

Introducing Coding through Tabletop Board Games and Their Digital Instantiations across Elementary Classrooms and School Libraries

Victor R. Lee
Graduate School of Education
Stanford University
Stanford, CA USA
vrlee@stanford.edu

Frederick Poole
Instructional Technology and
Learning Sciences
Utah State University
Logan, UT USA
frederick.poole@usu.edu

Jody Clarke-Midura
Instructional Technology and
Learning Sciences
Utah State University
Logan, UT USA
jody.clarke@usu.edu

Mimi Recker
Instructional Technology and
Learning Sciences
Utah State University
Logan, UT USA
mimi.recker@usu.edu

Melissa Rasmussen
Department of Computer Science
Utah State University
Logan, UT USA
melissa.ann.r@gmail.com

ABSTRACT

This experience report describes an approach for helping elementary schools integrate computational thinking and coding by leveraging existing resources and infrastructure that do not rely on 1-1 computing. A particular focus is using the school library and media center as a site to complement and enhance classroom instruction on coding. Further, our approach builds upon “unplugged” knowledge and practices that are already familiar to and motivating for students, in this case tabletop board games. Through these games, students can use their prior knowledge and ease with tabletop gaming mechanics to cue relevant ideas for core computational concepts. We describe a model and an instructional unit spanning across classroom and school library settings that builds upon board game play as a source domain for computing knowledge. Building on expansive framing, the model emphasizes instructional linkages being made between one domain (the tabletop board game) and another (specially designed Scratch project shells with partially complete code blocks) such that the reasoning activities and different contexts are seen as instantiations of the same encompassing context. We present the experiences of three elementary school teachers as they implemented the unit in their classrooms and with their school librarian. We also show initial findings on the impact of the unit on student interest (N=87), as measured by Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

SIGCSE '20, March 11–14, 2020, Portland, OR, USA

© 2020 Copyright is held by the owner/author(s). Publication rights licensed to ACM. ACM 978-1-4503-6793-6/20/03...\$15.00
<https://doi.org/10.1145/3328778.3366917>

pre- and post- surveys. We conclude with lessons learned about ways to improve the unit and future classroom implementations.

CCS CONCEPTS

• **Social and professional topics~K-12 education** • Social and professional topics~Computational thinking • Social and professional topics~Model curricula • Social and professional topics~Computer science education

KEYWORDS

Elementary School Coding, CS unplugged, Computational Thinking, Expansive Framing

ACM Reference format:

Victor R. Lee, Frederick Poole, Jody Clarke-Midura, Mimi Recker, and Melissa Rasmussen. 2020. Introducing Coding through Tabletop Board Games and Their Digital Instantiations across Elementary Classrooms and School Libraries. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3328778.3366917>

1 INTRODUCTION

Educators throughout the US are working to address the imperative to integrate computer science (CS) and computational thinking into the K-12 curriculum. In Utah, where our project is based, for example, state standards and requirements to include CS in elementary school are still in draft stages, but there is a heightened awareness that fluency with computation is necessary both for future workforce development (e.g., 92% of school principals surveyed in Utah 2014-2016 considered CS necessary and relevant for a multitude of future jobs [14]) and for developing a form of

computational literacy that is becoming an integral part of everyday life [9]. However, few teachers or school district leaders are aware of usable curricular resources to support these aims. Moreover, less than half of principals in Utah felt there was adequate teaching expertise in their school to support computer science [14].

As a result, teachers and school districts are turning to free coding resources such as Code.org and Scratch [24]. However, using these kinds of resources remains unfamiliar territory for most Utah teachers, especially those at the elementary level. An additional difficulty is that Utah has among the lowest per-pupil spending of all the U.S. states [28], meaning funds for computing resources are scarce. As a result, when computers are available for class use, it is typically a mobile laptop cart that is shared across several classrooms.

To address these issues, we have been working with local schools to find ways to integrate computational thinking and coding using existing resources and infrastructure and that do not rely on 1-1 computer to student ratios (e.g., [18]). One existing resource that we seek to leverage is the school library and media center, which often has its own set of computers for students to use.

School libraries are currently in the midst of a dramatic change in terms of the services and experiences they provide to the youth that they serve [17]. While they continue to be sites for schoolwide literacy programs and media instruction, they are also becoming sites for Maker activities where youth can create and customize digitally enhanced artifacts [19]. Relatedly, some are becoming sites for CS instruction and coding [20]. That latter move has been recognized and sanctioned by organizations like Google and the American Library Association, which have launched the “Libraries Ready to Code” initiative (librariesreadytocode.org).

A general premise of our work is that some of the time spent in the school library can be leveraged for coding instruction. However, we recognize that much is already asked of school librarians as educators and that the work of providing coding instruction may be best shared between the librarians and classroom teachers. Furthermore, it is our belief that coding should build upon “unplugged” knowledge and practices that are already familiar to and motivating for students.



Figure 1: Students playing board games in a school library

In our prior design-based research in school libraries [19], we observed that board game play is a popular, organic activity already taking place among students in the school library (Figure 1). Board game play has been noted elsewhere as a practice that has strong affinities with computational thinking [7] as it often involves development and implementation of

algorithms, debugging, and reasoning with conditionals. By leveraging the space of the school library and practice of board game play with an explicit eye toward computing, we hope to help establish and develop a new and more broadly accessible image of computing that is consistent with existing school resources and routines (e.g., weekly library instruction and students’ school library usage).

This experience report describes our efforts to create a unit that spans across classroom and school library settings and builds upon board game play as a source domain for computing knowledge. In this report, we present a description of the board game used and the accompanying unit that we developed. We describe the experiences of three elementary school teachers as they implemented the unit in their classrooms and with their school librarian. We also present initial findings on the impact of the unit on student interest (N=87), as measured by pre- and post- surveys. We conclude with lessons learned about ways to improve the unit and future classroom implementations.

2 LITERATURE REVIEW

2.1 Board games to screen-based coding

As exemplified in curriculum such as *CS Unplugged* [5] and research demonstrating how computing can be expressed through craft activity [6, 10, 18], there is growing awareness that computational thinking can be developed and experienced in non-digital environments and materials. Building on the premise that gaming supports complex forms of learning and reasoning [13, 26], research has demonstrated how tabletop board game play also provides a space for substantial computational thinking to be exhibited. One example comes from the work of Berland and Lee [7] who examined discourse interactions and coded them to identify concrete examples of conditional logic, algorithm building, debugging, simulation, and distributed computational processes. More recently, studies have examined computational reasoning and strategies related to classic tabletop games like Mancala [4, 23].

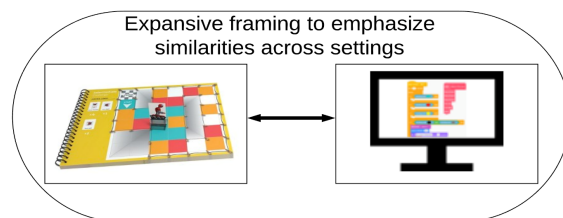


Figure 2: Expansive Framing as Instructional Model

Tabletop board games can be well-suited for learning computational thinking because they have defined rules and structures that players must enact. In a sense, learners execute a program in order to complete the game. At the same time, players engage in extensive sense-making conversations to understand what and how procedures and rules are to be followed, how to optimize strategy, and how to monitor each other’s actions.

The tabletop board gaming industry has taken notice of the current national interest in CS education and published several computer science-oriented board games for consumers (e.g., *Coding Farmers*, *Potato Pirates*, *Robot Turtles*). These games, like other recreational objects such as toys, are often marketed with language stating they will help children learn CS concepts. However, the research to support such claims has been scarce [15]. While some games may involve some aspect of computational thinking, this is not always made explicit nor connected to how one would think about writing or modifying actual computer code. It is possible that computational thinking is encouraged, but some additional work must be done to make that knowledge accessible and for it to transfer beyond the initial setting.

In our approach (see Figure 2), computational ideas are first encountered through tabletop board games where students can use their prior knowledge and ease with tabletop gaming mechanics to cue relevant ideas for core computational ideas, such as looping or conditionals. We then look to Engle’s model of expansive framing [11] to support transfer of those computational ideas to a digital environment. This model emphasizes instructional linkages being made between one domain (i.e., the tabletop board game) and another (i.e., specially designed Scratch project shells with partially complete code blocks) to the point that the reasoning activities and different contexts are seen as instantiations of the same encompassing context (i.e., computational thinking). Linkages are established largely by activity design and specific classroom discourse moves that promote student authorship, awareness that knowledge will be used in different contexts, and connections between different contexts. We hypothesize that this instructional approach of linking unplugged tabletop activities with screen-based programming activities where a form of computer-based coding is explicitly encountered, can overcome some of the limited computational learning observed when implementing unplugged activities alone [12].

2.1 Libraries as part of the learning ecology

Connected Learning, a perspective gaining prominence in education research, posits that much of the powerful learning taking place for youth today involves interest-driven pursuits that span across a range of different spaces [16, 21]. For instance, a life-wide learning ecology [3] might include experiences that take place at home, in school, and in other learning spaces, such as libraries. A student may learn web development from a combination of classroom instruction, workshops at an afterschool club at the local library, and from time at home using mobile devices. One of the greatest impacts of Connected Learning as a prominent perspective in education is resurgent interest in designing for youth learning in library settings [19, 27]. In this view, libraries are a place where interests are developed and nurtured.

To date, the most prominent research on Connected Learning for youth has emphasized public libraries [2], rather than school libraries, even though school libraries comprise over 98,000 of the almost 120,000 libraries in the United States [1]. These settings have unique features and affordances that, if properly utilized, can make them very supportive of school-based STEM learning [27]. For instance, co-designing and co-

teaching with classroom teachers is common practice of school librarians. While librarians do have their own content to cover, many schools are rethinking what roles physical libraries serve. This has made working with school librarians especially attractive to our school district partners who have already begun to deploy librarians as early childhood coding educators through lessons using Scratch Jr.

Another important aspect of school libraries is their role in collections management and circulation. In framing learning as taking place across a broader ecosystem of spaces, including both the school and home [3], there is an opportunity for educational board games to circulate and go home with youth on loan. The benefit of such an approach is that the circulated board game can serve as a boundary object [25] between the classroom, the library, and the student’s home. It provides opportunities for families to discuss what is being learned in school and for family members to share their expertise or for students to “learn by teaching.” By using libraries as a critical location for these board games to be played and circulated, we can begin to extend the boundaries of when and where relevant computational thinking can be developed and supported. This is, however, dependent on when and how these materials circulate and how involvement from family members can be encouraged and supported outside of the library and the school.

Thus, our goals are to design school-based computer science instruction that incorporates the resources of the school library. We are specifically concerned with the library’s ability to serve as a locus for interest development and maintenance. We are also interested in the ability for computational learning objects, in this case, CS board games, to circulate and go home with students.

3 METHODOLOGY

3.1 Design spanning the class and library

With input from teachers and librarians, we designed an eight-week unit using the unplugged-to-plugged sequence that includes CS board games and Scratch shells of those same games. The general sequence of the unit was for students to first learn and play a CS board game (two weeks), then work in Scratch instantiations of the board game where they make edits and amendments (three weeks), and then design their own levels of the same board game (two weeks) which they then share with their teachers and classmates (one week).

Instruction was organized such that students had a total of six preparatory lessons from their classroom teacher that lasted for 10-20 minutes. For instance, on the first day, the lesson introduced the board game and the basic rules. Then, during their weekly scheduled library time, the librarian would review the activity for the day and build on what was introduced in class. During the remainder of the library time, approximately 20 minutes, the students played the game in pairs while the librarian circulated around the room and assisted students with specific levels and answered questions. In later weeks, the classroom teachers introduced Scratch in a

mini presentation to the class. For instance, they showed how to operate the controls of the Scratch instantiation of the board game, following which the students played the digital versions of the same board game in the library. Or, before students designed their own levels of the board game, the teacher demonstrated how to use paint tools in Scratch to create custom levels.

3.1.1 The Computer Science Board Game. The board game used in this project was *//CODE: On the Brink* designed by Mark Engleberg and published by Thinkfun. *On the Brink* is one of three single-play or collaborative-play puzzle games in the *//CODE* series. The premise of the *On the Brink* is a robot needing instructions to navigate a series of rooms, each with colored tiles arrayed on a six-by-six grid. Each color of tile is associated with a sequence of actions that the player specifies by placing cards on a control panel (Figure 3). For instance, the instructions associated with a blue tile may be to move forward and then move forward again while the instructions for a red tile may be to move forward only once. The movement options involve moving forward, turning 90 degrees to the right or left, or doing nothing. All actions in a sequence must be executed before the next tile’s instructions are executed. In more complex levels, the additional actions are available, including moving forward two steps, moving backward, making a long turn left, or making a long turn right, the last of which involved moving forward, turning 90 degrees, and moving forward again. The game has forty levels of increasing difficulty. As advertised on the game box, the game is intended to teach procedures and other computational ideas.



Figure 3: Sample level from //On the Brink

We built a Scratch instantiation in which a sprite moved to various colored tiles and performed similar actions to the robot. To move the sprite, the player assigns moves and turns to custom procedures associated with specific colors on the board. The board was designed using a colored backdrop in the Scratch interface. To check what color tile was being occupied, a series of conditional statements was embedded in a forever-loop to continuously check what color the sprite was currently touching. Replicas of each board game level and a template for custom level creation were created. The main challenge for students using the Scratch shells was to populate and build procedures and to place procedures in conditionals (Figure 4).

3.2 Research Activities & Context

3.2.1 Participating Classrooms. Three fifth-grade teachers from the same rural-serving school implemented the unit concurrently, with each teacher doing their lesson and their class’s library visits on a different day of the week. Interviews of teachers showed that they differed in their prior experience and comfort levels with the unit (Figure 5). Shawn (a pseudonym) was quite familiar with block-based programming and expressed the least concern with being able to teach the unit. Shelly, the most junior teacher of the three, had some undergraduate exposure to web design but had not taught coding previously. However, she was very confident in her ability to both

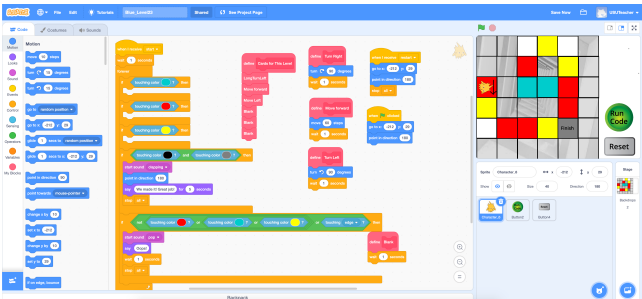


Figure 4: Example Scratch shell

learn and teach it. Mandy, a veteran teacher, had some prior experience through district professional development activities with code.org, but was otherwise known by her colleagues and students to find all kinds of technology to be very difficult to use in her teaching. She expressed the least confidence in her ability to teach the unit.

Teacher	Prior Coding Experience	Quote
Shawn	Was familiar with block-based coding and coached a small extracurricular robotics coding team after school	"I can block program pretty well, I can see things well, and figure out how things work and do that pretty well"
Shelly	Had some undergraduate exposure to web-based coding (HTML) but had not taught coding previously nor done coding independently	"I think more than anything, it's [teaching computing is] a confidence issue... I'd say I'm like a five out of ten [in skill]? Can I [teach computing]? I know I can. Ten out of ten, I definitely could do it, definitely can."
Mandy	Had brief exposure to code.org activities but had not worked with code, felt challenged with technology	"so, me personally, I need to know what I'm doing before I teach it to the kids... [but] I'm not confident in it [programming]"

Figure 5: Teachers background and comfort teaching computing

3.2.2 Collected Data. Video recordings of class library sessions were collected for all days of instruction (except one for Mandy). In the library, two video cameras were set up and each recorded a group of students doing their library activities. The teachers and the librarian also participated in pre- and post-interviews.

In addition, a pre- and post-survey related to interest in CS was administered to all students. The survey consisted of 32 Likert-scale items and measured intrinsic and future interest in

CS as well as perceived family member interest. The family member interest was obtained because of the possibility that the board games would be checked out from the library and played at home with family members.

4 RESULTS

4.1 Quantitative Analysis

Given the short duration of the unit, its distribution across classroom and library time, and the limited number of instructional minutes, our primary aim was to increase intrinsic interest in CS by helping students recognize that knowledge developed from playing a board game could be directly used to do basic coding in Scratch. Four items were used to measure intrinsic interest in CS, with high reliability on both pre ($\alpha = .92$) and post ($\alpha = .93$) surveys.

The three classes did not differ significantly in their initial levels of intrinsic interest ($\chi^2 = 2.43, p = .296$) (Figure 6). We found that despite having the teacher who was the most experienced with block programming, students in Shawn's class reported a significant decrease in their intrinsic interest in computing. Students in Mandy's class, the teacher who reported the least experience and the greatest discomfort, stayed about the same. Shelly's students reported a significant increase. A Kruskal-Wallis test confirmed a statistically significant difference between student gains in different classes ($\chi^2 = 19.77, p < .001$). A Dunn's post-hoc test showed that Shelly's class had significantly higher gains than both Mandy's and Shawn's classes, and that declines in Shawn's class were significantly lower than Mandy's class (see Figure 6).

	Pre ($\alpha = .92$)			Post ($\alpha = .93$)			N	Post-Pre Z	Dunn's Post-Hoc Test (Z)	
	M	Med	SD	M	Med	SD			Mandy	Shelly
Mandy	4.30	4.50	1.56	4.26	5.00	1.69	30	-0.93		
Shelly	3.84	3.88	1.76	4.16	4.50	1.66	28	2.25*	2.24*	
Shawn	4.45	5.00	1.61	3.45	3.44	1.78	29	-3.49***	2.28*	8.71***

Figure 6: Composite scores for students' pre-post survey of intrinsic interest in computing. Scores range from 1 to 6. Wilcoxon Signed-Ranked Test (Z) was used due to non-normal distribution of the data. Dunn's post-hoc test compares gain scores between teachers. * $p < .05$, ** $p < .01$, * $p < .001$.**

Interestingly, we found that those who took the board game home ($N=21$) reported higher levels of intrinsic interest ($M=5.20$) on the post-survey compared to those who didn't ($M=3.59$). Moreover, taking the board game home was positively and significantly correlated ($r=.30, p=.005$) with higher levels of perceived father support for computing. We hypothesize that by taking the games home and playing or talking about them with their families, this particular group of students received some positive messages about CS, coding, the project, and/or the game.

While these findings are encouraging and motivate us to explore more ways to increase game circulation from the library, we were still unclear as to why there were such

differences between Shawn, Mandy, and Shelly's classes. We turned to a small qualitative analysis to inquire further.

4.2 Qualitative Analysis

As implemented, the school librarian enacted the same lesson three times each week, once for each of the three fifth-grade classes. We observed that the librarian was consistent in her instruction, keeping to the same pacing and language regardless of which class she was teaching. However, we saw differences in how the classroom lessons were enacted by Shawn, Mandy, and Shelly. These differences appeared in both the amount of time spent on lessons and in the language used during the lessons. This led us to more closely examine differences in each teacher's implementation.

By and large, the three teachers adhered to the prescribed sequence of activities in the lesson plans we developed. However, they did most of the talking during their lessons rather than leaving things as open discussions with the students. In light of that, we decided to analyze teacher talk during three lessons. These lessons were chosen because: 1) we had video data for all teachers, 2) the data were comparable in length and content, and 3) they were all introducing a procedure in Scratch.

We performed a grounded discourse analysis to compare teacher utterances. We note that teacher talk has been extensively examined in the literature, whether it is in the form of specific discourse patterns when interacting with students such as inquire-respond-evaluate (IRE) [8] or in analysis of specific discursive moves such as revoicing [22]. Our interest in teacher talk is in statements the teachers made as they led a lesson and how the structure and content of these statements varied across teachers teaching the same lessons. We posit that it was through teacher talk that variation in implementation was largely realized, and thus was a contributor to the difference in student post-survey results between classes.

Qualitative coding involved assigning utterances with a label based on content and function. The coding scheme was developed bottom-up from the data. To illustrate, codes were assigned when teachers provided direct instruction demonstrating what to do in Scratch, and when teachers made connections between Scratch to some other environment to help familiarize students with what they would be doing. Other codes included establishing expertise in coding, whether it was stating that some outside coder was an expert, the teacher herself was an expert, or the students were experts, and narrating the lesson trajectory where teachers would provide an overview of what the class was doing or going to be doing. These codes were assigned by one analyst who then met with a second analyst who reviewed and offered modifications to assigned codes. This was then presented to a larger research group for review.

4.2.1. Direct Instruction. Students in Shawn's class received the most direct instruction and reported the greatest decrease in intrinsic interest scores. Specifically, we coded 40 instances

of direct instruction by Shawn, whereas Mandy and Shelly had 23 and 24 respectively. Thus, Shawn made almost double the number of direct instruction utterances. Moreover, his direct instruction tended to be lengthier and involved monologues, whereas Mandy and Shelly were briefer. In analyzing the classroom video, we noted that the students appeared to be more attentive to the instruction in Mandy and Shelly's classes whereas the students in Shawn's class appeared distracted and to be occupied with other things during direct instruction. Furthermore, there were occasions when Shawn provided instruction that the librarian was to provide, suggesting the delineation between classroom and library instruction was not always observed.

4.2.2 Connections. All three teachers made some effort to connect Scratch with something more familiar and relevant to the students. For Mandy and Shelly, it was to Minecraft, Code.org, and the Sphero toy coding environments. For example, Shelly described Scratch as: "it's a little bit like Minecraft and some of those other building games where you eventually are able to get certain rules and things to go." In contrast, Shawn likened Scratch coding to making a calculator program and how the different buttons on a calculator have different commands and operations associated with them: "Okay, just as an example, a calculator, if I pull the calculator up on my computer right now, what that is, is a set of buttons just like the buttons you had on this..." Mandy and Shelly's connections for the students were to more playful environments whereas Shawn's was more complicated and abstract. This may have been a situation where Shawn's more advanced knowledge of programming led him to pick a more complex example that was less relevant to most students.

4.2.3 Narrating lesson trajectory. Shelly more frequently narrated the lesson trajectory—stating what activities she was going over and what the students would be doing—relative to the other teachers (12 times with 24 instances of direct instruction, whereas Shawn narrated the trajectory 10 times with 40 instances of direct instruction). This suggests that Shelly expressed the goal and direction of the lessons more clearly and explained how the different activities cohered and connected with each other. We believe that providing this information allowed students to better see the purpose of what they were doing in class.

5 LESSONS LEARNED

Having completed this model of instruction in three classes, we identified a few things that need further improvement and that we plan to change for our next implementation. First, we decided that there was too much time between playing the board game and coding it in Scratch. Some students forgot the core game mechanics by the time they were to author their own levels of the game. Thus, we are resequencing the unit to move game re-design earlier, more immediately after initially learning the game. There were also some occasions where the

instruction provided in the classroom and in the library was redundant rather than complementary, as was the case for Shawn. More coordination needs to be established with the split instruction, which was necessarily split due to time constraints from the teachers and the librarian. Our lessons are trying to better delineate what happens in the library and what happens in the classroom. Finally, circulation of the game was high at the beginning of the unit but dropped abruptly. We intend to explore more ways to encourage game circulation in later weeks so that if there is indeed a positive effect from taking the game home, it can impact more students.

6 CONCLUSION

In this report, we have explored a form of unplugged-to-plugged instruction that involves classrooms, school libraries, computer science board games, and Scratch instantiations of those board games. This approach was taken to leverage existing resources at the school and pave a way for a district partner that was seeking to involve librarians in computer science education. Our initial outcomes from a first implementation found a mix of changes in intrinsic motivation toward computer science. A qualitative analysis suggested that these may be due to different amounts of direct instruction, especially redundant instruction, the kinds of connections that were made to Scratch, and how much the lesson trajectory was narrated. We found that there were some challenges in how the unit was sequenced, in how classroom instruction and library instruction were coordinated, and in how board games circulated only at the beginning of the unit. In the coming year, we will be implementing this classroom-library spanning unit at another school. It is our belief that by using our model in multiple schools in our partnering district, we demonstrate how school libraries can be involved in CS education. A future paper will share more about our successes and other lessons that we have yet to learn in our next iteration.

ACKNOWLEDGMENTS

This work was supported by National Science Foundation Grant no. DRL-1837224. Opinions, findings, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agency. We thank the partnering district and especially the participating teachers, school librarian, and students. Thanks also to Heidi Vincent and Kholtny Barney for their assistance in data collection. Helpful comments were provided by the anonymous reviewers.

REFERENCES

- [1] American Library Association. Number of Libraries in the United States. Chicago, IL, 2015.
- [2] Kimberly Austin, Stacy B. Ehrlich, Cassidy Puckett, and Judi Singleton. *YOUmedia Chicago Reimagining Learning, Literacies, and Libraries: A Snapshot of Year 1*. Consortium on Chicago School Research at the University of Chicago Urban Education Institute, Chicago, IL, 2011.

- [3] Brigid Barron. Interest and Self-Sustained Learning as Catalysts of Development: A Learning Ecology Perspective. *Human Development*, 49, 4 (2006), 193-224.
- [4] Rebecca Y. Bayeck. An Alternative Learning Space: The Case of Mancala Board Game Players. In *Proceedings of the Annual Meeting of the American Educational Research Association*, New York, 2018.
- [5] Timothy C. Bell, Jason Alexander, Isaac Freeman, and Mick Grimley. Computer Science Unplugged: School students doing real computing without computers. *Journal of Applied Computing and Information Technology*, 13, 1 (2009), 20-29.
- [6] Sophia Bender and Kate Samson. Re-Crafting Mathematics: Early Lessons Learned at the Intersection of Textile Crafts and Math. In *Proc. of Fablearn 2015* Stanford, CA, 2015.
- [7] Matthew Berland and Victor R. Lee. (2011). Collaborative strategic board games as a site for distributed computational thinking. *International Journal of Game-Based Learning*, 1(2), 65-81. doi:10.4018/ijgbl.2011040105
- [8] Courtney B. Cazden. Classroom discourse: the language of teaching and learning. Heinemann, Portsmouth, NH, 2001.
- [9] Andrea A. diSessa. Changing minds: Computers, learning, and literacy. MIT Press, Cambridge, MA, 2000.
- [10] Michael Eisenberg. Bead Games, or, Getting Started in Computational Thinking Without a Computer. *International Journal of Computers for Mathematical Learning*, 15, 2 (2010), 161-166.
- [11] Randi A. Engle, Diane P. Lam, Xenia S. Meyer, and Sarah E. Nix. How Does Expansive Framing Promote Transfer? Several Proposed Explanations and a Research Agenda for Investigating Them. *Educational Psychologist*, 47, 3 (2012/07/01 2012), 215-231.
- [12] Yvon Feaster, Luke Segars, Sally K. Wahba, and Jason O. Hallstrom. Teaching CS unplugged in the high school (with limited success). In *Proc. of the 16th annual joint conference on Innovation and technology in computer science education*, ACM, Darmstadt, 2011.
- [13] James P. Gee. *What video games have to teach us about learning and literacy*. Palgrave Macmillan, New York, 2007.
- [14] Google Inc. & Gallup Inc. K-12 Computer Science Education: States Reports. Results from the 2014-2016 Google-Gallup Study of Computer Science in U.S. K-12 Schools. 2017.
- [15] Megan Hamilton, Jody Clarke-Midura, Jessica F. Shumway, and Victor R. Lee (in press). An Emerging Technology Report on Computational Toys in Early Childhood. *Technology, Knowledge and Learning*.
- [16] Mizuko Ito, Kris Gutiérrez, Sonia Livingstone, Bill Penuel, Jean Rhodes, Katie Salen, Juliet Schor, Julian Sefton-Green, and Craig S. Watkins. *Connected Learning: An Agenda for Research and Design*. Digital Media and Learning Research Hub, Irvine, CA, 2013.
- [17] Victor R. Lee and Abigail L. Phillips (Eds.). (2018). *Reconceptualizing Libraries: Perspectives from the Information and Learning Sciences*. New York, NY: Routledge.
- [18] Victor R. Lee and Mimi Recker. (2018). Paper Circuits: A Tangible, Low Threshold, Low Cost Entry to Computational Thinking. *TechTrends*, 62(2), 197-203. doi:10.1007/s11528-017-0248-3
- [19] Victor R. Lee, Mimi Recker, and Abigail L. Phillips (2018). Conjecture Mapping the Library: Iterative Refinements Toward Supporting Maker Learning Activities in Small Community Spaces. In J. Kay & R. Luckin (Eds.), *Rethinking Learning in the Digital Age: Making the Learning Sciences Count, 13th International Conference of the Learning Sciences (ICLS) 2018* (Vol. 1, pp. 320-327). London, UK: ISLS.
- [20] Victor R. Lee and Heidee Vincent (2019). An Expansively-framed Unplugged Weaving Sequence Intended to Bear Computational Fruit of the Loom. In P. Blikstein & N. Holbert (Eds.), *Proceedings of FabLearn 2019* (pp. 124-127). New York, NY: ACM.
- [21] Crystle Martin. Connected learning, librarians, and connecting youth interest. *J. Res. Libr. Young Adults*, 6 (2015).
- [22] Mary C. O'Conner and Sarah Michaels. Aligning Academic Task and Participation Status through Revoicing: Analysis of a Classroom Discourse Strategy. *Anthropology & Education Quarterly*, 24 (1993), 4.
- [23] David Phelps, John Benner, Joshua Munsel, and Gabriel de los Angeles (2017).. How K-5 Students Leverage Computational Practices and Products to Become Expert Mancala Players. In *Proceedings of the 2017 Annual Meeting of the Jean Piaget Society*. San Francisco, CA.
- [24] Mitchel Resnick, John Maloney, Andrés Monroy-Hernandez, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. Scratch: programming for all. *Commun. ACM*, 52, 11 (2009), 60-67.
- [25] Susan L. Star and James R. Griesemer. Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. *Social Studies of Science*, 19, 3 (August 1, 1989 1989), 387-420.
- [26] Constance Steinkuehler and Sean Duncan. Scientific Habits of Mind in Virtual Worlds. *Journal of Science Education and Technology*, 17, 6 (2008), 530-543.
- [27] Mega M. Subramaniam, June Ahn, Kenneth R. Fleischmann, and Allison Druin. Reimagining the Role of School Libraries in STEM Education: Creating Hybrid Spaces for Exploration. *The Library Quarterly: Information, Community, Policy*, 82, 2 (2012), 161-182.
- [28] US Census Bureau. Annual Survey of School System Finances. 2017.