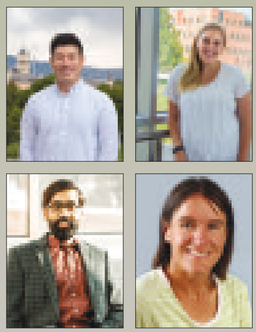
FEATURE ARTICLE



"There are a number of ways in which CS topics can be made more accessible to students and librarians alike."

Unplugged-to-Plugged Computer Science at the Library

VICTOR R. LEE, AUBREY ROGOWSKI, UMAR SHEHZAD, AND MIMI RECKER

Computer science (CS) is increasingly becoming a topic for teacher librarians to offer in their libraries. The American Library Association (ALA) has drawn attention to both the importance and opportunity for pursuing this through their *Libraries Ready to Code* initiative (Braun & Visser, 2017), launched with initial support from Google. Since its inception, resources that address CS, programming, coding, and computational thinking have been compiled for interested librarians and hosted on the ALA website (http://www.ala.org/tools/readytocode/home).

To further this momentum, at the end of 2019, the American Association of School Librarians (AASL) crosswalked some alignments between AASL standards and the Google CS First curriculum (https://standards.aasl.org/project/crosswalks/). The purpose of these efforts has been to highlight connections between CS education and school library learning. Yet, even though there are areas of overlap, many teacher librarians may find CS education to be unfamiliar and intimidating territory. It need not be that way. There are a number of ways in which CS topics can be made more accessible to students and librarians alike.

COMPUTER SCIENCE UNPLUGGED

First, there needs to be a recognition that knowing some of the important ideas and practices related to CS do not require months or years of computer programming experience nor hours of online tutorials. Instead, for example, computational thinking has been identified and described as an important skill set that was more about conceptualizing and solving problems than knowing any particular programming language (Wing, 2006). At its core, computational thinking is centered on breaking down a problem and designing solutions and systems in ways that could be executed by a computer or by another human. For instance, given the task of sorting a large pile of books in alphabetical order in a limited amount of time, how would we best get that work done? Would it be most efficient if each book was taken, one at a time, and placed it on the shelf in its proper location? Or, would it be better to go through the pile first, arrange the books into piles based on their general placement in the alphabet, and then do some more precise alphabetization of those groups?

Thinking about that example, many teacher librarians may be surprised to realize that in some of their most routine daily work, they are doing computational thinking. Sorting and optimization are precisely the kinds of problems that appear in CS classes around the country. These can be discussed as part of the topic of algorithms, or creating lists of rules, that one specifies for a computer when programming.

Ideas central to computing can be meaningfully explored without use of a digital computer—a realization that led to a new genre of computing education often referred to as "unplugged" (Bell, Alexander, Freeman, & Grimley, 2009). As the name implies, unplugged computing does not involve anything digital. In some circles, researchers have begun to identify sophisticated unplugged computing activities that can range from game play (Berland & Lee, 2011) or beadwork (Eglash, 2007) to styling hair (Eglash & Bennett, 2009). Just like with our book-sorting example above, it is possible to look at familiar activities and see them as being broadly approachable options for unplugged learning activities. Through these activities, students can work on developing their computational thinking and begin to understand some big ideas in CS.

DOABLE COMPUTATIONAL THINKING

Several unplugged computational activities have been identified, along with makerspace and making activities, and made publicly available for both public and teacher librarians (Lee & Recker, 2018; Lee & Vincent, 2019; Rogowski, Phillips, Recker, & Lee, 2019).

As shown in the illustrated examples below, the researchers encourage teacher librarians to pursue an approach that involves an unplugged-to-plugged sequence of instruction. This involves identifying an unplugged activity that supports or involves computational thinking and then following that with activities that help students use those ideas in a computational setting.

For instance, if we apply the book-sorting example, students could start by physically sorting piles of books while discussing sorting strategies and time one another using those strategies. Then, as a later activity, those students might work with an app that allows them to program those strategies in an online environment that lets them change the number of 'books' that are to be sorted and see how long it takes. An important part of achieving student success is to make explicit the connections between the unplugged and the plugged activity, and to create opportunities for students to reflect on how this concept applies broadly to both CS as well as to many other activities. In addition, it is important to provide an opportunity for students to showcase the programs they authored.

Knowing that budgets are limited, free computing tools like *Scratch* (Resnick et al., 2009) can be used to introduce programming in a very approachable way using colorful drag-and-drop blocks. The examples are publicly available projects that have been largely authored by K-12 students. In particular, the researchers of this article have also created and made available their own Scratch programs to fit in specific unplugged-to-plugged learning sequences, described in the two examples below. It is important to note that using Scratch is not required. Many free apps and applets that are available online can serve as the "plugged" part of the unplugged-to-plugged learning sequence.

RESOURCES FOR UNPLUGGED/ PLUGGED COMPUTER SCIENCE, CODING, AND COMPUTATIONAL THINKING

Online Resources:

Libraries Ready to Code: An Initiative of the American Library Association: http://www.ala.org/tools/readytocode/home

National School Library Standards Crosswalk with Code with Google's CS First Curriculum: https://standards.aasl.org/wp-content/uploads/2019/11/aasl-standards-crosswalk-cs-first.pdf (linked from the Crosswalks page: https://standards.aasl.org/project/crosswalks/

Making in Small Town Libraries: A Resource Site for Public and School Libraries (https://slli.usu.edu/)

Looming Code: https://slli.usu.edu/looming-code/

Board Games:

Robot Turtles by ThinkFun

Coding Farmers by MathAndCoding.org

//CODE: On the Brink by ThinkFun

US National Science Foundation: Unplugged to Plugged Computing (classroom-school library curriculum using On the Brink): https://sites.google.com/view/tabletop2screenscurriculum.

As originally designed, the unplugged-to-plugged activities described below were intended to be done in person at a physical school library space. However, when it is not possible for students, librarians, and teachers to convene in person, there are ways that these activities can be done remotely.

The essential components involve making sure students have access to unplugged materials (which can be provided by the library as circulating items or in librarian-prepared construction kits) and providing students with access to the plugged activities, which can be accessed online. It is also wise to come up with ways to provoke reflection from students about the big ideas that apply in both the unplugged and plugged settings and to provide them with an opportunity to showcase their programs or created artifacts. This can take place during a video meeting led by the teacher librarian, on a school library web page, or on a printout that accompanies circulated materials.

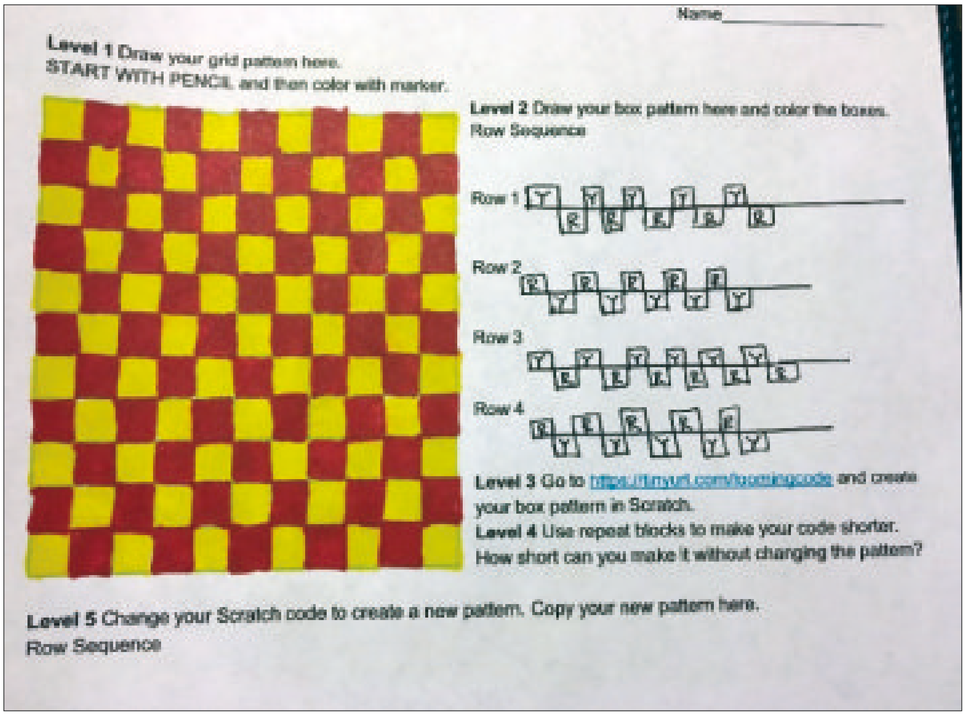


Figure 1: Replacing weaving pattern and identifying row sequence on a worksheet

LOOMING CODE: BLOCK CODING TO CREATE WEAVINGS

Crafting is a common hobby and one that many teacher librarians know and enjoy. Crafting can also serve as an entry point and tangible way to teach and learn about computing concepts. Some crafting skills, such as weaving, lend themselves especially well as pathways to understanding computing concepts, such as looping and repeating processes.

The researchers found that teachers identified looming as a popular activity for youth in the library, so looming was used to develop a library activity called Looming Code (see https://slli.usu.edu/looming-code/), which uses the crafting medium as a way to engage students in both tangible and digital computational thinking and programming.

In this activity, students begin with a physical weaving. They start by identifying the patterns in each row ('loops' in CS) in the physical weaving and then replicate the pattern by coloring a grid worksheet. Second, students write out each row sequence in terms of the represented colors (e.g., green, blue, green, blue) on a worksheet. Third, they make the pattern readable by labeling the vertical strings in the weaving as "up" and the horizontal strings in the weaving as "down" and writing the row sequence pattern (e.g., row 1, row 2, row 1, row 2) as repeating processes (see Figure 1).

Fourth, moving to the digital realm, students mimic this same pattern using the "Looming Code" shell developed in Scratch to support this activity (see Figure 2).

Fifth, students modify (or "remix") the pattern to create a new loom weaving pattern. Finally, students physically weave the new pattern that they designed and programmed in Scratch by using simple craft materials: pipe cleaners and a shower comb (see Figure 3). A shower comb and pipe cleaners are much easier for youth to manipulate than yarn, making this craft accessible to everyone.

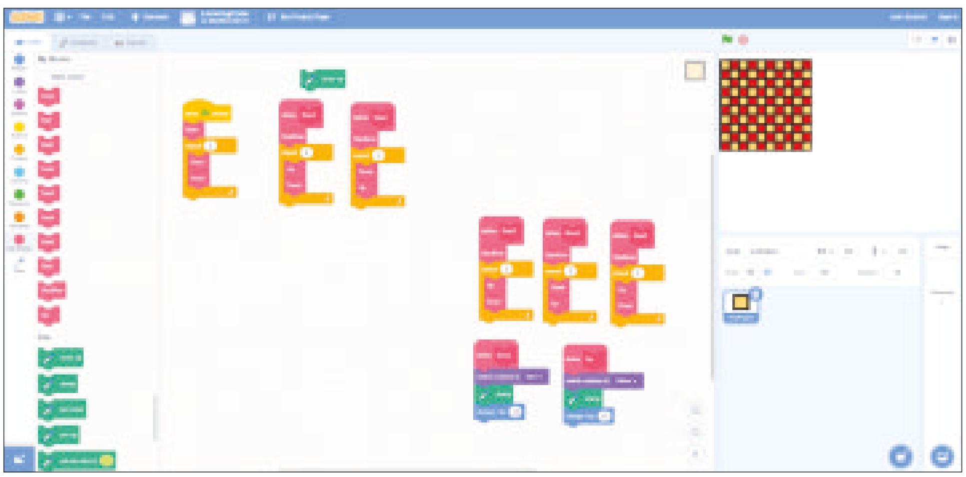


Figure 2: Mimicking a weaving pattern in the "Looming Code" Scratch shell

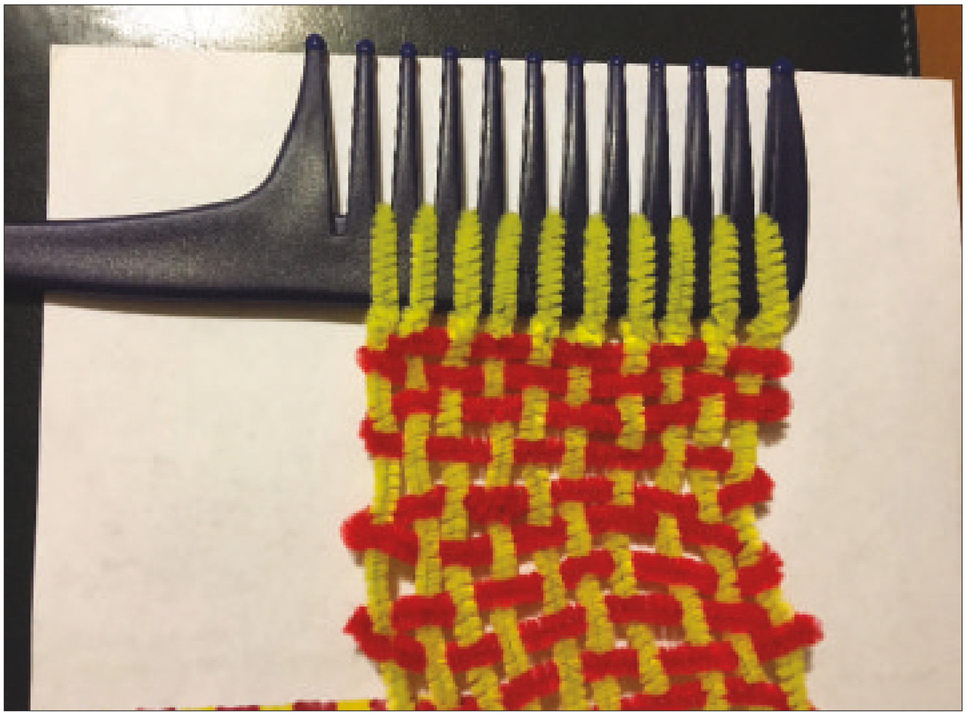


Figure 3: Example of weaving using shower comb and pipe cleaners

Due to the unplugged and plugged nature of this activity, it can also be done in person or in a remote setting. The physical materials (shower comb, pipe cleaners) can be provided as a check-out kit, and all other digital resources can be featured on the library website. The physical weavings created by students can be put on display in the library, or pictures of youth-created weavings can be shared on the library's social media accounts. In addition, youth can share their Scratch account usernames and "remix" each other's weaving patterns.

TABLETOP TO SCREENS: BOARD GAMES TO PROMOTE COMPUTATIONAL THINKING

Tabletop board games are commonly found in school libraries and offer a popular pastime for all students (see Figure 4).

At a time when consumption of digital media is raising concerns of social isolation, the renaissance of the board games industry (Roeder, 2015) seems timely. Board game play encourages social interaction and also promotes important 21st century skills such as problem solving and collaboration. Recent work has also shown how board games can serve as a rich source of computational thinking (Berland & Lee, 2011). Moreover, there are many board games on the market that are specifically designed to involve CS concepts through their rules and game mechanics.

One such game is *Robot Turtles*, in which players use a series of simple "move" cards (Forward, Rotate Left, and Rotate Right) to create a program that indicates in which direction a player wants their turtle to move. These simple instructions use physical cards to introduce players to fundamental programming concepts such as algorithm design, abstraction, and debugging in a fun and unintimidating setting (Poole, Clarke-Midura, Rasmussen, Shehzad, & Lee, 2020). Other board games such as *Coding Farmers* can also introduce players to programming syntax, because they include actual Java code snippets on playing cards that the players use during game play.

Another game, *//CODE: On The Brink* (published by ThinkFun), was used to develop a computing activity that can be co-taught by a teacher librarian and elementary school teacher (Lee, Poole, Clarke-Midura, Recker, & Rasmussen, 2020). In this activity, the classroom teachers first introduce students to the unplugged board game, its rules, game mechanics, and some of the underlying computing concepts (Figure 5).

During their next school library time, the students play the board game in the school library. Subsequently, they play the same game using a plugged Scratch version of the game, building on what they learned during board game play. Finally, students au-

Figure 4: Tabletop board game play taking place in school libraries

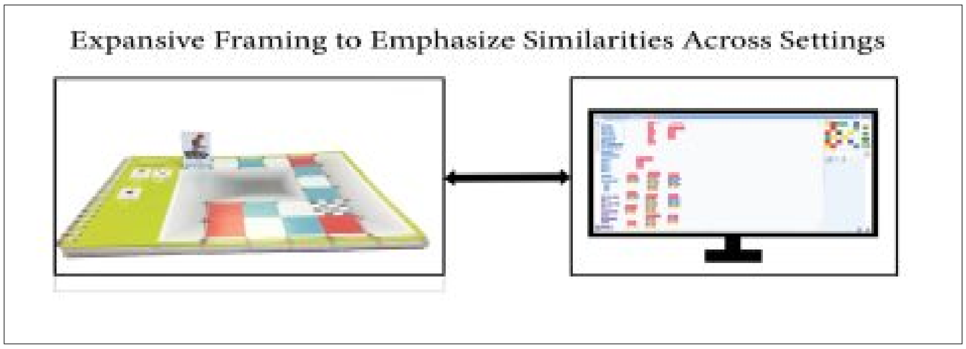


Figure 5: CS curriculum that uses "//CODE: On The Brink" Board game and its Scratch levels

thor their own game levels in Scratch and challenge their classmates to play them. As this activity shows, tabletop board games have a role in making computing more approachable and fun for students—all before they are introduced to the Scratch programming activity. This classroom-school library curriculum is available for free via https://sites.google.com/view/tabletop2screenscurriculum.

SUMMARY

Libraries have long been places for all kinds of literacy activities. Add to this list a new kind of literacy—computational literacy—which is rapidly gaining traction throughout the U.S. School libraries can be a fruitful place for introducing computational thinking and programming, and these low-barrier activities, which rely on an unplugged to plugged sequence, can support a wide variety of students learning, tinkering, creating, and sharing in participatory ways.

This work was supported by the Institute of Museum and Library Services grant number RE-31-16-0013-16 and National Science Foundation grant DRL-1837224. The opinions expressed herein are those of the authors and do not necessarily reflect those of either funding organization.

REFERENCES

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The Journal of Applied Computing and Information Technology, 13*(1), 20-29.

Berland, M., & Lee, V.R. (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

Braun, L., & Visser, M. (2017). Ready to code: Connecting youth to CS opportunity through libraries. *Overview of OITP Perspectives No. 5.* American Library Association. Retrieved from http://www.ala.org/tools/sites/ala.org.tools/files/content/Ready_To_Code_OVERVIEW_FINAL.pdf

Eglash, R. (2007). Ethnocomputing with Native American design. In L.E. Dyson, M. Hendriks, & S. Grant (Eds.), *Information Technology and Indigenous People* (pp. 210-219). IGI Global. https://doi.org/10.4018/978-1-59904-298-5.ch029

Eglash, R., & Bennett, A. (2009). Teaching with hidden capital: Agency in children's computational explorations of cornrow hairstyles. *Children, Youth and Environments*, *19*(1), 58–73.

Lee, V. R., Poole, F., Clarke-Midura, J., Recker, M., & Rasmussen, M. (2020). Introducing coding through tabletop board games and their digital instantiations across elementary classrooms and school libraries. *SIGSCE: Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 787-793. New York: ACM. https://doi.org/10.1145/3328778.3366917

Lee, V. R., & Recker, M. (2018). Paper circuits: A tangible, low threshold, low cost entry to computational thinking. *TechTrends*, *62*(2). https://doi.org/10.1007/s11528-017-0248-3

Lee, V. R., & Vincent, H. (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.

Poole, F.J, Clarke-Midura, J., Rasmussen, M., Shehzad, H., & Lee, V.R., (2020). Tabletop games designed to promote computational thinking [Manuscript submitted for publication].

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver J., Silverman, B., & Kafai, Y.B. (2009). Scratch: programming for all. *Communications of the ACM, 52*(11), 60-67. doi:10.1145/1592761.1592779

Roeder, O. (2015, August 18). Crowdfunding is driving a $196 million board game renaissance. *FiveThirtyEight*. Retrieved from https://fivethirtyeight.com/features/crowdfunding-is-driving-a-196-million-board-game-renaissance/

Rogowski, A., Phillips, A., Recker, M., & Lee, V.R. (2019, April 5-9). *Design capacity for informal learning within schools: An analysis of school librarians and maker activities.* [Paper session]. American Educational Research Association Conference, Toronto, ON, CA

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

Victor R. Lee, Ph. D, is an associate professor in the Graduate School of Education at Stanford University. His research focuses on using emerging digital technologies to support STEM learning and educational practices. His current research projects include supporting the development of STEM Maker programs for school libraries and using board games and block-based coding to teach elementary school students computer science, and teaching and learning about data in K-12 settings. Lee's most recent book is *Reconceptualizing Libraries: Perspectives from the Information and Learning Sciences.*

Aubrey Rogowski, M. Ed, is a doctoral candidate specializing in STEM-rich Making in the Department of Instructional Technology and Learning Sciences at Utah State University. She holds degrees in Elementary Education and Instructional Technology and Learning Sciences, and her research focuses on how to integrate technology and STEM-rich Maker activities to promote student learning.

Umar Shehzad, MS, is a doctoral candidate specializing in computer science in the Department of Instructional Technology and Learning Sciences at Utah State University. His current research projects include supporting the development of STEM Maker programs for school libraries and using board games and block-based coding to teach elementary school students computer science.

Mimi Recker, Ph. D, is a professor in the Department of Instructional Technology and Learning Sciences at Utah State University. Her research focuses on studying how people learn with technology, with an emphasis on computer science education, learning analytics, and teacher professional development. Her current research projects include supporting the development of STEM Maker programs for school libraries, using board games and block-based coding to teach elementary school students computer science, and designing a middle school science curriculum that integrates computational thinking and sensor technology.