physics, biology and materials science (Blikstein & Wilensky, 2009; Dabholkar et al., 2018; Levy & Wilensky, 2009; Sengupta & Wilensky, 2009). These units were largely designed by researchers and then taught by the researchers alone or with partner teachers. While this approach had some benefits, it did not emphasize teacher agency and growth or sustainability of adoption.

This paper presents preliminary findings from the Computational Thinking Summer Institute (CTSI) in which STEM teachers joined us to co-design "CT-ified" curricular units. In contrast to past CT-STEM projects in which teachers primarily implemented pre-designed curricula, in CTSI they co-created the curricula from the beginning. In *The Children's Machine*, Papert asked "What can be done to mobilize the potential force for change inherent in the position of teachers?" (Papert, 1994, pg. 79). The co-design process itself is a constructionist learning experience for teachers which can increase their CT skills and confidence. Additionally, when teachers co-design curricula, they are more likely to take ownership of the change, a core element for sustaining adoption (Coburn, 2003). Such opportunities for teachers are seeds for cultural evolution in education.

Co-design with teachers as a dual approach to curriculum development and professional development was rewarding for both teachers and the research team. Most teachers gained CT skills and were happy with the curricula they developed. However, the process was not without its tensions. Recent literature has identified tensions inherent in teaching teachers and supporting constructionism in the classroom (Brennan, 2015; Hickmott & Prieto-Rodriguez, 2018). We discuss a tension that arose for some teachers between designing their curricula and devoting time to develop their own computational thinking and comfort with computational tools.

Research Questions

In the rest of the paper we describe CTSI, present case studies of four teachers' co-design processes and present a preliminary analysis to answer the following questions:

1. How did teachers shift in their attitudes and confidence regarding computation?
2. What different co-design styles emerged and did any tensions arise?

The Computational Thinking Summer Institute

Participants

Eight science teachers from urban and suburban public high schools in the midwestern United States and one university professor representing physics, chemistry, biology, statistics, earth science and materials science participated in CTSI along with nine members of our research team (six graduate students, two post-docs and a curriculum developer). Some of the teachers have worked with members of our team in the past two years to implement a CT-STEM unit, but none of them had previously participated in full-fledged co-design. The teachers' prior experience and comfort with computational tools ranged from total novice to fairly experienced.

Setting and Timeline

The institute lasted for four weeks during the summer, five hours per day. Three days a week the teachers joined the research team on our campus for workshops and co-design. The other two days the teachers were expected to work on their units from home. The in-person days consisted of two hours of work in the morning, one hour for group lunch, and two hours of work in the afternoon.

Workshops and Computational Tools

The first week of the institute was devoted to a "crash course" in computational thinking in STEM. Both teachers and researchers participated as students in a CT-STEM lesson taught by a researcher and then analyzed it from a pedagogical perspective. They also participated in lessons aimed at showcasing the various computational tools available, including "unplugged" computational activities, NetLogo (Wilensky, 1999), NetTango (Horn & Wilensky, 2011), and CODAP (Finzer, 2016).

NetLogo is an agent-based modeling environment widely used in both scientific practice and education. NetTango is a blocks-based interface to NetLogo that can be integrated with a web-based version of NetLogo, allowing for an easier entry into creating agent-based models. This was the first time our group used NetTango in a professional development setting with teachers. NetLogo and NetTango were emphasized, because agent-based representations are a natural fit for many scientific phenomena, making them easier to understand and study (Wilensky & Papert, 2010). CODAP is a data analysis and visualization tool. It has been integrated with NetLogo Web to facilitate easy data collection from computational models. Together, these tools can be used to support the full range of practices in the CT-STEM taxonomy cited in the introduction.

After the first week, teachers could (and did) attend optional workshops to improve their skills with various computational tools. They also attended two workshops by the Principal Investigators of the project on the relationships between CT, science, and education.

Constructionist Co-design

The co-design aspect of the institute had two goals: for teachers (1) to have a constructionist learning experience to enhance their own learning of CT practices and constructionism, and (2) to incorporate constructionist design and pedagogical principles in the curricular units that they designed.

After the first week, the majority of the institute consisted of co-design with built-in time for feedback and group reflection. Each teacher was paired with a member of the research team as a co-designer. The teams each decided how they wanted to work together.

At the beginning of each week, small groups of three to five teachers and researchers gave one another feedback. Every teacher received feedback from one other teacher and at least one researcher. Throughout the feedback and discussion sessions, the researchers foregrounded constructionist approaches for designing the embedded computational models and tools as microworlds (Papert, 1980) and sometimes more specifically as Emergent Systems Microworlds (Dabholkar & Wilensky, 2019). At the end of each week all of the participants came together to reflect on their experiences, sharing both accomplishments and challenges.

Case Studies

Methodology

The following three cases involve four different teachers representing a range of experiences and roles throughout CTSI. To construct these cases, we combined our own observations with responses from teacher surveys conducted at the end of each week of CTSI and interviews conducted at the end of the institute.

Physics: David and Elizabeth

David and Elizabeth are physics teachers. David has eighteen years of experience teaching and Elizabeth has seven. David has been mentoring Elizabeth since she began working at David's high school two years ago. Since then, they have closely worked to align their teaching practices, leveraging the Modeling Physics Instruction curriculum (Hestenes, 1997). Neither David nor Elizabeth have prior experience participating with this research team. However, David has participated in other university education programs focusing on learning and teaching computational thinking in the sciences. In post-interviews, both David and Elizabeth described their teaching as using a pedagogy that foregrounds students as "creators and askers and formulators of knowledge" (Elizabeth, Post-interview) and themselves as scaffolders of materials and experiences to support student growth.

During the co-design institute, David and Elizabeth partnered to redesign two units on motion maps and electrostatics. Throughout CTSI, the pair focused substantial time on designing and programming NetLogo models and positioned their co-designers as providers of just-in-time help to support programming goals. Their experience in the summer program can be summarized as follows. After the first week of scaffolded experiences introducing computational tools, David and Elizabeth both expressed concern about whether they would be able to integrate agent-based approaches of CT into physics. However, during the first feedback and co-design sessions, they brought twelve ideas for integrating CT to deepen student understanding of charge behavior and interactions, 2D motion, and electric potential. They first modified an existing NetLogo model on electrostatics to better fit the needs of their instruction style and curriculum, then designed a NetLogo model on charge interactions (which a co-designer implemented), and finally modified a model to create a 2-D motion map. They leveraged three researchers as co-designers who supported the creation of NetLogo Models and CODAP activities.

This focus on programming appeared to be a result of David's prior interest in and exposure to programming in other languages, including Logo when he was a child. In surveys and interviews, David expressed increasing confidence and enjoyment in the programming focused design, while Elizabeth expressed uncertainty and decreasing confidence. Based on Likert responses to pre-/post-survey and post-interviews, both David and Elizabeth expressed increased comfort in computational data practices. On the pre-survey, both expressed comfort with computational modeling practices as well, reflecting their integration of many PhET (University of Colorado) models and other simulations throughout their curricula in the past. The post-survey, however, captures a divergence in the pair's comfort with computational modeling. Elizabeth expressed a decrease in her comfort with computational modeling practices, disagreeing with statements on comfort defining computational modeling, finding resources, and helping colleagues. Additionally, she selected neutral on statements about adapting lessons, creating new lessons, and identifying students' practices in computational modeling. Conversely, David expressed increased comfort in computational modeling practices, strongly agreeing that he could answer student questions and find resources about computational modeling. We hypothesize that Elizabeth entered the summer institute with a conception of computational modeling practices more aligned with the use of simulations, like PhET models, that she had often integrated in her curricula and then was exposed to a more nuanced conception and set of associated practices which disrupted her confidence.

More generally, this divergence in confidence appears to be influenced by David's and Elizabeth's differing backgrounds in programming and how this impacted their needs and perceptions of the co-design experience. Elizabeth initially expressed interest and enjoyment from coding activities but then expressed a shift in confidence as the pair transitioned into modifying and programming models: "you guys gradually exposed us to different things we could do and changing the colors of the turtles. I thought it was really fun and made me feel relatively confident at the time. Then the next week came" (Elizabeth, Post-interview). She expressed feeling uncomfortable with NetLogo code and that she either needed greater exposure to programming in NetLogo or for the researcher co-design partners to develop models. Conversely, David expressed that he "had some experience with coding and Netlogo.... So [he] sort of took on the role of trying to start modifying existing NetLogo models" (David, Post-interview). He expressed excitement throughout the weeks as he tinkered with, debugged, and developed functioning models with the support of his co-designers. For David, this process appears to have allowed him to push his own programming abilities while largely leveraging his co-design team and teaching partner for thinking about content or providing just-in-time help. For Elizabeth, this process appears to have highlighted the practices she had yet to learn, resulting in lower comfort in modeling despite leaving the program with personally identified skills in reading code and programming. However, despite decreases in comfort on the post-survey, Elizabeth was very positive about the experience in the post-interview, mentioning that she felt confident reading code in NetLogo models, helping her students, and teaching the unit overall but that she would need either more NetLogo programming education or support from researchers to make units with new models.

Chemistry: Clara

Clara has taught honors and AP chemistry for ten years. She worked with the research team two years prior to participating in CTSI, teaching CT-integrated chemistry units designed by the research team. Clara viewed her role in the classroom as being a facilitator of student-driven learning.

> I am a guide on helping them figure out what questions to ask to be able to answer the bigger question that I've posed for them at the beginning of any given unit. And then I'm there to sort of help them put the puzzle pieces together of how to answer that big question… But I also am there to help them to work with each other and to understand that I am not the only one in the room with any knowledge. (Clara, Post-Interview)

During the institute, Clara co-designed a new unit with one member of the research team focused on heat and energy transfer during chemical reactions. Clara's co-design process was curriculum-centric: she focused on designing the curriculum while providing her co-designer with the overall vision for the computational aspects of the unit. For example, Clara wanted a NetLogo model of the decomposition of potassium iodide in water to show the energy transfer taking place in the reaction. She described the phenomenon, what she wanted the model to do, and how she wanted it to look. Her co-designer then created the model and altered it based on her feedback. Clara's final unit used two NetLogo models and computational data manipulation.

Based on a Likert response pre-/post-survey and post-interviews, Clara identified an increase in her comfort with CT practices and their integration after CTSI, specifically with identifying, defining, and teaching computational data practices. She also reported feeling more confident in her ability to modify curricula to include computational data practices:

> I've been very excited that I'm integrating some CODAP this year, which I ...didn't use at all prior to this summer… I'm excited because I already see other possible places in my year that I can use this. (Clara, Post-Interview)

Similarly, after CTSI Clara felt more confident in her abilities to identify computational modeling practices and to adapt curricula or create new lessons to include computational modeling.

Clara and her co-design partner included several opportunities for students to engage in constructionist learning. In one lesson, students construct statistical models using the computational data tool CODAP. Students also use a NetLogo model to run experiments &

develop hypotheses on the nature of chemical reactions. This is in direct contrast to how Clara taught this lesson in the past when students wrote and balanced equations on a worksheet. Lastly, Clara and her co-designer included activities in which students discuss the underlying programming logic of the models to investigate how the models were constructed.

Biology: Tracy

Tracy has been a high school biology teacher for 30 years, teaching all levels of biology from basic to advanced and AP level classes. Prior to the institute, Tracy had worked with the CT-STEM team for two years in which her role and involvement in the design of the curricula expanded. Initially, she saw the implementation of curricula as a "research project" in which she did not want to "mess up" (Tracy, field notes). In the second year, Tracy identified natural selection as a content area that would benefit from computational integration and worked with a researcher to create a computationally enhanced version of the curriculum. During the implementation of this unit, Tracy took a more active role in the classroom, discussing a natural selection phenomenon using NetLogo models projected at the front of the classroom and connecting students' use of CODAP for data analysis with another phenomenon that she routinely taught.

Tracy entered CTSI with these two years of experience and, according to our pre-survey, moderately high comfort with integrating computational modeling and data practices into her classroom. During the institute, she chose to redesign a curricular unit on experimental design. Tracy saw integrating computational thinking into this curriculum as an opportunity to accomplish two goals: one, overcome obstacles that students encounter when performing physical experiments by integrating computational modeling practices, and two, make the student experience more authentic to actual scientific practice by integrating computational data practices.

The unit had students design experiments to find the preferred habitat conditions of the *pill bug* (a species of woodlouse colloquially also called roly-poly). Tracy decided that students would start out with a regular physical experiment using a simple environment of two connected chambers, one damp and one dry. The students would then place pill bugs inside of the environment and observe the change in population of the two chambers over time. After the physical experiment, Tracy then wanted students to digitally explore, modify and recreate the animal behavior experiment by using a NetLogo model and then creating a computational model using NetTango (Figure 1).

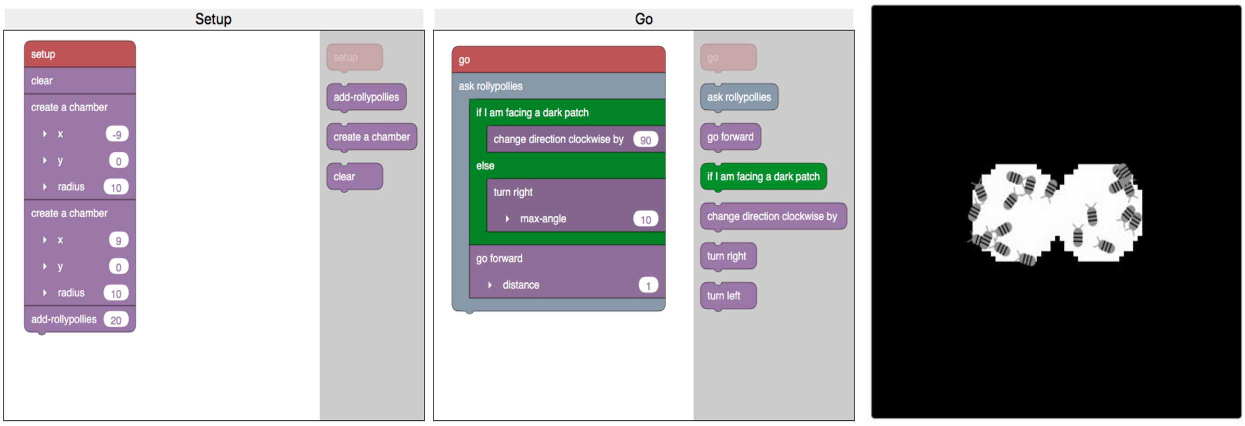


Figure 1: A screenshot from the model of pill bugs that Tracy built with the help of her codesign team. Students use the NetTango blocks (left) to program the behavior of the pill bugs (right).

Incorporating programming was a large change for Tracy who avoided this aspect of computational thinking throughout her two previous years of involvement in the project. In fact, she noted this as one of her largest advancements:

> [before] I would not attempt to code anything all by myself. I would go in now [and program myself]. And [before] like if the kids had a question [about programming], I would direct it

> to the CT-STEM team. Now I can do something over there [at the student's computer] and see if they can figure it out. I still need to work on that, but it's farther than I thought I would get. (Tracy, Post-interview)

In fact, while she saw her role in the classroom as a "facilitator" rather than as an "instructor," she mentioned feeling very intimidated by students' ability to quickly uptake new computational tools and surpass her own skills: "before they knew what my computer skills were like, so they just made fun of me" (Tracy, Post-interview). Rather than seeing programming as a realm of new opportunities for students to engage in the science content, she saw it as a blackbox in which she would not be able to independently support her students. The institute was the first time that she was able to engage with programming tools, specifically NetLogo and NetTango, as a learner herself. After having the experience of constructing a model from scratch, she felt able to support students in that process, even without being an expert programmer. While her co-design partner still supported her greatly in the development of the new model, Tracy took an active role in this process, designing features, contributing code, and actively choosing which programming blocks should be made available to students.

While gaining programming experience during the institute was key, Tracy also talked about how designing these models had informed her ideas of what it means to use a computational model:

> I just have to make sure that they (the students) understand that we coded the preference in there. We might be wrong ... and there are other things that might affect the behavior of the roly-poly that we're not aware of. So, you know, remember this is just a model. So, let's talk about the pros and cons of using the models. Just like there were in using the actual physical experiment. (Tracy, Post-interview)

Through co-designing the model, Tracy gained a deep understanding of how each design choice affects the outcome of the model. The model was no longer just a curricular tool to be *used* to understand the behavior of pill bugs; it was an artifact to understand what it *means to create* a computational model. She acknowledged that students could gain a deeper understanding of the phenomenon at play by building these models themselves.

At first, Tracy had major concerns about "deadlines" in her classroom—that by integrating computational activities, she would be losing time needed for other topics in the Advanced Placement Biology class. By the end of CTSI, she saw these computational open-ended design and investigation activities as *integrated* rather than *additional*: "I'm going to just really try to incorporate it by removing the direct lecture and incorporating the content. And it shouldn't take any longer. Like my evolution unit did not take any longer because I did [a CT-STEM unit]…and they learned just as much" (Tracy, Post-interview).

It can be difficult for even experienced teachers to change their pedagogical styles to integrate constructionist CT activities. However, Tracy's learning and co-design experiences during CTSI demonstrate her emerging understanding of how integrating CT in a constructionist way might fundamentally change her students' experiences and science learning.

Discussion and Conclusion

Regarding our first research question of how teachers shifted their attitudes and confidence regarding computation, most teachers gained considerable confidence and skills in computational tools and various aspects of CT in STEM over the course of CTSI. The one exception to this trend was Elizabeth's reported decreased confidence in computational modeling practices specifically. However, based on her responses in the post-CTSI interview, we interpret this as reflecting her more expansive notion of computational modeling at the end of the summer compared to the beginning. So, even this result can be seen as a positive development. Moreover, teachers expressed more nuanced understanding and appreciation of the ways that CT and computational tools can support students' experiences and deepen science learning.

In answer to our second research question regarding co-design styles, a number of different styles emerged, which can be placed on a spectrum of teacher roles. At one end of the spectrum, teachers such as Clara focused almost exclusively on curriculum design, leaving the construction of computational models and tools to their co-designers. On the other end of the spectrum are teachers who focused equally on curriculum design and constructing computational models/tools. Tracy and David were close to this end of the spectrum, as they were actively involved with co-constructing computational models. Ideally, teachers become as comfortable with the computational tools as they are with curriculum design.

The different co-design styles emerged naturally in the co-design teams. While this was largely positive, one tension should be noted. On the positive side, allowing for different co-design styles meant that all teachers were supported to engage meaningfully with creating computationally enhanced curricula, regardless of their progress towards becoming computationally sophisticated educators. Those with more CT experience were able to co-construct new computational models and tools, while those with less could focus on designing their curriculum and advising on the pedagogical role of the computational models which their co-designers implemented. However, this affordance, combined with the time pressure to create a whole curricular unit during CTSI, created a tension for some teachers between developing their own computational skills and working to finish their units. One teacher remarked at an end-of-week reflection “I kind of wish we just had time to learn NetLogo.” For teachers with less computational experience, it was natural to have their co-designer primarily create the computational tools for their lessons, but this may have inadvertently limited their opportunities to increase their own CT skills. Based on the surveys and interviews, all of the teachers still increased in their confidence with at least some elements of CT, but for some teachers this progress may have been curtailed by the tradeoff between curriculum development versus computational tool building and attendant learning. In future years, we may offer additional skills-based workshops outside of the four-week CTSI as one way to ease this tension.

In the school year after CTSI, the teachers implemented their units with support from the research team. Future work will analyze these implementations to further understand the role of co-design in shaping teacher practice and student outcomes in CT integrated science curricula. Building from the lessons and successes of this inaugural CTSI, the research team will host a second institute this summer for both new and returning teachers.

All in all, CTSI was a rewarding experience for both the teachers and the research team. Engaging teachers in constructionist co-design helps them grow as educators and helps us grow as researchers, especially in our sensitivity to the needs and tensions faced by teachers. Most importantly, co-designing with teachers empowers them to be agents for constructionist cultural evolution.

Acknowledgements

We thank Sally Wu, Golnaz Arastoopour Irgens, and Hillary Swanson for their role in helping design and execute CTSI and all of our participating teachers for co-designing with us and implementing the units. This work was made possible through generous support from the National Science Foundation (grants CNS-1138461, CNS-1441041, DRL-1020101, DRL-1640201 and DRL-1842374) and the Spencer Foundation (Award #201600069). Any opinions, findings, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

References

Blikstein, P., & Wilensky, U. (2009). An atom is known by the company it keeps: A constructionist learning environment for materials science using agent-based modeling. *International Journal of Computers for Mathematical Learning*, *14*(2), 81–119. https://doi.org/10.1007/s10758-009-9148-8

Dabholkar, S. & Wilensky, U. (2019). Designing ESM-mediated collaborative activity systems for science learning. Proceedings of International Conference of Computer Supported Collaborative Learning (CSCL) 2019, Lyon, France.

Dabholkar, S., Anton, G., & Wilensky, U. (2018). *Developing Mathetic Content Knowledge using an Emergent Systems Microworld*. Proceedings of Constructionism 2018, Vilnius, Lithuania.

Finzer, W. (2016). *Common online data analysis platform (CODAP)*. Retrieved from concord.org/codap

Hestenes, D. (1997). Modeling methodology for physics teachers. *AIP Conference Proceedings*, *399*(1), 935–958. https://doi.org/10.1063/1.53196

Horn, M., & Wilensky, U. (2011). NetTango [Computer Software]. Evanston, IL: Center for Connected Learning and Computer Based Modeling, Northwestern University. Retrieved from http://tidal.sesp.northwestern.edu/nettango/

Levy, S. T., & Wilensky, U. (2009). Students' Learning with the Connected Chemistry (CC1) Curriculum: Navigating the Complexities of the Particulate World. *Journal of Science Education and Technology*, *18*(3), 243–254. https://doi.org/10.1007/s10956-009-9145-7

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas* (2nd ed). New York, NY: Basic Books.

Papert, S. (1994). *The Children's Machine: Rethinking School In The Age Of The Computer* (Reprint edition). New York, NY: Basic Books.

Sengupta, P., & Wilensky, U. (2009). Learning Electricity with NIELS: Thinking with Electrons and Thinking in Levels. *International Journal of Computers for Mathematical Learning*, *14*(1), 21–50. https://doi.org/10.1007/s10758-009-9144-z

University of Colorado. (n.d.). PhET Interactive Simulations. Retrieved October 24, 2019, from PhET website: https://phet.colorado.edu/

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Wilensky, U. (1999). *NetLogo*. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering computational literacy in science classrooms. *Communications of the ACM*, *57*(8), 24–28. https://doi.org/10.1145/2633031

Wilensky, U., & Papert, S. (2010). *Reformulating Knowledge Disciplines through New Representational Forms 1*. Presented at Constructionism, Paris.

Student meanings on programming abstraction when they modify a ‘half-baked’ simulation game

Marianthi Grizioti, *mgriziot@ppp.uoa.gr*
Educational Technology Lab, Department of Educational Studies, National and Kapodistrian University of Athens, Greece

Chronis Kynigos, *kynigos@ppp.uoa.gr*
Educational Technology Lab, Department of Educational Studies, National and Kapodistrian University of Athens, Greece

Abstract

Abstraction is considered a key concept to programming and to computational problem-solving. It is also a soft concept, hard to be conceived and expressed by young and inexperienced students, who tend to focus on syntactic details and create non-abstract solutions when they program. At the same time, most educational coding approaches focus on teaching specific programming concepts through closed and unlinked exercises. It seems that there is a need for designs that would make the practice of programming abstraction accessible to students, by providing them with diverse computational affordances and representations for exploring and expressing ideas in a meaningful context.

In this paper, we discuss a design-based study that studies student meaning-making processes on programming abstractions as they collaboratively play, discuss, and modify a half-baked simulation game in ChoiCo (Choices with Consequences) environment. The game, that is called CT Chef, simulates the management system of a restaurant through several choices that affect several game attributes (e.g Money, customers, quality). The game is half-baked, that means that it has didactically engineered bugs that make the gameplay unbalanced, aiming to challenge the players to question and modify it. Students modified the game using the three computational affordances offered by the ChoiCo environment: a map designer, a database and block-based programming (Figure 1). The initial findings allowed us to develop an understanding of students learning strategies about powerful ideas related to programming abstractions, such as variables, classes, and data structures.

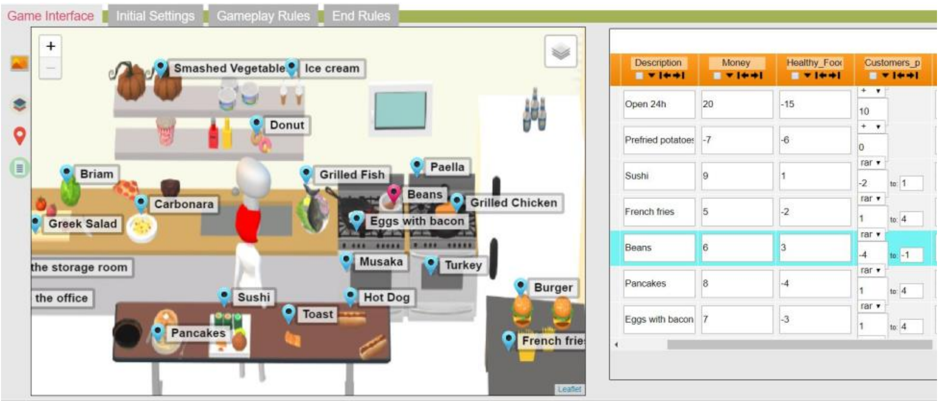


Figure 1. The 'CT Chef' game in the “design mode” of ChoiCo

Keywords

Game-based learning, programming abstraction, integrated affordances, half-baked artefacts, simulation game, computational thinking

Theoretical Background

Making programming abstractions accessible to students

Abstraction is a key concept of programming and computational thinking (Armoni, 2013, Kramer, 2007, Wing, 2008). It involves the process of creating general solutions from specific instances by recognizing patterns and ignoring details. A variable is an abstract representation of any possible value, a data structure or a class generalizes information from specific instances to abstract objects that stand for many, the user interface is an abstract level that 'hides' information form the user, and an algorithm is an abstraction of a step-by-step solution to a certain type of problems. Therefore, the ability to understand, create, and efficiently use abstractions is essential for dealing with complexity in computational problem-solving.

However, traditional educational activities focus on teaching programming concepts (e.g. loops, conditionals) as individual clusters of knowledge through closed, and usually unlinked, coding tasks and quizzes (Brennan & Resnick, 2012). In this strict approach, students don't have the opportunity to explore more general computational practices such as abstraction. On the other hand, young and inexperienced students usually have neither the knowledge nor the experience to perceive and use abstraction when they program an artefact (e.g. a game) from scratch (Armoni, 2003). It seems that there is a need for new designs that would enable students to progressively access, explore and use types of abstraction in a meaningful context for them.

To address the above challenge, we discuss the idea of students modifying a questionable digital game, instead of designing it from scratch, by using diverse integrated affordances, which so far have been used as individual learning tools. In the presented study students modified the simulation game 'CT chef' in ChoiCo game designer[34], a constructionist environment that integrates a database, a map GIS-based editor and block-based programming for game design (Kynigos & Grizioti, 2020). We exploit these three affordances as means for students to access types of abstractions integrated into the game system, such as variables, data structures and classes, and to experiment with practices related to abstract thinking, like pattern recognition and generalization.

Simulation games with questionable rules and content

'CT chef' belongs to the genre of choice-driven simulation games (Gee, 2008). That means that the game flow is formed by the player's actions upon the simulation in a non-linear way. Therefore, the players have to analyse the interconnections and the rules of the simulation system to improve their gameplay strategies. Simulation analysis as an educational activity allows students to deal with an abstract form of complex, real-world phenomena and to interact with their components in multiple levels of abstraction (Weintrop & Wilensky, 2014).

In order to engage students with computational abstraction in a meaningful context, we integrated into the game questionable rules and content. The game, that simulates the management of a newly opened restaurant, deals with subjective socio-scientific issues aiming to provoke students to doubt them and modify them (e.g. the game assumes that the junk food will bring new customers). Furthermore, the game was designed as a half-baked artefact (Kynigos, 2007). Half-baked game design involves the didactical engineering of bugs into the rules, the content, or the values, inherent in gameplay. Related studies have shown that these 'bugs' work as trigger mechanisms for students to discuss the game content and express computational solutions for improving or extending it (Grizioti & Kynigos, 2018; Kynigos & Yiannoutsou, 2018, Kynigos & Grizoiti 2020).

With this design, we expect the game to become the controversial 'topic' of a 'computational conversation' between the students. In this 'conversation' students would use programming, in conjunction with the other affordances, in order express and communicate their ideas on the questionable elements by creating and sharing a new, personalized, version of the original game.

[34] ChoiCo is available online: http://etl.ppp.uoa.gr/choico/

The game, as a 'living document', keeps evolving, transforming, and being reconstructed, waiting each time for new ideas to be expressed in the form of modifications or to be discovered through the gameplay. This design, embracing the pedagogical principles of constructionism learning theory, approaches programming as a means of communication and collaboration in an authentic social context and learning as the process of a continuous bricolage and sharing of digital artefacts (Papert, 1980; Kynigos, 2015; Kafai & Bruke, 2016).

'CT Chef': A game for Computational Thinking

CT Chef (Computational Thinking Chef) is a choice-driven game that simulates the management of a newly opened restaurant[35]. The player, having the role of the restaurant owner, makes choices that have positive and negative consequences on the five game parameters: money, healthy_food, customers_per _day, animal_products, vegetable_products. The goal of the game is to keep the restaurant open, that means to make as many choices as possible without losing from any of the five parameters. The available choices include the menu dishes (e.g. sushi, grilled fish, hot dog), the purchase of raw material (e.g. fresh vegetables, canned vegetables) and management actions (e.g. advertising). The gameplay involves balancing of the resources, decision making and analysis.

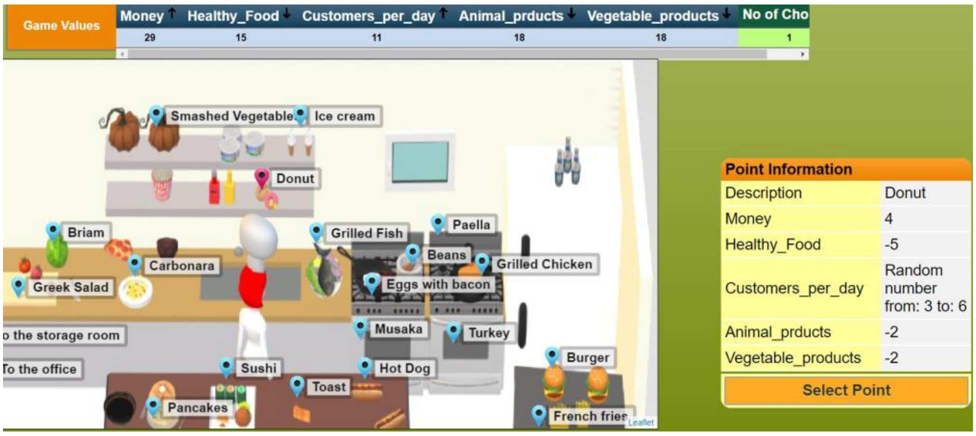


Figure 2: The main game layer of the 'CT Chef' game

Design Decisions

The following design decisions were made so that powerful ideas of programming abstraction were integrated into the gameplay. First, the interface is divided into three areas in the form of map layers, the kitchen, the storage room and the office, and each area includes a certain type of choices. in the kitchen the player can select between menu dishes, in the storage room there are raw materials to purchase and, in the office, there are available management actions. With this design, we aim to engage students with the idea of categorization and abstraction of information between interconnected interfaces. Students can access and modify the interfaces, the available choices and the graphics using the map editor (Image 3).

[35] CT Chef half-baked game is available online on: http://etl.ppp.uoa.gr/choico/?ctchefEng

Figure 25: Using the map editor and the interactive database to modify the area 'Storage Room' of CT Chef

Furthermore, the game choices can be abstracted to general categories that follow the same rules for their consequences, e.g. all menu dishes would increase the ‘money’ parameter and reduce the parameters ‘animal_products’ and/or ‘vegetable_products’, all raw materials have the reverse consequences, that is they would always decrease the money parameter and increase the parameters ‘animal_products’ or ‘vegetable_products’. The idea of choices being an instance of an abstract category with generic rules resembles the notion of classes in programming. Students can access and modify the choices and their consequences using the database affordance.
Finally, the gameplay rules were designed so that they include diverse uses of conditional statements and variables. For changing the ending rules of the game students would have to explore, compare and modify different types of conditionals; from simple conditional structures to complex ones such as nested ifs, Boolean conditions, conditions with mathematical equations.

Engineered Bugs

We designed the game to be a ‘half-baked’ artefact by engineering three intentional bugs to its structure (Grizioti & Kynigos, 2018). The bugs do not affect the game functionality, but they make the gameplay unbalanced and unfair. The first is related to the number of the available choices of a certain category. Only 3 out of 34 available choices increase the value of the parameter ‘vegetable_products’. This makes it very hard for the player to avoid running out of vegetables very soon.

The second bug is in the consequences that food choices have to the parameter ‘customers per day’. All choices have a random consequence to this parameter within a range of four numbers (e.g. 0 to 4, -2 to 2). However, three choices (Musakas, Grilled Chicken and Smashed Vegetables) have a much wider range of values as consequence for this parameter, making the game unstable (-20 to 9, -15 to 8 and -20 to -1 accordingly).

The third bug is a condition that ends the game if the value of parameter ‘Money’ is less or equal to 15. Even though this is correct in terms of coding, it is not logical in the game context, considering that the initial value of Money is 20.

With the above integrated ‘bugs’ we aim to provoke discussions about the game structure and challenge students to analyze and modify it. Through this study, we discuss the meanings that students expressed and the powerful ideas they explored about types of abstraction, such as variables, classes, patterns, data structures and interface layers, in order to improve these faulty behaviors.

Design-based study

To gain an understanding of student meaning-making processes we organized an empirical study with students of a public junior high school in Athens, Greece. The research method we used is that of Design-Based Research (Bakker, 2018; Edelson, 2002). The study was carried out through repeated cycles of design and implementation, utilizing every implementation as an opportunity for data collection, evaluation, and review in two levels of design: 1) regarding the tasks and 2) regarding the CT Chef game. During the data analysis we addressed the following research questions:

1. What meanings do students construct for the practice of abstraction in programming, when they collaboratively play, evaluate and modify the half-baked simulation game "CT Chef"?
2. How do students use the three affordances of ChoiCo (database, map interface and block-based programming) to explore and express these meanings?
3. What is the role of the half-baked and questionable elements of the "CT Chef" game in the emerging meaning-making processes?

Context and participants

The study was organized as an activity of the school mathematical club. The interested students filled in a form for participation that was signed by their parents. The duration of the study was 15 hours divided into 8 sessions. The study took place in the school computer laboratory with the presence of the researcher and two math teachers.

The participants were 7 students aged 13-14 years old from the second grade who were also members of the club. According to their schoolteacher, they had medium to good performance in programming subject and they had so far been taught basic programming concepts in Scratch environment, including variables. They had no experience on other abstract concepts like classes, functions and databases. Students worked in 3 groups of 2-3 students each, with one computer per group.

Task design

The tasks were organized into 3 parts according to the model of three-stages of progressive engagement with game modding 'Play-Fix-Create Mod', described by Kynigos & Grizioti (2020). For the first part the groups play the half-baked game 'CT Chef' and fill in a worksheet with open questions related to a) student opinions on the game e.g. 'write 3 things you don't like about the game', b) the game structure e.g. 'when does the game end?' and c) student gameplay strategies e.g. 'what did you do to achieve a high score?'. At the end of the first part, the researcher discusses the game with the whole aiming to raise discussion about the questionable elements.

For the second part, the groups are asked to improve the game by fixing the faulty behaviours they had detected in the first part. They can use the affordances of ChoiCo to make small modifications to the game structure that include adding new choices, modify the database and modify the ending and initial rules.

For the third part, they are asked to create a new version of the game. They can modify all game elements that they can access with the ChoiCo affordances but they have to keep the same gameplay ideas, that is a) to be a simulation of a real situation, b) the player to make choices with both positive and negative consequences, and c) the aim to be to remain in the game for as long as possible.

Data collection and analysis

During the study, we collected a set of data that included pre and post questionnaires from each student, interviews, approximately 20 hours of audio and screen capturing from all groups and for the total duration of the study, student games and worksheets. The interviews were semi-structured, artefact-based (Brennan & Resnick, 2012) with questions related to student modifications and uses of abstraction. The questionnaires were designed in google forms and they consisted of 12 computational thinking tasks designed by the researcher or adapted by existed computational thinking activities (Dagiene & Stupuriene, 2016). Before the study students and their parents gave written consent for staying extra hours after school, for audio and screen recordings and the interviews.

We did a qualitative analysis of the collected data in two stages: a) between every two sessions and b) a retrospective analysis at the end of the study (Bakker, 2018). As a unit of analysis, we used the critical incident (Tripp, 1993), that is a representative moment of student meaning-making process about computational abstraction as they are engaged with the game with the available affordances. For the identification and description of critical incidents, we correlated all data sources. After the first analysis, we had a set of approximately 200 critical incidents which were further analysed with a set of codes searching for patterns of critical incidents emergence that would inform us about possible patterns of student learning processes. The coding schema was reviewed and evaluated by an external researcher from the field of computer science education.

Findings

Game balance and meanings on abstraction

By the end of the study, each group had created 2 mods of the original game shown in table 1. As the activity unfolded students modified more game elements creating more sophisticated game versions. In the data analysis, we observed two types of student modifications concerning how they perceived the game system: micro and macro level modifications. With the term micro-level modification, we refer to a focused, element-specific adaptation or addition that doesn't take into consideration the game system (e.g. change an ending rule, modify the data in one database cell). Macro-level modification refers to a general modification or extension of a system's functionality or behaviour, that affects the related elements accordingly (e.g. create a new pattern for consequences, add a new game parameter). For macro-level modifications, students expressed powerful ideas of computational abstraction, including the ideas of variable, data structure and class.

Table 2: Game versions created by students

Group	Mod Title	Brief Description	Modified Elements
1	'The correct CT Chef'	Fixed version of the original game	Consequences values, New choices, end rules conditions
1	'CT Traveller'	You are a food critic. Travel in European cities and try different foods. Parameters: Joy, Health, Tiredness, Food_Experience, Money	Parameters, map interface graphics, new game areas, consequences patterns, new choices, end rules, gameplay rules
2	'CT Chef 2'	Fixed version of the original game	Consequences values, new choices, initial parameter values, end rules conditions
2	'CT Cine'	You manage your new cinema. Choose the movies you will show and the food you will serve aiming to keep the customers happy and make your cinema successful. Parameters: Money, Customers_per_Day, Cinema quality, Entertainment	Parameters, map interface graphics, consequences patterns, new choices, end rules, gameplay rules
3	'The best chef'	Fixed version of the original game	Consequences values, new choices, initial parameter values
3	'CT Life'	You just moved to a new city. Make new friends, earn money and live a balanced life. Parameters: Mood, Energy, Friends, Money	Parameters, map interface graphics, game areas consequences patterns, new choices, end rules, gameplay rules

In most cases, the trigger for the students to shift from micro to macro perception of the game algorithm was the problem of gameplay balance. The gameplay is formed by all game elements including the rules, the choices, the consequences, the interfaces and can be affected, slightly or highly, by micro-level modifications to them. Critical incident 1 is an example of how students shifted from specific to abstract thinking after several micro-level modifications had affected the gameplay balanced. In this episode, students from group 2 test their mod 'CT Cine' in which the player manages a cinema instead of a restaurant and sells movies instead of foods.

Critical Incident 1

S1: Hey! This movie costs very few money but gives a lot of customers and entertainment. Why did we do that? ***The game is very easy***
S2: And this one has only negative consequences. ***We must balance the game***
S1: We can think of ***some rules for the consequences*** *like the ones for the foods in the original game*
S2: Oh! Yeah! Like for example that ***the latest movies will cost more but will bring you customers while the old movies will do the opposite****.*
S1: Or good and bad movies. Let's write these rules and create some categories and then we will change the values in the database

At first, students made micro-level modifications in the database cells which resulted in unbalanced and easy gameplay. When students test the game and see the result of the micro modifications to the gameplay, they start to perceive and refer to the game as a system ('The game is very easy', 'We must balance the game'). Then, they express the idea of abstracting the choices from specific database instances to generic categories that follow some rules ('We can think of some rules for the consequences'). They describe the categories and the rules on paper and then they modify all the data records accordingly. Students here explore and construct meanings on the notion of classes and instances in programming, a concept difficult to be conceived by the students in a traditional programming context.

Later on, the same group generalized, even more, their solution as shown in critical incident 2. On that moment students had decided that the consequences of the parameter 'Entertainment' will not follow a rule but it will be based on their personal preferences.

Critical Incident 2

S1: Ok. It works for now. But ***if we want to have 100 or even more movies*** *we must every time check and decide if we like the movie in order to put the value of 'Entertainment'.*
S2: Yes that's right…Unless.. ***we create a relation between the game parameters****. Like Entertainment equals money/10 or something like that*
S1: Oh yeah! But let's think ***one relation that would make sense for the game****.*

Here we can see students shifting from a strict design strategy to an abstract one that they can use for x number of game choices ('100 or even more movies'). Using the database affordance, they invent a mathematical relation for connecting two game variables in a meaningful way for their game. The integrated affordances of the ChoiCo environment and the multiple representations of the game variables, namely the same variable is represented in the form of game attribute, code element and database element, seemed to foster student explore otherwise complex ideas related to the notion of programming variable.

Powerful ideas on the notion of programming variable

Variable is an abstract concept that novice programmers and young students have difficulties to understand and use efficiently in an algorithm (Samurcay, 1989, Robins et al 2003). Common misunderstandings include the initialization of a variable's value, the idea that a variable can have only one value each time and the process of changing the assigned value of a variable (Haberman & Muller, 2008). The analysis of our study showed that when students used the integrated affordances to create meaningful and balanced gameplay, they explored and expressed powerful ideas related to the concept of variable. Some common ideas were the notion of value variation, variable initialization, variable control and variable comparison.

In critical incident 3, group 1 students test their first modified game. They have changed the ending rule for the game parameter 'Money'(micro-level) but they haven't yet changed the code that initializes the game.

Critical Incident 3

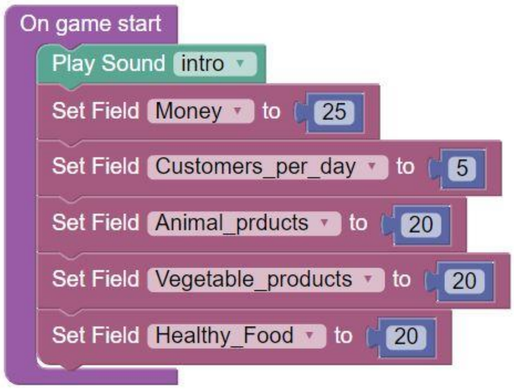
S1: Wait why do we have 25 Money? That's a lot of Money now that we changed the game over limit to 0.

S2: Oh! We forgot to change the ***initial value****.*

S1: How can we do this?

S2: ***I think there is a command that sets Money to a number*** *when the game starts. Look! It is here under the 'On game start'.*

S1: Oh! Now I get it. So, ***all these commands are executed only when the game starts****. Ok.* ***If the game ends when Money is below 0, then lets set the initial value of Money to 15, to be more difficult!***



In incident 3, students during playtesting realize the relation between the initial value of the variable 'Money' and the conditional that checks the variable's value and controls the game flow ('we changed the limit to 0'); two ideas that they had thought as separate game elements until this moment. Attempting to achieve meaningful gameplay, they express meanings on the process of value assignment (there is a command that sets Money to a number), of variable initialization ('We forgot to change the initial value', 'all these commands are executed only when the game starts') and of flow control ('If the game ends when Money is below 0, then let's set the initial value of Money to 15'). These concepts obtain meaning and value for the students in the context of their game ('That's a lot of Money', 'to be more difficult').

In critical incident 4, S2 uses the representation of consequences in the form of database records to explain to S1 (group1) how the value of variable 'Money' variates during the gameplay.

Critical Incident 4

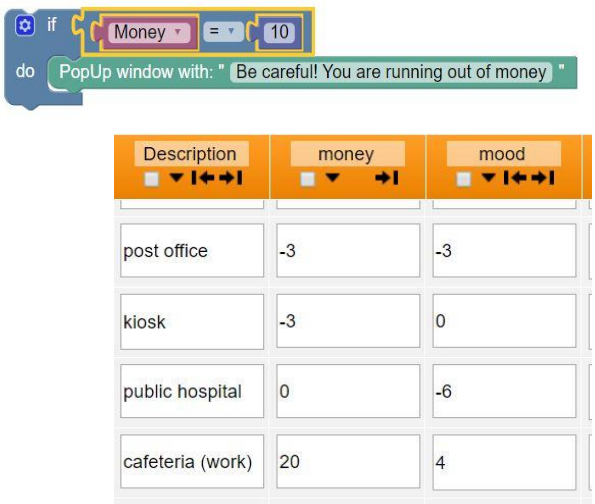
S1 programs a warning message to appear if Money = 10

S2: No not equal to 10. ***Make it less or equal to 10***

S1: Why?

S2: Because let's say you have 11 Money and you select a choice that reduces Money by -2 or -3 (opens the database)*. Then* ***Money will become 9 or 8*** *and the message will not appear.*

S1: Oh yes you are right! Money ***is not always decreased by 1****.*



In the case of incident 4, the database affordance allows students to visualize the variation of the variable ‘Money’ during the execution of the game algorithm. Students combine the representation of conditional statement in the code with the database affordance to develop understanding and to express ideas on the variable behaviour in their algorithm. During their discussion they express personal meanings on the idea of value assignment and variation (*Money will become, Money is not always decreased by 1*) and on the conditional statement (*less or equal to*).

Similar critical incidents were detected in all groups. Students usually expressed meanings on variables when implementing the following modifications: a) control the game flow b) balance the consequences and c) count the player’s choices. Even though at the beginning of the study, students used variables only in simple game rules, as the activity unfolded, they engaged with more sophisticated uses of variables, including mathematical comparisons, complex conditional statements and Boolean conditionals (Figure 4).

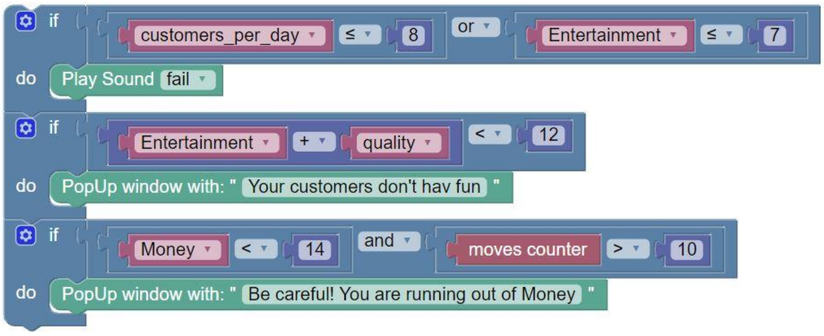


Figure 26: Variable uses in the game code

Discussion

In this paper, we discussed student meaning-making processes on programming abstraction when they collaboratively play and modify a half-baked simulation game. The presented study was small-scale, design-based research and the results cannot be generalized. However, they provide some first insights on how students with small programming experience, may explore, realize and communicate ideas on the concept of abstraction when they use programming in conjunction with other computational affordances as tools for self-expression.

The integration of logical bugs and controversial axioms into the game, motivated students to question its structure, analyse it, and modify it and to create their personalized version. The affordances of ChoiCo design tool enabled students to access and develop an understanding of otherwise complex concepts, such as variable variation and initialization, classes and conditional structures. For instance, they used the database representation of choices and consequences to describe the variation of game variables. They also used the data representation to think of and express abstract mathematical relations between the game variables, to balance the gameplay.

The issue of game balance, that is central in ChoiCo games, challenged students to shift from a specific and restricted perception to an abstract approach of the computational elements. During this transition from micro to macro level modifications, students expressed powerful ideas on forms of computational abstraction such as the concept of variables, classes and flow control.

It seems that game modding approach can provide a scaffold for students to explore concepts and practices with which are not usually engaged when they design a game from scratch. Through repeated cycles of modification, sharing and evaluation of the questionable game, students may progressively express powerful computational ideas on programming abstraction. In this context, the didactical integration of coding with other computational affordances can give students new means for expressing and communicating these ideas.

Acknowledgements

The presented research has been financially supported by the General Secretariat for Research and Technology (GSRT) and the Hellenic Foundation for Research and Innovation (HFRI) (PhD Scholarship Code: 531)

References

Armoni, M. (2013). On teaching abstraction in CS to novices. *Journal of Computers in Mathematics and Science Teaching*, *32*(3), 265-284.

Bakker A., (2018). *Design Research in Education*. Routledge, New York.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada*, 1–25.

Dagienė, V.& Stupurienė, G. (2016). Bebras - a Sustainable Community Building Model for the Concept Based Learning of Informatics and Computational Thinking. *Informatics in Education*, 15(*1*), 25–44. https://doi.org/10.15388/infedu.2016.02

Edelson D. C. (2002). Design Research: What We Learn When We Engage in Design. *Journal of the Learning Sciences*, 11*(1)*,105–121.

Gee, J. P. (2008). Learning and games*. The ecology of games: Connecting youth, games, and learning*, 3, 21-40.

Grover, S., & Pea, R. (2018). Computational Thinking: A competency whose time has come. *Computer Science Education: Perspectives on Teaching and Learning in School*, 19.

Grizioti, M. & Kynigos C. (2018) Constructionist Approaches to Computational Thinking The case of Game Modding. In *Proceedings of Constructionism conference 2018,* Vilnius. ISBN 978-609-95760-1-5.

Haberman, B., & Muller, O. (2008). Teaching abstraction to novices: Pattern-based and ADT-based problem-solving processes. In *Proceedings of the 38th Annual Frontiers in Education Conference*, https://doi.org/10.1109/FIE.2008.4720415

Kafai B. Y & Burke Q. (2014). *Connected code: Why children need to learn programming*. Mit Press.

Kramer, J. (2007). Is abstraction the key to computing?. *Communications of the ACM*, *50*(4), 36-42.

Kynigos, C. (2007). Half-baked Logo microworlds as boundary objects in integrated design, *Informatics in Education*, 6(*2*), 1–24.

Kynigos, C. (2015). *Constructionism: Theory of Learning or Theory of Design?* In Selected Regular Lectures from the 12th International Congress on Mathematical Education Sung Je Cho (Eds) 417- 438, Springer International Publishing Cham Heidelberg New York Dordrecht London, Switzerland 2015.

Kynigos, C., & Yiannoutsou, N. (2018). Children challenging the design of half-baked games: Expressing values through the process of game modding. *International Journal of Child-Computer Interaction*, 17, 16-27.

Kynigos, C., & Grizioti, M. (2020). Modifying games with ChoiCo: Integrated affordances and engineered bugs for computational thinking. *British Journal of Educational Technology*.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Robins, A., Rountree, J., & Rountree, N. (2003.). Learning and Teaching Programming: A Review and Discussion*. Computer Science Education,* 13*(2),* 137-172.

Samurcay, R. (1989). The concept of variable in programming: Its meaning and use in problem-solving by novice programmers. *Studying the novice programmer*, 9, 161-178.

Tripp, D. (2011). *Critical incidents in teaching (classic edition): Developing professional judgement*. Routledge Education, New York.

Weintrop, D., & Wilensky, U. (2014). Situating Programming Abstractions in a Constructionist Video Game. *Informatics in Education*, 13(*2*), 307–321. https://doi.org/10.15388/infedu.2014.10

Wilensky, U., & Resnick, M. (1999). Thinking in levels: A dynamic systems approach to making sense of the world, *Journal of Science Education and Technology*, 8(*1*), 3-19.

Wing J. M. (2008.) Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences,* 366 (*1881)*, 3717–372

Task Development for Discovery Learning of Informatics Concepts

Philipp Prinzinger, *philipp.prinzinger@tuwien.ac.at*
Institute Information Systems Engineering, TU Wien, Vienna, Austria

Gerald Futschek, *gerald.futschek@tuwien.ac.at*
Institute Information Systems Engineering, TU Wien, Vienna, Austria

Abstract

We describe our own learning process while developing tasks and scaffolding materials for discovery learning of informatics concepts. At our university we perform nearly every day a workshop with school classes of secondary schools. The main aim of the workshops is to involve the school children in interesting informatics activities that make fun and keep in mind for a longer time. We decided to perform mostly activities without use of computer to give more emphasis on thinking about informatics concepts than on information technologies. We quickly found out that a constructive way of learning that allows detection and re-invention of informatics concepts is much better appreciated by the students than presentation and explanation of selected informatics concepts. We present an exemplary development of a task and its scaffolding materials that support the learners in their discovery process.

We used a prototyping approach with very short development cycles. The first year of workshops performance shows not only essential changes in the offered tasks but also a rapid evolution of the general workshop format. All changes aim to better fit the ideal goal to achieve active, discovery learning of all individual students with use of constructive and tangible materials. The evolution of the error detecting activity is described in detail.

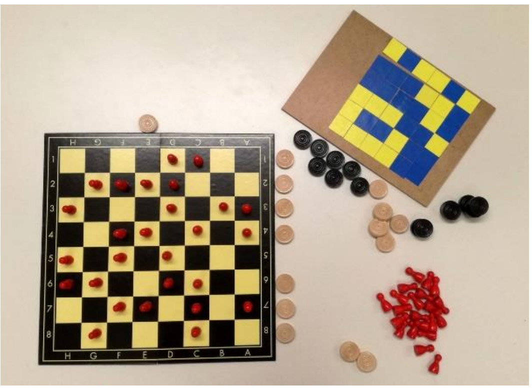


Figure 1. Different versions of learning material for the “error detecting” task

The learning materials changed while we learned how to pose the problem in such a way that the kids can better explore situations so that they find themselves solutions to the problem.

Keywords

Task development, discovery learning, informatics concepts, scaffolding, secondary schools

Introduction

In January 2019 we started at our university an outreach program for school classes in the field of informatics. We offer workshops for students from grade 5 to 12 (age 9 to 18). The goal was to involve students in informatics-related activities to show the beauty und fun of solving problems informatically. It is not only the goal to attract students to study informatics but mostly to propagate the idea that informatics is a creative science that allows problem solving that makes fun and can be applied to other topic and disciplines. The learning situation is different to a regular learning unit at school. The students come accompanied by their teacher to a place that is new for them. There an university teacher and some tutors await them and give them some tasks to learn more about informatics. A typical workshop lasts 90 Minutes and there is no follow up planned so far.

Our workshops are structured in 4 parts: Part one is the introduction, in which we welcome the classes and give them a short overview of the topics and the workshop itself. This takes about 10 minutes. In part two we start with our icebreaker activity that last 20 minutes on average. This is a simple common activity guided by the workshop leader, in which all participants are involved. Then we move on to the third part, the main activities in small groups. Here the students can solve different tasks within 45 minutes. In the last 15 minutes of the workshop we summarize the key messages in the fourth and last part.

So this learning intervention is extremely short compared to regular courses. There is also no grading of students involved. Students may explore given situations and discover properties, rules and even algorithms. There are no correct and false solutions. Important is a personal engagement and fun if something useful is detected. Since the time of the workshops is relatively short the situations and activities must be carefully planned and proper supporting material must be prepared. As we focus on informatics concepts learning and due to the short workshop time we provide mainly unplugged activities. So we have not to spend time for technology related issues.

When we started with our workshop programme in January 2019 it was not clear how to properly state the problems to the students to reach our goals. We had at that time just installed the exhibition “Abenteuer Informatik” (adventure informatics) (Gallenbacher 2012) in the newly restaurated aula of the informatics building. So we were tempted to use this exhibition for our goal. The exhibition comprises 44 large boards covering 11 informatics themes: Maximum Flow Problem (see figure 2), Travelling Salesman Problem, Binary Numbers, Ethiopian Multiplication, Codes, Sorting and Searching, Parity Bits, Limit of Computation, Decision Problem, Binary Search, Sorting Networks. The exhibition boards offer interactive tasks, but also solutions and a lot of informative text to the corresponding informatics themes.

Figure 2. Maximum flow problem: A tutor explains a hands-on task on a board of the exhibition

From constructionist learning we know that we should pose interesting problems and provide the students with a proper environment in which they can construct and try their own solutions. Scaffolding is allowed but offering the solution is not.

Related work

Webb and Rosson (2013) describe experiences with an outreach program where computing activities for school girls involve discovery learning.

In the last decade emerged many unplugged activities for learning informatics concepts, most well-known is "CS Unplugged" (Bell, T. et al., 2009) from where we took also the idea of the "error detecting" activity, its further development to discovery learning is explained later in this article. Bell & Lodi (2019) argue that these unplugged activities are constructive and involve in this sense constructionist elements.

In their article Weigend et al. (2019) classify creative unplugged activities in 4 categories: find an algorithm, find an application, find an example and find a visualization. Also Futschek & Moschitz (2010) argue that inventing and playing algorithms can be done with groups of school students where all of them are involved in a creative learning activity. In our workshops we also try to apply creative learning and in especially inventing algorithms.

Workshops for school classes

Since the "Abenteuer Informatik" exhibition offers opportunities to hands-on experiments that deepen an informatics concept for each of the themes, we started to use the exhibition boards for our workshops. We call our first approach the "exhibition workshops", since we used the exhibition boards to work with small groups of students.

In a short first phase we guided school classes, divided into small groups, through the exhibition. Our plan was to enable the students to work out the presented informatics concepts individually or in groups using the didactically high-quality exhibits. Apart from the above-mentioned challenges, there was also the problem that the exhibits did not allow any alternative variations to the intended concept. If the students did not develop exactly the expected results, they had to be guided to it, which did not correspond to our goal of a more individual and more independent learning.

The exhibition workshops

A typical school class of 24 students was divided in 4 groups á 6 students. Each group should explore 4 themes of the exhibition and was assisted by a tutor who explained the tasks related to the exhibition boards that should be solved using the interaction possibilities explained on the boards. But unfortunately the number of students directly interacting at an exhibition board is due to limited space 1 or 2 students (see Figure 1). So the other students are just observing instead of active interacting. To involve all students the tutors were tempted to explain the informatics concepts in detail instead letting the students discover the main ideas of the concepts themselves. This led to a more teacher-centered learning, where too many technical terms were used by the tutors, which led to a mental overflow of the students and a loss of attention. The different experimentation times of different student groups working on different exhibition themes confronted us also with another problem: additional waiting time when moving from one theme to another.

Figure 3. The “error detecting” problem presented as a magic trick on an exhibition board. A group of students is analysing and testing the concept by arranging and flipping the magnetic cards

To overcome this additional problem, more topics than groups were included in the workshops and the interaction time per participant was limited. Although these modifications were able to reduce the problems described above, it was still not in line with our ideal of constructionist learning. From the very beginning we tried to design the workshops in such a way that as many students as possible could become active and participate. Our vision is to give all students the opportunity to develop their own solutions and concepts. This ambitious goal could not be achieved for all students when working in this way.

Exploratory Workshops

To involve all students of a group in exploring a concept at the same time, we created additional tangible materials which were related to the themes of the exhibition. With these materials the students could work largely independently or with brief explanations of the tutors. The materials became tools and not, like the exhibits, a source of information. In this way we performed a change from guiding groups of students to a phase where we allowed each of the students individual analysis of each concept that led to a deeper understanding of this concept.

But there was still a major issue: the tasks and materials led to only one informatically perfect solution. We identified two problems for achieving our goals: little freedom for creative solutions and a still rather reproductive learning. By aiming at a certain concept, there was little or no room for own solutions and concepts. If the one solution was not found, the tutors had to present it. This led to purely reproductive learning and possible frustration. Although we were able to enable more students to explore the solution and research independently, there were still some limitations. The students may re-construct in their brain the concept but they do not construct a concept themselves.

Constructional Workshops

Therefore we moved to a third phase of our workshop format, where the students could develop the main ideas of informatics concepts themselves. The specific evolution of the workshop formats and their three stages of development are described in detail in the following section using as example the “error detecting” task, that involves the informatics concepts error detection and error correction using parity bits.

Development of scaffolding materials for discovery learning

The development of the error detection and error correction task took its origin in the activity "Parity magic" from CS Unplugged, which is similarly included at one of the boards of the “Abenteuer informatik” exhibition (figure 3). In its original form presented in the exhibition, the activity consists of the demonstration of a magic mind reading trick presented by a tutor who plays the magician. The students are given 25 magnetic cards showing a black cross on a yellow background on one side and a white circle on a blue background on the other side. The students have the task of creating a graphical information, represented by a grid of 5 by 5 cards. Through adding one row and one column of cards by the magician under the cover of making the information more complicated, this grid is extended to 6 by 6 (see figure 3). Now it's the students' turn again: they have the options either to flip any card and thus change the original information, or to do nothing and leave the information untouched. While the students may manipulate the information, the magician turns away with closed eyes and then claims to find out with the power of his thoughts, if and which tile was flipped. Because the magician has added the cards according to a special but secret system, he is always able to determine whether a card has been flipped, and he even can tell exactly which card has been flipped.

After the demonstration the secret was revealed: the 11 additional cards were added in a way, error detection and correction is made in computers. Also the term "parity bits" was introduced. Afterwards it was the students' turn to try out the presented concept for themselves.

We redesigned the task for our workshops several times: In the first variation the students were given a random pattern of the two sided magnetic cards on a 5x5 grid. They were asked to design a concept, how they could place the remaining 11 parts, in order to recognize if a card was flipped and ideally also which card was flipped.

The students didn't see this as a big challenge, as most of them were able to remember the given pattern on the rather small field. This reduced the motivation to consider an appropriate error detection or correction concept. Apart from the low motivation, we had the problem that almost none of the students came up with the “right” concept on their own. So it was always necessary to supervise the students by a tutor who led them to the solution.

In order to increase the motivation of the students and provide them more freedom in exploring, discovering and developing the concept of parity bits, we made another variation. We created “message boards” (as shown in figure 4) with yellow and blue bits on a 5x5 grid. But now one of the bits was not fixed. The loose bit was yellow on one side and blue on the other. The students were asked to put this loose bit with the right colour into the grid. They got the hint: The L-shaped grid can help them to solve the task. This situation should enable the students to analyze the pattern and to recognize that the L-shape creates an even number of blue and yellow parts in all rows and columns. Subsequently, the recognized system could be used to create parity bits for other square patterns of different sizes.

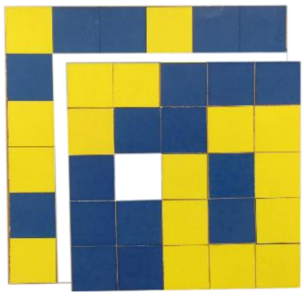


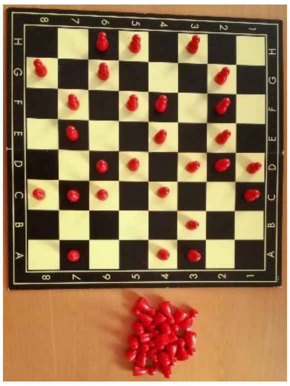
Figure 4. Message boards with added parity bits (L-shape) and one missing bit. The task is to find out the systematic behind the coloring of the bits of the message board. And finally also the color of the missing bit is asked. (The parity bits are such that the number of blue and yellow bits is even in every row and every column. So the missing bit has to be yellow.)

At this stage of development, the students were able to do some research and discovery, but could not discover a system for error detection by themselves. So we had not yet arrived at a fully discovery learning task. The goal was to find materials to engage the students to develop their own solutions and strategies to notice, if anything in the original information (pattern) will be changed. In the sense of Tissenbaum, Sheldon and Abelson (2019) these kind of activities do not have such a long-term impact compared to concrete implementations, but we are convinced that developing, testing and presenting one’s own creative solution has a great impact on motivation and learning behaviour.

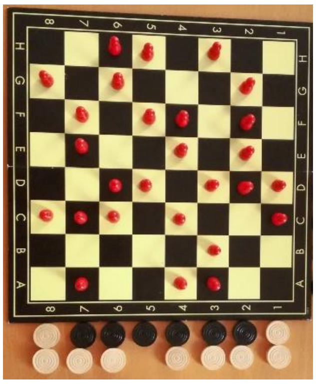
We reached the final step towards discovery learning of the concept with help of a chessboard, some game figures and checkers.

A chessboard as scaffolding material

Subtask 1: students are asked to place some game figures randomly on a chessboard (see figure 5a). Subtask 2: the students should develop a concept for placing the checkers (next to or even on the chessboard) to be able to detect if a single game figure will be added or removed or if nothing has been changed. Subtask 3: they should use the placement of the checkers to determine which game figure has been added or removed. Subtask 4: a placement strategy should be developed using a maximum of 16 checkers.

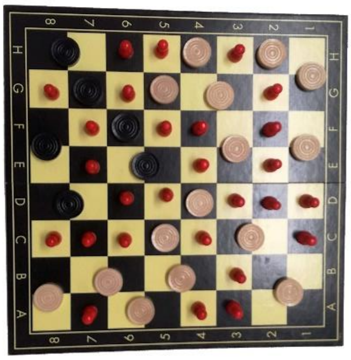


a. Subtask 1

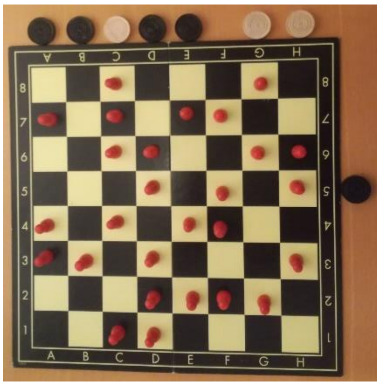


b. Subtask 2

c. Subtask 2 & 3



d. Subtask 2 & 3



e. Subtask 4

Figure 5 a-e. Students' solutions of the subtasks 1-4 (before any changes were made)

The scaffolding: The task described above introduces students to the intended concept of error detection, in particular the subtasks 2, 3, 4 for determining whether something has been changed or not. The first subtask serves to create an initial situation, that could be interpreted as a 2-dimensional code of a message. If a single error occurs during transmission of this message, this error should be detected with help of the checkers. Students' ideas, concepts and suggested solutions are discussed with the tutors after each subtask, e.g. by the tutor pointing out possible advantages and disadvantages. The tutor also explains the connection between the overall task and the computer science concept of error detection and error correction, including the representation of a message by the arrangement of the chessboard. In the second subtask only a single error (adding or removing of a single token) should be detected. It is not yet relevant to localize the error or to use as few checkers as possible.

Figure 5b shows a students' solution to identify whether a game figure will be removed or added before the tutor has made any changes. The checkers are used to count the number of game figures in each column. To reduce the number of checkers needed, the students used the black checkers for three game figures and the white ones for one.

In the third subtask the position of the error should be discovered too, the number of checkers is still limited just by the provided number of checkers.

The solutions in 5c and 5d have a similar approach for error detection and error correction: marking fields. While in 5c all occupied fields are marked with a checkers underneath the game figure, the method in 5d marks all unoccupied fields. The students who designed the solution 5d had too little checkers to mark all fields, so they positioned the checkers on common edges or corners of neighboring fields to mark more fields with just one checker.

Through the markings an error (adding or removing a game figure) can be detected and corrected easily. If a game figure is removed, the error is obvious: either an empty square remains without a marker or only the marker remains. Adding a game figure is also immediately obvious: either it is on an already marked square after being added or the marking below it is missing. A decisive difference between the two methods lies in the informatics feasibility: for the method shown in figure 5c some kind of overhead could be used to store, that this memory unit contains data; although the basic principle of the method in figure 5d is basically feasible, its presented design cannot be implemented informally.

Both methods have in common, that they won't work in reality. Although the method in figure 5d can detect some errors in practice, since all data (game figures and checkers) would be lost in the case of a faulty memory block or message block, error correction becomes impossible in both cases.

In the fourth subtask the number of checkers used for any placement of the game figures should be as low as possible. Students are informed that a maximum of 16 checkers can be used, for any number of figures and for any placement on the chessboard, if only one game figure is removed or added.

The solution given in figure 5e is a representation of the parity bits. Through the placement of the checkers outside the chessboard, one additional line and one additional column is added. In every row and column with an uneven number of game figures a checkers is placed next to the chessboard. In the extended grid every row and every column now has an even number of pieces. If the game figure on the field D5 would be removed, the simulated error can be determined and corrected easily. Row D and column 5 both contain an uneven number of pieces, ergo there must have been a game figure on the square D5 before any changes were made.

Through this kind of scaffolding the students can develop their own concepts, test them,discuss them within the group and with the tutor and refine them step by step if needed.

Discussion of findings

In each phase of the development of our workshops we designed them with close attention to our goals under consideration of the given circumstances. During the performances of the workshops, we often found that the workshop design did not allow us to fully achieve all of our goals. Through constant observation and reflection we were able to identify concrete problems and make adjustments.

To make our findings comprehensible, we first describe our goals and then the actions, observations and adjustments of the respective development stages.

Although the workshops have changed over time, our goals have remained the same throughout: to inspire students for informatics, to show the wide variety of informatics (informatics is more than programming), to spread the joy of informatics, to offer a high degree of interaction, to provide impulses for informatics teaching.

In order to achieve these goals we used at the beginning the interactions of the exhibition boards in the exhibition workshops and a wide range of different topics of the exhibition, which were presented by the tutors with enthusiasm and excitement.

With this tutor-centered approach we could observe that the limitation of the interaction possibilities to one or two persons led to disinterest of the uninvolved persons. Since the tutors presented the contents and transferred knowledge, it was essential that each group was supervised by a tutor. This meant that we had a high personnel effort and hardly any possibilities to shorten the waiting time for the students. Due to the constant attendance and presentation by the tutors, the motivation to find the solution for a given task was rather low. The students often preferred to ask the tutor if they could not solve a problem quickly. When the solution was explained, the secret was revealed and the desire to try it out for themselves was rather low.

In order to address the observed problems of passivity and tutor-centeredness, we developed additional materials based on the exhibition and our own tasks, which can be worked on more independently by the students.

The discovery workshops covered fewer topics, each with several tasks. The information from the exhibition could be used by the pupils to work on the tasks. This meant that the tutors no longer had to do the entire knowledge transfer.

With these changes, we could see that the students dealt with the tasks more intensively. Due to the additional materials, several students could work on one task at the same time. In the small groups, discussions on the topic of the task were held and more joint work on a solution was carried out. For some tasks we had to find out that the pupils only worked on them superficially or not at all. The students said that the tasks were too difficult for them or that they had not yet learned this particular topic.

By varying the tasks we found out that the complexity was one reason, but also aiming at a certain solution increased the difficulty of a task.

After we found out that the additional materials had a positive influence on the students' motivation, we created further tasks with our own materials, which went beyond the scope of the exhibition. With these changes we arrived at the current stage, the constructional workshops.

The new open-ended puzzle-like tasks offered a lot of creative freedom. The students could carry out and try out their own ideas. They spent more time on the tasks and showed a deeper interest in the topic. At these student-centered workshops we observed a lot of fun and instead of a variety of topics we had a variety of solutions.

References

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, *13*(1), 20-29.

Bell, T., & Lodi, M. (2019). Constructing Computational Thinking Without Using Computers. *Constructivist Foundations*, *14*(3), 342-351.

Futschek, G., & Moschitz, J. (2010). Developing algorithmic thinking by inventing and playing algorithms. *Proceedings of the 2010 Constructionist Approaches to Creative Learning, Thinking and Education: Lessons for the 21st Century (Constructionism 2010)*, 1-10.

Gallenbacher, J. (2012). Abenteuer informatik: hands-on exhibits for learning about computational thinking. In *Proceedings of the 7th Workshop in Primary and Secondary Computing Education* (pp. 149-150). ACM.

Tissenbaum, M., Sheldon, J., & Abelson, H. (2019). From Computational Thinking to Computational Action. *Communications of the ACM,* Vol. 62, No. 3, 34–36.

Webb, H., & Rosson, M. B. (2013, March). Using scaffolded examples to teach computational thinking concepts. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 95-100). ACM.

Weigend, M., Vanicek, J., Pluhar, Z., & Pesek, I. (2019). Computational Thinking Education through Creative Unplugged Activities. *Olympiads in Informatics*, Vol. 13, 171–192.

The constructionist nature of an instructor's instrumental orchestration of programming for mathematics, at university level

Ana Isabel Sacristán, *asacrist@cinvestav.mx*
Department of Mathematics Education, Cinvestav, Mexico

Marisol Santacruz-Rodríguez, *marisol.santacruz@correounivalle.edu.co*
Institute of Education and Pedagogy, Universidad del Valle, Colombia

Chantal Buteau, *cbuteau@brocku.ca*
Department of Mathematics and Statistics, Brock University, Canada

Joyce Mgombelo *jmgombelo@brocku.ca*
Faculty of Education, Brock University, Canada

Eric Muller, *emuller@brocku.ca*
Department of Mathematics and Statistics, Brock University, Canada

Abstract

In this paper we describe and analyse the orchestration of an instructor in a Canadian university program, called *Mathematics Integrated with Computers and Applications* (MICA), where students program computer microworlds for mathematical investigations. More specifically, we analyse the particular constructionist elements of the instructor's instrumental orchestration that aim to support students use of computer programming as an instrument for thinking mathematically. We analyse how this instructor conducted his classes in terms of the three components of instrumental orchestration (see Figure 1) and illustrate, briefly, how it is reflected in the activity of one of his students.

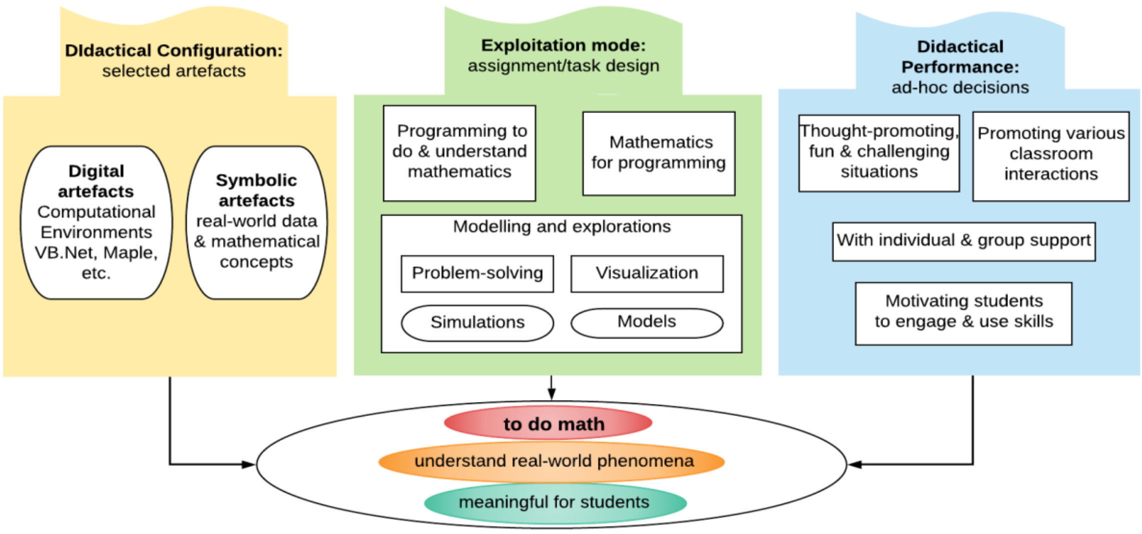


Figure 1. Selected components of a MICA instructor instrumental orchestration related to Constructionism

Keywords

Computer programming; mathematics; university; instrumental orchestration; constructionism.

Introduction

In the past 2018 constructionism conference (Buteau et al., 2018), we discussed the roles and demands on instructors to create constructionist environments in university computer programming for mathematics courses –specifically, the *Mathematics Integrated with Computers and Applications* (MICA) courses implemented, since 2001, at Brock University, Canada (Buteau et al., 2015b). In those courses, math majors and future mathematics teachers learn to design, program, and use interactive environments –called *Exploratory Objects* (EOs)– for the investigation of mathematical concepts, conjectures, and theorems or real-world situations (Buteau et al., 2015a). As explained in the latter paper, those MICA courses fit the constructionist paradigm (Papert & Harel, 1991), by requiring students to design and program the EO computational objects for mathematical investigation and learning.

Since our 2018 paper, we have been analysing the work of MICA instructors in their courses and how they steer their students' investigations and learning, as part of an on-going five-year naturalistic (i.e., not design-based) research project. In that project, we study students' activity and their instructors' work using as framework the instrumental approach (Rabardel, 1995; Guin et al., 2005) and the instrumental orchestration theory (Drijvers et al., 2010), with the following research questions:

- How do students appropriate programming as an instrument for mathematics investigations?
- How do instructors create and orchestrate a learning environment that supports students' instrumental geneses development?

In this paper we focus on the latter through a case study. We look specifically into a MICA instructor's practice in terms of: what artefacts he integrates into his courses, and why; what are the considerations and aims underlying the design of his assignments; what does he expect his students to learn, and in what ways does he support his students. We attempt, specifically, to identify particular constructionist elements in this instructor's instrumental orchestration.

Background: The constructionist design of MICA courses

MICA is a sequence of three semi-annual courses (MICA I-II-III). In Buteau et al. (2018) we described how MICA undergraduate students have to create (i.e., design and code) Exploratory Objects (EOs), including, as final course projects, original EOs for which they select the topic – created individually or in groups of 2-3 students. The EOs can be thought of as mathematical microworlds, which Mavrikis et al., 2008 (cited in Buteau et al., 2015a, p. 138) define as exploratory learning environments that "allow students to explore not only the structure of accessible objects in the environment, but also construct their own objects and explore the mathematical relationships between and within the objects, as well as the representations that make them accessible". That is, EOs are "digital environments to explore mathematics concepts and their relationships represented in diverse forms (in the code and in the interface)" (Buteau et al., 2015a, p. 144).

In MICA I, students learn to create and use mathematical EO microworlds; while in MICA II-III courses, EOs serve to investigate more sophisticated mathematics, with MICA II focusing on applications of mathematics using technology, including simulations and modelling. As can be seen in the MICA guidelines URL (n.d.), the central intent is for students to learn mathematics through creating EOs: Buteau et al. (2015a) connect to constructionism principles (see Appendix), the dominant elements of the MICA courses: 1. The EO mathematics projects require engagement with programming where students learn to do mathematics (learning by making) conducting their own mathematical explorations. 2. Students are empowered when they develop their own strategies and make their own choices for finding solutions to a diverse range of mathematical problems. 3. The course fosters and values students' creativity in mathematical work. And, as explained in Buteau et al. (2018), MICA instructors design and implement a constructionist environment in order to support the intended students' experiences.

Aim of the paper and theoretical elements

In this paper we explore, more in-depth, the constructionist nature of that design, through a case study in which we analyse how a MICA II instructor creates (orchestrates) a constructionist environment. For that, we carried out a literature review to identify some main constructionist principles and ideas from Papert's writings and from the broader constructionist literature.

Main constructionist principles

We organized the identified constructionist principles into four themes, and put them in a table (see Appendix):

- (1) Epistemology, and conceptions of mathematical knowledge and of mathematics;
- (2) Conception of learning and of the role of the student;
- (3) Pedagogy and design; and
- (4) Computer programming and microworlds.

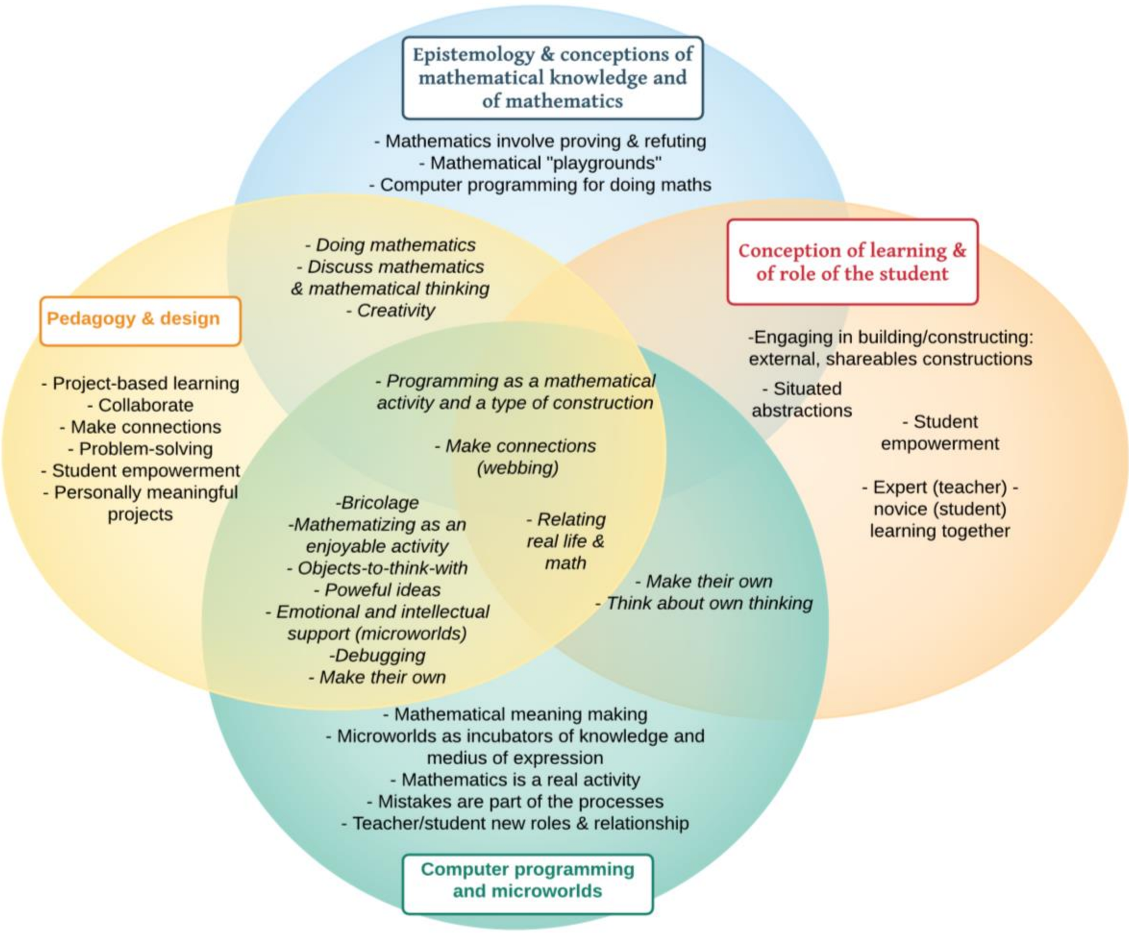


Figure 2. Some main constructionist principles by themes, as presented in the Appendix.

In Figure 2, we present a diagram that represents the main relationships between the constructionist principles that we identified. These principles served as guiding elements for our analysis, for which we used concepts from the instrumental approach (Rabardel, 1995), specifically those of instrumental orchestration (Drijvers et al., 2010).

The instrumental approach and the notion of instrumental orchestration

The instrumental approach, proposed by Rabardel (1995), focuses on how human activity with technology mediates the construction of knowledge: i.e., how people, through a process called instrumental genesis, appropriate an artefact (not necessarily physical), through its use, and turn it into an instrument –a psychological construct developed by a subject over time through the use of the artefact– (Drijvers et al, 2010).

Trouche (2004) considers that students' instrumental geneses may need to be steered by a teacher. He thus proposed the concept of instrumental orchestration to refer to the teacher's intentional and systematic organization and didactic use of various artefacts in the class (including digital ones), that steer students' instrumental geneses. He explains that, through the teacher's orchestration, not only individual aspects are mobilized in the class, but also aspects of a social and collective nature that influence student learning. As an extension to this theory of instrumental orchestration, Drijvers et al. (2010) introduce the idea of didactical performance to explain the different modifications, adjustments and changes to the didactic configurations, which are made in response to the events of the class. Drijvers et al. (2010; p. 215) consider three instrumental orchestration components:

- (i) *Didactical configuration* – "an arrangement of artefacts in the environment, or, in other words, a configuration of the teaching setting and the artefacts involved in it";
- (ii) *Exploitation mode* – "the way the teacher decides to exploit a didactical configuration for the benefit of his or her didactical intentions [it] includes decisions on the way a task is introduced and worked through, on the possible roles of the artefacts to be played";
- (iii) *Didactical performance* – which "involves the ad hoc decisions taken while teaching on how to actually perform in the chosen didactic configuration and exploitation mode."

Using the Instrumental Approach for analysing constructionist activities

Relating the instrumental approach (including the idea of instrumental orchestration) with constructionism is not new. Kynigos and Psycharis (2013) drew connections between constructionism and instrumental theory, highlighting the process of instrumentalisation for the generation of mathematical meanings. Previously, Hoyles and Noss (2004) noted that work with computational tools and the development of learning communities around their use, pointed the way "towards a consideration of the complex process of instrumental genesis, the role of the teacher, and the connection of tool use and traditional techniques" (p. 214). Also Hoyles, Noss & Kent (2004) drew links between the notion of situated abstraction, with those of instrumental genesis and orchestration, elaborating the ways in which technological artefacts can provide means of mathematical expression; they discussed how the process of orchestration at a first level could foster the growth of situated abstractions, by establishing a ''cognitive scaffolding'' for a second level of orchestration taking place over an extended period through a combination of collective activity in the classroom and individual work by students. Later, Trouche and Drijvers (2014) showed how the notion of "webbing" proposed by Noss and Hoyles (1996) –the process of constructing and using connecting structures, such as those provided by digital tools– and the notion of orchestration, are interrelated views on how learners develop a relationship with mathematics through technology, and the role of the teacher to support, through orchestration, such a development.

Analysis of the constructionist nature of an instructor's instrumental orchestration

Data and methodology

In the case study presented in this paper, we analyse the instrumental orchestration of Bill, a MICA II course instructor. As data, we had: (i) From the instructor: course design; EO assignment guidelines; EOs grading rubric; instructor interviews (6 in total) after each assignment (coded A1I to A4I), and after the final projects at the end of the course in two parts (FI1-F12). And (ii) from the students: lab session reflections (LR1-LR9), the completed EOs and accompanying reports (R1-

R5), and individual task-based interviews after each EO (A1I-A5I).

As said above, the analysis of the instructor's orchestration was guided by the main principles of constructionism, which were summarized as keywords (preset codes), as presented in Figure 2 and in the Appendix. Those helped us identify the constructionist elements (marked in bold in the analysis in the following section) in Bill's course orchestration, organised according to Drijvers et al.'s (2010) instrumental orchestration components: didactical configuration, exploitation mode and didactical performance. We also did a preliminary analysis of the effect of Bill's orchestration on one of his students (Kassie –pseudonym); her comments illustrate some ways in which students' responded to his orchestration.

In Figure 1, we present a schematic representation of selected aspects of Bill's instrumental orchestration components, with his main goals, showing some constructionist principles (see Appendix) involved, and the relationships between the elements.

Bill's MICA II course: structure, conception, orchestration and constructionist elements

MICA II's stated objective is for students to learn basic methods of **mathematical modelling** and of "experimental mathematics", with emphasis on computational and algorithmic methods, and computer simulations (Buteau et al., 2015a). We assume that Bill adhered to this course aim and that it guided his orchestration; this course aim can be re-phrased as: to learn **programming for modelling and investigating mathematical problems in real-world contexts**.

Bill's semester-long MICA II course consisted of:

- Four individual programming-based EO mathematical investigation assignments
- A final original project where students, working in pairs (see MICA guidelines URL, n.d.), were given three options to choose from, for the type of object they had to program:
  (i) "an **investigation** of a mathematical problem" (an EO);
  (ii) "a **learning object** (LO) designed to teach/test a mathematical concept";
  (iii) "a 'real world' **application** of mathematics".

Students receive guidelines for each assignment, including, in written form: guidelines with several steps or parts (see as example, the description of Assignment 4, below). Bill's assignments **focus on programming activities, where students investigate mathematical ideas, through simulations, graphics, modelling**. For the programming EOs to be produced, he asks that all the "code should be carefully structured and very easy to read with all variables, functions and subroutines labelled in a helpful way"; with a user-friendly and attractive interface. The proposed topics come from contexts he considers interesting and relevant for his students, but that require a computer to investigate properly: he wants his students to find the **projects meaningful** so that they become engaged in it. The assignments use content resources –"**objects-to-think-with**" (Papert, 1980a)– from daily life, which provide the mathematical programming activities and explorations with **realistic** significance, and may **promote engagement and excitement** in his students. How the task design and context were conceived and presented, is part of the *exploitation mode* of Bill's orchestration.

- Assignment 1 dealt with probability investigations: Students investigated, through programming EOs, four situations, including the Buffon Needle problem where they explored the probability that a needle dropped onto a plane of lines, touches one of the lines (for this problem, students were given a start-up piece of code in VB.net that they could modify and build upon).
- Assignment 2 presented three discrete deterministic dynamics situations to be modelled for investigating and predicting: two population-related phenomena (the spread of a disease; how parents socioeconomic status affects income-earning); and daily return percentages, using real stock exchange Excel data.
- Assignment 3 focused on producing an EO for exploring the chaotic system of the bifurcation diagram for the logistic map (a discrete dynamical system) –Figure 3a– through its bifurcation and fixed points (found through calculations in Maple).

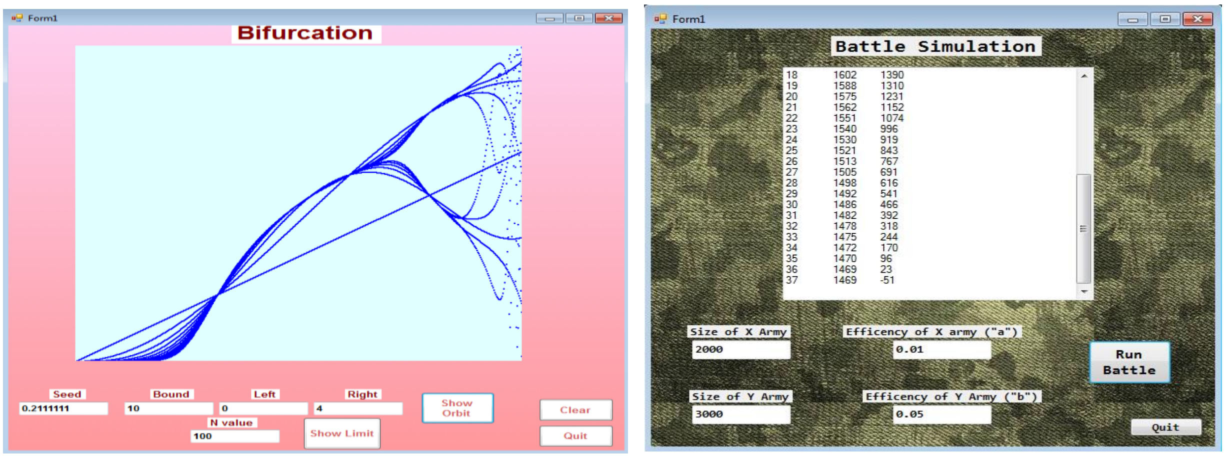


Figure 3. Screen captures of some of Kassie's programs (a) L: Logistics map; (b) R: Battle simulation.

- Assignment 4 focused on the discrete Lanchester equations that model battles (which deal with differential equations and with a modification involving stochastic processes).

Bill and several of his students said that this Assignment 4 was the one that they engaged in, and enjoyed the most. Bill explained some reasons of why he proposed it:

> *I went into the literature of warfare and discovered that at the heart of a lot of military planning are several equations called the Lanchester equations; and it turns out that the world is still using these equations […]; so, this is very current!* (Bill; A4I)

He also explained that there were many mathematical (and scientific) ideas and activities involved in this assignment, such as drawing trajectories, finding probabilities, integrals, differential equations, hyperbolas, and energy conservation, etc. Assignment 4's guidelines outlined the Lanchester equations, provided assumptions, and asked students to create an EO for a "Battle Simulation" (Figure 3b) in order to determine how long a battle would last, depending on the initial size of the participating armies. The program would also show the size of each army on each day of the battle. The guidelines take students through several steps of writing programs to investigate the effects of changing different inputs and aspects; they also ask students several questions, such as making estimations and conclusions on the basis of the explorations using the programs. Furthermore, the guidelines provide a piece of computer code to help students in writing one of the programs.

Bill's *didactical configuration* consists of the artefacts (material or symbolic) that he provides his students with, and their organisation. The "symbolic artefacts" are the topics, problems and mathematical contents that he chose for his assignments, including the data provided and other problem variables. He looks for symbolic artefacts that are engaging: **"**I'm thinking 'what's fun?', 'what can I do?', 'what can I play with?'". He provides **interesting** elements to work with (e.g., **real-world data**, such as from the stock; certain mathematical elements, such as the Lanchester equations, etc.) –i.e., **objects-to-think-with–**, that he himself enjoys (he says: "I teach a topic that I'm interested in, and doing something I care about") and that can lead to **engaging explorations**.

The material artefacts include the computational tools to use in the assignments (mainly VB.Net, but also Excel and Maple). From Bill's interviews, we inferred that he took into account two aspects, in order to decide the selection of the artefacts:

- He considers them effective tools (instruments) for ***doing* mathematics**.
- He considers it important that his students learn to use tools that are normally used in the workplace and in **"real-life".**

For example, with regards to Excel, he explains:

> *I think, how could a math major not be able to [download the past ten years of the stock, and chart its behaviour] in Excel. I also think […] that every math major has to be able to use*

Excel, because this is one of the standard tools in the outside world. It is everywhere. And Excel will do […] almost all the statistics that we teach an undergraduate to do. It has all the tasks, it has everything you could imagine. So, the point [is]: I want them to work with data that they believe is important, and I want them to believe it's important, and I want them to learn how to use Excel. […] We're using another technological tool, we're using the power of Excel. And, of course, the graphics in Excel are extremely powerful, so it's trivial that they learn all the graphics. They learn how to do statistics using Excel. (Bill; A2I)

Behind his selection of the artefacts for his course, Bill shows a conscious reflection for his decisions, that include several constructionist elements. For instance, he added that it was important for his students to learn about the characteristics (e.g., commands) of all the computational environments that were part of the course and that they learn to use them. We infer that he considers those **computational environments as mediums of expression** where students could **do mathematics** and **model**. These are decisions that involve both the *didactical configuration* but also its *exploitation mode*.

In his *didactical configuration,* the selected artefacts are meant to help students explore the complex mathematical ideas of his assignments; they are means to help connect and build relationships between those ideas and previous ones (**webbing**).

And the way in which Bill promotes that building of relationships (the making connections) is through the *exploitation mode:* an **exploration of mathematical ideas** through **modelling,** with students working both independently and **collaboratively.** In this *exploitation mode*, Bill designed programming tasks for **doing mathematics**, that would force students to go beyond what they would normally do in regular mathematics courses. He said: "my first thought in designing […] a lot of the MICA assignments, is to move them as far away as possible from standard mathematics teaching." Thus, Bill selects task contexts using challenging social and science topics related to interesting mathematical ideas, that he thinks students would **enjoy** (so that they become **engaged** in the projects, finding them **meaningful),** but also make them **reflect on their own thinking:**

*I want this to **be part of their life in a real** way […], something they say, "oh, wow! This is something good right now". […] One reason why I like the MICA courses, is because it can be a reflection of themselves. […] I think it's so important that students see a reflection of themselves in their work.* (Bill; FI-1)

His student Kassie's comment below reflects her own **thinking about her learning**:

"Doing mathematics" means to understand why, practice how, and then relate the answer back to the question. This helps to absorb learning something and remember why you did it. If you can understand why, then you will know how to do an advanced question in the future. (Kassie; LR1)

Bill's goal is for his students to use mathematics to **investigate problems, through programming models** (e.g., simulations and models of real-life phenomena). In learning to program, he expects his students to understand better the math, how to apply it and use it to explain the phenomena (**do mathematics**); it implies, **relating programming and mathematics,** as well as **relating various mathematical ideas:**

*[Mathematical concepts are] covered in other courses. My goal […] is to show people the power of introducing computing into **doing mathematics**, teaching mathematics, and so on. […] Part of this course is to try to understand the idea that we can take a real-world situation, and we can distil from that the mathematics, and then take that mathematics and write a simulation based on that mathematics. […] So it's a kind of a two-part process: the real world to the model, write the formal model that we do in the classroom, and then finally the computer simulation that we do in the lab. […] We are looking at the data first, we are building the programs […], observing what happens, and then we go and we try to prove and try to find mathematical evidence for what's going on. (Bill; A1I)*

Bill explains the **back-and-forth process** and relationships between the **real-world phenomena**, its **model** and **mathematics,** in the case of Assignment 1 (the Buffon needle problem),:

> *We can actually toss straws in the computer […], we can write a simulation. And part of this course is to try to understand the idea that we can take a real-world situation, and we can distil from that the mathematics, and then take that mathematics and write a simulation based on that mathematics, so it's a kind of a two-part process, the real world, to the model, right the formal model that we do in the classroom, and then finally the computer simulation that we do in the lab. (Bill; A1I)*

He emphasized this aspect of **making connections** in one of the last interviews:

> *Sometimes the connection is purely through programming skills. […] Sometimes I Bill say: "In this course, what are the kinds of things we've done, what types of techniques have we learned? You're going to show me an amazing way in which someone could learn mathematics using all the resources of a computer."* (Bill; FI-1)

His student Kassie explained the challenges she faced and how, in writing the computer program of the assignment, she had to understand the mathematics and programming concepts and relate them, and how that enhanced her learning:

> *[Some] parts of the assignment, I understood the mathematical parts […] and just needed to do the coding part of each. […] Certain topics and necessary components of the program that I had remembered and were reminded were crucial. This actually helped me to learn more […], which is pretty amazing. I completed the program. […]*
> *…it teaches you a different way of thinking and allows you to expand your horizons in a subject that you may tend to find repetitive. Learning programming is so important in this current society, as technology is a huge part of society and will become an even bigger on in the future.* (Kassie; LR1)

The explorations also take place in environments where students can **try** and **test** their programs**,** simulations and **models,** but where it is **safe to make mistakes, debug** and try again –as Papert (1980) described in his **microworlds'** idea. In this respect, Bill said:

> *How we get people to explore is that we create an environment where they are completely safe to do that and: "Oh, I can just generate anything, and look at anything. And I don't think I did that well enough here…."* (Bill; A3I)

In his *exploitation mode* and *didactical performance*, Bill also takes into account the **affective aspects.** For instance, encouraging students' curiosity and creativity, and being supportive**:**

> *The students […] were fascinated by the warfare stuff. […] I can get them to go a lot further in programming because they're curious, and they believe that this is important. […] Especially in MICA, […] they've written a creative project in mathematics. This is the first time. They need support and encouragement. They really do need that.* (Bill; A4I)

Bill's *didactical performance* involved classroom interactions where he helped students who were "stuck". His **approachable guiding** role **(**that support his students, gives them ideas or suggestions, discusses with them, and ask questions) helps promote students' connections of **programming** and **mathematical** knowledge and skills, without giving out all the information ahead of time, so students have to **play** and **discover**, in a **fun** and **engaging** environment:

> *I mean, it was really fun to do the lab, and their engagement is fun. So, it makes it fun […] because they're doing it, and they're working on it and asking me questions. And it just makes it fun. […] I get to have good conversations.* (Bill; A2I)
> *Sometimes I will say: "In this course, what are the kinds of things we've done?, what types of techniques have we learned?" […] They have to build a mathematical model that makes sense, and [if] that goes terribly wrong […] I have to pull them back from that and say "No, no. no, OK?, On paper, just you and I." And I sit with them with the paper to: "show me, draw me a picture, show me what happens" […]: the model first, the programming second. […] I*

had a lot of nice conversations about [what don't you know about this?" –the math problem]. (Bill; A3I)

Bill guides and supports his students, but lets them be responsible for their work (**student empowerment**), encouraging student exchanges and discussions. In that respect, Kassie said:

> *It helped definitely because […] everyone has different ideas […] if you look at like someone else's code and like maybe it's not working, then you […] kind of understand more why yours is.* (Kassie; LR1)

The Final Projects, in particular, are a **collaborative** process with both **novices and expert working and learning together**:

> *Right at the beginning when we start the projects, I tell them this: "It's not something you're going to do by yourself. You're working in a team" […] But I tell them I'm going to be an integral part in this process, we're all going to work together.* (Bill; A5I)

Concluding remarks

Our analysis focused on identifying the constructionist elements that Bill integrated into his orchestration; we did not attempt to critically appraise Bill's approach in any other respect. We observed that Bill's *didactical configuration* involved mathematical and social considerations, and a web of ideas and actions that provide a creative structure (a webbing) for drawing connections between programming and mathematics. Yet, it is in his *exploitation mode* and *didactical performance* that these provisions manifest in a constructionist way. In Figure 1 (see abstract), we illustrate selected aspects of Bill's orchestration components that aim to support students' programming for investigating math problems in real-world contexts.

The *didactical configuration* involved the computational artefacts that Bill selected for each assignment, the real-world data used in the tasks and the symbolic artefacts at play (e.g., codes, mathematical concepts, representations). The *exploitation mode* relates to his aims and the didactical design of the assignments, which involve modelling, problem-solving, simulations, and explorations; and where "mathematics is used in programming", and "programming is used to do and understand mathematics". And Bill's *didactical performance* aims at supporting and empowering his students, while also taking into account and promoting affective aspects (e.g. fun, motivation, creativity).

Our analysis shows provides insights into how the constructionist elements in Bill's orchestration helped achieve his, and the MICA course's, aim: to engage students in programming for mathematics investigations. This analysis informs our on-going work, where we study MICA students' instrumental geneses of programming, pointing to how the constructionist elements in the MICA approach may promote those geneses.

Acknowledgment

This research is funded by SSHRC grant #435-2017-0367, with ethics clearance REB #17-088.

References

Barabé, G. & Proulx, J. (2017). Révolutionner l'enseignement des mathématiques: le projet visionnaire de Seymour Papert. *For the Learning of Mathematics*, 37, 2.

Buteau C., Muller E. & Marshall N. (2015a). When a university mathematics department adopted core mathematics courses of an unintentionally constructionist nature: Really? *Digital Experiences in Mathematics Education*, 1(2–3): 133–155.

Buteau, C., Muller, E. & Ralph, B. (2015b). Integration of programming in the undergraduate mathematics program at Brock University. In *Online Proceedings of Math+Coding Symposium*, London, ON. http://researchideas.ca/coding/proceedings.html

Buteau, C., Sacristán, A.I & Muller, E. (2018). Teaching in a Sustained Post-Secondary Constructionist Implementation of Computational Thinking for Mathematics. In Dagienė, V. & Jasutė, E. (Ed.). *Constructionism 2018. Constructionism, Computational Thinking and Educational Innovation: conference proceedings*, 528-535.

Drijvers, P., Doorman, M., Boon, P., Reed, H., & Gravemeijer, K. (2010). The teacher and the tool: instrumental orchestrations in the technology-rich mathematics classroom. *Educational Studies in Mathematics*, 75(2), 213-234.

Guin, D., Ruthven, K., & Trouche, L. (Eds.). (2005). *The didactical challenge of symbolic calculators: Turning a computational device into a mathematical instrument*. NY: Springer.

Hoyles, C. & Noss, R. (2004). Situated abstraction: Mathematical understandings at the boundary. *Proceedings of Study Group 22 of ICME-10*, 7, 212-224.

Hoyles, C., Noss, R., & Kent, P. (2004). On the integration of digital technologies into mathematics classrooms. *International Journal of Computers for Mathematical Learning*, 9, 309-326.

Kynigos, C. (2015). Constructionism: Theory of Learning or Theory of Design? In S.J. Cho (ed.), *Selected Regular Lectures from ICME 12*. doi: 10.1007/978-3-319-17187-6_24

MICA guidelines URL (n.d.) http://ctuniversitymath.files.wordpress.com/2019/10/mica-ii-assignments.pdf

Noss R. & Clayson J. (2015) Reconstructing constructionism. *Constructivist Foundations*, 10(3): 285–288. http://constructivist.info/10/3/285

Noss, R. & Hoyles, C. (2017). Constructionism and Microworlds. In E. Duval, Sharples M., Sutherland R. et al. (eds.), *Technology Enhanced Learning*, (eds). Cham: Springer. DOI 10.1007/978-3-319-02600-8_3

Noss, R., & Hoyles, C. (1996). *Windows on mathematical meanings: Learning cultures and*

Papert, S. (1980a). *Mindstorms: Children, computers, and powerful ideas.* Basic Books, NY.

Papert, S. (1980b). Teaching children to be mathematicians vs. Teaching about mathematics. In R. P. Taylor (Ed.), *The Computer in the School: Tutor, Tool, Tutee* (pp. 177–196). New York, NY: Teachers College Press, Columbia University. (Original work published in 1972).

Papert, S. & Harel, I. (1991). Situating constructionism. In I. Harel and S. Papert (Eds), *Constructionism*. NY: Ablex Publishing Corporation, 1-11. http://www.papert.org/articles/SituatingConstructionism.html

Rabardel, P. (1995). *Les hommes et les technologies; approche cognitive des instruments contemporains.* Paris, France: Armand Colin.

Trouche, L. (2004). Managing complexity of human/machine interactions in computerized learning environments: Guiding students' command process through instrumental orchestrations. *International Journal of Computers for Mathematical Learning*, 9(3), 281–307.

Trouche, L. and P. Drijvers (2014). Webbing and orchestration. Two interrelated views on digital tools in mathematics education. *Teaching Mathematics and its Applications*, 33, 193–209. doi:10.1093/teamat/hru014.

Kynigos, C., & Psycharis, G. (2013). Designing for instrumentalisation: Constructionist perspectives on instrumental theory. *International Journal for Technology in Mathematics Education*, 20(1), 15–20.

Appendix: The Main Constructionist Principles and Keywords

Themes	Main constructionist principles	Keywords
Epistemology & conceptions of mathematical knowledge and of mathematics (How is mathematical knowledge constructed and what it means to do mathematics)	"Constructionism (in mathematics education) is an epistemology of (i) mathematics as a discipline and (ii) of doing mathematics" (Kynigos, 2015).	Doing mathematics
	"Mathematics is conceived as something that needs to be proved or refuted" (Barabé & Proulx, 2017).	Mathematics involves proving & refuting
	"it is possible for [students] to do creative mathematics (that is to say: to do mathematics)" by creating new branches of mathematics where beginners can play" (Papert, 1972/1980b; p.178) and that "enable beginners to discuss his mathematical thinking in a clear articulate way" (ibid, p. 180)	Mathematical "playgrounds" Creativity Doing mathematics Discuss math & math thinking
	"mathematical thinking involves communication irrespective of the (virtual or real) presence of others. It is the mediating tools – and especially representations – that […] form a major focus" (Hoyles & Noss, 2004; pp. 212-213)	Communicate (discuss) math / math thinking Mediating tools / representations (computational infrastructures)
	"The everyday-life experience of procedures and programming becomes a resource for doing mathematics" (Papert, 1980a, p. 154).	Computer programming for doing math
Conception of learning & of role of the student	"Principles of mathetics – ideas that illuminate and facilitate the process of learning": 1. "relate what is new and to be learned to something you already know. 2. "take what is new and make it your own: Make something new with it, play with it, build with it." (Papert, 1980a, p. 120).	Make connections (webbing) Make their own Play Building/constructing
	"mathematics might be learned in [….] **settings that are real**, socially cohesive, and where experts and novices are all learning [….] **Learning is not separate from reality**" (Papert, 1980a; p. 179).	Real settings / mathematics is a real(-life) activity Expert (teacher) – novice (student) learning together
	Students learn better when they build personally meaningful objects	Meaningful objects/projects
	-For learning to take place, knowledge has to be **(re) constructed** by the subject -"Meanings are created by experiences" (Noss & Hoyles, 1996, p. 46) i.e., when students consciously engage in constructing (e.g., program) The student's learning is facilitated by building external, "tangible" (in the sense of perceivable by the senses), shareable, constructions (Papert & Harel, 1991)	(Re)-construction by the subject Engaging in building/constructing: external, shareable constructions
	Students should have the opportunity to **engage with their own learning** (be conscious of own learning process): i.e. think (e.g. reflect about the problems as well as **think about own thinking** - analyze) (Noss & Hoyles, 2017) **make connections (webbing)** (Noss & Hoyles, 1996). "Meanings are constructed by **action** on virtual objects and relationships. Within a computational environment, some at least of these objects and **relationships become real** for the learner […]: **learners web their own knowledge and understandings** by actions within the microworld, and simultaneously articulate fragments of that knowledge encapsulated in computational objects and relationships - abstracting *within,* not *away from,* the situation." (Noss & Hoyles, 1996; p. 125)	Think about own thinking Make connections/relationships (webbing) through actions (Mathematical) meaning-making Situated abstractions
	The action of the student is central in his/her learning: s/he should have a central and active role and be in charge of the activity.	Active role in learning process Student empowerment
Pedagogy & design (e.g. of the learning environment)	**A project-oriented approach:** gives "time to **talk about it**, to establish a common language with a **collaborator** or an instructor, to **relate it** to other interests and problems" (Papert 1972/1980b; p. 179) "A project is long enough to have recognizable phases—such as planning, choosing a **strategy** of a attempting a very simple case first, finding the simple solution, ***debugging*** *it*, and so on." (Papert, 1972/1980b; p. 180)	Project-based learning (long-term) Discuss (talk) Collaborate Make connections (webbing) Strategizing Problem solving Debugging
	"the learning processes of an individual must be considered in **a meaningful context** of goal-directed, socially-situated activity." (Hoyles & Noss, 2004; p..212). E.g.: Personally meaningful projects	Personally meaningful projects Student empowerment
	"In a learning environment with the proper **emotional and intellectual support**, [students] can learn […] not only that they can **do mathematics** but that they can **enjoy it as well**". (Papert, 1980): Lovable mathematics (Papert, 1980b)	Emotional and intellectual support Doing mathematics Enjoyable (lovable) math
	Provide "**objects-to-think-with**": objects that children **can make theirs** for "themselves and in their own ways" (Papert, 1980a)	Objects-to-think-with Make their own
	-The learning environment and activities should provide opportunities for **students' explorations ("bricolage)**: **building, adapting, testing** and **rebuilding,** and allow them to **express themselves** (Papert & Harel, 1991). - Provide opportunities to **adapt** ,have **new ideas**, **test** and **share** them: **discuss** and **collaborate** with others (Kynigos, 2015; Noss & Hoyles, 2017).	Bricolage / exploring Building / constructing (new objects/ideas) Testing

	- Provide a **medium of expression** for students, e.g. the **construction of models**. The actions are expressed in a model (e.g. computer programs or codes, texts, diagrams, narratives, sketches, etc.). (Noss & Clayson, 2015). -**Modelling** promotes the learning of **powerful ideas** through use (Noss & Clayson, 2015); it emphasises the utility of a mathematical concept (Kynigos, 2015)	Adapting /re-mixing Have medium of expression Sharing / Discussing /Collaborating Construction of models / modelling Powerful ideas
	Testing the models Bill lead to a need of **fixing** (**debugging)** (Papert, 1980): that implies analysis and reflection.	Debugging / Analysing Adapting/ Re-mixing
	The teacher's role is to support students' "bricolage" and encourage their creativity to use, explore, build, adapt, test and rebuild their code.	Teacher/student new roles & relationship Creativity
	"the **relationship of the teacher to learner is very different**: the teacher introduces the learner to the microworld in which **discoveries** Bill be made, rather than to the discovery itself." (Papert, 1980b, p. 209). "The **flow of ideas and even of instructions is not a one-way street**," with **a culture that "enriches and facilitates the interaction between all participants** and offers opportunities for more articulate, effective, and honest teaching relationships" (p. 180), where the line between learners and teachers can fade. The […] teacher Bill answer questions, provide help if asked, and sometimes sit down next to a student and say: 'Let me show you something.' […]. Sometimes it is something the student can use for an immediate project. Sometimes it is something that the teacher has recently learned and thinks the student would **enjoy**. (p.179)	Teacher/student new roles & relationship Enjoyable
Computer programming and microworlds	The computer is "a mathematics- speaking being in the midst of the everyday life" that can provide **links between everyday life and fundamental and engaging mathematics**. Computer programming can bring us into **a new relationship to mathematics**": entering into a "**mathematical conversation**," showing possibilities that may have previously seemed "too hard." (Papert 1980a; p. 47) It provides a webbing structure that learners can draw upon and reconstruct for support for creating meanings (Noss & Hoyles, 1996)	Mathematical meaning-making Making connections with mathematics (webbing): "mathematical conversations" Relationship real-life & mathematics
	Programming is a type of construction (e.g. of a model) that allows students to **connect with insights into mathematics**, **science and other fields** (Papert, 1980).**Programming as a mathematical activity**: "**Programming** offers the student **an environment for mathematizing.** When mathematizing familiar processes is a fluent, natural and enjoyable activity, then it is about mathematizing mathematical structures, as in a good pure course on modern algebra." (Papert, 1972).	Programming as a mathematical activity and a type of construction Mathematical meaning-making Making connections (webbing) Mathematizing as an enjoyable activity
	Microworlds are **incubators of knowledge and powerful ideas** (Papert, 1980a): "… a successful microworld is both an epistemological and an **emotional universe**, a place where **powerful (mathematical, or scientific, or artistic) ideas** can be explored; but explored "in safety", acting as an **incubator** both in the sense of fostering conceptual growth, and a place where it is **safe to make mistakes** and show ignorance. And, centrally these days, it is a place where **ideas can be effortlessly shared, remixed and improved**" (Noss & Hoyles, 2017; p. 32). In "computational learning environments […] the process of generating and **expressing meanings** with the available representational infrastructure tends to produce individual and collective understandings and ways of working that are divergent from standard mathematics." […] "computational systems [are] a setting in which new kinds of representation for mathematical objects **generates new possibilities for mathematical expression**" (Hoyles & Noss, 2004; p. 213). In computer programming environments "mathematics is a **real activity** that can be **shared by novices and experts**" where "start interacting mathematically because the product of their mathematical work **belongs to them and belongs to real life**" (Papert 1980a, p. 179)	Microworlds: Contain objects-to-think-with /powerful ideas - incubators of knowledge and powerful ideas (epistemological universes) - emotional universes Experiment / investigate Mistakes are part of the process Testing/fixing/ debugging Adapting/ Re-mixing Mathematics is a real activity Mediums of expression Mathematical meaning-making Make their own Relationship real-life & mathematics Teacher/student new roles & relationship
	The (digital) tools play an important role in learning, since in themselves they are **expressions of mathematical meaning**. In addition, through the **feedback** that digital tools do to students' work, they allow them to **express their ideas**, individually or collaboratively. (Noss & Hoyles, 2017)	[Computational environment / digital tools are] mediums of expression Feedback [of computational environment]
	Computer feedback: impersonal, leads to **debugging** and reflection. (Papert, 1980): It allows students to **explore how** they **think (themselves)** and understand their mistakes (understand what went wrong, why and how to fix it). (Papert & Harel, 1991)	Feedback [of computational environment] Debugging / fixing / Analysing Think about thinking

The “Creative Learning Challenge Brazil” from the Perspective of Constructionism

Ann Berger Valente, *annbv@media.mit.edu*
Lifelong Kindergarten Group, MIT Media Lab, Cambridge, MA

Abstract

The “Creative Learning Challenge Brazil” is a fellowship program that builds upon the country’s broad base of hands-on and innovative educational practices. The fellowship is a project of the Lifelong Kindergarten group of the MIT Media Lab in partnership with the Lemann Foundation, a Brazilian educational foundation. In designing the Fellowship program, the question was, how can we structure opportunities for adults to learn in meaningful and context-relevant ways that more closely resemble the exploratory, idiosyncratic explorations of young children? Paramount was to support this group of educators in their ongoing work and to help them reframe their experience through the collective input of a highly innovative and engaged group of professional colleagues.

The objective of the fellowship program is twofold – to support educators working in socio-economically vulnerable communities while simultaneously fomenting the development of the Brazilian Creative Learning Network. The Network is composed of over 3000 artists, designers, inventors, entrepreneurs, educators, researchers, and school leaders nationwide dedicated to making Brazilian education more hands-on, relevant, playful and collaborative for all.

The fellowship program is currently in its 4th edition, with a total of 54 Fellows across Brazil. The Fellows develop a wide range of projects in schools, community centres and afterschool/non-formal learning spaces, innovating new materials to use in hands-on learning experiences. For example: braving the school kitchen in which the curriculum subjects of chemistry, physics and math are reinterpreted through the implementation of student recipes; photography as a tool to build identity in remote communities founded by former runaway slaves; building low cost robotics toolkits for public schools, and exploring art and cognition through the development of creative computing projects with children with disabilities.

Figure 1. Photography, Memory, and Identity / Creative computing with children with disabilities

Based upon the documentation of reflective practices, as well as materials produced by the program and its participants, this paper analyses the fellows’ development through the lens of Constructionism. It describes a journey as the fellows develop their projects, objects to think with that are used to build a personal understanding of Constructionism and Creative Learning. Constructionism based practices are revealed at multiple levels of the program, including the selection process, project development, and fellowship design, thereby providing a robust immersive learning and leadership development experience.

Keywords

Hands-on Learning, Creativity, Professional development, Fellowship Program, Mediation

Introduction

No one walks without learning to walk, without learning how to make his way by walking, redoing and refining the dream by which he set out to walk.[36]

Paulo Freire

The Brazilian educational system is marked by a strong tradition of progressive leaders not the least of whom is Paulo Freire whose critical pedagogy put into question the "banking" model of education in which children were seen as vessels, or bank accounts, to be filled with information aligned with the dominant norms of society (Freire, 1970). Brazil's history of digital technologies in education is equally marked by innovative public policies which deliberately opposed the notion of computers as teaching machines in favour of strategies for learning with technology (Almeida & Valente, 2016). Despite the persistence of major structural problems, inefficiencies, and inequity in Brazilian public education, as well as recent reactionary political tendencies nationwide, there exists a growing cohort of educators committed to promoting hands-on, innovative, relevant and collaborative educational opportunities engaging all children and youth. Often these are isolated educators, artists, designers, entrepreneurs and school leaders dedicated to making a difference. The Creative Learning Challenge was born in this context.

The objective of the current paper is to explore the ways in which Creative Learning Challenge Brazil aligns with the theory of Constructionism and to show how Fellows experience first-hand the educational philosophy that they are attempting to put into practice in their own projects. The methodology adopted to produce this paper is based on the documentation of various face-to-face and virtual reflective practices, as well as materials produced by the program and Fellows.

The Fellowship Program

The Brazilian Creative Learning Network[37] was created in 2015 as an initiative of the Lemann Creative Learning Program of the MIT Media Lab's Lifelong Kindergarten group, with support from the Lemann Foundation, a Brazilian education foundation. The Lemann Creative Learning Program is based on three premises: 1) that there exists a wealth of initiatives in the spirit of Creative Learning throughout Brazil yet disconnected from one another; 2) that the adoption of new educational practices is more sustainable when building upon already existing organic transformational practices; and 3) that teachers learn better from one another than from external specialists. Therefore, the Program set out to identify initiatives and to connect them with one another, and with the Lifelong Kindergarten, in order to disseminate best practices and find solutions to common problems. In this way, the Lemann Creative Learning Program has promoted the Brazilian Creative Learning Network, enabling it to engage over 3000 members in a process of collective learning through events, meetups, discussions, networking, sharing of best practices, and resource development. The fellowship program, the Creative Learning Challenge Brazil, is one of these efforts.

The Creative Learning Challenge is a strategy to give visibility to the creative efforts of a select group of Brazilian educators working in socio-economically vulnerable communities and to advance their experiences in Creative Learning. Simultaneously, the Fellowship aims to expand Creative Learning practice, to share this new knowledge with the network, and to inspire the development of new Creative Learning initiatives with different groups and regions of the country.

The Challenge consists of a yearly competition, awarding approximately 10 fellowships each year. Awarded projects receive a grant of US$3000 to invest in their implementation. The Challenge is

[36] Ninguém caminha sem aprender a caminhar, sem aprender a fazer o caminho caminhando, refazendo e retocando o sonho pelo qual se pôs a caminhar.

[37] Rede Brasileira de Aprendizagem Criativa http://aprendizagemcriativa.org

currently in its 4th edition. The first cohort was awarded in 2015, the same year as the formation of the Brazilian Creative Learning Network. Currently there are a total of 54 Fellows from 42 projects across Brazil, as seen in Figure 2.

Figure 2. Location of the 42 projects awarded in the Creative Learning Challenge since 2015

The program consists of three face-to-face events. The kick-off event is in March and coincides with a major Creative Learning festival in São Paulo City organized by the Brazilian Creative Learning Network. Here, newly initiated Fellows participate in several hands-on workshops including Scratch and creative computing. They also have the opportunity to get to know a wide range of innovative projects and to share their experiences with like-minded educators. In May, the Fellows visit the Lifelong Kindergarten group at the MIT Media Lab as well as several Boston-based formal and informal educational spaces. The visits include cutting edge learning environments as well as less sophisticated yet innovative solutions in education. Fellows take from these experiences new ways to think about facilitating learning, with or without technology, and come to challenge presumptions that contribute to traditional educational conventions in Brazil. The third and final face-to-face meeting is in late September at the national Brazilian Creative Learning Conference. At this point the Fellows have initial results from their own projects that they can share and discuss more deeply with each other and with the larger Network community.

Over the course of the year, the program holds monthly online meetings on topics relevant to the emerging interests and needs of the Fellows. In addition, there is continuous support from peers and the program staff to address the challenges and to celebrate the successes that arise during the process of project development. Finally, in late November, there is a virtual meeting for feedback and reflections. Following the year of intense activities, Fellows continue as members of the Brazilian Creative Learning Network and many become agents fomenting local community groups of Creative Learning in their cities as well as nation-wide.

A broad range of topics have been explored by the Fellows over the course of the program. For example, in the traditional public-school setting, a physics and chemistry teacher makes tangible her theoretical curricular content using the school kitchen. She weaves in the social sciences and mathematics as they explore regional recipes and comfort foods. Students test the effects of tweaking the ingredients and they make the calculations to scale up and down for various events and campaigns. Figure 3.

Figure 3. Braving the School Kitchen [38]

In another example, a not-for-profit organization devoted to educational and cultural communication brings photography to a remote community founded by former runaway slaves. Together they mobilize school children, teachers, youth advocates, artisans and the senior population to re-discover and document their heritage through photography exhibits. The goal is to empower the community with the knowledge and tools to build their own cultural museum with the support of public funding. Figure 4.

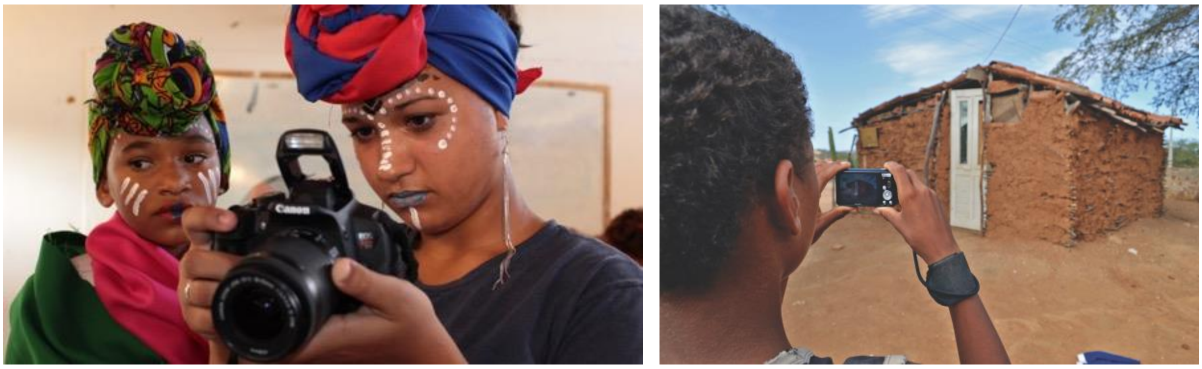


Figure 4. Photography, Memory, and Identity [39]

Yet another project integrates psychology and engineering departments at a public university for the exploration of art and cognition through the development of creative computing projects with children with disabilities. Engineering undergraduates develop a series of scenarios in which the children can build simple contraptions using robotics and physical computing. Psychology students investigate the children's behaviours as they navigate the various activities. In the end, engineering and psychology teams combine their tools to create meaningful educational opportunities. Figure 5.

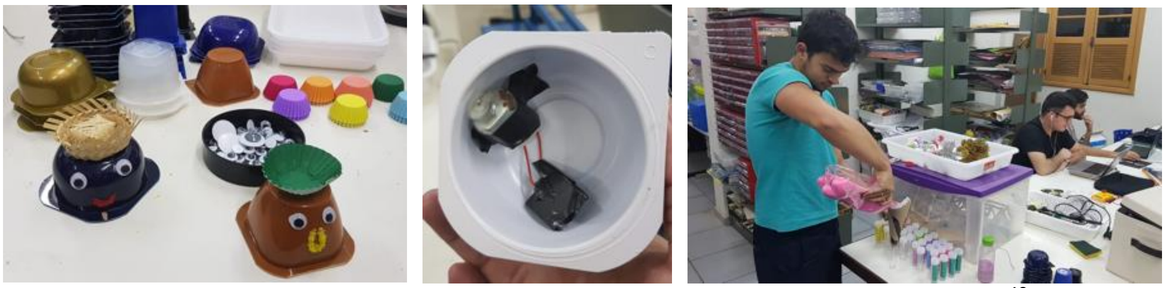


Figure 5. Develop creative computing with children with disabilities [40]

[38] Ellen Regina Romero Barbosa
[39] Centro de Documentação e Comunicação Popular - CECOP
[40] Eduardo Bento Pereira

A brief description of each of the projects awarded through the Creative Learning Challenge Brazil is available at the Lifelong Kindergarten group of the MIT Media Lab (Burd, 2019; Burd, 2018; Burd, 2017). A sample of the experiences of the 2018 cohort can be found at Creative Learning in Practice (Rede Brasileira de Aprendizagem Criativa, 2018). Digital technologies have a special role in these projects; however, they are not a necessary requirement. Learning can be creative using paper, wood or cloth, depending on the context and objectives. Nevertheless, programming and digital technologies expand the repertoire of phenomena that can be explored through projects, making the concepts manipulable and transparent. Likewise, programming provides a unique record for reflective practice.

Creative Learning and Constructionism

The term, "Creative Learning" was coined by Mitchel Resnick and his colleagues at the Lifelong Kindergarten. It calls attention to the quality of learning experiences afforded by the Scratch programming language in tandem with the variety of devices, solutions and the online community designed to "help children learn to think creatively, reason systematically, and work collaboratively" (Resnick, 2012). These skills make up a (digital) fluency which is seen as essential for successful navigation in a constantly changing workplace and increasing complex society. Engine of Creative Learning is the “Creative Learning Spiral” (Resnick, 2017) in which the learner, who could be a kindergartener but could also be a willing adult, engages in a continuous process of imagining an idea, creating an instantiation of the idea, experimenting with the creation in a playful fashion, sharing the creation with other people, using this collective experience to reflect on the original idea, and then start all over again imagining the next iteration. The sequence, ‘imagine - create - play - share - reflect - imagine again’, is a spiral in the sense that each iteration is informed by, and elaborates on, the previous one (Figure 6).

Figure 6. Creative Learning Spiral

The spiral encourages creative thinking in so much as the activities follow, to greater or lesser degree, four essentials principles - the 4P’s. “In short, we believe the best way to cultivate creativity is to support people working on projects based on their passions, in collaboration with peers and in a playful spirit.” (ibid, p.16). The 4P’s are central parameters for defining what we mean by Creative Learning activities in the Brazilian Creative Learning Network.

Creative Learning has its foundation in Constructionism as defined by Seymour Papert, which in turn has its roots in Piagetian Constructivism. Both Constructionism and Constructivism are defined by the self-organizing aspect of human intellectual development as the child constructs and modifies her mental schema through interaction and manipulation in the physical world. However, Constructionism diverges from Constructivism on several tenets, one of which is that Constructionism places particular importance on building things - the creation of objects, or public entities, through which budding knowledge can be expressed (Harel & Papert, 1991). The activity of programming the Logo Turtle, ‘teaching’ the turtle how to perform a particular task, is a clear instantiation of this idea. With this comes a new appreciation of the concrete, not as a stage to pass through on the way to abstract thinking, but as a way to explore and manipulate complex ideas in physical entities.

This notion that knowledge is constructed through the exploration of numerous concrete examples leads to a second divergence of Constructionism from Constructivism. In contrast to Piagetian stage theory, Papert asserts that learning is essentially nonlinear and context dependent - it is the connectedness and continuity of meaning that is embedded in a particular material or setting (Papert, 1993). Furthermore, he associates this nonlinearity with different learning approaches, from planning on the one hand to tinkering, or the "bricolage" on the other (idem). This focus on the individual learner and her idiosyncratic learning path raises the importance of building on one's own passions and interests. When the public entity that she constructs is personally significant, this influences the level of engagement, the understanding of the concepts, and the depth of learning.

Building on this intellectual tradition, Creative Learning goes on to expand significantly the notion of public entity, the wealth of concepts that can be manipulated, their shareabilitiy, the role of collaboration, and the building of communities. Creative Learning highlights the social dimension of Constructionism by emphasizing the role of collaboration and peer interaction in regard to personal engagement and understanding of new perspectives.

The Creative Learning Challenge Brazil embodies the Creative Learning and Constructionism educational philosophies on several levels. First, each Fellow is deeply engaged in a project that is personally meaningful. This project becomes their public entity, their "object to think with". Second, the projects themselves promote Creative Learning practices with children in a variety of formal and informal learning settings. Therefore, the Fellows are providing opportunities for others to experience their own constructions. Third, based upon their practice with children, Fellows create, test and present their own novel Creative Learning activity to be included in the project gallery of the Brazilian Creative Learning Network. As designers of Creative Learning, the Fellows share their activity with the online community and become part of the learning spiral. Fourth, the Fellowship program itself is conducted in the spirit of Creative Learning. The mentorship and facilitation provided by the program constitutes an instantiation of Constructionism based pedagogy. Finally, the program is embedded in the Brazilian Creative Learning Network, an organization designed to champion for Creative Learning practices through a process of collective learning.

Understanding through examples

Papert goes to great lengths to define Constructionism through examples, explaining that the stories of concrete situations constitute the most (only) effective way to express meaning given that comprehension of any new concept is a personal construction. Furthermore, as more and more examples are shared, from very distinct domains of knowledge and experience, these instantiations will more likely than not converge on the essential nature of Constructionism.

> "I find an interesting toe-hold for the problem in which I called the playful facet-- the element of tease inherent in the idea that it would be particularly oxymoronic to convey the idea of constructionism through a definition since, after all, constructionism boils down to demanding that everything be understood by being constructed. The joke is relevant to the problem, for the more we share the less improbable it is that our self- constructed constructions should converge." (Harel & Papert, 1991, p.2).

Papert's joke is the basis for the selection criteria adopted in the Fellowship's selection process. The rubrics used by the evaluation committee to evaluate the project proposals centre on various parameters, most importantly, alignment with the principles of Creative Learning as defined by hands-on, significant to the target population, collaborative, and promoting playful experimentation. The projects are analysed with regard to the relative emphasis on promoting learning as opposed to direct teaching. By concentrating on the quality of the learning experience as defined in the 4P's, the rubric allows for a tremendous amount of freedom in terms of the nature of the application. Indeed, many of the applicants have never heard of Creative Learning before the call for proposals. The fellowship itself helps them give a name to, and thereby reflect upon and express, many aspects of their practices.

The idea is that through the accumulation, discussion, and reflection on a growing mass of examples that contemplate diversity in local cultures, thematic content, and implementation settings, "as bricoleurs we can come to agreement" (Harel & Papert, 1991, p.2). Through building personal meaning around Constructionism, the diversity of perspectives can substantiate the original idea and make it more easily understood.

A project narrative

Imagine a small village called Pau D'Arco near the Araguaia River in the Amazon. In the municipality of Santa Bárbara, Pau D'Arco is located 45 Km from the Pará State capital city of Belém. Proximity to the metropolis brings high rates of violence and narcotics, while poor access leads to inadequate sanitation, lack of schools, and limited opportunities. Raimundo das Graças Lima Xavier was raised in Pau D'Arco and founded the organization Ação Parceiros (Partners in Action) which for over 10 years has provided social and educational support for the children and their families in a community forgotten by the public and private sectors. Xavier and his colleague, Sebastião Borges Fonseca, submitted a proposal to bring Social Educational Robotics to Ação Parceiros, Figure 7.

Figure 7. Ação Parceiros community centre / Educational robotics materials [41]

The project proposal described hands on activities in the subject areas of Reading, Math, Electronics, Programming, Robotics, English as a Second Language, and Civic values. The curriculum content would be developed through 'Creative Education' defined as creative activities and problem solving focused on solutions to the adverse situations encountered in the children's daily lives. For example, the proposed description of the course in Robotics would use materials from obsolete computers and electro domestics in conjunction with recycled materials. The description was very much in the spirit of Creative Learning. The 4P's were evident in the elaboration of projects that were personally significant and developed in teams in a playful spirit of re-inventing discarded objects. The project was selected for its intuitive alignment with the ideas of Creative Learning in the candidates' areas of expertise (electronics, programming and robotics) and for its potential to bring these ideas to such a vulnerable, remote community.

Nevertheless, in the same breath, the proposal also described the use of curricular booklets they had produced to navigate the content in each of the core curriculum subject areas. In the beginning of the Fellowship, the relationship between these artefacts was not clear nor how the divergent educational paradigms would be reconciled. Both existed harmoniously in the original project proposal.

[41] Projeto Social Ação Parceiros

The cognitive dissonance began to emerge at the first meeting in São Paulo and ruptured into despair during the week in Boston. Fellows Xavier and Fonseca returned to Ação Parceiros and proceeded to knock down the internal walls of their community centre, literally breaking down the divisions between disciplines, unifying the methodology. It was through the renovation of the physical space that they aimed to express the integrated essence of learning through building external entities. Their rooms needed to be open with easily accessible materials in order to foment the autonomy and creativity they envisioned. Renovation of the outdoor area into an adventure playground provided learning experiences as important as building an electrical circuit.

Figure 8. Integrated learning spaces. Ação Parceiros

The challenges continue for Fellows Xavier e Fonseca as they struggle to guarantee the sustainability of Ação Parceiros. Workshops, courses, and speaking engagements are all part of the dissemination and fundraising process. With each event the materials become more ingenious as they invent new ways to embody the curriculum content in the physical constructions. Gradually the pedagogical methodology becomes increasingly adapted to the particular audience. Their learning spiral progresses.

Walk the talk - a Constructionist Fellowship

> [...] the goal is to teach in such a way as to produce the most learning for the least teaching.” (Papert, 1993, p.139).

In designing the Fellowship program, the question as proposed by the Lifelong Kindergarten group was, how can we structure opportunities for adults to learn in meaningful and context-relevant ways that more closely resemble the exploratory, idiosyncratic explorations of young children? Paramount was to support this group of educators in their ongoing work and to help them reframe their experience through the collective input of a highly innovative and engaged group of professional colleagues.

The Creative Learning Challenge Brazil is a journey that begins with the work that the Fellows have been developing in their respective settings, as synthesized in their project proposals. The project becomes their ‘object to think with’. It is the backdrop as the Fellows come in contact with Constructionism in a wide variety of settings. What each Fellow takes away from these experiences is highly individualized, and assimilation of the ideas will depend upon the person’s current frame of mind. For example, some people focus on the technology, others want to expand their repertoire of activities, others are looking for how to connect with the classroom curriculum. As they test these new ideas in their projects there is a constant sharing of partial results online with the group, as well as the monthly online meetings when the program mediators have an opportunity to delve into each of the projects. The program mediators use targeted questions to get Fellows to describe their practice and thereby express their assumptions. Adapting or even letting go of certain perspectives is part of the process.

The program includes two moments for more structured sharing with a larger audience - at the MIT Media Lab Lifelong Kindergarten and at the Brazilian Creative Learning Conference. Here, each Fellow sets up a table with artefacts from their project and a brief description of their journey - an exercise in synthesizing their identity and purpose. The audience circulates among the tables and the participants engage in conversations, as shown in Figure 9.

Figure 9. Interactive project presentations

These interactions with a genuinely interested and supportive audience of experts are extremely rich opportunities for expanding perspectives on their own work while simultaneously legitimizing their accomplishments. They are essential moments that support learning and creativity.

The disposition to become critically reflective of one's own assumptions is highly valued in the Fellowship program. Space for revisions in the original project proposal is built into the program. The ideas described in the original project proposals are just a starting point. Fellows are encouraged to experiment with new ideas and to remix practices that others have used. In fact, if the project does not evolve over the course of the year, it constitutes a missed opportunity for learning, both by the Fellow and the program.

The Fellowship program is designed to be a microworld of Creative Learning, in the same vein as Papert (1993) defines learning French in France or learning math through playing with mathematical objects. The Fellowship can be thought of as an environment to be explored in order to understand its internal machinations – a safe place to build new practices. There is little information delivery, rather the program works with what the Fellows bring to the table. The “curricular agenda” is the process of the fellowship itself. There are no lectures, rather a plethora of optional resources. Over time, the intentionality of mediation through Constructionism becomes an object to think with. The Fellowship becomes a model that can be referenced when decisions are made for how to deal with situations that arise in their own practice.

Concluding remarks

The Creative Learning Challenge Brazil adopts Constructionism as its model for learning. Based upon the premise of knowledge construction as an iterative process of externalizing one’s ideas as a public entity that can be shared, reflected upon, and refined, the fellowship explores Creative Learning as defined by the MIT Lifelong Kindergarten. Fellows are chosen on the basis of project proposals that may or may not explicitly align with Creative Learning principles, but that embody the elements of promoting projects that are personally significant to the learner in a collaborative and playful environment. These essential principles allow for a wide range of themes in a variety of contexts. In this way, the notions of Constructionism and Creative Learning are instantiated through the examples.

This paper demonstrates various ways in which the Fellowship itself is an example of Creative Learning in the tradition of Constructionism. Fellows learn about these notions as they refine their practice with children and other educators. Simultaneously they themselves are involved in a process of Constructionism as they share, reflect upon and refine their own projects. As future leaders in the Creative Learning community it is essential that they construct their own meaning of Creative Learning in their context. The Fellowship itself is a process that is tied to its theoretical roots and becomes an object to think with. In this way it offers the stability of a set of shared references. A transformative experience that strengthens the intellectual, social and affective affinities that define the community of practice.

References

Ação Parceiros. (2019, October 20) *Home page*. Retrieved from https://www.acaoparceiros.org/

Almeida, M. E. B. and Valente, J. A. (2016) *Políticas de Tecnologias na Educação Brasileira: Histófico, Lições Aprendidas e Recomendações.* Centro de Inovação para a Educaçãp Brasileira, Estudos #4. Retrieved from http://cieb.net.br/wp-content/uploads/2019/04/CIEB-Estudos-4-Politicas-de-Tecnologia-na-Educacao-Brasileira-v.-22dez2016.pdf

Burd, L. (2019, October 20) *Resultado do Desafio de Aprendizagem Criativa Brasil 2019*. Retrieved from https://www.media.mit.edu/posts/resultado-do-desafio-aprendizagem-criativa-brasil-2019/ .

Burd, L. (2019, October 20) *Resultado do Desafio de Aprendizagem Criativa Brasil 2018*. Retrieved from https://www.media.mit.edu/posts/resultado-do-desafio-aprendizagem-criativa-brasil-2018/ .

Burd, L. (2019, October 20) *Resultado do Desafio de Aprendizagem Criativa Brasil 2017*. Retrieved from https://www.media.mit.edu/posts/resultado-do-desafio-aprendizagem-criativa-brasil-2017/ .

Freire, P. (1970) *Pedagogy of the Oppressed.* A Continuum Book The Seabury Press, New York.

Harel, I. and Papert, S. (Eds). (1991) *Constructionism: research reports and essays 1985-1990 by the Epistemology & Learning Research Group*. The Media Laboratory, MIT. Ablex Publishing Corporation, Norwood, NJ.

Papert, S. (1193) *The Children's Machine: rethinking school in the age of the computer*. Basic Books, New York.

Rede Brasileira de Aprendizagem Criativa. (2018) *Aprendizagem Criativa na Prática: DAC Brasil 2018*. Rede Brasileira de Aprendizagem Criativa, Fundação Lemann, MIT Media Lab. Retrieved from http://porvir.org/publicacao-propoe-viagem-por-8-projetos-inspiradores-de-aprendizagem-criativa/

Rede Brasileira de Aprendizagem Criativa. (2019, October 20) *Home page.* Retrieved from http://aprendizagemcriativa.org

Resnick, M. (2017) *Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play.* MIT Press, Cambridge, MA.

Resnick, M. (2012) *Mother's Day, Warrior Cats, and Digital Fluency: Stories from the Scratch Online Community.* Paper presented at Constructionism 2012, Athens, Greece. Retrieved from http://web.media.mit.edu/~mres/papers/mothers-day-warrior-cats.pdf

The gradual construction of a constructionist kindergarten program

Ruthi Aladjem, *rutrutal@gmail.com*
Knowledge Technology Lab, Tel-Aviv University

Asi Kuperman, *asikuper@gmail.com*
Knowledge Technology Lab, Tel-Aviv University

David Mioduser, *miodu@tauex.tau.ac.il*
School of Education, Tel-Aviv University

Abstract

This paper describes the process and outcomes of a decade long implementation of a constructionist learning program. The program focuses on advancing the technological thinking of kindergarten children and promoting the acquisition of basic programming concepts. During the ten years since its initial implementation, the program has undergone substantial changes and shifts in focus. Each of the six strands composing the program have evolved, and have undergone multiple transformations. The vast majority of changes that have made their way to the program through the years were a direct outcome of events arising from the field (i.e. the day to day experience at the kindergarten). Many have surfaced from the kindergarten children, either through their explicit suggestions or by way of observations of their activities. On a previous paper, the program rationale and theoretical grounding were introduced; eight years on, with the program actively operating in 8 kindergartens, this paper presents reflections, conclusions and main takeaways from the years of implementation.

Keywords

Technological thinking, Kindergarten, Smart-artifacts, Design and Learning, Implementation

Introduction - Program rationale and theoretical grounding

This paper discusses a decade-long implementation of a constructionist learning program focused on advancing the technological thinking of kindergarten children, and on promoting the learning of basic programming concepts. The program is based on the "Design and Learning" (D&L) model (Mioduser, D. 2009), a constructionist learning model based on the premise that "*mindful interaction with the designed world (the human-mind-made-world), and active involvement in designing objects for this world, serve as an intellectual and practical platform for promoting young children's learning about contents, processes, and skills related to the artificial world*" (Mioduser, Kuperman and Levy, 2012). The models' rationale is built around the encounter between technology and learning, both taken in their broadest possible sense (Mioduser, D. 2009; Mioduser, Kuperman and Levy, 2012).

Program scope and implementation

For the past decade, the program has been implemented in 8 kindergartens across Israel. Each kindergarten is attended by 35 children on average, between the ages of 5-7, with an educational team comprising one teacher and one assistant. Before and during the Implementation, the team undergoes comprehensive training consisting of an introduction to core ideas in constructivist and constructionist theory and an introduction to the D&L model. The team also receives active ongoing support during the school year.

The program is composed of 6 strands (Problem-solving; the designed/artificial world; Design; Notations; Smart artifacts and the Integrative project). An additional strand, "Special design for special needs", is not addressed in this paper. Although the program is presented as six separate strands, the order of the strands is not hierarchical, and the strands are in fact closely intertwined. Accordingly, the educational team is encouraged to experiment, change the order of strands and activities or combine among them, and make their personal interpretations - according to what they believe will best suit the children and the unique character of each kindergarten.

Following is a description of the strands and the ways in which they have evolved over the course of a decade-long implementation.

Strand 1- Approaching problem-solving (from haphazard to budding systematically in planning and implementing solutions)

The main objective of the problem-solving strand is to establish a "problem-solving culture" in the kindergarten, and to encourage problem-solving (PB) processes as a commonplace practice and an integral part of the kindergarten life. When engaged in PB, children and teachers act together as co-creators engaged in a structured process, using a PB framework. The framework consists of a simplified version of PB and design processes from the world of engineering. The adaption to kindergarten has fewer steps, posed as a set of four consecutive stages- starting from problem definition, followed by initial solutions, then selection of the best solution and finally, implementation of the selected solution (Kimbell et al. 1966; Mioduser & Dagan 2007). The simplified framework was created in order to facilitate a structural way of thinking but also to encourage freedom and creativity through the process.

Main extensions gradually constructed along the implementation

With the implementation of the program, PB processes have become a key element in kindergarten life, serving as a tool in multiple areas. "*There is a problem*" is a statement often heard across the kindergarten, expressed by the children as an initiation of a problem-solving cycle. The children and educational team regularly use the PB process for tackling dilemmas in versatile fields, ranging from technological tasks (product building and repair, building games, etc.) through solving dilemmas that arise spontaneously during the day (setting rules and establishing kindergarten norms) to social challenges (solving quarrels and disputes, reaching compromises, reaching agreements etc.). Often, children use problem-solving tools independently, without adult mediation on multiple issues and areas. In addition, this practice has expanded beyond the kindergarten walls and parents often report that their child initiates a "problem-solving" processes at home, when they encounter conflicts (such as issues with their siblings).

Initially a simplified, four-stage process scheme, the framework has been extended and further elaborated. Three stages (**Reasoning, Control, Research and exploration**) were added, as a result of practices arising from the field, as detailed below.

- **Research and exploration -** "*I want to build a robot*" declared S, 6-year-old. Her friend offers to Google "how to build a robot". The exploration phase was not a part of the original process, but in many kindergartens, the children turned to search engines to explore and understand and ask questions. As a result, circles of inquiry were formed tangentially to the subject being investigated, in a kind of threading process.
- **Control (debugging and problem fixing) -** "*I want to explain why my solution is the best*" Y. 5-year-old told the group of children engaged in a thought process to solve the "tallest tower" problem. Control processes were initiated intuitively by the children as part of the process.
- **Reasoning -** "*I want to explain why my solution is the best*" Y. The 5-year-old girl told the group of children engaged in a thought process to solve the "tallest tower" problem. The children often express their wish to explain and elaborate on the strengths of their solution and a need to explain and justify the choice. As a result, the "reasoning" step was added to the PS process schema.

The evolution of the PB process sprouted in a bottom-up manner. Eventually, some of the changes from the field made their way to the (ever-evolving) framework, but variations of this process remain and are encouraged.

Strand 2 - The designed/artificial world (artifacts and their use and context)

The main objective of this strand was for the children to gain knowledge and understanding of the world of objects, our material culture, through the exploration of different artifacts and structures. An additional goal was for the children to gain an understanding of mechanisms, through disassemble and repair of objects and through engaging in classification processes based on the objects' structural and functional properties.

Main extensions gradually constructed along the implementation

The program created encounters with a rich variety of objects brought by the children. The children were actively engaged in complex classification processes that went beyond the standard kindergarten curriculum and enabled an exploration in a wide technological, cultural and social contexts.

- **Broader and unique categories for sorting** - "*They are both connectors… the phone between people and the stapler between papers*". R. (6 years old) reflected while holding a phone in one hand and a stapler in another. Standard kindergarten curriculum usually calls for external visual parameters such as colour, material, size and shape as criteria for classing objects. The classification process conducted seeking a common basis for seemingly very different objects gave rise to ideas for non-trivial categories suggested by the children, e.g., on properties related to their functionality at a quite abstract level.

- **A holistic exploration of the world of objects**- *"Let's build a plane just like the wright brothers!*" proposed A. (6 years old.) The exploration went beyond the object itself, on gaining a broad understanding of the object, its origin, history, context and more, reaching a more holistic understanding of the object's story and evolution. (See Figure 1).

Figure 1. Holistic exploration of the world of objects.

Strand 3 - Design (from free-form building to reflective construction)

The strand involves construction and assembly using building kits and games (such as Lego blocks, Mechanic Duplo, K'nex, Magnetic tiles etc.) and creating sketches of the constructed object, as documentation of it. The main objective of the sketches was to initiate a reflective process, (both on the 3D construction, as well as on the transition from 3D construction to a 2D sketch), in order to support the development of fine-tuned perception of details as well as representational skills.

Main extensions gradually constructed along the implementation

This strand has gained momentum and became a central activity in the kindergartens. As with the problem-solving process described above, the process has expanded beyond its initial scope, due to "natural evolution" in the kindergarten environment.

- **Sketching for planning, not just documentation**- Originally sketching was used for documentation and reflective objectives but it spontaneously became a part of the planning process. The children's sketches serve as designs for constructions by their peers who were interested in creating constructions which they liked. In cases where the sketch (now serving as plan) was not clear (not detailed enough or not accurate enough to build from) the children tried to explain (or "translate") the sketch for their peers. This attempt to decipher the sketch for construction often let to a rich discourse between the child who made the sketch and their peers. Some kindergartens created "catalogues" of past sketches that allow anyone interested in doing so, to select a sketch and use it as a plan for construction. Furthermore, Sketching was used spontaneously by the children in many additional situations during the day, as a means for planning an activity, process or a game.

- **Use of colour to depict structure**- initially the sketches were created in a simple pencil, but the children spontaneously decided to add colours, to better depict the colours of the parts used for the physical model. This was also an effective way to better communicate the sketch to the peers interested in reconstructing it.

Figure 2. Left- original model, right a model- constructed from the sketch.

- **Authentic Construction-**The use of construction and sketching techniques has expanded beyond the building games activity. They have become tools, used on authentic areas in the children's lives for planning the socio-dramatic game (Bretherton & Beeghly 1989) and also as tools for co-planning different constructions (see figure 3).

Figure 3- Sketching and building a "smiling robot".

Strand 4 - Notations (from conventional signs to computer programs)

This strand involves the construction of visual and conceptual representations and symbols, that serve as 'epistemic tools' encapsulating both the represented content as well as the representing means (Mioduser, Kuperman and Levy, 2012), effectively creating a physical, symbolic language. Similarly to "sketching" described above, this strand focuses on symbolic representations, with the symbolic language not constricted to a two-dimensional representation on paper.

Main extensions gradually constructed along with the implementation

The original goal was to introduce the children to notations as an initial, initiation, stage to be followed by the programming (smart artifacts) phase. As part of the activities, the children create routes and mazes; they also experiment and practice receiving and providing instructions. During the program, multiple tracks appeared in the kindergarten built from anything available (blankets, chairs, cubes, and more), representing real and imaginary spaces. Building routes and mazes as part of the sociodramatic play and representing reality became a common practice. Children often engage in discussion with their peers about the choices of symbols and how best to represent objects, facilitating an abstract understanding of language and communication.

The children also create representations (maps) of the routes taken from their home to the kindergarten and include details such as significant objects that they encounter on the way. Maps have effectively become a tool used in multiple context and topics; the children use the map as a tool to explain and represent knowledge.

Figure 4- building a track and a map.

Strand 5 - Smart artifacts (Understanding and constructing artificial-behaviour)

The main focus of the strand is to encounter programmable "smart" artifacts and experience behaviour building for smart artifacts while engaging in collaborative processes involving planning, problem-solving and repair processes (Papert, 1987, 1993). During the program, the children are exposed to the world of smart artifacts and coding becomes a part of the daily routine of kindergarten life. The programming (of EV3 robot LEGO) is done through a programming interface (Kinderbot, Scratch etc.). The process includes three phases: building a (physical) route, documenting it (sketching) and programming the robot to pass through the route.

Main extensions gradually constructed along the implementation

This strand has also expanded and was essentially reshaped due to new tools, and to events in kindergarten reality.

- **A shift in focus**- from programming to storytelling - The original objective was to transition from creating a route for the robot to programming the robot's behaviour. In effect, much (if not most) of the focus of the activity was placed on the route (or multiple routes) creation. The route, in fact, became more of a goal of itself, then (as originally planned) a means to an end. Routes were created around varying themes and topics serving as grounds for socio-dramatic play and storytelling, with the smart artifact becoming just one of the players. This shift was not preplanned, but the importance of creating a rich context for the smart artifact, describing its background and "motivation" for travelling through the route was evident in all kindergartens.

- **The move from desktop to mobile-based programming –** As reported on our previous paper (Aladjem, Kuperman, & Mioduser, 2017), the transition from a desktop to mobile-based programming environments contributed to substantial changes including changes in the programmers' perspectives, changes in foci and learning patterns in different programming modes, and changes in patterns of collaboration among peers.

Figure 5- Mobile programing.

Strand 6- The integrative project

The main objective of this strand is to create an opportunity to concretely implement the knowledge (construct), tools and skills that the children have acquired, by creating a tangible, integrative project. The project is created through collaborative work, leveraging creative thinking in order to plan, explore, solve, and finally to create an end product. In other words, the integrative project effectively "ties" all the strands together into a hands-on, holistic product-oriented product.

Main extensions gradually constructed along the implementation

The plan for this strand has emerged impromptu since there was room for interpretation; there was substantial variation between kindergarten classes. The ideas for the projects were raised by the children or identified together by analyzing situations, events or needs in the daily life of the kindergarten. Projects carried out in all kindergartens reflected the skills that the children acquired during the school year of the program. Starting with the planning stage - the organization, the equipment lists, the process, collaborative work, exploration processes, problem-solving processes and more. Examples of projects included different types of factories (such as a chocolate factory and a notebook factory), different customer-facing businesses (such as a café and a restaurant) and complex physical environments (such as a palace and mazes). The second part of the school year in all the kindergartens revolved around the integrative project, beyond constructing the project environment, the project was discussed and approached from multiple perspective and angles (such as historical, political and economic perspectives) and became the centre and focal point for much of the kindergarten activities.

Summary and Conclusions

During a decade long implementation, the program has undergone evolutionary transformations both in scope and focus as well as in some of the tools used. This paper is an opportunity to revisit the fundamental principles upon which the program was constructed (constructionist theory and the D&L model) and to reflect upon the agents and processes who have contributed to its gradual construction.

The paper illustrates how integrating a constructionist program as part of the kindergarten curriculum can create a unique experience and an added value. Beyond the acquisition of programming skills, the program supports the development of an extensive skillset, authentically arising from children's curiosity and needs (Papert, 1987; Resnick, 2007). Skills and practices acquired throughout the program become tools (or a tool-set) used in multiple contexts in kindergarten and beyond.

The program has undergone a gradual construction since its inception up until its current state, with the vast majority of changes emerging from the field (i.e. the kindergartens where the program has been running) or more specifically, from ideas, opinions and needs raised by the children. Flexibility is an inherent principle of the program, and it allowed for different interpretations and implementations in each kindergarten. This principle, combined with the sensitivity of the educational team, gave room for the children to express themselves and in turn, contributed to substantial changes which made their way to the "official" program.

Each of the six strands composing the program evolved, expanded and transformed in ways which have exceeded our plans and expectations. Furthermore, in each of the kindergartens where it has been implemented, the program has gradually become a core activity and its underlying philosophy has, in many ways, permeated all activities. Following, we briefly present what we view as three of the more substantial takeaways observed in the past decade, across program strands and kindergartens.

- **The child as an equal partner and co-designer**- A guiding principle of this program and of constructionist thought in general, places the child in the centre, acting as an explorer, a researcher, following their curiosity and learning through hands-on experience with technological tools (Piaget, 1975, Papert 1980, 1993). Over the years, through our observations, we found fundamental support to the perception that the child should be a part of all aspects of the program. This was observed on all strands, with the children expressing needs and ideas and acting as active agents and significant contributors to the design and construction of the program.
- **Collaboration amongst peers-** Learner interaction can stimulate cognitive development, as the individual contributes to the group knowledge and in turn, the group knowledge contributes to the individual's understanding (Kolodner et al. 2003). Collaboration among peers was observed on all the strands, often happening naturally and spontaneously with no need for encouragement or mediation from the educational team. For example, while Sketching, in many cases, the children shared the process of "deciphering" a sketch and using it as a basis for construction. Problem-solving was also conducted in spontaneous cooperation.
- **Use of multiple tools –** Using tools for designing, creating, and manipulating objects, both in the physical and virtual world is a powerful idea that can empower the individual (Papert 1980, 1993; Resnick, 2008). It seems that the type of tool used often dictates the nature of the artifact. For example, each type of building game, ked to similar constructions across kindergartens. Each game has different rules and structures (connections stabilizing mechanisms etc.) and effectively requires the use of different techniques. This observation highlights the importance of using a variety of tools for developing concrete thinking and learning about abstract phenomena.

In closing, a decade long implementation brings with it many insights, many of which were presented in this paper. As our next steps, we plan to expand the program to more kindergartens with diverse populations and to expose more educational teams, children and communities to constructionist philosophy, thinking and learning.

References

Aladjem, R., Kuperman, A., & Mioduser, D. (2017). Kindergarten Programming Goes Mobile: Should The Next Years Be About Ubiquity?. In *Proceedings of the 2017 ACM Conference on Interaction Design and Children* (pp. 304-309).

Bretherton, I., & Beeghly, M. (1989). Pretense: Acting "as if." In Bridges, J. J. & Hazen, N. H. (Eds.), Action in social context: Perspectives on early development (pp. 239–271). New York: Plenum Press

Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., ... & Ryan, M. (2003). Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting learning by design (tm) into practice. *The journal of the learning sciences*, 12(4), 495-547.

Mioduser, D., & Dagan, O. (2007). The effect of alternative approaches to design instruction (structural or functional) on students' mental models of technological design processes. *International Journal of Technology and Design Education*, 17(2), 135-148.

Mioduser, D. (2009). Design and Learning (D&L) by Kindergarten Children. *Educational Research and Review*, *3*, 9-13.

Mioduser, D., Kuperman, A & Levy, T, S (2012). Design and Learning (D&L) in the Kindergarten. In *Proceedings of the Constructionism 2012 Conference*. (pp.620-624) Athens, Greece.

Papert, S. (1980, 1993). *Mindstorms: children, computers, and powerful ideas.* 1st and 2nd eds. (pp. 3-176). Cambridge, MA: Basic Books.

Papert, S. (1987) Computer Criticism vs. Technocentric Thinking. *Educational Researcher*. 16(1), 22-30

Piaget, J. (1975). Comments on mathematical education. *Contemporary education*, 47(1), 5.

Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. In *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition* (pp. 1-6). ACM.

Resnick, M. (2008). Sowing the Seeds for a More Creative Society. *Learning & Leading with Technology*, 35(4), 18-22.

The Magic Word: A Coding Tutorial-Game to Engage Female Teenagers in App Design

Bernadette Spieler, *bernadette.spieler@uni-hildesheim.de*
Institute for Mathematics and Applied Informatics, University of Hildesheim, Hildesheim, Germany

Naomi Pfaff, *pfaff@student.tugraz.at*
Institute of Software Technology, Graz University of Technology, Graz, Austria

Stefania Makrygiannaki, *it164703@it.teithe.gr*
International Hellenic University, Nea Moudania, Greece

Wolfgang Slany, *slany@tugraz.at*
Institute of Software Technology, Graz University of Technology, Graz, Austria

Abstract

Educational games are commonly used to motivate students and provide enhanced learning opportunities. Apps and mobile games play an increasingly important role in education and smartphones are part of the daily lives of most female teenagers: Half of mobile gamers are women and 64% of women prefer smartphones to other platforms. However, gender differences in playing behaviour and preferences raises concerns about potential gender inequalities when games are developed for education. In order to develop a tutorial game that suits the female target group and provides challenging tasks to solve, girls were involved at a very early stage of the development cycle and the idea was developed on the basis of surveys and focus group discussions. A first prototype of the game has been tested in a mixed-gender group to get feedback about the learning content, the worked examples, and the whole structure of the game. Finally, a tutorial game with six worked examples has been released in our Luna&Cat app, a programming tool that has been designed for our female target group in particular.

Figure 1. Tutorial-Game for Luna&Cat: The Magic Word

Keywords

Game Design, Gendered Design, Mobile Learning, Constructionism, Gaming Literacy

Introduction

The acquisition of digital skills is more important than ever and represents a key professional qualification of future generations (European Commission, 2018). There is a great potential for young women to counteract the acute (and growing) shortage of qualified professionals in ICT (European Commission, 2016). However, the absence of female students who are interested in Information and Communication Technology (ICT) related fields can be observed at all levels of education as well as in the industry (European Statistics Eurostat, 2018). To allow women to improve their lives by offering them better career choices and more quality jobs, opportunities must be visible. Consequently, many girls leave school without any meaningful knowledge in Computer Science (CS), never quite understanding what CS is and how it relates to algorithmic thinking or problem-solving (Giannakos et al., 2014). These girls will be less likely to choose a career in CS or study it as a major. One accessible way is to catch their attention with attractive, supportive tools and games. To appeal to female teenagers in particular, a tailored version of our app Pocket Code, with the name "Luna&Cat" has been developed (Spieler and Slany, 2019). For this paper we design and develop a new coding tutorial for this app. The tutorial game should be extremely engaging and motivating for this challenging target group of younger female teenagers between 13 to 14 years old. At the same time, it should teach coding on small smartphone screens.

This paper is organised as follows: First, we provide an overview of the literature that informed the design of the tutorial game. The preferences of the game's target audience, the strengths of previous comparable games, and the psychological background of the game's learning mechanism are presented. In the following section, we outline the methods used to tailor the game to the game preferences of our target group as well as describing the results yielded by these methods. Next, we describe the first prototype and the procedure that was implemented in testing this prototype. The changes made to the finished version of the game based on our experiences with the prototype are also discussed in this section. Finally, our findings are presented and their implications and restrictions are discussed. Ongoing studies of the game are described and future research designs are suggested.

Literature that Informed the Tutorial Game

For the development of the tutorial game, it was first essential to consider the preferences of our target group of female teenagers between 13 to 14 years old. Second, the game must be challenging and follow game design strategies and concepts, and third, it should deliver meaningful coding concepts and help girls to develop their own games in the future. Beside explaining important game design concepts and gender conscious strategies in games, this section provides an overview to Constructionism and tutorial games, and highlights challenges in problem- based learning by considering cognitive load and applying worked examples.

Gaming Perceptions and Experiences in Teenager Girls

The ESA annual report contains statistics on video game player demographics (ESA, 2006-2013). Here, we see a slight change from 2006, where female players accounted for 38% of video game players, to a total of 45% female video game players in 2013. The flow state, described by Csikszentmihaliy (1975), is the state in which people are fully absorbed in their task, forget space and time, and only care about the activity - they are in a flow experience. This state of mind is very common while playing games. When comparing self-documented data with actual data from game servers, it was found that women underestimated their playing time by an average of 3 hours per week, while men overestimated their playing time by an average of 1 hour per week (Williams et al., 2009a). That users reach this state in the context of an educational tutorial game is very desirable for developers. Furthermore, William, et. al. (2009b) find out upon examination that men were significantly more motivated by achievement and women were significantly more motivated by social and immersion factors such as geographic exploration, role-playing, avatar customization, and escapism. In describing game style preference, Kinzie and Joseph (2008) suggest that girls at a mean age of 12 prefer creative and explorative aspects of a game, and

Quaiser-Pohl, Geiser and Lehmann, (2006) continue that female high school girls choose more non-players or logic/skill training games.

To conclude, a closer look at game designs in reference to the different motivations for gameplay discussed above, as well as full consideration of the different gaming style preferences between gender, can prevent developers from excluding potential player groups.

Constructionism and Tutorial-Games

Tutorial games are available for many computer games, but rarely for learning programming. The few examples available are usually playable for a few minutes and not very exciting. A great exception to this common problem that appeals to teenagers is Nintendo's tutorial game in "Wario Ware[42]: Do It Yourself." Another example tutorial game for coding designed especially for girls is the one with Wonder Woman[43].

The constructionist theory by Semour Papert (1985) explains the benefit if students use computational technologies to construct knowledge through the act of constructing personally meaningful projects. In Mindstorms (1985), Papert suggests the use of MicroWorlds in which learning about specific principles occur. Tutorials can be organized in such a way that instructions help students to develop technological knowledge to further acquire the skills to express themselves and their ideas through new tools (Stager, 2001). The Luna&Cat tutorial will enable girls to feel intellectually powerful by solving challenges after challenges, perhaps for the first time in their lives.

Problem Solving Instruction

Problem Schemas are mental representations of solutions to a type of problem. They contain the operators that can be applied to a problem, the problem states that lead from the initial problem to its solution, and the effect of operators on the problem during different problem states (Sweller and Cooper, 1985). This information can be used to solve any problem belonging to the same category. Expertise in a problem domain can be understood as the number of problem schemas acquired by an individual. The main learning outcome of problem solving is the acquisition of problem schemas.

Cognitive Load: To solve a problem, one has to mentally represent all components of the problem and manipulate them by applying operators to the problem (van Gog et al., 2006). The strain a problem puts on cognitive resources is the cognitive load imposed by this problem. The cognitive load imposed by problem solving affects novices and experts differently. Experts have automatized the mental representation of the problems' components so that they are left with sufficient cognitive resources to acquire problem schemas during problem solving. Novices' representation of the problems components is not automatic and therefore imposes a high cognitive load on their working memory.

Worked Examples are solved problems. For example, in Algebra, a worked example is a line by line documentation of every problem state from the initial problem to its resolution. They show learners how the initial problem state is transformed to the goal state through the use of operators. By definition, worked examples contain all the information that is necessary to form a problem schema. Worked examples impose a lower cognitive load than problem solving while providing the same information (van Gog et al., 2006). Therefore, learners have more cognitive resources available for the processing of problem schemas, leading to a better retention of problem schemas. This has been shown to increase learning outcomes in both expert and novice learners when compared to problem solving (Carroll, 1994; Sweller and Cooper, 1985; van Gog, Kester and Paas, 2011; van Gog et al., 2006). Novice learners receive an even greater benefit from the study of worked examples than experts because the high cognitive load that is imposed by problem

[42] Wario War: https://www.youtube.com/watch?v=4lSQqUoqQ9Q&feature=youtu.be&t=88
[43] Wonder Women: https://www.madewithcode.com/projects/wonderwoman

solving impacts their ability to acquire problem schemas acquisition more severely (Sweller and Cooper, 1985). Therefore, worked examples are especially beneficial to our target group.

Research Design

For the design and story of the tutorial game a bottom up approach was used by developing personas first. A persona is a description of user characteristics and her or his aims (Cooper, 2003). According to Cooper, a persona should be presented in text and/or image and it is usually generated to help designers to understand, describe, and define user preferences and behavior patterns. The data for the personas for this tutorial game has been constructed on the basis of questionnaires and focus discussions. In May 2018, a survey was created which included questions surrounding girls' game preferences, internet use (mobile games, social media), and general information (hobbies, interests, TV series, movies). The survey was handed out to 21 female students in Grade 4 (between 13 to 14 years old). In addition, a focus group discussion in which four students attended was conducted in order to ask more specific questions about the topics which emerged.

Furthermore, in June 2019, a first prototype of the tutorial game which already included three worked examples, was tested in a class of 24 students. This was important to get insights on how a tutorial game could be integrated in our Luna&Cat app as well as how to approach further difficulties during solving the tutorial.

Building a Gender-Conscious Tutorial Game by using Personas

The app Pocket Code has been developed at the Graz University of Technology in Austria under the Catrobat association (https://catrobat.org). It has a media-rich programming environment for teenagers to learn coding with a visual programming language very similar to the Scratch (https://scratch.mit.edu/) environment, which has been used for creating games and apps (Slany, 2014). Pocket Code is freely available on the Google Play Store (https://catrob.at/pc) and on iTunes (https.//catrob.at/PCios) and allows for the creation of games, stories, and many types of other apps directly on phones, thereby teaching fundamental programming skills. In addition, a new version of our app with the name Luna&Cat is available on Google Play (https://catrob.at/luna) since April 2019. This version has been developed by considering gender-sensitive aspects from gender studies with the goal to reinforce female teenagers. To engage more female teenagers in coding, a tutorial game has been created especially for this app version. The focus was on the conception, design, and development and testing the tutorial game.

Personas were used to determine the game genre of the tutorial and the narrative genre of its storyline (Cooper, 2004). In an interaction with a digital game, the primary goal of any user is to have fun. Students were categorised based on their requirements for fun; the game genres that students named as their favorites were used to for categorisation. Personas were developed from the answers given by group members on our questionnaire. We then constructed a questionnaire filled out by the persona by inserting real answers of group members belonging to the same group into a single questionnaire.

The first group was used to create the primary persona. This group favours "Jump&Run" games. Their favorite feature of their favorite games is that "you have to be clever and lucky". They prefer light games that emphasise manual and perceptual skill over story and strategy. Several members of this group listed "Candy Crush" as their favorite game and two students stated that they liked "Candy Crush" because "you have to think". The persona for the "Jump&Run" group is **Natalie**. Her favorite movies are the Harry Potter movies. Her favorite singer is Billie Eilish and she has written Billie Eilish's darker song lyrics into her the sidelines of her school notebooks in dramatic calligraphy. She likes the tone and imagery of "Alice in Wonderland" for the same reasons she likes Billie Eilish. She is very good at braiding hair. She has calm hands and is good at fiddling with bits, which is why her family asks her to open small clasps on necklaces or repair small earrings.

The second group was used to create the secondary persona. Its members stated that their favorite game genre was adventure games. Its quote regarding its favorite feature of its favorite game was “The world is huge, so you can always discover something new. It’s not boring because you can make your own free decisions”. This group did not like games which had issues such as “not enough space, not free enough, no more quests”. The persona for the adventure game group is **Laura**. Laura’s favorite book is the Alexander Rider series. She enjoys physics and the feeling of understanding a new concept. She likes Marvel movies and StarWars. She also associates the science fiction elements of StarWars and Marvel with physics and the sciences and therefore feels a connection to their audience. She is also interested in and fascinated by the science fiction technology featured in these movies.

The design of the game combined the interests of these personas. It is true that it is considered good practice to keep personas as specific as possible as it is thought to be better to create a product that is very appealing to a small group than to create a product that is mildly appealing to a large group (Cooper, 2004). However, our study was addressing the preferences of a previously specified group, that of 13 to 14 year old girls, so that it was possible to combine personas without significantly broadening the target group.

The exercises in the first game would appeal to Natalie. They are puzzles and their increasing difficulty gives Natalie a sense of self-efficacy. Programming in visual languages such as the Catrobat language involves perceptual skills such as the recognition and memorisation of patterns. The individual tasks designed for her were to serve as minigames which are lighter and closer to games like “Candy Crush”. These minigames were placed within a larger and more immersive adventure game.

The storyline of Luna’s quest was designed for Laura, but it would appeal to both Laura and Natalie. Laura would enjoy it because of its immersive effect and the fictional world created by the narrative; both of these features are central to adventure games. Laura would prefer if there was more choice of actions within the game; this should be considered in future versions of the game. Laura would be excited to learn to code because of the connection between programming and the development of new technologies. She would be intrinsically motivated to understand the worked examples.

Case Study: Application of the Tutorial Game

The prototype focused on having users test some in-app features on an educational application, while on the other hand, the finished version concentrated on having users code missing parts of the story in a game. A first user test of the prototype involved 24 students (11 female, 13 male) in an academic high school in Graz, Austria. The goal was to test if the concepts delivered by the tutorial could be transferred to similar tasks. Therefore, half of the class was randomly assigned a tutorial game that included animations and pictures of code snippet to explain the worked example and the other group had only a description in the form of a text, see Figure 2. The results of this first test did not produce a large range in values, making the interpretation of the effect of the two conditions difficult to decipher. The mean number of errors made by the text example group was 0.83 while the mean number of errors made by the worked example group was 1.60. The text example group took an average of 9 minutes to solve the problems while the worked example group took 11 minutes. One possible interpretation of these results would be that the worked example group produced more errors because they wrote more code. This is supported by observations during the grading of the programs written by students. Students in the text example group did not finish their programs or handed in empty programs. As the method of grading had previously been defined as counting errors it did not allow for this to be included in grading.

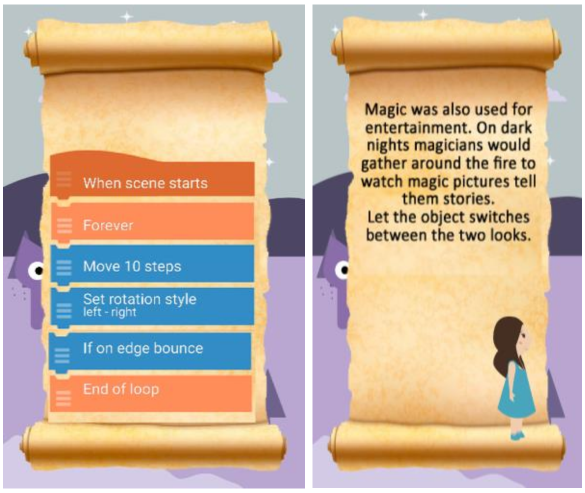


Figure 2: Left: animations and code; Right: text description only.

The Magic Word: A First Concept

The final version of the game displays a major and drastic improvement in comparison with the prototype. The storyline and teaching method were kept the same theoretically, as friendlier graphics and easier to use interface was developed. General add-ons to the game were a menu for easier navigation through the game, see Figure 3.

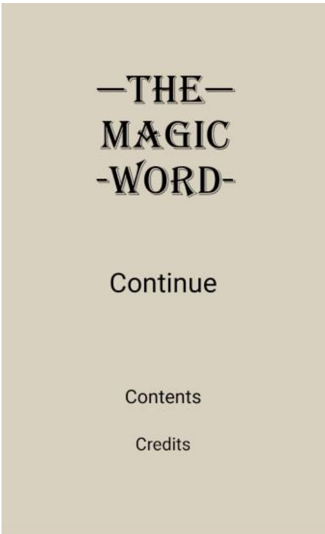
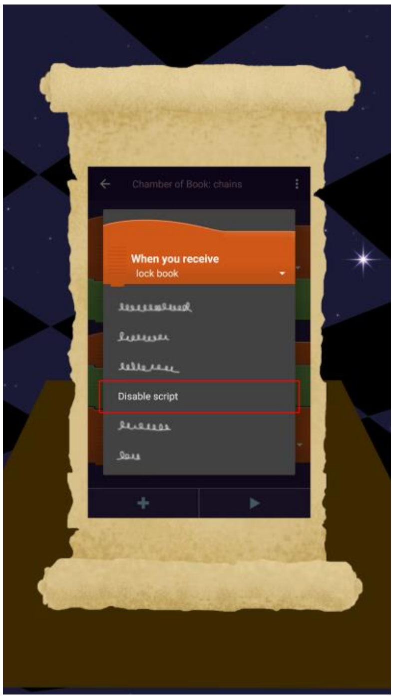
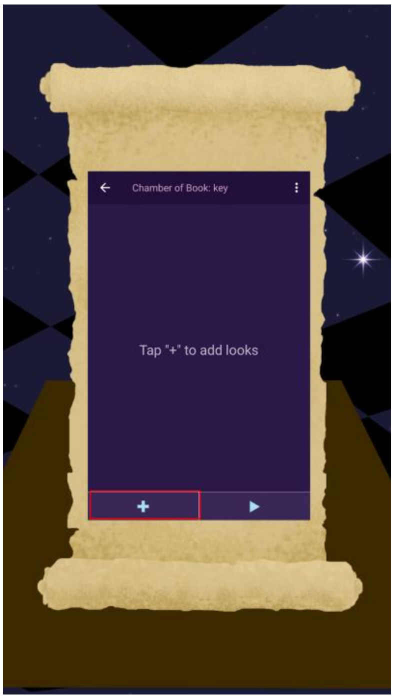
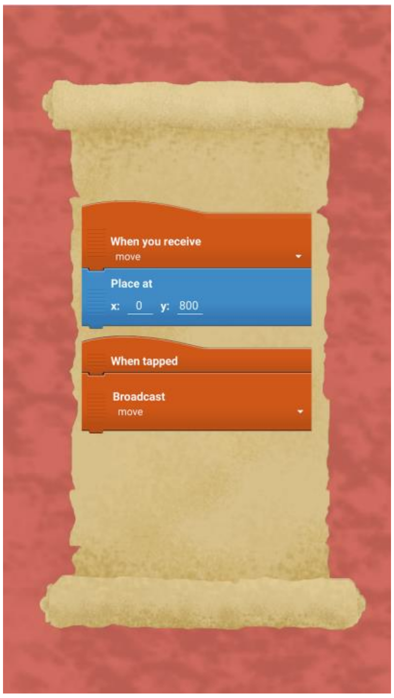


Figure 3. The Magic Word: Main Menu

The story of the game blends programming into the game world and makes programming ability central to the success of the main character's quest. The story is also designed to get the user emotionally invested in the main character's success. In the prologue the user helps Luna's mentor Cat to steal a book. This book appears in Lunas room. We see Luna examine to book. She is interrupted by Noodles, her puppy. They play and during the game, the puppy falls through the book which is a portal to a magical world. Cat appears and explains that Luna is a descendant of magicians from this world and that it is time for her to learn magic. The magic Cat teaches Luna is directed by the user through coding. The "spells" Cat shows Luna are worked examples of code. The user's ability to apply the code to the problem in the game world determines the success of the spell cast by Luna. As the story progresses Luna and Cat get closer and closer to Noodles. The obstacles in their way can only be removed by magic. Table 1 presents the worked examples:

Table 3: Worked examples (WE) of the tutorial game

		
WE 1: disable code blocks.	WE 2: add looks to game objects.	WE 3: function of motion blocks.
		
WE 4: function of loops.	WE 5: function of broadcasts.	WE 6: function of conditions.

WE 7: function of motion sensors.

WE 8: function of visual sensors.

WE 9: function of variables.

Discussion & Conclusion

At the time of writing, a study of the tutorial's effect on programming ability is still in progress. This one following the first case study, tested more students and counted the number of correct code blocks in order to account for this issue. For the first case study the usability issues with the prototype definitely had a negative impact on learning outcomes. The clear step-by-step instructions provided in the finished version of the game have resolved this issue. There was also confusion about the object-oriented design of the Luna&Cat app. In the finished version this issue was addressed through a unit that is dedicated to explaining this concept as well as other important functionalities of the app. The effect of the tutorial's learning mechanism should be more clearly visible in future studies which will use the improved final version of the tutorial.

Testing the game with a larger sample would provide a clearer picture of the tutorial's learning outcome. While the tutorial's learning outcome is central to an assessment of its quality, it is true that the tutorial was only intended to teach basic programming principles that could then be built on through the active use of the Luna&Cat app. While the understanding of basic programming principles is a prerequisite for the successful use of the the Luna&Cat app, willingness to engage with the app is also needed. Future studies should investigate the tutorial's effect on participants' willingness to engage with programming and the Luna&Cat app. For example, the tutorial's motivational effects could be investigated by testing the usefulness of the tutorial game as a predictor of future programming experience in a longitudinal study.

References

Carroll, W.M. (1994). Using worked examples as an instructional support in the algebra classroom. Journal of educational psychology, 86(3), 360.

Csikszentmihalyi, M. 1975. Beyond boredom and anxiety, San Francisco: Jossey-Bass.

Cooper, A. (2004). The inmates are running the asylum (1st ed.). Indianapolis, Indiana: Sams Publishing.

European Commission (2018) Increase in gender gap in the digital sector - Study on Women in the Digital Age. [online] Available at: https://ec.europa.eu/digital-single-market/en/news/increase-gender-gap-digital-sector-study-women-digital-age. Last access: 18.09.2019

European Commission (2016) Women in digital — a gap and an opportunity. [online] Available at: https://ec.europa.eu/digital-single-market/en/blog/women-digital-gap-and-opportunity. Last access: 28.08.2019

European Statistics (2018) Women in science and technology. [online] Available at: https://ec.europa.eu/eurostat/web/products-eurostat-news/-/EDN-20180210-1?inheritRedirect=true Last access: 10.09.2019

Entertainment Software Association. (2006). Essential facts about the video and computer game industry.Washington DC: Author. [online] Available at: http://www.theesa.com/facts/pdfs/ESA_EF_2006.pdf Last access: 16.09.2019

Entertainment Software Association. (2013). Essential facts about the video and computer game industry.Washington DC: Author. [online] Available at: http://www.theesa.com/facts/pdfs/ESA_EF_2013.pdf Last access: 16.09.2019

Giannakos, M.N., Jacceri, L, and Leftheriotis, I. (2014) Happy Girls Engaging with Technology: Assessing Emotions and Engagement Related to Programming Activities. In Learning and Collaboration Technologies. Designing and Developing Novel Learning Experiences, Vol 8523, 398–409.

Papert, S. (1985) Mindstorms. Children, Computer, and Powerful Ideas. In Basic Books Inc.

Quaiser-Pohl, C., Geiser, C., & Lehmann, W. (2006). The relationship between computer-game preference,gender, and mental-rotation ability. Personality and Individual Differences, 40(3), 609-619.

Kinzie, M. B., & Joseph, D. R. D. (2008). Gender differences in game activity preferences of middle school children: Implications for educational game design. Educational Technology Research & Development, 56(5/6), 643-663.

Spieler, B. and Slany, W. (2019) A Customised App to Attract Female Teenagers to Coding. Conference on Gender Research (ICGR 2019). April 11-12, 2019, Rom, Italy.

Stager, G.S. (2001) Constructionism as a High-Tech Intervention Strategy for At-Risk Learners. National Educational Computing Conference, “Building on the Future”.

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. Cognition and instruction, 2(1), 59-89

Slany, W. (2014). Tinkering with Pocket Code, a Scratch-like programming app for your smartphone. In Proceedings of Constructionism 2014.

Van Gog, T., Paas, F., & Van Merriënboer, J. J. (2006). Effects of process-oriented worked examples on troubleshooting transfer performance. Learning and Instruction, 16(2), 154-164.

Van Gog, T., Kester, L., & Paas, F. (2011). Effects of worked examples, example-problem, and problem-example pairs on novices’ learning. Contemporary Educational Psychology, 36(3), 212-218.

Williams, D., Yee, N., & Caplan, S. (2009a) Who plays, how much, and why? Debunking the stereotypical gamer profile. Journal of Computer-mediated Communication, 13(4), 993-1018.

Williams, D., Consalvo, M., Caplan, S., & Yee, N. (2009b). Looking for gender: Gender roles and behavior among online gamers. Journals of Communication, 59(4), 700-725.

Thinking in Levels Across Multiple Levels

Arthur Hjorth, *arthur@cs.au.dk,*
Department of Computer Science, Aarhus University

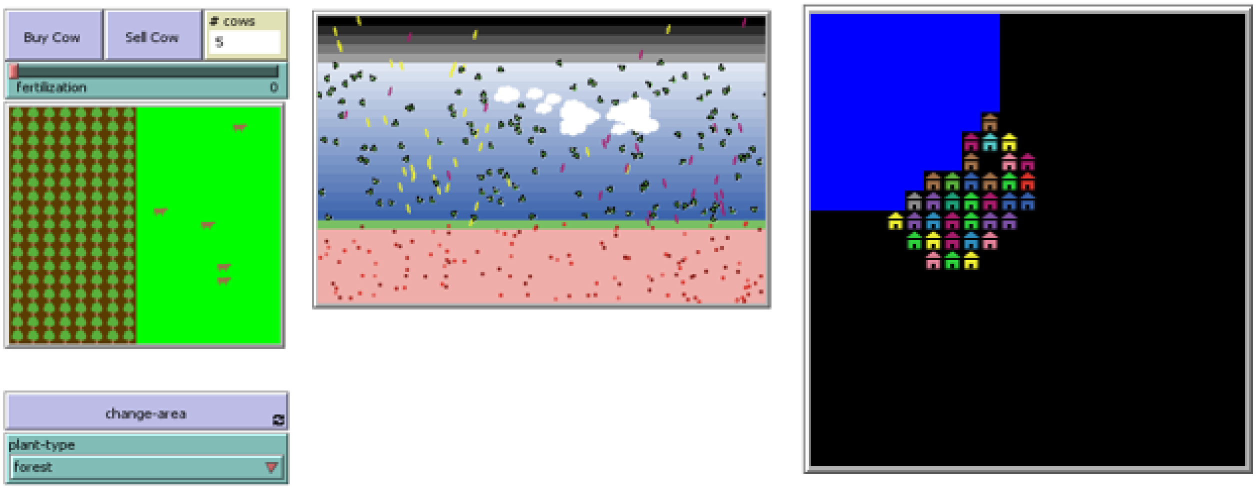
Corey Brady, *corey.brady@vanderbilt.edu,*
Peabody College of Education and Human Development, Vanderbilt University

Uri Wilensky, *uri@northwestern.edu,*
School of Education and Social Policy & Department of Computer Science, Northwestern University

Abstract

In this paper, we present a study of our first implementation of a classroom learning activity in which a Multi-Level Agent-Based Model (ML-ABM) is used to support learners in developing Complex Systems Thinking about issues of sustainability and food production. Using Agent-Based models in complex systems education is a well-established approach. Agent-Based Models (ABMs) have been used as learning tools for teaching complexity and complex systems for decades. ABMs are particularly well suited for this task because they allow learners to break down phenomena into their constituent parts, and then to piece together the interactions between these individual parts at the micro-level into causal explanations of the phenomenon at the macro-level.

ABMs restrict the representational scope of phenomena because they always exist at one temporal and spatial scale. Recently, Multi-Level Agent-Based Modelling (ML-ABM), in which many concurrent and interdependent ABMs interact to create a larger system-of-systems has started to gain attention in modelling research. However, to our knowledge, using a multi-level model for educational purposes has not been done before. We therefore first discuss the potential educational benefits of using ML-ABMs, and present an exemplar ML-ABM learning activity.



We then present a feasibility study on data collected in two different Environmental Science AP classrooms. We look at whether students are able to make sense of these complex, multi-level systems, and whether they are able to achieve the goals set for them in two learning activities. We find that students are indeed capable of doing so, and raise questions for future research in the use of ML-ABMs for education.

Keywords

Complex Systems Thinking, NetLogo, Multi-Level Agent-Based Models, education

Multi-Level Agent-Based Models as an Object-to-Think-With for Thinking in *Many* Levels

The importance of "thinking in levels" (Wilensky and Resnick 1999) has gained increasing attention over the past two decades as we are realizing how many phenomena can be fruitfully understood as emergent, complex systems. Complex Systems are systems that distributed across many autonomously acting entities (Holland 1995; 1998; Bar-Yam 1997), and thinking in levels means being able to construct full and correct casual explanations about how interactions between these entities at the micro-level can produce observed behaviour at the aggregate, system-level (Levy and Wilensky 2008; 2009; Wilensky and Resnick 1999). Complex systems are, due to the many interacting parts and the often non-linear, non-monotonic, and feedback-driven behaviours at the system level, difficult to reason about. In the Constructionist design and research tradition, a goal in complex systems education has therefore been to develop *objects-to-think-with* that can act as external representations and reflective spaces for leaners, as they grapple with the often surprising and counter intuitive behaviour of complex systems. Agent-based models have shown to be particularly well-suited for this task across many different domains (Goldstone and Wilensky 2008; Wilensky and Jacobson 2014).

A relatively new innovation to ABMs is so-called "multi-level agent-based models" (ML-ABMs) (Morvan 2012). The theoretical and technical development of ML-ABMs has been driven by the acknowledgment that many complex systems are connected with other systems, and that traditional, 2-level ABMs for some purposes are not capable of capturing these multiple systems in sufficient details. ML-ABMs allow for multiple emergent systems to interact, and for the emergent, system-level behaviour to come about as a result of interactions at more than one micro-level. An interesting feature of ML-ABM is that adding a level to a system can fundamentally change how the system behaves, even keeping the behaviour of the original system constant. From an educational point of view, this poses a potential challenge to students, as it means they need to revise their causal understanding of the system when adding a level.

Over the past eight years, we have been developing LevelSpace (Hjorth, Head, and Wilensky 2015), a multi-level agent-based modelling extension to NetLogo (Wilensky 1999). LevelSpace allows modelers and educational designers to connect any number of NetLogo models, and to program the ways in which models affect each other (Hjorth et al. 2015). This extension has been used in ML-ABM research, but here we present our (and, we believe, *the)* first study of using ML-ABM in an educational setting. We present a design of an ML-ABM learning activity, and an analysis of students work with a model on sustainability, specifically focusing on whether students were able to manage this increased complexity.

Research Questions and Structure of Paper

This paper, we first present a structured learning activity in which students first use a two-level system, and *then* use the same system, but with one new level added to it to learn about sustainability. In starting to explore the use of ML-ABM in education, we ask a few foundational questions in this paper:

1. How can we design good learning activities on Complex Systems Thinking that utilizes the new possibilities in ML-ABM?
2. Are students able to manage the increased complexity of multiple-levels?

Setting & Sample

The study was conducted in a private, religious high school outside of Chicago in an elective, AP-level Environmental Studies Class. The school is located in a suburban, primarily white, affluent area, and the sample in the present study reflected the composition of the rest of the school. 99% of students go on to college upon graduation, 79% are white, 6.4% black, 8.3% Hispanic, and 4.0% Asian.

We collaborated with a teacher over the span of two and a half years, spending the first half year on designing and developing materials, and the next two years iteratively improving learning materials as we were able to test them in classrooms. During this period, we did two rounds of implementation, first in just one classroom, and then in two simultaneously. The data presented in this paper are from the second of two iterations, and was collected in two classrooms with a total of 42 participating students.

The ML-ABM activities presented in this paper are parts of a larger, three-week unit on "Sustainability and Food Production", designed for high school AP-level Environmental Science in the United States. In the two weeks leading up to these activities, students worked in a PartSim (Wilensky and Stroup 1999) in which they played the role of cowherds who had to manage and share a common grazing ground. (For a more complete description of this unit and other learning activities, please see Hjorth, Brady & Wilensky (2018).) Students spent about one hour on each of the two activities.

Description of the ML-ABM Model System and Learning Activity

The aim of the learning activities was to help students reason about the ways in which, and the degree to which, the causal interactions between different parts of a food production system are interconnected and interdependent. We collaborated with the teacher to design and build an ML-ABM of three interacting models relating to food production and consumption:

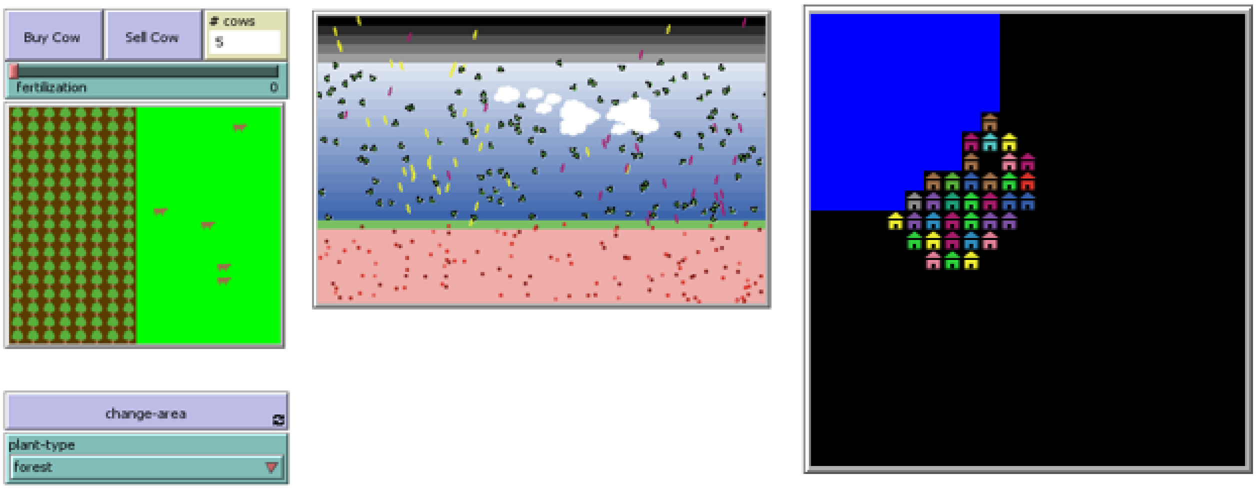


Figure 27: Screenshot of each of the three models that comprise the ML-ABM

1. Food Production Model (FPM): This model consists of an area of land on which students can plant forest or grazing grounds. Grass grows over time on each of the parts of the world that are designated grazing ground. Additionally, students can place any number of cows on the grazing grounds. These cows move around, eat the grass, metabolize energy over time, and produce milk if they are well-fed. If cows are not well-fed (i.e. if their energy level falls below a certain threshold), they produce no milk. If they run out of energy outright, they die. Finally, students can change how much fertilizer is spread on the grazing ground. Fertilizer makes grass grow back faster, making it possible to feed more cows with a smaller combined area of grazing grounds.
2. Climate Change Model (CCM): This model shows the interactions between CO2, visible and infrared light, clouds, and the surface of Earth. In brief, by modeling these interactions, the model is able to calculate temperature changes as a result of changes to CO2.
3. Human Population Model (HPM): In this model, we see a human town represented by houses. The more food this town has, the more people will move to it. Additionally, the human population reproduces. The town is located on a lake, and this lake provides water

to the human population. If the lake is polluted, there is a probability that individual members of the human population will fall ill and die. The more populated the lake is, the more probable this is.

Students are given an overview of all variables in this sytem in order to help them make sense of it.

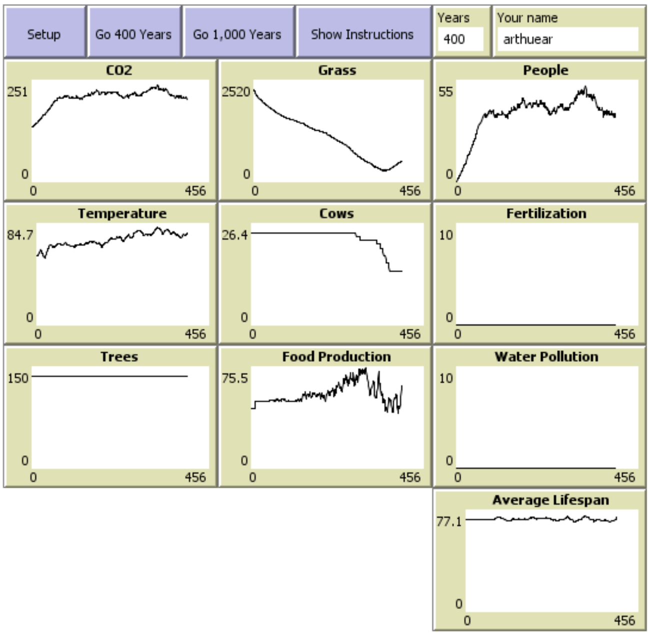


Figure 28: Graphs showing all system variables to students during simulation

In addition to the interactions *within* each model, the models also affect each other in the following ways:

1. Food is sent from the FPM to the HPM model and determines how large the human population can grow.
2. The amount of fertilizer used in the FPM will determine how polluted the lake in the HPM is.
3. Forest in the FPM ties CO2 in the CCM, so the more forest there is, the less CO2 there is, and vice versa.
4. The temperature in the CCM determines how quickly the grass grows back in the FPM. Grass has an optimal growth rate at 12C, and declines linearly as temperature differs from 12C.
5. Humans produce CO2, and the more people there are in the HPM, there more CO2 is added to the CCM.

Learning by Adding a Level: From a two- to a three-systems Model

It has always been possible to add complexity to an educational agent-based model by adding interactions between agents/entities, or by adding new ways for students to interact with or manipulate the system. What is new about ML-ABM is that one can change how a system works, by attaching it to another system, but keeping agent- and student-model interactions constant in the "original" system. Below in Figure **29**, we illustrate the difference between the 2-system and the 3-system.

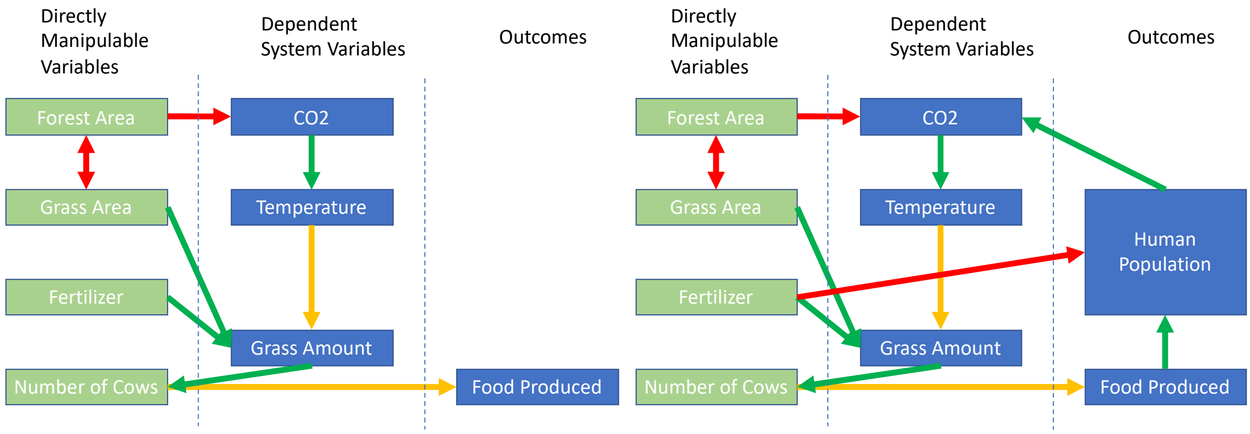


Figure 29: Causal diagram of the 2-model system (left) and the 3-model system (right)

In the figure, the left column contains variables that students can directly manipulate (in light green). The centre column shows variables in the system that students can only manipulate indirectly, and the right column shows the learning activity outcomes. The color of arrows show how variables affect each other: green means that a variable increases the other variable; red means it decreases it; and yellow means that they have a non-monotonic relationship. Importantly, these diagrams also show that adding the human population model in Activity 2 keeps intact the original system and the possible ways in which students can interact with it. It simply adds that the amount of food produced will increase the human population; the amount of fertiliser used will decrease the human population; and finally, that the human population will increase the amount of CO2.

We believe that this exemplifies a good use of ML-ABM in education. It allows students to first build an understanding of *a part* of the system, and then draw on their understanding of a part of the system to build an understanding of the system in a larger context, and with added levels.

We built two learning activities around these models, both of them aiming to scaffold students in exploring these connected systems. In the first activity, students had to optimize the amount of food produced in the FPM. In the second activity, and with the 3-model system, the task for students was to maximize the human population size. For each activity, students were given a worksheet that listed all of the causal mechanisms in the models. Students were then given approximately an hour with the first activity, and 40 minutes with the second activity. During this time, students would iteratively change variables and run the simulation to try to optimize the relevant outcome. One simulation takes around 1 minute, so students were able to run many simulations for each activity.

Managing Complexity

In order to assess the feasibility of students working with these complex, multi-level systems, we look at whether students were able to improve on the outcomes specified in each of the two activities as this, to us, would indicate that students successfully understood the system.

Classroom Level Measurements of Improvement

In activity 1, the 42 students produced a total 184 simulation runs, for an average of 4.52 per student and standard deviation of 2.16. All 42 students produced at least two runs in activity 1, and 21 produced four or more. In activity 2, this number was 130 or 3.11 on average with a standard deviation of 1.78, and 13 out of 42 students produced 4 or more runs. We spent the first of three periods and the majority of the second period on activity 1, so while it may look like student slowed down for activity 2, they actually spent less time *per* simulation. As can be seen in *Figure **30***, a plurality of students managed to run two or three runs for each of the two activities.

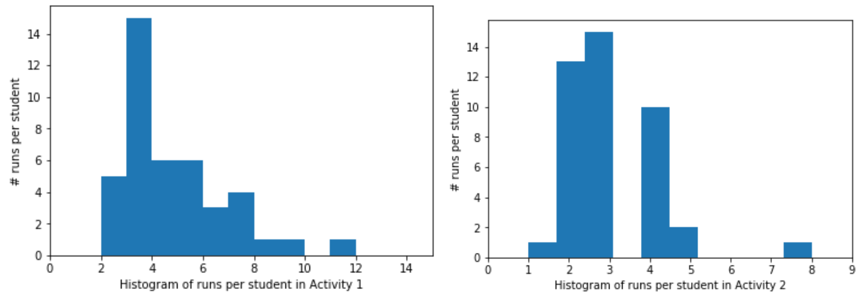


Figure 30: Model runs per student per activity

Measuring whether the class got better overall is made complicated by two interacting factors: first, students each ran a different number of runs. Second, students would sometimes “crash” their ecological system, resulting in poor outcomes. However, this is of course part of the iterative nature of the task of optimizing, and so a crash that results in a poor outcome should not necessarily be seen as indicative of the student failing, as much as indicative of the student trying to squeeze as much optimization out of the system as possible. For this reason, it is important to look at classroom level indicators of improvement both with *and* without crashes. Out of the 184 model runs in activity 1, 35 runs crashed, and 149 did not. Out of a total of 130 runs in activity 2, 53 resulted in crashes, and 77 did not. A histogram of the number of runs per student without crashes can be seen in figure *Figure **31***. We see here that if we remove all data from runs that crashed, 39 out of 42 students produced more than two runs, and 20 produced four or more runs in activity 1. For activity 2, 27 students produced more than two runs, but only seven students produced three or more runs.

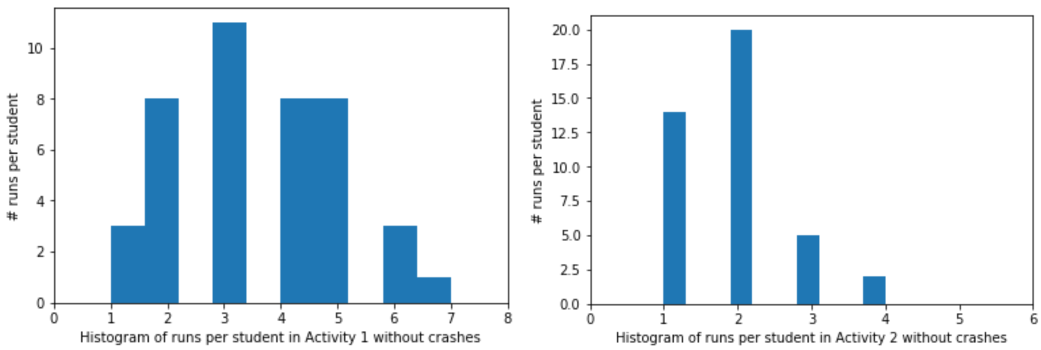


Figure 31: Number of runs per student without crashed runs

Measuring Primary Outcome in Activity 1

One way of looking for improvements at the classroom level is to look at whether students’ primary outcome got better in each of the two activities. Figure **32** shows the mean and standard deviation of the amount of food produced at the end of each simulation run at the classroom level. In the left-hand side in Figure **32** we see all simulation runs, including those in which the population of cows at some point crashed. On the right-hand side, those runs have been removed.

Removing model runs with ecological system crashes, (left in Figure **32**), we see that the classroom starts out with a lower outcome, then does better during their second simulation, and then gradually increases for the rest of the runs. While the variance does not show statistical significance, numerically we see clear improvements with students almost tripling the average outcome at the classroom level from their first to their second run. Given that 39 out of the 42 students made it through at least two runs without crashes, this is not just a small group of outliers, but shows a systemic improvement at the classroom level. There are two interesting things to note: the first one is that if we remove the runs in which students crash their system, their outcomes very clearly improve over time. The second interesting thing is that this trend is not completely monotonic over time, and while the trend is upwards, there is a drop going from students second to their third attempt. Again, this is not an outlier – 31 one students ran a 3rd simulation run without a crash, so it is interesting that we see an overall drop, and with less variance, suggesting that this may have been a point in their work on the activity where several of them tried something new that didn't quite work out.

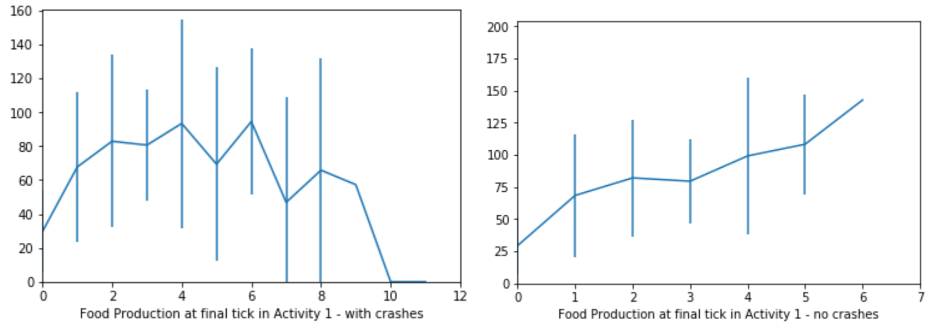


Figure 32: Activity 1 outcomes at the classroom-level. X-axis shows the run-#, and the Y-axis shows the food produced

Measuring Primary Outcome in Activity 2

In Activity 2, students were asked to optimize the population in a three-model system. *Figure **33*** shows a plot of the means and standard deviations of the population size at the end of students' simulation run.

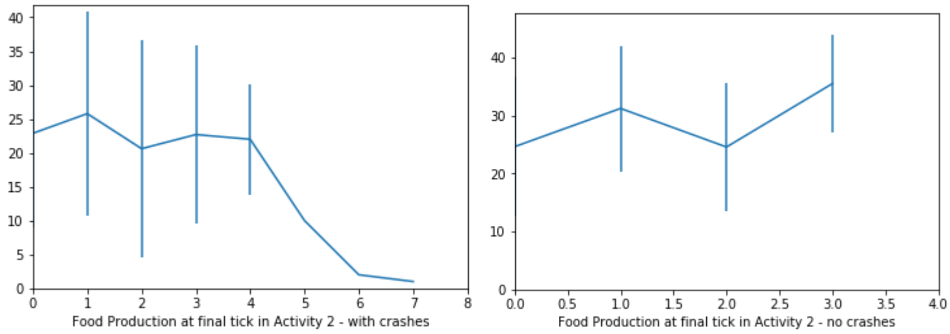


Figure 33: Activity 2 outcomes at the classroom-level. X-axis shows the run-#, and Y-axis shows the population size

Importantly, we again see a clear improvement of students outcomes when we ignore the runs that resulted in crashes. Maybe even more interestingly, we again see that there is an improvement from the first to the second run, then a drop from the second to the third run, and finally an increase again from the third to the fourth. Potentially this may suggest a more general trend of outcomes in iterative computational data optimization tasks as this one, because the iterative nature will invite cycles of failure. However, students produced fewer simulation runs in the second activity and even fewer without crashes. Only 27 out of the 42 students produced more than two runs, and only seven produced three or more. Consequently, the data from activity to may speak less to this than the data from activity 1.

Changes in the probability of a crash at the Classroom Level

To investigate the very poor outcomes in both activity 1 and activity 2 when including the crashed runs, we were curious about whether students became better or worse at preventing crashes in their systems. We therefore calculated the proportion of runs that resulted in crashes over time. The results can be seen in *Figure **34***.

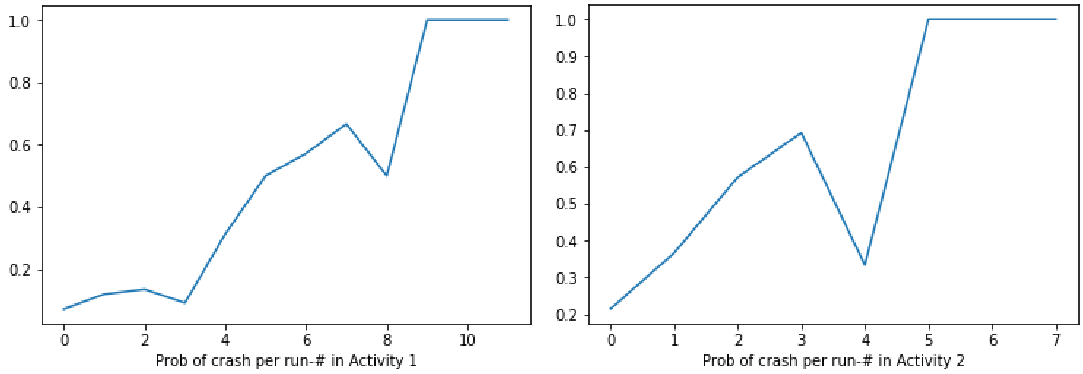


Figure 34: Probability of the system crashing, per run-#

Before extracting these data, our intuition was that students would become increasingly better at preventing crashes. However, clearly this is not the case, and this answer is maybe even more interesting: it seems that students are careful in the beginning, when they don't know the system very well, and then start to take more risks when optimizing the system.

Measurement: Quantitative Improvement for individual students

Another way to measure whether students improved on their outcomes, is to see if students at any one point ever produced a higher number in their primary outcome – food, and population, respectively - than they did in their very first attempt. This would indicate that students at least were able to improve on their initial outcome and show some understanding of the system.

In activity 1, 36 out of the 42 students (83.3%) **did improve** their outcomes. However, this includes the three students who only ran one model run, and who therefore simply could not improve them because they only had one data point. Removing those students that only produced one model run, we see that 3 out of 39 students, or 93.3% **did improve** their outcomes.

In activity 2, 20 out of 42 students or 47.6% **did improve** their outcome. The primary reason why this number is so much lower in activity 2 is because, as mentioned, a lot fewer students produced more than one simulation run that did not crash, and therefore only had one data point. Removing those students that only produced on data run, we see that 20 out of 27 students, or 74%, **did improve** their outcome. Of course, this number is still lower than we see in activity 1. But activity 2 is much more difficult because of the increasing number of interactions between human populations, CO2, and food production.

Discussion and Conclusions

In this paper we first presented an example of a ML-ABM based learning activity. We discussed how the affordance of LevelSpace to connect models into larger systems can help students make sense of complex systems in a larger context, and we showed how this idea was designed into the present learning activity. We are planning to build more of these ML-ABM based learning activities and expand on the proven record of ABMs to act as objects-to-think-with for Complex Systems Thinking education.

We then presented an analysis of student outcomes when working with the activity to assess the feasibility of these more complex, multi-level systems in education. The data presented in this paper suggest a few things. First of all, students improved on their outcomes as a whole, suggesting that students are indeed capable of working with multi-level ABMs. This was the case both in activity 1 and activity 2. Given the complexity of the system that students were asked to manipulate, this was not a straightforward task and it was not immediately obvious that they would able to do this. This indicates that the design of the simulation activity worked well as a *multi-level* object-to-think-with that supported students in constructing increasingly precise mental models of the complex system. A future, qualitative analysis of students' written responses during their work with the simulation should focus on trying to understand what parts of the system they attended to, and how they decided on their manipulations of the system that resulted in the incremental improvements that we saw. We also saw that the probability of students crashing their system went up, rather than down as one might expect. This raises the question, what is the role of crashes in students' progression through the activity?

We hope that this early work on using ML-ABMs for educational purposes will take hold. Agent-based models have shown for decades now that they help students explore complex systems, and being able to expand on this by attaching more levels is intriguing to us. We are in particular interested in using ML-ABMs' ability to invite discussions about what the boundaries of a particular phenomenon are, and to create a low-threshold, high-ceiling interface for students to build their own connected systems.

The activity in this paper showed just one approach to using ML-ABM in education: to increase complexity by adding a level in the form of a model. We believe that there are many more ways of letting learners play with and explore complexity, and our future work will focus on exploring these. We hope that others will join us in these efforts to explore other and novel uses of ML-ABMs for Constructionist learning and design.

References

Bar-Yam, Yaneer. 1997. *Dynamics of Complex Systems*. Vol. 213. Addison-Wesley Reading, MA.

Goldstone, Robert L., and Uri Wilensky. 2008. "Promoting Transfer by Grounding Complex Systems Principles." *Journal of the Learning Sciences* 17 (4): 465–516.

Hjorth, Arthur, Corey Brady, and Uri Wilensky. 2018. "Sharing Is Caring in the Commons – Students' Conceptions about Sharing and Sustainability in Social-Ecological Systems." In . Vilnius, LT.

Hjorth, Arthur, Bryan Head, Corey Brady, and Uri Wilensky. 2015. "LevelSpaceGUI: Scaffolding Novice Modelers' Inter-Model Explorations." In *Proceedings of the 14th International Conference on Interaction Design and Children*, 453–57. ACM.

Hjorth, Arthur, Bryan Head, and Uri Wilensky. 2015. *LevelSpace NetLogo Extension. Http://Ccl.Northwestern.Edu/Levelspace. Center for Connected Learning and Computer-Based Learning. Evanston, IL.* www.github.com/NetLogo/LevelSpace.

Holland, John H. 1995. *Hidden Order: How Adaptation Builds Complexity*. Reading, Mass.: Addison-Wesley.

Holland, J. 1998. *Emergence: From Chaos to Order*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA.

Levy, Sharona, and Uri Wilensky. 2008. "Inventing a 'Mid Level' to Make Ends Meet: Reasoning between the Levels of Complexity." *Cognition and Instruction* 26 (1): 1–47.

Levy, S & Wilensky, U. 2009. "Crossing Levels and Representations: The Connected Chemistry (CC1) Curriculum." *Journal of Science Education and Technology* 18 (3): 224–42.

Morvan, Gildas. 2012. "Multi-Level Agent-Based Modeling-a Literature Survey." *ArXiv Preprint ArXiv:1205.0561*.

Wilensky, U. 1999. "NetLogo: Center for Connected Learning and Computer-Based Modeling." *Northwestern University, Evanston, IL*.

Wilensky, Uri, and Michael Jacobson. 2014. "Complex Systems in the Learning Sciences." In *The Cambridge Handbook of the Learning Sciences*. Cambridge, UK: Cambridge University Press.

Wilensky, Uri, and M Resnick. 1999. "Thinking in Levels: A Dynamic Systems Approach to Making Sense of the World." *Journal of Science Education and Technology* 8 (1): 3–19.

Wilensky, Uri, and Walter Stroup. 1999. "Learning through Participatory Simulations: Network-Based Design for Systems Learning in Classrooms." In *Proceedings of the 1999 Conference on Computer Support for Collaborative Learning*, 80. International Society of the Learning Sciences.

Towards a Noisier Constructionism: Reimagining Experimental Music as Learning Context

Peter J. Woods, *pwoods2@wisc.edu*
Dept of Curriculum & Instruction, University of Wisconsin-Madison, Madison, USA

Abstract

While extant literature on constructionist approaches to music education has provided invaluable insight into the ways people learn through making musical artefacts (including instruments, recordings, etc.), this research often prioritizes certain forms of musical knowledge derived from western classical and popular music. In doing so, researchers overlook important forms of knowledge that individuals construct within other genres. To address this oversight, new research that considers constructionist practices within a broad range of musical traditions needs to occur. As an initial step into this wide-reaching intellectual project, I use this paper to address the following research question: how do noise musicians conceptualize learning and knowledge within this musical tradition? Because noise music (also known as noise) rejects the tenets of western music (i.e. rhythm, melody, repetitive structure, etc.) and utilizes a wide array of invented and repurposed instruments (Novak, 2013), noise provides an excellent site of research into alternative music education practices.

I approach the research question at the centre of this study from two angles. First, I develop a theoretical argument that intertwines constructionist learning practices within not only noise but experimental music more broadly. Placing experimental music literature and learning theory in conversation, I argue that noise music and its practitioners constantly engage a practice of redefining the boundaries of music that include socialized performance practices. This process relies on the mechanisms of sociocultural knowledge production at the centre of situated learning theory (Lave & Wenger, 1991) and constructionist approaches to learning. To this end, I argue that making noise music is an inherently constructionist practice that results in the construction of the musical boundaries that surround this tradition. Second, I analyse and present empirical data collected from interviews with the four featured artists from the 2017 Experimental Education Series, all of whom engage noise music and its surrounding scene to varying degrees. Within these interviews, participants discuss the process of knowledge construction they engage within noise music, intrinsically framing noise as a constructionist environment. Participants also discuss the types of knowledge they construct within these spaces including musical dispositions and “broken knowledge” (a deep understanding of the supposed incorrect way of doing things). This positions noise as a space in which community members produce deep, sociocultural forms of knowledge and meaning within musical contexts through an inherently communal approach to constructionist practices.

By focusing on an educational context that prioritizes communal and cultural knowledge production, this study broadens the scope of constructionist research beyond what Ames (2018) describes as “an atomized and often oppositional understanding of … learners” (p. 2) that prioritizes individualized forms of knowledge. Through this analysis, I define learning within noise as the cultural practice of developing new musical knowledges through the act of performance, a finding that situates informal music communities as a fruitful space for future research into constructionism. In doing so, this study challenges researchers to explore other educational contexts beyond technocentric educational spaces that engage similar forms of cultural knowledge production.

Keywords

Noise, Experimental Music, Free Improvisation, Informal Music Education, Situated Learning Theory

Introduction

Within extant literature on constructionist approaches to music education, scholars have largely engaged two distinct lines of research. Representing one approach, Bamberger (1991, 2013) and Gargarian (1991) examine how students develop musical knowledges within constructionist contexts. In the other (far more populated) camp, researchers unearth the kinds of knowledge students acquire when building electronic instruments and digital artefacts (see Peppler & Kafai, 2009a; Peppler & Kafai, 2009b; Rosenbaum, 2015; Milner, 2009). While these studies have provided invaluable insight, some limitations do exist. First, this research prioritizes the western musical canon, overlooking important forms of knowledge that individuals construct within other musical traditions. Second, centring on the construction of electronic instruments shifts the focus of research away from musical knowledge and towards computer-based education (e.g. how to create technologies, not music). This technocentrism within constructionist research leads to what Ames (2018) describes as “an atomized and often oppositional understanding of… learners” (p. 2) that prioritizes individualized forms of knowledge over culturally situated forms of meaning making.

In response, constructionist research into music education needs to expand beyond formal and western music education environments and investigate the kinds of knowledge created within other music traditions and social contexts. To this end, I use this study to examine noise music, a caustic subgenre of experimental music that draws from punk and industrial traditions (Bailey, 2009), through the constructionist lens. More specifically, I address the following research question: how do noise musicians conceptualize learning and knowledge within this musical tradition? Because noise music (also known as noise) emerges outside of the tenets of western music (i.e. rhythm, melody, etc.) and utilizes a wide array of invented and repurposed instruments (Novak, 2013), noise provides an excellent site of research into alternative music education practices.

To engage this research, I first develop a theoretical argument that aligns formations of noise with constructionism. I then present findings from research into the 2017 Experimental Education Series, a quarterly workshop and concert series that features noise artists from around the United States. Through this analysis, I position noise as an art form fully (yet unintentionally) intertwined with constructionism. Moreover, constructionist elements of noise foreground culturally situated knowledge and collaborative acts of meaning making, enacting a form of constructionism that sits at odds with Ames’ (2018) reading of this theory.

Theoretical Context

A Brief Introduction to Noise Music

To reframe noise as a constructionist environment, it helps to situate noise within the broader scope of experimental music. Here, I rely on Gilmore’s (2014) ideological definition of experimental music which includes all new music created outside of the organizing tenets of the western musical canon (i.e. pitch, rhythm, repetitive structure, harmony/melody, etc.). While this definition includes the mid-century American composers that have historically defined experimental music (see Nyman, 1974), this ideological definition expands the boundaries to include overlooked but interrelated traditions, e.g. free jazz (Lewis, 2002). Returning to noise, the origins of the contemporary noise scene exist within two specific music subcultures: the Japanese harsh noise scene and the European power electronics scene, both of which emerged in the late 70s and early 80s (see Bailey, 2009; Taylor, 2016). Although certain tropes have emerged from these contexts (i.e. an overwhelming use of volume, the use of dissonant electronic sound, a reliance on taboo subject matter), authors have pushed back on defining noise through these aesthetic markers (Novak, 2013; Thompson, 2017). In response, Atton (2011) contends that noise emerges discursively, a process that “entails a move away from a static categorization… and toward a continual working-through of membership, features, meaning, and evaluation” (p. 327). Within this discursive definition of the genre, fans and musicians constantly position noise at the extreme edge of music, pushing the sonic envelope as far as it can possibly go through multiple aesthetic approaches. However, the process of challenging the boundaries of music is inherently iterative since fringe ideas often become the norm through repetitive use (Novak, 2013). In turn, noise music constantly reconstructs itself through the production of new musical artefacts (albums, performances, etc) as fans discursively engage the boundary work associated with defining a genre.

Turning to performance practices, noise musicians align themselves with both the mid-century composers and free jazz musicians in multiple ways. The misuse of electronic sound-making devices, incorporation of non-musical objects, and purposeful loss of control over instruments help define both noise and the historicized tradition of experimental music as unique musical forms (Gottschalk, 2016; Keep, 2009; Novak, 2013). Regarding the loss of control, John Cage defined this practice as indeterminacy, or the purposeful inclusion of chance within compositions that allowed various actors (performers, technologies, etc.) and procedures to shape a performance in real time (Nyman, 1974). This mirrors a common practice in noise where musicians "perform their own loss of control as authoritative human subjects" (Novak, 2013, p. 159) by creating uncontrollable electronic instrumentation systems. Moreover, noise engages the practice of free improvisation, or the spontaneous and unconstrained creation of music in the moment of performance, utilized by free jazz musicians (Fischlin et al., 2013). Although theorists like Bailey (2004) contend that freely improvised music works best within collaborative settings, noise reimagines this collaborative practice in non-anthropocentric terms as performers respond to and work with partly uncontrollable instruments in the moment of performance (Novak, 2013; Thompson, 2011). These approaches to music making position noise as a perpetually evolving space, one in which performers and audiences constantly redefine what counts as "musical knowledge" in the form of both novel music technologies and theoretical understandings of music itself.

Although I have argued previously that the culturally situated process of knowledge construction described above positions noise within a situated learning theory framework (Woods, 2019), one famously defined by Lave & Wegner (1991), this does not account for the mechanisms through which individuals create knowledge or how participants conceptualize and formulate that knowledge (only that it remains culturally situated). Research into the learning practices of noise therefore needs to extend beyond situated learning theory. With this in mind, I now turn towards constructionist literature.

Towards a Constructionist Understanding of Noise

If "constructionism views learning as building relationships between old and new knowledge in interactions with others while creating artefacts of social relevance" (Kafai et al., 2009, p. 3), it follows that noise exists as a constructionist environment. The constant reinvention of music at the heart of noise emerges from the ongoing and iterative creation of homemade instruments, non-traditional performance techniques, free improvisations, and, subsequently, new musical knowledges that circulate between other cultural bodies and ways of knowing (Novak, 2013). Upitis (1990) might therefore describe noise and its associated knowledges as the outcome of musicians playing in a musical playground, developing internalized musical structures and practices through open ended engagements with instruments and musical technologies (including cultural technologies such as notation). These processes then produce artefacts of social relevance that artists, both experienced and novice, share through the informal infrastructure that surrounds noise (Bailey, 2009).

A reinterpretation of noise as constructionist environment also invokes what Turkle and Papert (1991) call bricolage, or the development of knowledge through the manipulation of and negotiation with materials. Noise establishes sound as this material, a proposition Cage (2004) implied when he defined the composer as an "organizer of sounds" (p. 27). Noise musicians then manipulate/negotiate with that material within freely improvised or indeterminant performances, a process that results in musicians "inventing the message at the same time as language" (Attali, 1985, p. 134). This contrasts with most western music traditions: instead of material sound acting as the building block of western composition, “it is the pitch relationships and their occurrence in time (that is, rhythm) that are the basic material” (Smalls, 1996, p. 25), a relational material that discourages manipulation. Despite rooting her work within a western musical context, Bamberger (1991) takes a similarly critical approach by focusing on musical phrases (not atomistic notes) as the material of music. By reframing the material beyond pitch altogether, noise builds on Bamberger’s constructionist framing by allowing performers to continually construct and reconstruct musical knowledge in the form of performance technique, musical meaning, and the boundaries of the genre or music itself (see Atton, 2011) as opposed to positioning socialized western musical knowledge as the ultimate end.

Noise also reimagines another important aspect of constructionism: the creation of “objects to think with” or “objects in which there is an intersection of cultural presence, embedded knowledge, and the possibility for personal identification” (Papert, 1980, p. 11). Gottschalk (2016) contextualizes this alignment between noise and constructionism by claiming that experimental musicians and composers constantly learn through the process of making instruments, inventing technique, and creating new works. Noise therefore moves beyond technocentric research into constructionist music education by situating the musical performance (and not just technologies) as the object to think with. Research into the BlockyTalky, a collaborative coding tool used to create electronic music, points to a similar outcome: while coding with BlockyTalky helps build computational thinking skills, researchers also found that performing with newly coded instruments produced new collaborative understandings of music (Kelly et al., 2017; Shapiro et al., 2017). Building from Attali’s (1985) theory that knowledge via noise emerges through the performative act (and not a priori), noise expands on Papert’s original formulation of objects to think with by situating both new instruments and performances as sites of reflection or launching points for new explorations, similar to what Fields et al. (2018) describe as “objects to learn with.” Through this lens, creating noise music (and not just noise making technologies) exists as a constructionist activity in and of itself.

While a theoretical connection between constructionism and noise music provides a conceptual frame to examine the learning practices of noise musicians, it does not provide evidence of the mechanisms through which individuals and communities learn and what knowledges they construct in the process. With this in mind, I now turn to empirical research to further explore the learning processes of noise musicians.

Methods

To better understand how noise musicians conceptualize learning, I interviewed the four featured artists (Sarah Hennies, Angel Marcloid, Bianca Marcia Naves, and Amanda Schoofs) from the 2017 Experimental Education Series (EES). The EES is a quarterly music series that occurs in Milwaukee, WI at the Jazz Gallery Center for the Arts. Each instalment consists of a concert and a workshop taught by the featured artist, all of whom regularly perform at noise concerts and identify as noise or experimental musicians. Interviews relied on a semi-structured approach that focused on the planning and enactment of workshops and their experiences learning about and through experimental music. I recorded and transcribed all interviews and coded this data utilizing an open and iterative approach to both descriptive and thematic coding (see Saldaña, 2015). I focused primarily on instances where the artists discussed learning both in the EES workshop and in their own development as an artist, looking specifically for moments where the artists discussed the development of specific performance techniques, their artistic practice as a whole, and their practice as a listener or how they uncovered/constructed meaning within noise music. This approach allowed me to investigate how these artists conceptualized both knowledge and learning within this context. I then used this analysis to construct what Glaser & Strauss (1967) define as a substantive theory of learning and knowledge shared between the participants in this study.

Findings

Types of Knowledge

When describing what they learned in and through noise, the participants acknowledged two distinct kinds of knowledge: dispositions and “broken” knowledge. First and most prominently, the participants in this study described the process of developing new dispositions towards both experimental music and the world at large. Schoofs described her disposition towards experimental music in relation to other vocal music traditions: “to refine the technique of, for example, operatic singing, you're excluding a lot of different types of vocal sounds… you're saying no to all of these other possibilities. And I see a lot of young artists as they become more and more specialized dismiss all those other possibilities. Not everyone obviously, but I'm really excited about people who don't dismiss those other sounds as valid.” In this sense, learning how to make noise involves developing an appreciation for sounds neglected by other traditions. The participants also discussed shifting dispositions for experimental music listeners. For Naves, this involved a heightened awareness of her sonic surroundings: “that was kind of my first introduction to it, like, ‘wow hearing that dude working on that bridge over there, that's actually a beautiful sound.’ You can create your own experimental music if you just listen to the way that sounds work together.” Developing a disposition towards experimental music largely involves a broadening of the definition of music itself, a process that emerges from the active process of constructing musical contexts through both performing and listening.

The participants also discussed acquiring what I term “broken knowledge.” This category of knowledge involved developing a detailed understanding of the *incorrect* way to utilize some skill. Schoofs engages this theme when discussing the act of singing while inhaling: “Our musculature is not designed to vocalize on an inhalation… Because of that, it's an exciting sound that I like to use in my improvisations because I don't always know exactly what will happen.” Working against the “correct” way our bodies function, Schoofs developed a new and exciting vocalization technique. Marcloid also connects with this type of knowledge when circuit bending gear, discovering sounds when her pedals are “plugged into equipment and on and in a feedback loop while I work on it. I don't think that anybody else really does that or would think to do that because it’s kind of the wrong way to do it in order to get intentional results.” This marks a distinct shift from most approaches to instrumentation in which performers look to develop a clear understanding of how the instrument works.

Knowledge Construction

Drawing a connection to constructionism, the participants in this study ssituated the development of these knowledges within one overarching mechanism: rather than foregrounding technique before performing or composing, participants constructed knowledge through the act of making music. Marcloid related to this notion, stating that “I would just try to poke around the gear that I had at any given time and try to find new ways of doing things. But not with huge foundation of knowledge on how to properly attain these things.” Schoofs also developed her artistic practice through making during her time in graduate school: “We were just performing a lot in the halls on a weekly basis, so you got to fail a lot. You can't learn how to improvise well unless you've had a lot of failures and you understand what it means to not do something very well or make a poor choice. What that means to you personally, what that means to a group dynamic, and then how do you recover.” Looking beyond her own practice, Naves articulates this sentiment when she says, “the local experimental music artists around me, I've seen them evolve immensely by just doing it.” In all cases, the participants position the act of making music (and sound more broadly) as the mechanism through which they developed their own personal knowledge and practice.

In discussing this process of crafting knowledge through making, the artists brought up two somewhat contradictory origin points in that construction. First, all of the participants discussed their development as artists in relation to some outside skill development: Marcloid was an accomplished musician in various metal and punk subgenres; Naves started in theatre and make up design; Hennies discovered experimental music after playing drums in various rock bands; and Schoofs had trained classically as an operatic singer. All participants then relied on these backgrounds when developing new forms of experimental music. For example, Marcloid discusses how her ability to play guitar in a different tradition allows her to create interesting free jazz: “Because [I’m] really only just getting the very tip of jazz… I don't know what to do. So that makes my jazz really different from the free jazz that someone who knows jazz theory really well.” This experience making unique sounding free jazz with tools outside of the jazz realm indicates the value of crafting new knowledge with a disparate skill set.

The participants also discussed the value of creating something from a complete lack of skills or knowledge. Hennies discussed this concept in relation to a collaborative piece she performed at the EES, claiming that “the ideas in the piece are what dictated the material, which I think is why it came out so strange and unusual. We were very consciously not trying to play a style or to do something that was necessarily connected to anything else that we did.” Through this process, Hennies avoided relying on familiar musical knowledges to craft a new and unique performance. Marcloid addresses this sentiment even more directly: “Sometimes when you don't have a foundation of knowledge in something you may not necessarily get better results, but they're different because that foundation is not there. I think that's really exciting.” Without adding a value judgement, Marcloid acknowledges the unique knowledges that materialize from engaging foreign practices.

Communal Approaches to Learning and Knowledge

Shifting away from the individualist conceptions of learning, both Naves and Schoofs discussed the distributed knowledge construction they encountered between community members and performers. Naves in particular conceptualized her work through her creative community: “We have an interest in generally the same types of things. We find ourselves working together on projects that fit into this kind of experimental realm and we've developed a rapport where we know everyone's role.” This understanding of her community and how she fits into it has allowed her and her colleagues to create a number of collaborative works she could not have done on her own. As for Schoofs, this communal knowledge manifests itself during the moment of free improvisation. “when you're playing with other people it's amazing how people think very differently than one does as an individual. If I'm vocalizing with another person, they might do something that I didn't expect at all and that triggers me to change my sound… there's the exciting element that they have their own unique ideas that they are bringing to the table. And you can learn a lot from that process.” Fischlin et al. (2013) connect to this description when they describe free improvisation as a space to construct knowledge between performers, some of whom meet for the first time on stage, through the act of spontaneously creating new music and musical languages. This positions certain forms of culturally situated knowledge within noise as a communally developed product, one that forms beyond the scope of a single individual.

However, Hennies disagreed with the importance of this sentiment, stating that she often feels “on the outside” of any music community. However, this disagreement played out in an interesting fashion. Hennies claimed that “the result [of being an outsider] is that I can go play at a university or a music school or I can go play at the noise fest. And… when you go to [a noise festival] and play that piece that I played in Milwaukee surrounded by ten different sets of harsh noise... all of a sudden you're like ‘Wait, where are we now?’ Because now all of a sudden the festival is different than you thought it was.” While Hennies may not employ a communal approach to developing her own work, the juxtaposition of her work within a broader community of different artists challenges preconceived notions of the immediate musical environment of the festival. Since this understanding emerges from the relationship between her music and others, musical knowledge still forms through communal interactions.

Discussion

Mirroring the argument proposed in the theoretical context, the overwhelming focus on constructing knowledge through a process of making musical artefacts (i.e. performances) as a mechanism for learning firmly situates noise (and experimental music more broadly) within the context constructionism. Especially considering Marcloid's assertion that specific knowledges about noise emerge through engaging in practices one knows nothing about, I would argue that educators and researchers cannot separate experimental musical traditions (including noise) from a constructionist understanding of music creation. Since the construction of new artefacts acts as the dominant means towards creating new forms of socialized knowledge, making noise music inherently and tangibly enacts the theoretical foundations of constructionism described by Papert (1980; 1993). Moreover, the reliance on non-noise approaches to making music or performing connect to Kafai et al.'s (2009) assertion that "constructionism views learning as building relationships between old and new knowledge" (p. 3). While participants may have developed new dispositions while making noise and experimental music, these dispositions did not emerge out of thin air. Instead, they generated in part through the circulation of artefacts between cultural bodies described by Novak (2013) in his definition of noise. In turn, this process of relationship building between old and new knowledges contained within different cultural bodies allowed these artists to reclaim "broken" knowledges and reengage them as meaningful parts of a newly invented language, to use Attali's (1985) phrasing.

Moreover, the conceptualization of communal knowledge construction proposed by participants responds to Ames (2018) critique of constructionism that frames this approach as an atomized or individualistic form of learning. When Schoofs describes her experience of developing new knowledge in the act of performing with other people (one in which new ideas come from the interaction between freely improvising performers), musicians distribute not only knowledge but the act of learning across the community of performers. For both Attali (1985) and Fischlin et al. (2013), this communal knowledge construction sits at the heart of noise and free improvisation. Similar to students who "construct theories by arranging and rearranging, by negotiating and renegotiating with a set of well-known materials" (Turkle & Papert, 1991, p. 136) through bricolage, noise musicians do the same by working with sound in inherently collaborative contexts. Whether this involves a collaboration between artists (as described by Naves and Schoofs) or in the context of negotiating meaning with audiences (like Hennies describes in her experience at the noise festival), engaging bricolage within noise exists as a communal and situated process. In turn, noise positions constructionism in opposition to the individualized formation proposed by Ames (2018) and defines this learning theory as an inherently communal process.

Conclusion

If learning "happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it's a sandcastle on the beach or a theory of the universe" (Papert & Harel, 1991, p. 1) within constructionist learning environments, it behoves education researchers to investigate not only the sandcastles but the theoretical constructions as well. This becomes especially important when considering Ames' (2018) critique of the intellectual history of constructionism, one that ignores cultural forms of knowing in favour of individualistic approaches to knowledge construction. In this paper, I position noise music as an especially fruitful site of research because of its intrinsic reliance on the reinvention of culturally situated musical knowledges through collaborative, performative acts. In doing so, this study challenges education researchers to expand beyond formal, technocentric educational contexts to further understand the scope of constructionism. Noise music provides one example of a learning context that not only embraces but relies on the construction and sharing of theories of the (musical) universe as a cultural practice. However, that noise music in particular fits the constructionist model so well should not be interpreted as an implication that other forms of music do not fit the model as well or better. Instead, I position this analysis as a call for similar research into other musical traditions that challenge the overly socialized forms of musical knowledge that sit at the heart of the western musical canon. By exploring other spaces that replicate this process, scholars can fully realize the promise of constructionism across all cultural contexts.

References

Ames, M. G. (2018). Hackers, Computers, and Cooperation: A Critical History of Logo and Constructionist Learning. *Proceedings of the ACM on Human-Computer Interaction*, *2*(CSCW). https://doi.org/10.1145/3274287

Attali, J. (1985). *Noise: The Political Economy of Music*. University Of Minnesota Press.

Atton, C. (2011). Fan Discourse and the Construction of Noise Music as a Genre. *Journal of Popular Music Studies*, *23*(3), 324–342. https://doi.org/10.1111/j.1533-1598.2011.01296.x

Bailey, D. (2004). Free Improvisation. In C. Cox & D. Warner (Eds.), *Audio Culture: Readings in Modern Music* (pp. 255–265). New York: Continuum.

Bailey, T. B. W. (2012). *MicroBionic: Radical Electronic Music and Sound Art in the 21st Century*. Belsona Books Ltd.

Bamberger, J. (2013). *Discovering the musical mind: A view of creativity as learning*. New York: Oxford University Press.

Bamberger, J. (1991). *The mind behind the musical ear: How children develop musical intelligence*. Cambridge, MA: Harvard University Press.

Cage, J. (2004). The Future of Music: Credo. In C. Cox & D. Warner (Eds.), *Audio Culture: Readings in Modern Music* (pp. 25–28). New York: Continuum.

Fields, D. A., Shaw, M. S., & Kafai, Y. B. (2018). Personal Learning Journeys: Reflective Portfolios as "Objects-to-Learn-With" in an E-textiles High School Class. *Constructionism 2018 Constructionism, Computational Thinking and Educational Innovation, Conference Proceedings*, 213–223.

Fischlin, D., Heble, A., & Lipsitz, G. (2013). *The fierce urgency of now: Improvisation, rights, and the ethics of cocreation*. Durham: Duke University Press.

Gargarian, G. (1991). Towards a constructionist musicology. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 311–333). Ablex Publishing Company.

Gilmore, B. (2014). Five Maps of the Experimental World. In D. Crispin & B. Gilmore (Eds.), *Artistic Experimentation in Music: An Anthology* (pp. 23–30). Retrieved from https://muse.jhu.edu/book/39256

Glaser, B. G., & Strauss, A. L. (1967). *The discovery of grounded theory; strategies for qualitative research*. Chicago: Aldine PubCo.

Gottschalk, J. (2016). *Experimental Music Since 1970*. New York: Bloomsbury Academic.

Kafai, Y. B., Peppler, K. A., & Chapman, R. N. (2009). The Computer Clubhouse: A Place for Youth. In Y. Kafai, K. A. Peppler, & R. N. Chapman (Eds.), *The Computer Clubhouse: Constructionism and Creativity in Youth Communities* (pp. 1–14). New York: Teachers College Press.

Keep, A. (2009). Instrumentalizing: Approaches to Improvising with Sounding Objects in Experimental Music. In J. Saunders (Ed.), *The Ashgate Research Companion to Experimental Music* (pp. 113–130). Burlington, VT: Routledge.

Kelly, A., Finch, L., Bolles, M., & Shapiro, R. B. (2018). BlockyTalky: New programmable tools to enable students' learning networks. *International Journal of Child-Computer Interaction*, *18*, 8-18.

Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge England ; New York: Cambridge University Press.

Lewis, G. E. (2002). Improvised Music after 1950: Afrological and Eurological Perspectives. *Black Music Research Journal*, *22*, 215–246. https://doi.org/10.2307/1519950

Milner, A. (2009). Interface Design with Hook-ups. In Y. B. Kafai, K. A. Peppler, & R. N. Chapman (Eds.), *The Computer Clubhouse: Constructionism and Creativity in Youth Communities* (pp. 58–70). New York: Teachers College Press.

Novak, D. (2013). *Japanoise: Music at the Edge of Circulation*. Durham: Duke University Press Books.

Nyman, M. (1974). *Experimental Music: Cage and Beyond*. New York: Cambridge University Press.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. New York: BasicBooks.

Papert, S., & Harel, I. (1991). Situating constructionism. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 1–11). Ablex Publishing Company.

Peppler, K., & Kafai, Y. (2009a). Making Games, Art, and Animations with Scratch. In Y. B. Kafai, K. A. Peppler, & R. N. Chapman (Eds.), *The Computer Clubhouse: Constructionism and Creativity in Youth Communities* (pp. 47–57). New York: Teachers College Press.

Peppler, K., & Kafai, Y. (2009b). Youth Video Productions of Dance Performances. In Y. B. Kafai, K. A. Peppler, & R. N. Chapman (Eds.), *The Computer Clubhouse: Constructionism and Creativity in Youth Communities* (pp. 71–80). New York: Teachers College Press.

Rosenbaum, E. (2015). *Explorations in musical tinkering* (Thesis, Massachusetts Institute of Technology). Retrieved from http://dspace.mit.edu/handle/1721.1/97970

Saldaña, J. (2015). *The Coding Manual for Qualitative Researchers* (3rd edition). Los Angeles ; London: Sage Publications Ltd.

Shapiro, R. B., Kelly, A., Ahrens, M., Johnson, B., Politi, H., & Fiebrink, R. (2017). Tangible Distributed Computer Music for Youth. *Computer Music Journal*, *41*(2), 52–68. https://doi.org/10.1162/COMJ_a_00420

Small, C. (1996). *Music, Society, Education* (Revised edition). Hanover, NH: Wesleyan.

Taylor, P. (2016). The Genesis of Power Electronics in the UK. In J. Wallis (Ed.), *Fight Your Own War: Power Electronics and Noise Culture* (pp. 187–198). Headpress.

Thompson, M. (2011, June). *Voicing the cyborg: The political potential of noise music*. Presented at the International Association for the Study of Popular Music Postgraduate Conference, University of Liverpool. Retrieved from http://eprints.lincoln.ac.uk/16858/

Thompson, M. (2017). *Beyond Unwanted Sound: Noise, Affect and Aesthetic Moralism*. New York: Bloomsbury Academic.

Turkle, S., & Papert, S. (1990). Epistemological Pluralism: Styles and Voices within the Computer Culture. *Signs: Journal of Women in Culture and Society*, *16*(1), 128–57.

Upitis, R. (2018). *This too is music*. Oxford University Press, USA.

Woods, P. J. (2019). Conceptions of teaching in and through noise: A study of experimental musicians' beliefs. *Music Education Research*, *21*(4), 459–468. https://doi.org/10.1080/14613808.2019.1611753

Towards a Robotics Self-Efficacy Test in Educational Robotics

Georg Jäggle, *jaeggle@acin.tuwien.ac.at*
Institute of Automation and Control, Vienna University of Technology, Vienna, Austria

Lara Lammer, *lara.lammer@tuwien.ac.at*
Institute of Automation and Control, Vienna University of Technology, Vienna, Austria

Jan-Ove Wiesner, *jan-ove.wiesner@tuwien.ac.at*
Institute of Automation and Control, Vienna University of Technology, Vienna, Austria,

Markus Vincze, *markus.vincze@tuwien.ac.at*
Institute of Automation and Control, Vienna University of Technology, Vienna, Austria,

Abstract

Educational robotics (ER) is a pedagogical tool that evokes the interest and motivation of students in STEM fields. Perceived self-efficacy is one factor to increase the interest of children in STEM fields. However, there are no evaluation tools to prove the transformation of self-efficacy during educational robotics activities. The paper aims to show the first step of developing a Robotics Self-Efficacy (RSE) test with a pilot study in a robotics summer camp (RSC) with the title "Robots as we children wish them to be". The research question is "How reliable is the Robotics Self-Efficacy Test?". This paper describes the robotics summer camp that was held over five days with 18 Activity Blocks and 30 participants from 7 to13 years old. The RSC is part of the outreach concept of the university to support technical literacy. In addition, we extended this concept to promote also scientific literacy and self-efficacy. We conducted a mixed methods evaluation of the RSC with quantitative questionnaires and a qualitative analysis of documents and artefacts the children produced during the RSC. All participants filled out a robotics self-efficacy questionnaire at the beginning and the end of the RSC. The internal consistency of the Robotics Self-Efficacy questionnaire was found to be highly reliable (10 items: $\alpha = .842$). The correlation of the means with the values $r=0.59$ and $p<.01$ shows a significant positive influence on Robotics Self-Efficacy. For an in-depth analysis two participants, with highest RSE Score and lowest RSE Score, were chosen and their documents and artefacts were analyzed and compared with their RSE Score. The comparison shows a relation between the RSE Score and their documents and artefacts.

Keywords

Scientific Literacy, Robotics Self-Efficacy-Test, Constructionism Approach, Technological Literacy, Innovative Evaluation Tool

Introduction

The Outreach Team from the Institute of Automation and Control developed a Robotics Summer Camp (RSC) for one week for young children from 7 to 13 years old. The search about robotics camp offerings in Austria showed that most of them implemented activities of constructing and programming robots with commercial robotic sets. Our goal was to develop a summer camp on our scientific approach, which is based on constructionism, and give children the chance to discover the world of robotics as scientists in one week. The goals of the concept included engaging young children in the world of robotics that they will take an active part with hands-on activities in designing a robot, to foster scientific and technological literacy and to increase their perceived self-efficacy in robotics. On the one hand is it necessary to increase and assess the self-efficacy, because it is a factor about the belief in own capabilities to organize and execute courses of action required to produce given attainments. (Bandura, A., 1997). On the other side is the evaluation and fostering of self-efficacy in educational robotics an extension to empowering students in problem-solving in the robotic field and to increase the interest in STEM. Therefore, self-efficacy measures the assessment of robotic activities, which could improve the understanding between impact beliefs of educational robotics and student performance in act. We developed a Robotics Self-Efficacy (RSE) questionnaire with 10 items and 5 scales to measurement the self-efficacy in educational robotics activities. The whole evaluation design was based on mixed methods with quantitative and qualitative analysis. The quantitative method was conducted with questionnaires at the beginning and at the end of the summer camp to measure and assess. The qualitative method was a document analysis with the scientist diary (working sheets).

Literature Review

This section defines the constructionist approach, scientific literacy and perceived self-efficacy in the context of educational robotics and studied in the robotics summer camp.

Constructionist Approach

Papert (1980) defines constructionism as following:

> *“Constructionism—the N word as opposed to the V word— shares constructivism’s view of learning as ‚building knowledge structures’ through progressive internalization of actions… It then adds the idea that this happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it’s a sand castle on the beach or a theory of the universe.”*

The constructionist approach focuses on the art of learning on the importance of constructing artefacts in learning. The essential priority is on learner’s conversation with others about the artefact, and how these dialogues facilitate new knowledge building. This is the reason why tools (e.g. Educational robotics) and media are important to influence human development. The knowledge is actively constructed by the children in interaction with their world, for this we are tempted to offer opportunities for children to engage in hands-on explorations that fuel the constructive process to get a personal experience. (Ackermann, 2001).

A further and optimized understanding for our educational robotics context comes from Kafai and Resnick (2012), who builds his constructionism approach on the constructivist theories of Jean Piaget, claiming that knowledge is not simply transferred from the teacher to the student, but actively constructed by the learner's mind. Children do not get ideas, they make ideas. In addition, constructivism suggests that learners are more likely to develop new ideas if they are actively involved in producing a type of external artefact that they can think about and share with others.

Our RSC design is based on this concept. Children receive a chance to develop their own ideas and to construct their own robotic artefacts with the 5 step-plan (Lammer et al., 2015). They write about their experiences in their scientist diary and share their ideas in a presentation during the RSC and at the very end of the RSC in a presentation in front of a plenum of parents.

In our case, we bring children of different backgrounds and interests in the context of constructionism, so that they are more open and likely to engage with others and with situations. They enjoy discovering and learn from personal experience rather than from being told. They enjoy gaining understanding from singular cases and like to be engaged in situations. The children travel for that through the world of robotic summer camp, by adopting different perspectives and beginning a dialogue with initially incompatible experiences. This dialogue is shown in the learning artifacts (working sheets).

Scientific Literacy

The term "scientific literacy" has become increasingly prominent in discussions of the aims and purposes of school science education. The 1990 UNESCO World Conference on Education for All argued that science education should promote *"a world community of scientifically and technologically literate citizens"* (Millar, 2006).

In a sense, literacy means the skills to read and write. In the other sense, literacy means knowledge transfer, learning, and education. The two senses are interconnected. A person can be knowledgeable without being able to read and write: individuals can learn a lot through trial and error, word of mouth and apprenticeship (Norris and Phillips, 2003).

It has been established that learning science in context can improve students' scientific literacy as they come to an understanding of science, the way that it progresses and the interplay between it and their everyday lives. Thus, the strategy for the RSC to foster scientific literacy is learning science in context. Additionally, (Palincsar et al., 1993) found that collaborative problem-solving was effective in the promotion of scientific literacy. Students' naive views of science shifted to more scientific views after the students were provided with contextual problems.

Scientific Literacy implies that a person can inquire, discover or decide answers to questions determined by interest. It implies that such an individual has the capacity to depict, clarify and foresee characteristic marvels (Archer-Bradshaw, 2017).

Nine research questions guided the children through the camp. The different Activity Blocks fostered the children to develop scientific literacy and to answer these research questions with their own experiences, experiments and observations.

Technological Literacy

Technological literacy is far more than being able to use technology in everyday life. Thus, we have adopted the definition of the ITEEA:

> *„Technological literacy is the ability to use, manage, assess, and understand technology. A technologically literate person understands, in increasingly sophisticated ways that evolve over time, what technology is, how it is created, and how it shapes society, and turn is shaped by society."* (ITEEA, 2000).

The Standards for Technological Literacy emphasize the importance of understanding the basic elements that go into any technology. One of these elements is the design process to create solutions to problems by transforming the solution into a finished product. Besides understanding the development and use of particular technologies, students should also be able to evaluate their effects on society, the environment, and also other technologies.

The children are guided through camp to find a robotic solution for a character from a famous children's book, which they first develop and draw, and then create in form of a low-tech prototype, while also evaluating if the first desired solution the character is asking for is a good one, and if there can be other solutions that solve the underlying problem better.

Self-Efficacy

Rittmayer and Beier (2008) explained self-efficacy *"….as a significant predictor of both the level of motivation for a task and task performance."*

For that, we will develop an evaluation tool to measure the self-efficacy in the context of tasks and activities in robotic topics and learning through educational robotics, which we call Robotics Self-Efficacy.

We use the self-efficacy test because it is usually a better predictor of task-specific performance than the self-concept (Bong and Skaalvik, 2003). It is believed that self-efficacy influences task fulfillment through goal setting and self-regulation during accomplishment (Bandura, 1991).

In addition, levels of self-efficacy is associated with greater self-regulation, efficient problem-solving strategies and better management of working time (Zimmerman, 2000).

Wood and Bandura (1989) stated that "self-efficacy refers to beliefs in one`s capabilities to mobilize the motivation, cognitive resources, and courses of action needed to meet given situational demands."

Self-efficacy beliefs are an important aspect of human motivation and behaviour as action that can affect one's life. Regarding self-efficacy, Bandura (1995) explains that it "refers to beliefs in one's capabilities to organize and execute the courses of action required to manage prospective situations". More simply, self-efficacy is what an individual believes he or she can accomplish using his or her skills under certain circumstances (Edwards et al., 2007).

The basic principle of self-efficacy theory are more likely to engage in activities for which they have high self-efficacy, and less likely to be those who do not (van der Bijl and Shortridge-Baggett, 2001).

This means children with high Robotics Self-Efficacy set more challenging goals and work harder and more efficient to accomplish goals related to robotics (e.g. building a robot prototype) than students with low Robotics Self-Efficacy.

Bandura (1977) referred four sources of information that children help to judge their efficacy. They are performance outcomes, vicarious experiences, verbal persuasion, and physiological feedback. Table 4 is developed related to Rittmayer and Beier (2008).

Sources	**Description**	**Strategies for RSC**	**Mission for RSC**
Performance Outcomes/Mastery Experience	Mastery experiences are openings to memorize and hone the rules and methodologies essential to perform successfully.	Many STEM teachers or tutors have mastery experiences, such as lab work, experiments, design projects, and other applied activities that are part of the curriculum and hands-on activities.	The RSC incorporated hands-on activities in several Activity Blocks
Vicarious Experience	Rolemodels are particularly formative when they are perceived as similar to the viewer, suggesting that interaction with female tutors and advanced students in STEM would positively impact the self-efficacy of female STEM students.	Invite more STEM tutors and STEM professionals into classrooms to collaborate with students (for example, solve math problems or conduct a science experiment) or share their STEM experiences and experience and achievements.	Rolemodels lead the RSC. The team introduce themselves and their different technological background. The background of the team is interdisciplinary, mixed gender and cross generation.
Verbal/Social Persuasion	Positive feedback and encouragement, especially from tutors, teachers or parents, increase self-efficacy.	Provide feedback and support that is positive but genuine, appropriate, and realistic. Give students and their families with information about extra-curricular STEM activities, such as after-school clubs, camps, local	Tutors gave positive feedback. The children presented their results on the end of the summer camp in front of their parents plenum.

		lectures and exhibits, and encourage them to participate.	
Physiological Reaction	A person deciphers his or her enthusiastic and physical states to decide his or her self-efficacy convictions.	Feeling calm and composed, instead of anxious and stressed, when planning for and performing an assignment leads to higher self-efficacy.	Empower students to go to completely to errand at hand, which ought to diminish consideration paid to misgivings and fears subsequently decreasing task-related less common.

Table 4: Four Sources of self-efficacy in the RSC-Context (Rittmayer, 2008)

Research Questions

RQ1: How does the robotics summer camp influence the Robotics Self-Efficacy (RSE) of the students?

RQ2: Is the RSE-Test a statistically reliable analysis tool for measuring self-efficacy in educational robotics contexts?

RQ3: What has to be modified at the RSE-Test?

Robotics Summer Camp Concept

The robotics summer camp concept - a five-days comprehensive and coherent curriculum based on the approach of constructionism - was designed for young learners to increase their technological and scientific literacy as well as their perceived self-efficacy regarding robotics. We already analyzed educational robotics as a tool for increasing technological literacy (Jäggle et al., 2020). In this paper, we focus on the influence of a week-long intervention in an educational robotics context on the perceived self-efficacy of the children regarding robotics.

Overview of the Activity Blocks

The robotics summer camp lasted from 9 a.m. to 4 p.m. from Monday to Thursday, and from 9 a.m. to approximately 3 p.m. on Friday. It was designed for 7-13-year-olds. It consisted of 18 connected Activity Blocks. These activity blocks were placed in the context of the story of the Little Prince by Antoine de Saint-Exupéry (2010). The story is about a prince and his exploration journey as well as his relationship to a rose on his planet.

TIME	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
9:00-9:50	Daily Morning Circle AB1	Daily Morning Circle AB1	Daily Morning Circle AB1	Daily Morning Circle AB1	Daily Morning Circle AB1
9:50-10:50	Research diary AB2	Design Thinking AB6	Prototyping AB9	Presentation of Artefacts AB11a	Human-Robot Interaction AB14
11:05-12:05	Thymio / Humanoid robot pepper AB3	Inventors Lab AB7	Prototyping AB9	Propeller AB12	Presentation of Artefacts AB11a

12:05-13:15	Lunch break	Lunch break	Lunch break	Lunch break	Lunch break
13:15-14:45	Urban Scavenger hunt AB4	Scientist Interviews AB8	Robots in Applications AB10	Problem-Based Learning AB13	Plenum Presentation AB11b
15:05-16:00	Reflection 1 AB5a	Reflection 2 AB5b	Reflection 3 AB5c	Reflection 4 AB5d	

Table 5: The time schedule of the robotics summer camp

The aim was to let the children slip into the role of a scientist and find out using scientific methods, what technology is, what people need it for, what robots are, how they work and in which ways they can be used or be harmful to our society.

To investigate questions, make hypotheses and test them on their own, they were placed in the child-friendly context of the story of the Little Prince by Antoine de Saint-Exupéry.

In the activity-blocks, they dealt with the following nine questions, which were mostly derived from the "Standards for Technological Literacy: Content for the study of Technology" (Dugger Jr, 2001) seen in Table 6.

Q1: "What are the individual parts of a robot?"
Q2: "What is technology? What is nature?"
Q3: "What are the characteristics of a researcher?"
Q4: "What is good about robots, what is bad?"
Q5: "What are the characteristics of an inventor?"
Q6: "What problems does the rose have?"
Q7: "As a robot researcher, how can you help solve the problems? What ideas do you have?
Q8: "Find examples for tools or technology, where good comes out and where bad comes out.
Q9: "According to which plan do you design a robot?

Table 6: Questions related to Technological Literacy

The children investigated questions, made hypotheses and tested them on their own. Moreover, they had to apply their newly acquired knowledge directly in projects that ended with the design of their robot prototypes.

Activity Blocks (AB)

The RSC had different Activity Blocks (AB) with different activities. This section explains the different Activity Blocks and which Activity Block helped to answer the questions (Q) related to the technological literacy standards.

Title	Description	Questions	Shortcut
Daily Morning Circle	The children have received a letter in response to the letter they have written one day before. This process helped them to recall their experiences of the last day and prepares them for the current day. Their thought was discussed with the tutors.	Q3, Q4, Q5, Q6, Q7, Q8, Q9	AB1
Research diary	The children got to know each other and became familiar with the concept of documenting observations in a research diary, as their role models for the week, the researchers at the Vienna University of Technology would do. For a better identification with their personal research diary, they personalised the front paper with drawings.	Q3	AB2
Thymio / Humanoid robot pepper	The children explored the behaviour of the Thymio robot in each of the programs and have identified the necessary sensors and actuators for this purpose. Afterwards the humanoid robot Pepper was presented to them, and again they identified how its sensors and actuators led to its behaviour. They recorded their observations and findings in their research diaries.	Q1	AB3
Urban Scavenger hunt	The children searched in the local park for clues (e.g. dates, objects, words, symbols) that guide them to different stations. At each station, they got familiar with objects that are either classified as nature or technology to learn the distinction between the two of them.	Q2, Q3	AB4
Reflection 1	The children wrote a letter to the fictive character of the rose and told it about their experiences, feelings and thoughts about the first day in the RSC.	Q1, Q2	AB5a
Design Thinking	The children thought themselves into the problems of the rose and came up with solutions that would solve these problems. They sketched a robot that would solve this problem.	Q5, Q6, Q7	AB6
Inventors Lab	The children visited a lab where they were familiarised with the concept of an iterative design process, and they were shown several prototypes of a propeller that would let to the product that they would use at a later date in this robotics summer camp.	Q1, Q5	AB7
Scientist Interviews	The children interviewed passengers on the street about their view of robotics and its applications, noted the answer and drew their conclusion about the passenger's views and their needs for robot applications.	Q3, Q4	AB8
Reflection 2	The children reflected the day by drawing a picture of a robot that would solve the problems of the rose and contained all their knowledge about building	Q1, Q9	AB5b

	robots they have collected over the day. This activity happened based on the 5-step plan.		
Prototyping	The children built a low-fidelity robotic prototype and used their Design Thinking Skills and materials like cardboard, duct tape, strings, wire and LED lights.	Q1, Q7, Q9	AB11b
Robots in Applications	The children saw some real robotic applications at our Automation and Control Institute.	Q4, Q8	AB10
Reflection 3	The children reflected about what it is like to be a scientist.	Q5, Q8	AB5c
Presentation of Artefacts	The children learned about the importance of communicating their insights and results to others and invented a presentation for their prototypes.	Q7, Q9	AB11a
Propeller	The children received a presentation of a 3D printing program. After that, they observed the 3D printer working. When it was finished, they glued the components together and made experience with their propellers. It demonstrated how an idea comes from a model into a real product.	Q2	AB12
Problem-based Learning	The children had to solve the problem of transporting a little figure from one point to another point along a five-meter line. For that instance, they were subdivided into developer teams and invented different types of drives. They built their ideas and made the figure move along the line.	Q5	AB13
Reflection 4	The children completed the sentences “I liked…”, “I want to indicate…”, “I did not like…”, “I want to remember…”, “I miss out…”	Q6	AB5d
Human-Robot Interaction	Finally, the children learned about the behavioural design of the humanoid robot pepper. The reasons for the specific design solutions of the robots behavioural and physical design were discussed, and they thought about how a robot should interact with living beings and especially humans	Q4	AB14
Plenum Presentation	Finally, the children presented their prototypes and functions in several use-cases in form of short product presentations or stories in front of a plenum of parents.	Q7, Q9	AB11b

Table 7: The description of the 18 Activity Blocks (AB)

Research Design

The robotics summer camp was evaluated using a mixed method design. This evaluation is a combination of a quantitative method with PRE- and POST Questionnaire and a qualitative method with document analysis on two cases.

The questionnaires measured the changing of the self-efficacy during the robotics summer camp. To measure the influence of the robotics summer camp (RSC) the children got a PRE-Questionnaire before the RSC and a POST questionnaire on Thursday afternoon after their robot prototypes were finished. The qualitative method supported the significance of the questionnaires. The first step was to analyse the statistics about the significance and reliability of the questionnaire. After that, a case study compared the documents and artefacts of two students with the highest and lowest score of RSE-Test. This depth analysis proves the relation of results between the questionnaires and the documents and artefacts.

Robotics Self-Efficacy-Test

Self-Efficacy has an important impact to increase the interest in STEM. The question is if a week long activities in an educational robotics context based on constructionism also has an influence on the self-efficacy of students. First pilot-phase was with the standard self-efficacy-test, with 10 items and 4 scales from Schwarzer and Jerusalem (1999) with 32 students. The result shows that the self-efficacy increase during robotic activities but not significant. (Jäggle et al., 2019) The next step was to develop a self-efficacy test in context of robotic activities and not in general. This test was developed related to (Beierlein et al., 2012). Beierlein (2012) extended the self-efficacy test from Jerusalem and Schwarz (1999) from 4 scales to 5 scales, because the re-analysis showed that the four-step answer scale proposed by Jerusalem and Schwarzer (1999) was accompanied by a lack of differentiability of the answers at the upper end of the scale (Bandura, 1997). To address this problem, a five-step response scale was chosen. The Robotics Self-Efficacy questionnaire from Riggs and Enochs (1990) was reduced for an efficient evaluation from 25 items to 10 items, similar to the standard self-efficacy-test from Jerusalem and Schwarzer (1999) and with 5 scales.

RSE α=0.842 average scale score = 41.17 SD = 6.336 N=30

Item number	Item-total Correlation	Self-Efficacy Context	Item
RSE1	0.142	Positive Feeling (Lead robots)	I'm confident that a robot will move the way I want it.
RSE2	0.612	Competence (Built robots)	I'm sure I can build a robot.
RSE3	0.669	Achievement (Content about robots)	It's easy for me to understand the parts of a robot.
RSE4	0.535	Achievement (Difficulties with robots)	It would not give me any difficulties to let a robot drive along a line.
RSE5	0.506	Positive Feeling (Control robots)	If I don't know what to do, I'll find a way to control the robot.
RSE6	0.224	Effort (Create robots)	I can create a robot that solves other people's problems if I effort.
RSE7	0.673	Achievement (Working with robots)	I will work well with robots when I have a chance.
RSE8	0.854	Positive Feeling (Solving robotics tasks)	I am someone who immediately solve robotic tasks.
RSE9	0.510	Positive Feeling (Robotic Researcher)	I think that I will be able to everything that a robotic researcher has to do.
RSE10	0.724	Achievement (Function of robots)	I'm the one who will explain how robots work.

Table 8: Item correlations for RSE-Test at the Summer Camp

This Robotics Self-Efficacy-Test was filled out before the summer camp and after the summer camp from 30 students. The result is shown in *Table **9*** “Comparison RSE-PRE and RSE-POST”.

	Mean	N	Std.-Deviation	Standard error of the mean
RSE-PRE	37.00	30	4.291	.783
RSE-POST	41.17	30	6.336	1.157

Table 9: Comparison RSE-PRE and RSE-POST

The *Table **10*** “Correlation of the RSE-Test” show a positive correlation with .590 and with .000 is this result significant.

	N	Correlation	Significance (2-sides)
RSE-PRE RSE-POST	30	.590	0.000

Table 10: Correlation of the RSE-Test

For reliability analysis, Cronbach’s alpha was calculated to assess the internal consistency of the subscale for positive affect, which consists of 10 items. The internal consistency of the questionnaire is good (α= .842) (Streiner, 2003). Internal consistency can be increased to .859 by removing Item 1 from the questionnaires.

Reliability Statistic	
Cronbachs Alpha	Number of Items
.842	10

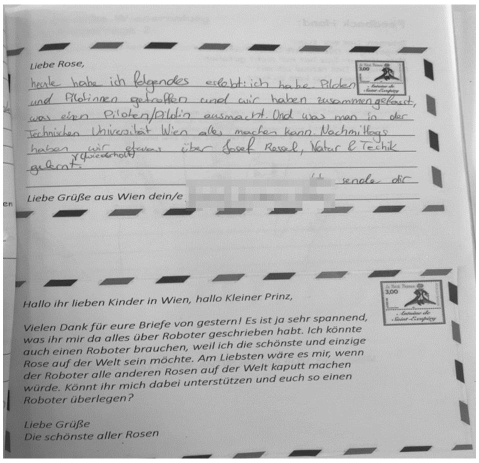
Table 11: Cronbachs Alpha of RSE-Test

The qualitative analysis proved reliability of the RSE-Test by comparing the outcome of documents and artefacts that require self-efficacy related to the context of robotics.

We compared the sketches for the prototypes and documents of the two children with the highest and lowest RSE-Score. The first comparison is shown in Figure 1 “Letter to the rose” from the Activity Block “Reflection 1” (AB5a). The children wrote a letter to the rose about their experience and learnings of the first day at the RSC.

It revealed that the child with the higher RSE score wrote more details, about performance outcomes and vicarious experiences, while the child with the lower did not mention personal performance outcomes, vicarious experience, verbal/social persuasions or physiological reaction, that would lead to a change in the perceived self-efficacy (Bandura, 1977).

Highest RSE-Score

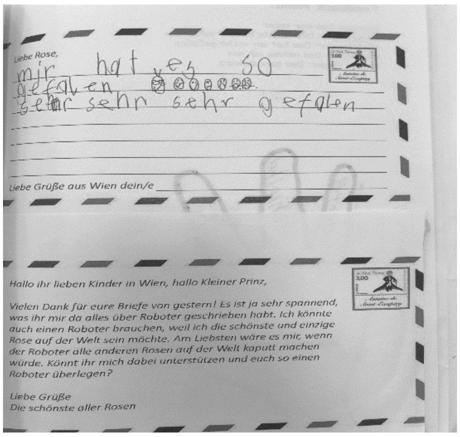

Liebe Rose,
heute habe ich folgendes erlebt: ich habe Piloten und Pilotinnen getroffen und wir haben zusammengefasst, was einen Piloten/Pilotin ausmacht. Und was man in der Technischen Universität Wien alles machen kann. Nachmittags haben wir etwas über Josef Ressel, Natur & Techik gelernt. (wiederholt)
[redacted] sende dir
Liebe Grüße aus Wien dein/e

Hallo ihr lieben Kinder in Wien, hallo Kleiner Prinz,

Vielen Dank für eure Briefe von gestern! Es ist ja sehr spannend, was ihr mir da alles über Roboter geschrieben habt. Ich könnte auch einen Roboter brauchen, weil ich die schönste und einzige Rose auf der Welt sein möchte. Am Liebsten wäre es mir, wenn der Roboter alle anderen Rosen auf der Welt kaputt machen würde. Könnt ihr mich dabei unterstützen und euch so einen Roboter überlegen?

Liebe Grüße
Die schönste aller Rosen

Lowest RSE-Score


Liebe Rose,
mir hat es so gefallen sehr sehr sehr gefallen
Liebe Grüße aus Wien dein/e

Hallo ihr lieben Kinder in Wien, hallo Kleiner Prinz,

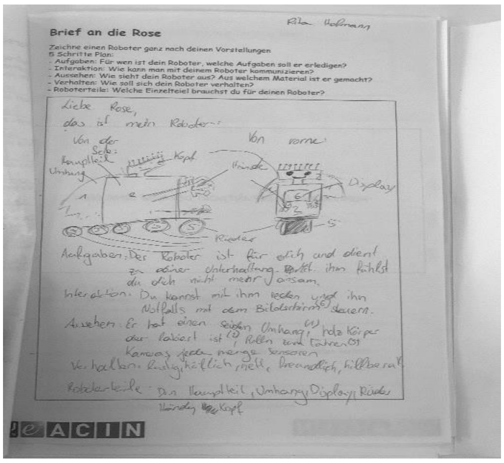
Vielen Dank für eure Briefe von gestern! Es ist ja sehr spannend, was ihr mir da alles über Roboter geschrieben habt. Ich könnte auch einen Roboter brauchen, weil ich die schönste und einzige Rose auf der Welt sein möchte. Am Liebsten wäre es mir, wenn der Roboter alle anderen Rosen auf der Welt kaputt machen würde. Könnt ihr mich dabei unterstützen und euch so einen Roboter überlegen?

Liebe Grüße
Die schönste aller Rosen

Figure 1 Letter to the rose

The second comparison is shown in Figure 2 Sketch about their robot. This document was written at the Activity Block Reflection 2 and based on the 5-step plan (Lammer et al., 2015). The child with the lowest RES-Score also invested less time and effort in drawing the sketch and even just scribbled. The child with the highest RSE-Score drew a sketch with more details and clear description of the different parts of the robot.

Highest RSE-Score



Lowest RSE-Score

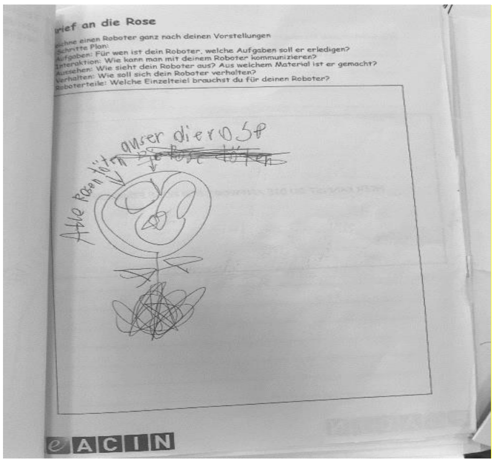


Figure 2 Sketch about their robot.

The Figure 3 The prototypes of the children, which were presented on the last day is shown the artefacts, which were done at the Activity Block “Prototypes” and the children presented in the Activity Block “Plenum Presentation”. The child with the highest score built the function of the robot linked to the sketch. The artefact of the child with the lowest score shows a prototype not related to the sketch, actually he helped his brother building his brother’s idea. Discussion: age difference and gender difference certainly play a role between the difference of the artefacts of the two children. The child with the lower RSE score has clearly less developed cognitive and fine manipulative abilities. However, the intention is not to compare the two children but to see if the high or low RSE score can be supported by the qualitative findings of the artefact analysis.

Highest RSE-Score

Lowest RSE-Score

Figure 3 The prototypes of the children, which were presented on the last day

The statistical analysis shows that the RSE-Test is significant and has a good internal consistency ($\alpha = .842$). In the qualitative analysis is shown that the RSE-Score is also seen in the documents and artefacts of the children. The next step is to implement the new RSE-Test in more educational robotics activities in constructionism approach to get more data from participants and better statistic results.

Conclusion and Outlook

The robotics summer camp with the title “Robots as we children wish them to be” influenced the Robotics Self-Efficacy significant positively (RQ1). The statistical analysis shows that the internal consistency of the questionnaire is significant ($\alpha = .842$). The correlation between the two questionnaires shows a positive influence on the mean of Robotics Self-Efficacy ($r=.590$; $p<.01$) (RQ2). The item-total correlation of RSE 1 has the lowest value (seen in table 5). It could be optimized with rewriting this item, through changing the general statement in rewriting in a more robotic specific statement. This rewritten RSE-Test needs to be assessed in educational robotics activities with constructionism approach in the future (RQ3). In the qualitative analysis, we show that the RSE-Score is also trackable in the documents and artefacts of the children needs to be tested in the future. The next step is to investigate in the next robotics summer camp the influence of the different Activity Blocks on the RSE and to create an evaluation tool for better measurement of scientific literacy.

Acknowledgement

The authors would like to thank Hannah Hieber, Birgit Seiler, Martin Piehslinger and Habibe Idiskut for their supporting the designing, realization and evaluation of the robotic summer camp.

References

Ackermann, E., 2001. Piaget’s constructivism, Papert’s constructionism: What’s the difference. Future of learning group publication 5, 438.

Archer-Bradshaw, R.E., 2017. Teaching for scientific literacy? An examination of instructional practices in secondary schools in Barbados. Research in Science Education 47, 67–93.

Bandura, A., 1997. Self-efficacy: The exercise of control. Macmillan.

Bandura, A., 1995. Exercise of personal and collective efficacy in changing societies. Self-efficacy in changing societies 15, 334.

Bandura, A., 1991. Social cognitive theory of self-regulation. Organizational behavior and human decision processes 50, 248–287.

Bandura, A., 1977. Self-efficacy: toward a unifying theory of behavioral change. Psychological review 84, 191.

Beierlein, C., Kovaleva, A., Kemper, C.J., Rammstedt, B., 2012. Ein Messinstrument zur Erfassung subjektiver Kompetenzerwartungen: Allgemeine Selbstwirksamkeit Kurzskala (ASKU).

Bong, M., Skaalvik, E.M., 2003. Academic self-concept and self-efficacy: How different are they really? Educational psychology review 15, 1–40.

Dugger Jr, W.E., 2001. Standards for technological literacy. Phi Delta Kappan 82, 513–517.

Edwards, L., Rand, K.L., Lopez, S.J., Snyder, C., 2007. Understanding hope: A review of measurement and construct validity research.

ITEEA, International Technology Education Association (Ed.), 2000. Standards for technological literacy: content for the study of technology. ITEA, Reston, Va.

Jäggle, G., Lammer, L., Hieber, H., Vincze, M., 2020. Technological Literacy Through Outreach with Educational Robotics, in: Merdan, M., Lepuschitz, W., Koppensteiner, G., Balogh, R., Obdržálek, D. (Eds.), Robotics in Education, Advances in Intelligent Systems and Computing. Springer International Publishing, pp. 114–125.

Jäggle, G., Vincze, M., Weiss, A., Koppensteiner, G., Lepuschitz, W., Stefan, Z., Merdan, M., 2019. Educational Robotics—Engage Young Students in Project-Based Learning, in: Auer, M.E., Tsiatsos, T. (Eds.), The Challenges of the Digital Transformation in Education, Advances in Intelligent Systems and Computing. Springer International Publishing, pp. 360–371.

Kafai, Y.B., Resnick, M., 2012. Constructionism in practice: Designing, thinking, and learning in a digital world. Routledge.

Lammer, L., Weiss, A., Vincze, M., 2015. The 5-Step Plan: A Holistic Approach to Investigate Children’s Ideas on Future Robotic Products. Presented at the Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts, ACM, pp. 69–70.

Millar, R., 2006. Twenty first century science: Insights from the design and implementation of a scientific literacy approach in school science. International journal of science education 28, 1499–1521.

Norris, S.P., Phillips, L.M., 2003. How literacy in its fundamental sense is central to scientific literacy. Science education 87, 224–240.

Palincsar, A.S., Anderson, C., David, Y.M., 1993. Pursuing scientific literacy in the middle grades through collaborative problem solving. The elementary school journal 93, 643–658.

Papert, S., 1980. Mindstorms: children, computers, and powerful ideas. Basic Books, New York.

Riggs, I.M., Enochs, L.G., 1990. Toward the development of an elementary teacher’s science teaching efficacy belief instrument. Science Education 74, 625–637. https://doi.org/10.1002/sce.3730740605

Rittmayer, A.D., Beier, M.E., 2008. Overview: Self-efficacy in STEM. SWE-AWE CASEE Overviews 1–12.

Saint-Exupery, A., 2010. The little Prince. www.bnpublishing.net

Schwarzer, R., Jerusalem, M., 1999. Skalen zur erfassung von Lehrer-und schülermerkmalen. Dokumentation der psychometrischen Verfahren im Rahmen der Wissenschaftlichen Begleitung des Modellversuchs Selbstwirksame Schulen. Berlin.

Streiner, D.L., 2003. Starting at the beginning: an introduction to coefficient alpha and internal consistency. Journal of personality assessment 80, 99–103.

van der Bijl, J.J., Shortridge-Baggett, L.M., 2001. The theory and measurement of the self-efficacy construct. Sch Inq Nurs Pract 15, 189–207.

Wood, R., Bandura, A., 1989. Impact of conceptions of ability on self-regulatory mechanisms and complex decision making. Journal of personality and social psychology 56, 407.

Zimmerman, B.J., 2000. Self-efficacy: An essential motive to learn. Contemporary educational psychology 25, 82–91.

Twenty Things to Make with Biology

Yasmin B. Kafai, *kafai@upenn.edu*

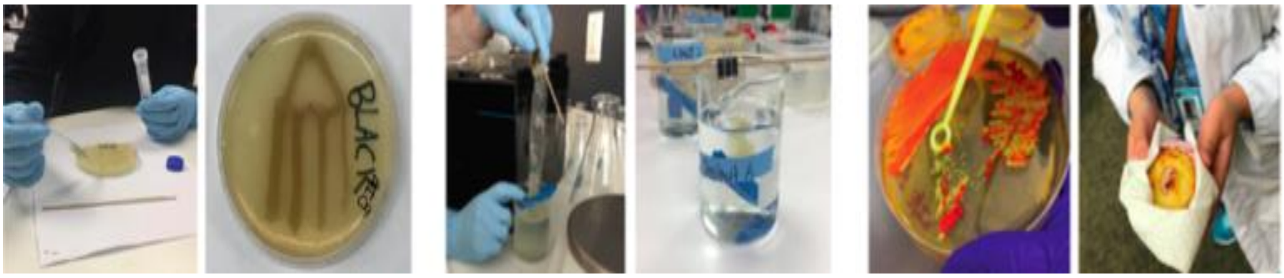
Graduate School of Education, University of Pennsylvania, Philadelphia, PA, USA

Justice T. Walker, *justicew@upenn.edu*

Graduate School of Education, University of Pennsylvania, Philadelphia, PA, USA

Abstract

A 1971 memo by Papert and Solomon introduced twenty things to do with a computer which became the foundation for constructionism. In this paper, we propose bringing constructionist activities into making with living materials. Significant developments in tools and methods have turned biology into a design science: it is now possible to make things *with* biology—or biodesign—rather than just observing processes and behaviours. Our list of twenty things to make with biology includes examples from making colours, toys, games, insulin, batteries, sensors and more. In the discussion, we review how making with biology addresses key affordances of constructionist learning: “tinkerability,” the ability to experiment; “perceptibility,” the immediacy of feedback on learning process; “expressivity,” the personal customization of products; and “usability,” the ability to use learning designs in everyday contexts. We conclude with an overview of accessible and affordable tools available to K-12 education.



A. BioLogo design using bacterial pigment to make colours: (left) ‘Painting’ with bacteria; (right) A completed logo design. Source: Kafai et al., 2017.

B. BioSensor construction using bacteria as detector: (left) Putting the transformed bacteria into the dialysis bag; (right) The completed sensor contraption. Source: Kafai et al., 2017.

C. BioCake using yeast with vitamin A: (left) Petri dish with mixed colonies of yeast cells; (right) Student holding her freshly baked enriched cake. Source: Walker et al., 2018.

Figure 1. Making with Biology: Colour (A), Sensor (B), and Food (C).

Keywords

Biology, Making, Materials

Introduction

In 1971, Seymour Papert and Cynthia Solomon wrote a memo titled "Twenty Things To Do With A Computer," where they outlined a bold vision of how children could be introduced to programming, the more general knowledge of computation, and other formal subjects ranging from physics to music. The programming language LOGO would allow learners to converse and interact with a computer, and in the process introduce new ways of learning. In the memo, they suggested a variety of activities children could program in LOGO: making a turtle draw images on paper by programming a pen to lift up and down; programming behaviours so that the turtle could follow along walls and navigate corners in a room; engaging in geometry by writing program to draw spirals; making an online movie by programming change of petals in a flower; programming sounds to play a song; playing a game called Spacewar and then programming a new game; and many more. The last item on the list was called "recursion line" asking the reader to come up with twenty more things to do with a computer!

These ideas became the foundation for *Mindstorms,* the book that Papert (1980) would publish a few years later. Here, he introduced the education community to how computers could be used by children for learning about powerful ideas such as recursion, variables, mathematics, and cybernetics among others. The activities suggested—a computer that could carry out such processes as spinning motors, activating electromagnets, switching on lights, or even reading the state of light sensitive cells—must have seem far-fetched for most readers in the early 1970's. But Papert and Solomon insisted that it was easy to make the computer do all these things, and readers didn't need to know *how* the computer worked. Instead they needed to describe *what* they wanted to do in an appropriate language such as the LOGO programming language, as if they wanted to give instructions to a person. They concluded while some might balk at the current high cost, that the price of terminal time could come down significantly if more schools would sign up, and that ultimately, every child should be entitled to experience the world of computers.

In "Twenty Things to Make with Biology" we are extending the constructionist vision of engaging learners to converse, interact and design with living materials in new ways. While computers in the 1970's introduced computation with 0's and 1's, today's world of biology as design uses A's, T's, C's, and G's as their building blocks. In bioengineering, designers can make their own DNA—gene by gene—and then grow their designs into real applications by inserting them into living things such as microorganisms (Endy, 2005). In the following sections, we describe twenty things to make with biology. More than half of our suggestions have already been implemented with middle and high school students in schools and community labs. Some of these activities make use of everyday materials such as yeast, kombucha, soil, sand, and tea found in people's homes and pantries while others use mycelium (i.e., mushroom roots) or Escherichia *coli* bacteria which can be ordered online. In some instances, they require lab setups such as petri dishes, plastic droppers, and incubators while others use home kitchen materials such as pots of warm water or baking sheets. Most importantly, readers not need think about how cells will actually make the things but more about how they can use general biology and practical knowledge to design new applications. In the last section of this paper, we share some of our observations about how *making things with biology* is either the same or distinct from doing things with a computer.

Twenty Things to Make with Biology

1. Create a Smell

In *Eau that Smell* learners can genetically modify bacteria (e.g., Escherichia *coli*) to selectively emit a banana scent at different stages of cell growth (Kuldell, 2015). Smell functions like an indicator and showcases how genetic perturbations can be introduced and programmed very precisely. It also illustrates how synthetic aromatics or flavour food additives can be sustainably produced.

2. Grow a Brick

The company BioMason (2019) grows bricks by the thousands by combining sand and bacteria in a cast. By feeding the bacteria with a liquid cocktail that generates a binding substance and letting them dry for a few weeks, the bricks are formed. This approach uses far less energy than existing methods that require stone/mineral extractions, transport and kiln for curing.

3. Bake Enriched Food

Take a plasmid, a pre-coded segment of DNA, and insert it into yeast (Saccharomyces *cerevisiae*), to reprogram the cells to produce beta carotene, also known as vitamin A (Kuldell, 2015). Growing more yeast with Vitamin A this way can be used to bake a cake, or bioCakes (Walker et al., 2018), which is enriched with important nutrients.

4. Build a GMO Detector

To find out whether food contains Genetically Modified Organisms (GMOs), collect DNA from uncooked fruits or vegetables and add a mix of DNA strands that react with known GMO elements. If your food contains a GMO, the DNA strands are designed to fluoresce in the presence of UV light (GMO Detective, 2019).

5. Feed a Battery Light

To make a battery, collect a soil sample from your garden. Place the soil in a container that has conductive chicken mesh at the bottom, attach to this chicken mesh an insulated wire made of zinc, and connect a LED at the top. Take a second mesh/chicken wire—this time attached to an insulated copper wire—and place it at the surface of the soil. Provide water for the microbes in the soil. After two days, the bacteria residing in the lower part of the container where there is much less air will produce enough electricity to turn on the LED light (Magical Microbes, 2019).

6. Grow Insulin

The Open Insulin project (2019) has reprogrammed yeast to produce human insulin at large scales. The yeast needs to be grown in standard nutrient broth to produce purified insulin hormone molecules. This makes insulin very compatible with humans and more affordable.

7. Spin Fibers for Fabric

Spider silk is not only light weight, but also incredibly strong which makes a very durable and versatile fabric. To grow silk with similar features, bacteria are genetically reprogrammed to produce the strong and elastic collagen proteins found in spider silk. This is protein is then purified, dried and spun into thread to weave fabric. Adidas (Wired Magazine, 2017) and The North Face (Forbes, 2019) already used this approach for making shoes and jackets.

8. Dynamic Colors

Make a canvas covered with colourful yeast nutrients that change colours overtime as the yeast consume, grow, and age (Yeast Art Project, 2019). Yeast cells are very good at producing beta carotene that can be scrambled up by adding a hormone to produce various pinks, violets, blues, and even black.

9. Power Gears

Rod-shaped bacteria known as Bacillus *subtilis* can be assembled to rotate microscopic gears and control machines. Tiny gears and screws can be assembled and placed in a liquid environment to keep the bacteria alive and mobile. When enough bacteria are present and move in a common direction—this is called a swarm—they can collectively force the gears to move in predictable directions. Photosensitive bacterial swarms can also be directed by using light (Sokolov et al., 2019).

10. Biodegradable Home Goods

Grow biodegradable home goods and accessories like pots, pencil holders, lamp shades, picture frames and other accent pieces (Ecovative, 2019) using mushroom roots, also called mycelium. Once Mycelium are fed flour and water, they become active after a few days. To make a shape, fill a container with active mycelium and mix in small wood chips, saw dust, or other materials. After a week, the shape is ready and can baked at low heat to stop the mycelium from working.

11. BioSensors

Bacterial cells can be genetically modified and grown to function as sensors and start to glow in the presence of a contaminating substance. Students can build their own sensor with dialysis tubing (see figure 1b) wherein they put the bacteria and place in a cup filled with water that may or may not have the contaminating substance (in this case a sugar called arabinose). If the cup contains arabinose, then the cells in their biosensor tubes will glow under ultraviolet light (Kafai et al., 2017).

12. Kombucha Plastic

Make a bioplastic using a blend of yeast (Saccharomyces *cerevisiae*), kombucha bacteria (Gluconacetobacter *kombuchae*), and lukewarm black tea in a pan. These two organisms work together to produce a biofilm or bioplastic in the presence of nitrogen-rich substances like tea (Shade et al., 2011). After waiting for about 2-4 weeks, a 1-2 inch layer will form in your pan which can be dried and then shaped in many ways.

13. Make an RGB Device

Bacteria such as Escherichia *coli* can be reprogrammed to glow in red, green, and blue (RGB colours) when exposed to ultraviolet light (Tsien, 2010). These glowing bacteria can be put together in different combinations to create new colours and designs. As long as the bacteria are fed, they will continuously produce these fluorescent colours.

14. Play a Game Under the Microscope

Single-celled amoeba-like organisms called Euglena *gracilis* are mobile and respond to specific light colours (Lee et al., 2015). It is possible to control their direction. This means that with the right configuration, two players could race to direct their organism across a finish line or compete to trap (or guide) them in a maze. The only thing needed here is a microscope to visualize the race.

15. Make Vegetables Savory

The Impossible Burger is made out of plants but tastes like a burger made of beef (Burger King, 2019). This is made possible by adding the DNA for a protein found in red blood cells, called heme, in plants. Then plant-based produce like tomatoes are not only more savoury, but also contain more protein content.

16. Dye Fabric

Manufacturing fabric colours like indigo with petrochemicals is harmful to the environment. The bacteria Streptomyces *coelicolor* can produce a large amount of rich, long-lasting, and environmentally friendly indigo pigments to dye thread and whole fabrics (Faber Futures, 2019).

17. Make a Photocell

Phylum algae are very effective at producing electricity using sunlight. These cyanobacteria use photosynthesis to generate this energy. They can be collected and placed in printer cartridges to print on conductive paper. By adding a transistor to a printed circuit arrangement, they can be powered and create a sustainable and recyclable household energy source (Phys.org, 2017).

18. Grow Construction Kits

Many construction kits are made of plastic that is non-degradable. By using mushroom roots (i.e., mycelium) and fermented kombucha, students can grow biodegradable materials to make a biodegradable toys such as a kaleidoscope, doll clothing made with kombucha bioplastic, or Lego compatible 3D printed wings covered in kombucha bioplastic (GIY Biobuddies, 2019).

19. Engage in Critical Discussions

Making things with biology can raise a whole host of thorny issues related to transparency, impact on environment and humans. Those include, evaluating the risks, impact, safety and moral acceptability of designs such as perils of plastic waste in the toy industry and the value of sustainable manufacturing. There are a number of topics to discuss around these issues including those related to food security, environmental sustainability, agriculture, and climate change to name a few.

20. Recursion Line

Think up twenty more things to *make with biology*!

Discussion

We described a wide variety of things that learners of all ages can make with biology using living materials. One attraction of many digital or physical constructionist activities—such as designing games, printing in 3D, building robots, or crafting electronic textiles—is that students are generating, re-making, or augmenting artifacts with physical and digital tools that are already present in their environment. While biomaking also involves materials and tools that are present in students' homes and science classes, the actual fabrication processes and outcomes are distinct in ways that confront core tenets of constructionist theory. Making things with biology differs in sometimes significant ways in terms of tinkerability, perceptibility, expressivity, and usability (Lui, Kafai, Walker, Hanna, Hogan, & Telhan, 2019). In the following sections, we discuss these distinctions but also similarities in more detail and what insights provide for constructionist learning designs and tools in making with biology.

How Making with Biology is Different

Constructionism has always valued tinkering (Resnick & Rosenbaum, 2013), a playful, experimental iterative style of engagement wherein makers are continually reassessing their goals, exploring new paths and imagining new possibilities, and having "a conversation with the material" (Schön, 1983). However, tinkering with biology is much more difficult since microbiological processes involve liquids and require a full run of the entire lab procedure before one can see any result. In biology, processes often occur in a holistic fashion and thus fixing a 'mistake' frequently means doing a lab procedure all over again and waiting for the result, whereas tinkering in engineering and coding involves discrete processes such as iterating on a gear mechanism or developing a specially defined procedure. The specificity of lab procedures and limitations of materials make it somewhat difficult to engage with on-the-spot messing around so popular in maker activities on and off the screen (Lui, Anderson & Kafai, 2018).

Another valued aspect in constructionist activities is that computer or physical designs can yield immediate feedback either on the progress or results of making. For instance, a coder can see the result of a bug they fixed in a program whereas in biomaking this process occurs more slowly. While microorganisms grow quite rapidly, it often takes hours or more for any genetic transformation to yield an outcome. More importantly, due to scale and colourlessness of the microorganisms, learners often cannot immediately see the outcomes of their designs or changes. In making with biology, it is also much more difficult—but not impossible—for learners to personalize artifacts or designs. Whereas consumer-grade electronics kits have created opportunities for lay people to create personalized computational designs, people with limited biological knowledge and background are not yet as able to produce biodesigns that fulfill their individual goals and purposes. Instead, learners must often (but not always) depend on existing protocols and materials developed by experts.

Finally, constructionist activities foster designs that learners or others can immediately use such as playing a game made in Scratch (Resnick, Maloney, Monroy-Hernández, Rusk, Eastmond, Brennan, Millner et al., 2009), making music on a banana piano made with MaKey MaKey (Silver, Rosenbaum, & Shaw, 2012) or a turn-signal hoodie made with the LilyPad Arduino can be worn while biking and signal directions with flashing LEDs (Buechley, 2006). In biomaking, usability comes with its own set of constraints. Some living designs can perish at some point, so careful consideration must be taken to, when necessary, keep the organism alive, such as supplying them with enough nutrients and at appropriates temperature. From this perspective, making with computers affords numerous ready-made situations for usability while biomaking has not yet reached this point of development in its short history.

What Making with Biology Shares with Things To Do with a Computer

We also saw similarities and connections to constructionist learning. While making with biology activities are limited in tinkering with regard to the scripted steps of the lab procedures needed to create the right conditions for, as an example, bacteria to flourish and produce a desired result, the actual hands-on construction and crafting of applications provides considerable degrees of freedom. For instance, students engage with crafting while: "painting with bacteria" by using hot glue guns to mold shapes for their petri dish logos (Kafai et al., 2017), making kombucha plastic and leather clothes for their paper dolls (GIY Biobuddies, 2019), or even colouring fabric.

We also noticed that in many of the suggested applications bacteria were chosen that would reveal a visible change, thus promoting the "perceptibility" dimension prominent in constructionist activities. For instance, in Eau that Smell (Kuldell, 2015) bacteria signal change by emitting a banana smell, in Faber Futures (Faber Futures, 2019) and the Yeast Art Project (Yeast Art Project, 2019) microorganisms signal change when they produce pigments, or—in another case luminescence (Tsien, 2010) to make outcomes more 'visible' to students. While not all biomaking activities provid the expected feedback, it was sometimes precisely the lack of feedback (beakers that did not glow and "stinky" bacteria) that provided contexts for conversations around the science of the process.

Finally, in terms of usability making with biology involved product designs that reached beyond the personal. For instance, in BioLogo, it involved a company focused on sustainable product design, while BioSensors involved researching contexts in which sensing pollution would be of importance, and BioCakes involved thinking about food products that could benefit from nutritional enrichment. It is here where we saw the imagination of students flourish as they recognized the usability—both personal and societal—of their designs. Other examples include Ecovative (Ecovative, 2019) and GIY Biobuddies (GIY Biobuddies, 2019) that both leverage mycelium properties to build a whole swath of products including furniture, home accessories, toys and constructions kits. Or bioMason (bioMason, 2019) and the North Face (Forbes, 2019), who use bacteria to construct building material and clothing. These examples illustrate new frontiers in biology wherein products are not only usable, but they also provide a space for student discourse around manufacturing, sustainability, material life cycles and their collective impact on the planet.

How to Make Things Happen

The development of programming languages and construction kits that let learners do things with computers both have been a driving force in promoting constructionist learning. Previous constructionist efforts focused on making digital designs by controlling a turtle on the computer screen or on the floor. The design of portable and programmable bricks (Resnick, Martin, Sargent & Silverman, 1996) allowed learners to move designs into the physical world and build autonomous creatures no longer tethered to terminals.

Recent developments of simple to use portable lab tools make it possible for K-12 students to genetically alter a wide range of cells for designing a variety of applications. For instance, the *biomakerlab* (Kafai et al., 2017) is a low-cost portable wetlab device that makes it possible to easily genetically modify and grow bacteria cells. Another even simpler example is *BioBits* (Stark et al., 2018) which eliminates cells altogether and provides freeze-dried pellets made of cellular parts that, when hydrated, assembled, and incubated, express unique gene designs, such as a full palette of colours that fluoresce in the presence of ultraviolet light. Other examples include Bento

Labs (Bento Labs, 2019) which provides a portable wet lab device that enables users to construct, isolate, enrich, and analyse genetic parts that can later be introduced into living cells. Amino labs (Amino Labs, 2019) is yet another example that enables young students to transform (i.e., genetically modify), grow and analyse newly engineered organisms.

Some work is even targeting younger students—namely elementary grades. CRISPEE (Verish et al., 2018) is an example of such an effort as researchers developed a low cost block-based simulation device that allows young learners to mix and match wooden blocks in a device that illuminates a simulated firefly bulb with a colour that is representative of the block combinations created by the user. This activity is meant to help young learners understand what synthetic-based genetic modifications are as a concept and the various ways it impacts living things and their traits. When learners introduce their own genetic perturbations (represented by different block combinations), they gain a sense of how to manipulate and—to an extent—control the design of living things.

Our examples of making with biology provided a glimpse into the foreseeable future in which we engage students with 'making' or 'growing' their designs in petri dishes—just like several decades ago students were first invited to making or 'coding' their designs on computers. Realizing making with biology in K-12 education will require significant efforts but learners themselves have already taken charge. In 2019, for the first time, two teams of high school students participated in the BioDesignChallenge which brings together international teams of college students who compete in developing biodesign applications that solve global challenges related to the environment and manufacturing sustainability. To everyone's great surprise, one high school team of three girls took home the first runner up by creating a biodesign toy kit for other K-12 students. The kit provided microbial-based and mushroom-based packaging for new toy designs to address the perils of plastic waste in the toy industry with more sustainable manufacturing. Indeed, making with biology can introduce learners to 21st century ways of doing and thinking just like computers did in the era before.

Acknowledgments

This work was supported by a grant from the National Science Foundation to Yasmin Kafai, Orkan Telhan and Karen Hogan (#1840933). We want to thank Orkan Telhan and Karen Hogan for many inspiring discussions. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF or the University of Pennsylvania.

References

Amino Labs (2019). Available at: https://amino.bio/.

Bento Labs (2019). Available at: https://www.bento.bio/.

BioDesignChallenge (2019). Available at: https://biodesignchallenge.org/.

bioMason (2019). Available at: https://biomason.com/.

Buechley, L. (2006). A construction kit for electronic textiles. In Proceedings of 10th IEEE International Symposium on Wearable Computers (pp. 83-90). Montreux, Switzerland.

Burger King (2019). *Impossible Whopper*. Available at: https://www.bk.com/menu-item/impossible-whopper.

Evocative Design (2019). Available at: https://ecovativedesign.com/.

Endy, D. (2005). *Foundations for engineering biology*. Nature, *438*(7067), 449.

Faber Futures (2019). *Colour Coded Forbes Pigment Collection 2018*. Available at: https://faberfutures.com/.

Forbes (2019). Available at: https://www.forbes.com/sites/johncumbers/2019/08/28/new-this-ski-season-a-jacket-brewed-from-spider-silk/#49b5cd94561e.

Franklin-Wallis, O. (2017). *These lightweight adidas shoes are made from spider silk grown in a lab*. Wired. Available at: https://www.wired.co.uk/article/adidas-futurecraft-biofabric-shoes.

GIY Bio Buddies (2019). Available at: https://giybiobuddies.weebly.com/.

GMO Detective (2019). Available at: https://gmodetective.com/about-2/.

Instructables (2019). Kombucha Fabric. Available at: https://www.instructables.com/id/Kombucha-Fabric/.

Kafai, Y. B., Telhan, O., Hogan, K., Lui, D., Anderson, E., Walker, J.T., & Hanna, S.Growing Designs with biomakerlab in High School Classrooms. In Proceedings of the 2017 Conference on Interaction Design and Children (IDC '17). ACM, New York, NY, USA, 503-508. DOI: https://doi.org/10.1145/3078072.3084316.

Kuldell, N. (2015). *Biobuilder*. San Fransciso, CA: O'Reilly.

Lee, S. A., Bumbacher, E., Chung, A. M., Cira, N., Walker, B., Park, J. Y., ... & Riedel-Kruse, I. H. (2015). Trap it!: A playful human-biology interaction for a museum installation. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 2593-2602. New York, NY: ACM.

Lui, D., Anderson, E., Kafai. Y.B. (2018). Is Making all about Tinkering? A Case Study of High School Students' Activities in Biomaker Workshops. In J. H. Kalir (Ed.). 2018 Connected Learning Summit, 158-167. Pittsburgh, PA: ETC Press. Available at: https://connectedlearningsummit.org/cls2018/proceedings/.

Lui, D., Kafai, Y.B., Walker, J.T., Hanna. S., Hogan, K., and Telhan, O. (2019). A Revaluation of How We Think about Making: Examining Assembly Practices and Artifact Imagination in Biomaking. In Proceedings from FabLearn '19: The 8th Annual Conference on Creativity and Fabrication in Education. New York, NY: ACM. https://doi.org/10.1145/3311890.3311895.

Magical Microbes (2019). Available at https://www.magicalmicrobes.com/collections/kits/products/mudwatt-clean-energy-from-mud?gclid=Cj0KCQjw6eTtBRDdARIsANZWjYaHhANF-s1jsS1UDPXCcBlh7m89MyEYeSqmtziN_64hdaN4bYQ9ZEwaAiypEALw_wcB.

Open Insulin Project (2019). Available at: https://openinsulin.org.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.

Peppler, K. & Bender, S. (2013). *Maker movement spreads innovation one project at a tim*e. Phi Delta Kappan, 95(3), 22-27.

Resnick, M. & Rosenbaum, E. (2013). *Designing for Tinkerability*. In Margaret Honey and David Kanter (Eds.), Design, Make, Play: Growing the Next Generation of STEM Innovators (pp. 163-181). New York, NY: Routledge.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A. & Kafai, Y. B. (2009). *Scratch: programming for all*. Communications of the ACM, 52(11), 60-67.

Schön, D. (1983). *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, New York.

Shade, A., Buckley, D. H., & Zinder, S. H. (2011). *The kombucha biofilm: a model system for microbial ecology*. Final report on research conducted during the Microbial Diversity course. Marine Biological Laboratories, Woods Hole, MA.

Silver, J., Rosenbaum, E. & Shaw, D. (2012). Makey Makey: improvising tangible and nature-based user interfaces. In Proceedings of the sixth international conference on tangible, embedded and embodied interaction (pp. 367-370). New York, NY: ACM.

Sokolov, A., Apodaca, M. M., Grzybowski, B. A., & Aranson, I. S. (2010). Swimming bacteria power microscopic gears. In Proceedings of the National Academy of Sciences, 107(3), 969-974.

Stark, J. C., Huang, A., Nguyen, P. Q., Dubner, R. S., Hsu, K. J., Ferrante, T. C., Anderson, M., Kanapskyte, A., Mucha, Q., Packett, J.S., Patel, P., Patel, R., Qaq, D., Zondor, T., Burke, J., Martinez, T., Miller-Berry, A., Puppala A., Reichert, K., Schmid, M., Brand, L., Hill, L. R., Chellaswamy, J.F., Faheem, N., Fetherling, S., Gong, E., Gonzalzles, E. M., Granito, T., Koritsaris, J., Nguyen, B., Ottman, S., Palffy, C., Patel, A., Skweres, S., Slaton, A., Woods, T., Donghia, N., Pardee, K., Collins, J. J., & Jewett, M. C. (2018). BioBits™ Bright: A fluorescent synthetic biology education kit. *Science Advances*, *4*(8), eaat5107.

Tsien, R. Y. (2010). Nobel lecture: constructing and exploiting the fluorescent protein paintbox. *Integrative Biology*, *2*(2-3), 77-93.

Verish, C., Strawhacker, A., Bers, M., & Shaer, O. (2018). CRISPEE: A Tangible Gene Editing Platform for Early Childhood. In Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction (pp. 101-107). New York, NY: ACM.

Walker, J. T., Shaw, M., Kafai, Y., & Lui, D. (2018). Biohacking Food: A Case Study of Science Inquiry and Design Reflections about a Synthetic Biology High School Workshop. In Kay, J. and Luckin, R. (Eds.) Rethinking Learning in the Digital Age: Making the Learning Sciences Count, In Proceedings of the 13th International Conference of the Learning Sciences (ICLS) 2018, Volume 3. London, UK: International Society of the Learning Sciences. https://repository.isls.org//handle/1/733.

Yeast Art Project (2019). Available at: http://www.yeastart.org/.

Zyga, L. (2017). *Digitally printed cyanobacteria can power small electronic devices.* Phys.org. Available at: https://phys.org/news/2017-11-digitally-cyanobacteria-power-small-electronic.html.

Using Programming to Improve Elementary School Children's Mathematical Understanding: Based on the Study of Hexagons

Sayaka Tohyama, *tohyama@inf.shizuoka.ac.jp*
Faculty of Informatics, Shizuoka University, Hamamatsu, Japan

Yugo Takeuchi *takeuchi@inf.shizuoka.ac.jp*
Faculty of Informatics, Shizuoka University, Hamamatsu, Japan

Abstract

There are a variety of programming education practices for elementary school children. One of the most common is the use of programming to deepen understanding of subject matter. In Japan, the Ministry of Education has decided to use programming in existing subjects like mathematics and science. To deepen students' understanding, collaborative learning is possible even with the use of programming. Our research objective in this study is to propose the design of a knowledge-construction jigsaw (KCJ), a collaborative learning methodology based on the jigsaw technique, for using programming in elementary schools and to suggest analytical points in the improvement of students' understanding.

For the research purpose, we conducted collaborative learning classes based on a KCJ to study the nature of regular hexagons and learn how to draw a regular hexagon using Scratch. The main question for the class was 'How can we draw a regular hexagon using Scratch, and why can we draw the operation you suggested? Please explain this'. If we want the students to understand the nature of a regular hexagon, creating a program to draw a regular hexagon works to do so (Figure 1).

Ninety-nine students from three classes in a Japanese public elementary school attended the practice. We compared the answers to the main question between pre- and post-tests to see how they changed after the class. The students' answers suggested that they knew how many degrees and how many times the steps should be repeated; however, they had difficulty explaining why it was repeated six times to draw a regular hexagon. Furthermore, discourses in the classes suggested that the students gradually understood the meaning of the program, and finally, some of the students noticed that the program could be used to draw regular polygons if the appropriate numbers for angle and repeating were defined.

Our results show that collaboration with programming works to deepen students' understanding of concepts and not only to write bug-free codes. Furthermore, the KCJ functions if we use programming.

Keywords

Mathematics, Regular Polygon, Scratch, Collaborative Leaning, Jigsaw, Elementary School

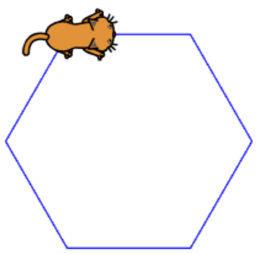


Figure 1. Regular hexagon drawn by Scratch

Background

In recent years, programming or coding has been globally accepted as the new educational domain for kids, especially for elementary school students (European Schoolnet, 2015; Jitsuzumi, Tanaka, Aizawa, Tohyama & Uchiyama, 2018; Resnick, 2017). In this paper, we will refer to programming and coding education as programming education for simplification.

There are a variety of programming education practices for elementary school children in the world. Because programming has some dimensions for scaffolding children's development, it can be used in broader ways to design educational practices.

We focused on a domain which is using programming for students' studying mathematics. There are two reasons that we are in this programming for mathematical problem-solving position: the first is that using programming to deepen understanding about subject matter is strongly connected to educational researches. Using programming to deepen understanding of subject matter is based on programming "as new media" (Kay Goldberg, 1977). From the viewpoint, programming functions for introducing people to attend digital society (Kafai & Burke, 2016). Some people will be introduced to computer science domain who are motivated by programming in subject matter or daily-experiment of using ICT. According to previous studies, improvement of higher-order cognitive skills will be observed when people engage in exploring or studying specific contents like "the nature of light" (Griffin, McGaw & Care, 2012). The second reason is that Japanese educational policy has decided to use programming in existing subjects. In the next section, we will introduce the situation in Japan regarding programming education for elementary school children.

Programming for scaffolding students' learning of mathematics

Programming is now used for deepening students' understanding of subject matter, especially in mathematics-relating topics. ScratchMaths (Hoyles & Noss, 2018) and diSessa's 'big picture' theory (diSessa, 2018) are great examples of this category. Lawler (1981) showed his daughter's deepening mathematical understanding process. The daughter discovered how to do two-digit addition through programming using Logo. Additionally, Totsuka's practices in the 1980s to 1990s were the first in Japan to use programming to deepen children's understanding in elementary school subjects (Totsuka, 1989; Totsuka, 1995). Totsuka is a former Japanese elementary school teacher who created a programming environment like Logo (Papert, 1980) in Japanese.

Recently, the Japanese Ministry of Education decided to use programming in elementary schools in above-mentioned way (MEXT, 2017). The context of Japanese programming education is a little different from that of schools in England. England, one of the pioneers of programming education for elementary school students, has allocated 'computing' as a mandatory subject in programming education (Department for Education in England, 2013). Japanese approach of programming education seems to follow the approach of ScratchMaths. The team of ScratchMaths pointed to the function of programming as deepening students' mathematical understanding (Benton, Saunders, Kalas, Hoyles, & Noss, 2018). On the other hand, England's decision looks to emphasise learning programming itself with computer science matters.

Japanese programming education will be held in existing subjects in all the public elementary schools beginning in April 2020. Infusing programming concepts into subject matter is characteristic of Japanese programming education in elementary schools. The subjects on which most of the focus is placed are mathematics and science because mathematical iterative calculation is one of the strongest areas of programming. In the context of mathematics in elementary schools, activities like drawing regular polygons using paper and pencil can be replaced by programming (Figure 1). This means that Japanese elementary school students will be introduced to programming activities in mathematical contexts rather than technological images. Fifth grade students study the nature of regular polygons by thinking about the size and incremental regularity of each angle in a regular polygon.

Another characteristic of Japanese programming education is the use of programming to deepen students' understanding of subject matter. To meet this requirement, we should investigate how to use programming to deepen students' understanding about the nature of regular polygons or the efficient use of electrical power, without using paper and pencil. This means that Japanese elementary school students can experience programming within the context of subject matter learning, not to be as a professional programmer. Educational theory and methodology for an effective use of programming is needed.

The Combination of Collaborative Learning and Programming

The effect of collaborative learning is widely accepted (Hmelo-Silver, Chinn, Chan & O'donnell, 2013; OECD 2013; Fischer, Hmelo-Silver, Goldman & Reinmann, 2018). There are a variety of methodologies in scaffolding collaborative learning such as the jigsaw technique (Aronson & Patnoe, 1997; Brown & Campione 1996), reciprocal teaching (Palincsar & Brown, 1984), knowledge building (Scardamalia & Bereiter, 1994), cooperative learning (Johnson & Johnson, 1994), and so on. Of these methodologies, we focused on the jigsaw technique. The first reason is that jigsaw is already used to deepen each student's understanding in elementary schools in Japan (CoREF, 2019; Miyake & Kirschner, 2014; Shirouzu, Scardamalia, Saito, Ogawa, Iikubo, Hori & Rose, 2016; Tohyama &Takeuchi, 2018). The second reason is that there is a 'knowledge-construction jigsaw' (KCJ) which is designed to improve understanding of mechanisms based on the theory of constructive interaction (Miyake, 1986). Not only is the analysis method explained in Miyake's research capable of revealing an iterative, progressive problem-solving process, but participants have also deepened their own understandings when constructive interaction occurred in their discussions. Miyake pointed out that constructive interaction is well-produced if the participants externalize their own understanding and the depths of their understanding differed.

Furthermore, we intend to contribute to the pursuit of a modern version of Webb's work (Webb, 1984; Webb, Ender & Lewis, 1986) using the viewpoints of Miyake's work through a KCJ. Webb's paper discussed concerns about familiarity with BASIC and students' typing skills as reasons for why they could not observe quality interactions. Higher-level problem-solving interactions can be expected if we add constructive interaction points of view.

Research Objectives

Our research objective in this study is to propose a KCJ design for using programming in elementary schools and suggest analytical points in the improvement of students' understanding. Of the above-mentioned categories of programming education, we focused on deepening students' understanding of subject matter. In this study, we focused on using programming to draw regular polygons in mathematics because it could be done during a one-hour class and does not require physical devices.

Research Method

In our study, we used the KCJ, a specialized jigsaw method, to improve students' understanding based on the theory of constructive interaction. The main objective of our designed classes was for students to understand the nature of a regular hexagon through the KCJ process using the Scratch 3.0[44] online editing software. The students were evaluated using pre- and post-tests as objective assessments, as suggested with KCJ, which is an evaluation method comparing students' answers to the main question of the class on the pre- and post-tests.

Summary of the Classes

The classes were held as a regular class for fifth graders in the computer room of a Japanese public elementary school on February 18, 2019. We conducted our lesson for each of three classes (class 1, class 2, class 3). Each class consisted of almost thirty-five students and had

[44] https://scratch.mit.edu/

almost an equal number of girls and boys. There was no segregation of classes according to students' individual levels of academic achievement. Each class lasted 45 minutes. The first author conducted all three classes, and there was no difference in procedure between the classes. One teacher from the elementary school and two university students with programming experience helped the students during the classes. Each group worked on one laptop PC to create and test their own programs using Scratch. Both before and after the classes, we asked the students to answer the main question individually. All of the students had played Blockly Games[45] for 45 minutes in their regular classes prior to our class to introduce our programming activity. Furthermore, all the students already studied the nature of regular hexagon and regular polygons without programming, and drew hexagons using a pair of compasses.

The main question of the class was 'How can we draw a regular hexagon using Scratch, and why can we draw the operation you suggested? Please explain this'. If we want the students to understand the nature of a regular hexagon, creating a program to draw a regular hexagon works to do so.

Detailed Activity

The KCJ consists of three stages: expert, jigsaw, and crosstalk. The timetable for our class is shown in Table 1. To begin, the first author demonstrated drawing a square and then failed in drawing an equilateral triangle using 60 degree. The 60 degree was suggested by some students. The 60 degree was inferred by an interior corner of equilateral triangle which the students have studied prior to our class. The first author prompted the students to use their bodies to clarify the angle degrees, and then tried to create a correct program using 120 degrees which some students suggested (Figure 1). The function and how to use Scratch were explained with the above-mentioned introduction. After that, the first author showed the main question and the students individually wrote down their own ideas in free-format on worksheets (pre-test).

Before starting the KCJ activity, the students were assigned to one of three roles in each group, and then they designed their own program together. During the first step of the KCJ activity, the three members were divided into one of three expert groups to be the 'experts' for each part: A, B, or C (details are shown in the appendix). Part A was a scaffold for finding the same operation-patterns in relation to drawing an equilateral triangle. Part B was a guide for using the 'repeat' operation based on the program for drawing an equilateral triangle. Part C helped to clarify which angle was appropriate when drawing a regular hexagon, focusing on exterior angles. They studied their assigned parts collaboratively in each expert group. Second, the students reunited with their original group members to teach the others about their own assigned parts. At this time, the students were prohibited from using Scratch to avoid trial and error without a hypothesis. The students were encouraged to explain each function of their own program for drawing a regular hexagon. This was the jigsaw stage. Then, they integrated all three parts into their own programs. Third, some groups presented their programs to the whole class and discussed them in the crosstalk stage of the KCJ activity. The students were not allowed to test their own programs for drawing a regular hexagon before the presentation, so they had to confirm their ideas before this crosstalk. An example of the right program is shown in Figure 3.

After the three steps of the KCJ activity, the students individually wrote their answers to the main question in free-format on the worksheets.

All of the groups in the classes used an IC recorder so we could later analyse the students' discourses. Two video-cameras, which were located in the front and back corners of the PC room, recorded the whole class activity. Two PCs in the PC room were recorded to see how the students made their programs. All worksheets were collected when the classes were over and were returned to each student after being electrically scanned by the first author.

[45] https://blockly.games/

Table 1. *Timetable of each class using KCJ*

Time (min)	Activity
10	Introduction to the study of regular hexagons using programming
5	Pre-test
10	1. Expert: Members become experts in study groups for part A, B, or C
10	2. Jigsaw: Students explain their own part to their group and how to integrate it into drawing a regular hexagon **without using Scratch**
5	3. Crosstalk: Some groups present their programs to the whole class, without testing their programs, using Scratch
5	Post-test

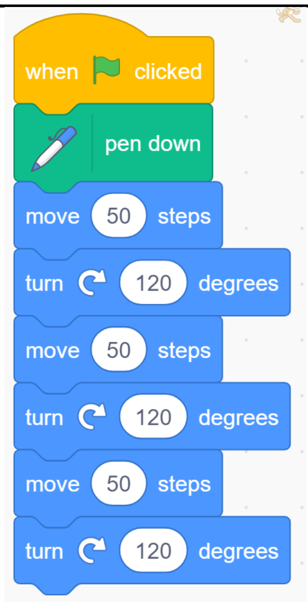


Figure 2. An example program for drawing an equilateral triangle

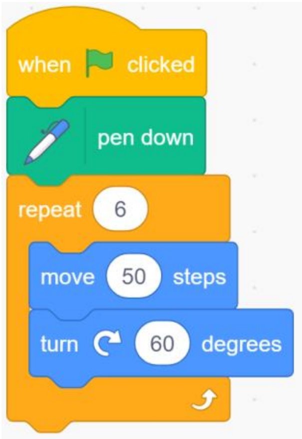


Figure 3. Program for drawing a regular hexagon

Analytical Method

Firstly, we analysed the pre- and post-tests from the viewpoints of the students' understanding of the program's meaning and the necessary operations in drawing a regular hexagon. To answer the main question of the practice, we defined three analytical points:

1. Did the students identify the correct angle (60 degrees) to draw a regular hexagon?
2. Did the students explicitly describe how to draw a regular hexagon using the 'repeat' command (to repeat 6 times)?
3. Did the students explicitly explain the meaning of '6' (6 means the number of angles or sides to draw a regular hexagon)?

Of these points, 1 may be easiest to answer because part C gave the key information. Point 2 is comparatively more difficult because the students would have to transfer their understanding of part B to a regular hexagon situation. Point 3 is the most difficult because the students had to think about the relationship between the nature of regular hexagons and their program.

Secondly, we analysed the crosstalk discourses to determine how deeply the students understood the nature of regular polygons not limited to regular hexagons.

Results

Ninety-nine students from three classes (class 1: 34 students, class 2: 35 students, class 3: 30 students) participated in the present study. Thirty-two groups (class1: 11 groups, class 2: 11 groups, class 3: 10 groups) were organized in the practices.

Result 1: Evaluation of Pre- and Post-Tests

We compared the answers to the main question between pre- and post-tests to find how the students' answers were changed by the practice. The results are shown in Figure 4. Following the three analytical points, 84.8% of the students wrote "turn 60 degrees" on their post-tests while 48.5% of the students had written it on the pre-test. This means that almost half of the students could answer correctly without the practice. For 'repeat 6 times', 68.7% of the students answered 'repeating' on their post-tests. On the pre-test, however, only 10% of the students were able to explain this. The comparison of pre- and post-test answers for 'to repeat 6' suggested that the programming experience successfully taught them the 'repeat' concept in programming. The third analytical point showed that 42.2% of the students understood this point on their post-test, while only 2% had understood it on their pre-test.

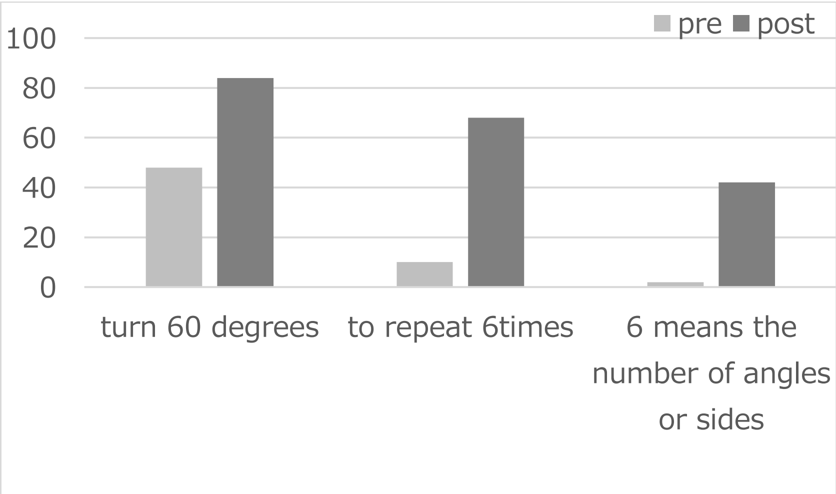


Figure 4. Results of the worksheets analysis

These results suggested that the students could describe the fragmental numbers or concepts of programming for drawing a regular hexagon during the jigsaw. However, explaining the meanings of the program was not easy for the students. In Japanese, the word for 'regular hexagon' includes '6' clearly ('正6角形' is the Japanese word for a regular hexagon), so the students were easily motivated to use 6 without thinking about the meaning of the number. The third analytical point revealed the difficulty of explaining the program even though the students were able to discuss the 'correct' numbers and concepts in the program.

Result 2: Crosstalk Discourses

During the crosstalk activity in class 2, the students pointed out that the program for drawing a regular hexagon can be changed to draw a regular hectogon. The actual discourse was as follows.

(T: First author as a class teacher, S: students)

- T: (showing the correct program for drawing a regular hexagon which was created by one of the students and drawing a regular hexagon via screen) So, now we confirmed that this program draws a regular hexagon. But why did the program repeat 6?
- S1: There are six diagonal lines, I think.
- T: Diagonal lines, hmm. Any other opinions?
- S2: There are six sides in regular hexagons.
- T: Six sides, hmm. Any other opinions?
- S3: I think there are six corners in regular hexagons, as shown in 'rotate X' and 'repeat 6'.
- T: Hmm…can we draw a regular octagon using this program?
- Ss: Yes, we can! (almost 10 students said)
- T: How about drawing a regular decagon?
- Ss: Yes, we can! (almost 10 students said)
- T: How about drawing a regular hectogon?
- Ss: Yes, we can! (almost 5 students said)
- T: How do we draw using the program?
- S4: (A boy raised his hand and stood up.)
- T: Can you show how should we change the program?
- S4: (Pointing to the number of repeats and the number of an angle turns in Figure 3) We should change the turning angle and repeat times.
- Ss: I think so too. (almost 5 students said)

The discourse suggested that the students gradually understood the meaning of the program, and finally, some of the students noticed that the program can be used to draw regular polygons if we define the appropriate numbers for angle and repeat. In the process, there were misconceptions, as shown in S1's remark. Furthermore, the students may be motivated to try to understand the meaning of '6' because there were two different opinions (S2, S3). Because they both looked correct, the students tried to explain the detail of the repeated blocks (S3).

Discussion

We designed a collaborative learning experience using a KCJ with programming to deepen students' understanding of mathematics and evaluated the students' understanding using a pre- and post-assessment framework. As mentioned above, collaborative learning, especially a KCJ, can be used with a programming activity even though the students were elementary school children. Furthermore, collaborative learning with programming scaffolds the students' relation-making between the nature of hexagons and the program, as shown in result 1. How the collaborative learning functioned was shown in result 2 as a case study. Even though previous studies have suggested that there is a limited effect of collaborative learning with programming (Webb, 1984; 1986), our results show that collaboration with programming worked to deepen students' understanding of the concepts that appeared in the program.

The process of collaborative learning with programming resembles the constructionism approach. The students may repeatedly re-construct their own understanding during collaborative learning. A gradual process of understanding can be observed in collaborative learning with programming.

As shown in the discourse in result 2, the students' understanding did contain misconceptions. This means that the students did not understand the nature of regular hexagons perfectly even though they could write a program for perfectly drawing a regular hexagon. In a programming activity, the gap between writing a correct program and understanding the meanings of the program often appears and is not limited to this study. If we eliminate this gap, students may be scaffolded to deepen their understanding because programming has the power of externalization. Collaborative learning is one of the solutions that can be used to eliminate the gap.

Because we chose our research method as design-based research, the limitation of this study is that it is not comparable to non-collaboration classes. Qualitative evaluation such as discourse analysis for each group in the present practice is one of the future works of this study. In addition, our future plans include analysing the reasons that some students were able to describe the meaning of 'repeat' in the program for drawing polygons and some students could not.

Acknowledgements

We would like to thank the Japan Society for the Promotion of Science KAKENHI for their grant (No. 17K17786; No. 20K14079) to the first author. We appreciate Koji Tominaga, who is a teacher in Fujimi elementary school in Japan, the teachers, and the students who cooperated in this research.

References

Aronson, E., & Patnoe, S. (1997). *The jigsaw classroom (2nd edition.)*. New York: Longman.

Benton, L., Saunders, P., Kalas, I., Hoyles, C. & Noss, R. (2018). Designing for learning mathematics through programming: A case study of pupils engaging with place value. *International Journal of Child-Computer Interaction*, 16, 68-76.

Brown, A. L. & Campione, J. C. (1996). Psychological theory and the design of innovative learning environments: On procedures, principles, and systems. In L. Schauble & R. Glaser (Eds.), *Innovations in learning: New environments for education* (pp. 289–325). Mahwah, NJ: Lawrence Erlbaum Associates.

CoREF (2019). Handbook of Designing Lessons on Collaborative Learning - 3rd edition. Consortium for Renovating Education of the Future, The University of Tokyo. (In Japanese)

Department for Education in England (2013). *National curriculum in England: computing programmes of study*. https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study (visited:10/20/19).

diSessa, A. A. (2018). Computational Literacy and "The Big Picture" Concerning Computers in Mathematics Education. *Mathematical Thinking and Learning*, 20(1), 3-31.

European Schoolnet (2015). *Computing our future: Computer programming and coding. Priorities, school curricula and initiatives across Europe*. http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf/d3780a64-1081-4488-8549-6033200e3c03 (visited:10/20/19).

Fischer, F., Hmelo-Silver, C. E., Goldman, S. R. & Reinmann, P. (Eds.) (2018). *International Handbook of the Learning Sciences*. New York: Routledge.

Griffin, P., McGaw, B., & Care, E. (Eds.). (2012). *Assessment and Teaching of 21st Century Skills*. Dordrecht: Springer.

Hmelo-Silver, C. E., Chinn, C. A., Chan, C. K. K. & O'donnell, A. (Eds.) (2013). *The International Handbook of Collaborative Learning*. New York: Routledge.

Hoyles, C. & Noss, R. (2018). Scratchmaths: A Positive Outcome for Constructionism at Scale. *Conference Proceedings of Constructionism 2018*, 68-69.

Lawler, R. W. (1981). The Progressive construction of mind. *Cognitive Science*, 5, 1-30.

Jitsuzumi, T., Tanaka, E., Aizawa, S., Tohyama, S. & Uchiyama, Y. (2018). *Approaches to Fostering 21st-Century ICT Capabilities for Future Generations in APT Countries Asia-Pacific Telecommunity*. Asia-Pacific Telecommunity.

Johnson, D. W., Johnson, R. T. & Holubec, E. J. (1994). *Cooperative Learning in the Classroom*. Assn for Supervision & Curriculum.

Kafai, Y. B. & Burke, Q. (2016). *Connected Gaming: What making video games can teach us about learning and literacy*. MIT Press.

Ministry of Education in Japan (MEXT). (2017). *New curriculum guidelines*. http://www.mext.go.jp/a_menu/shotou/new-cs/1384661.htm (visited: 7/25/19)

Miyake, N. (1986). Constructive interaction and the iterative process in understanding. *Cognitive Science*, 10(2), 151-177.

Miyake, N. & Kirschner, P. (2014). The social and interactive dimensions of collaborative learning. In Hmelo-Silver, C. E., Chinn, C. A., Chan, C. K. K. & O'donnell, A. (Eds.) *The International Handbook of Collaborative Learning, pp.* 418-438*.*

OECD (2013). *PISA 2015 collaborative problem solving framework*. OECD Publishing.

Palincsar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction*, 1(2), 117-175.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. NY: Basic Books.

Resnick, M. (2017). *Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play*. The MIT Press.

Scardamalia, M., & Bereiter, C. (1994). Computer support for knowledge-building communities. *The Journal of the Learning Sciences*, 3(3), 265-283.

Shirouzu, H., Scardamalia, M., Saito, M., Ogawa, S., Iikubo, S., Hori, N., & Rosé, C. (2016). Building on Cultural Capacity for Innovation Through International Collaboration: In Memory of Naomi Miyake. In Looi, C. K., Polman, J. L., Cress, U., and Reimann, P. (Eds.). *Transforming Learning, Empowering Learners: The International Conference of the Learning Sciences (ICLS) 2016, Volume 2*.

Tohyama, S. & Takeuchi, Y. (2018). How to Improve Children's Understanding of Code?: A Preliminary Study using Jigsaw Method for Computer Programming in Elementary School. Oral presented at IFIP TC3 Open Conference on Computers in Education.

Totsuka, T. (1989). *Eri-chan and Logo-kun in Kun-kun city*. Tokyo: Russel company. (In Japanese)

Totsuka, T. (1995). *Galaxy of computer education*. Tokyo: Bansei-publisher. (In Japanese)

Webb, N. M. (1984). Microcomputer learning in small groups: Cognitive requirements and group processes. *Journal of Educational Psychology*, 76, 1076-1088.

Webb, N. M., Ender, P. & Lewis, S. (1986). Problem-Solving Strategies and Group Processes in Small Groups Learning Computer Programming. *American Educational Research Journal*, 23(2), 243-261.

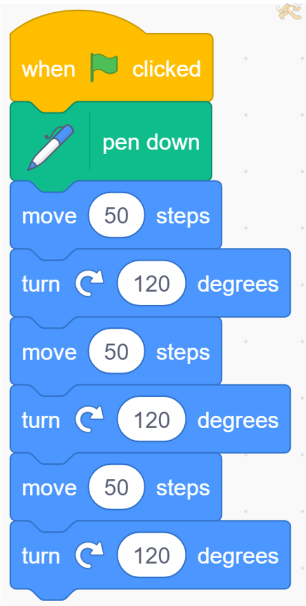
Appendix

<Part A>

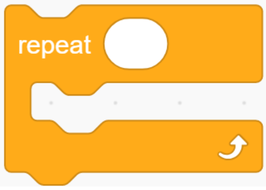
Based on the program for an equilateral triangle, we will draw a regular hexagon using Scratch.

How many times do you use these blocks? Why? Please explain your idea.

<Program for drawing equilateral triangle>



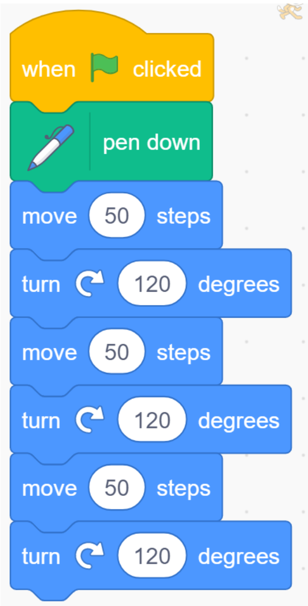
<Part B>

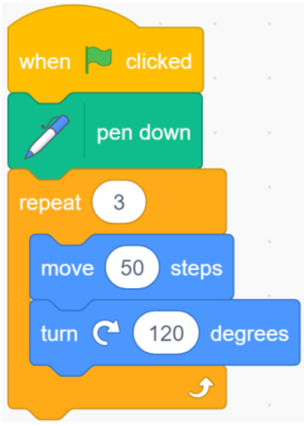
We will draw a regular hexagon using 

in Scratch.

When we run the right program, we will get the same result as running the left program.

<Program for drawing an equilateral triangle>

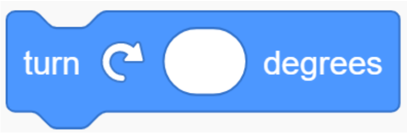




Please explain what you think ‘3’ means in the ‘repeat 3’ block in the right program.

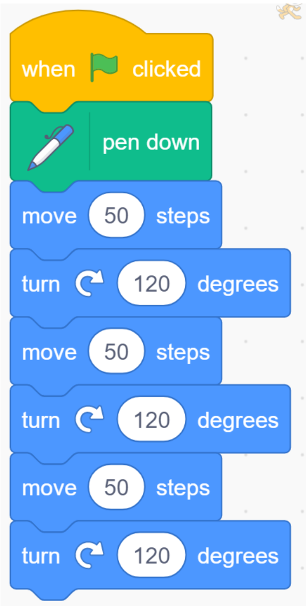
<Part C>

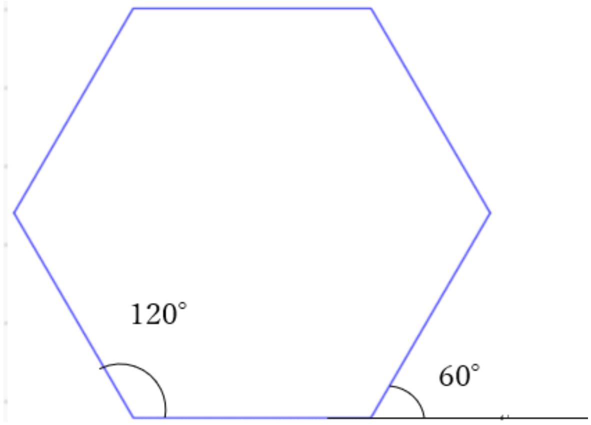
Based on the program for an equilateral triangle, we will draw a regular hexagon using Scratch.

Which number is appropriate in 

? Why? Please explain your idea.

<Program for drawing an equilateral triangle> <The shape of a regular hexagon>





Index of Authors