Changing the Game: Teaching Elementary Mathematics Through Coding

Paula Jakopovic
Michelle Friend
Betty Love
Victor Winter
University of Nebraska at Omaha
United States
paulajakopovic@unomaha.edu
mefriend@unomaha.edu
blove@unoamaha.edu
vwinter@unomaha.edu

Abstract: In the current climate of a technology-centered world and standards-based educational system, the vision of including computer science and computational thinking at the elementary level has gained momentum in recent years. This paper examines the similarities between elementary mathematics and computer science content standards and practices, and describes a hands-on, visual coding curriculum that allows teachers to integrate the two into mathematics instruction that meets the requirements of the standards using research based instructional strategies.

Introduction

Since *A Nation at Risk* (National Commission on Excellence in Education, 1983) outlined the need for major reforms in K-12 education in the United States, including the recommendation to adopt rigorous and measurable content standards, the standards movement has progressed steadily over the past 35 years in the U.S. As a result, organizations such as the National Council of Teachers of Mathematics (NCTM) have developed and revised K-12 standards (1999; 2000). Reforms have continued into the current decade with the addition of the *Common Core State Standards* for English/Language Arts and Mathematics (National Governors Association Center for Best Practice, 2010), the Next Generation Science Standards (NGSS Lead States, 2013), and more recently, the addition of K-12 computer science standards authored by the Computer Science Teachers Association (CSTA, 2017). These standards provide structured content and processes to be mastered by students at a given grade level, and afford opportunities to assess student achievement gains over time. In our current technology-infused society, there has been an increase in the desire for computer science education K-12, however there are few existing programs to promote vision shifting to action in U.S. schools (Google & Gallup, 2015). The development of K-12 computer science standards is a step forward in addressing this challenge, however additional work needs to occur for these standards to begin to permeate elementary school settings in coherent, meaningful ways.

Standards and Practices in STEM Education

The most current standards in mathematics, science, and computer science all focus on interweaving content standards, processes, and interdisciplinary practices for students during instruction (Computer Science Teachers Association, 2017; National Governors Association Center for Best Practice, 2010; NGSS Lead States, 2013). For example, the Mathematical Practices of the Common Core Standards (2010) require that students not only understand mathematical processes and procedures, but also know how to reason about and make sense of mathematical problems and structures and communicate their ideas effectively with others. Recent literature on mathematics teaching and learning suggests a list of reform-oriented teaching practices that are necessary to develop students' mathematical knowledge: establishing clear mathematical goals, implementing rich tasks that promote problem solving, fostering connections between mathematical representations, promoting productive struggle by posing questions and fostering student discourse, and helping students develop procedures based on conceptual understandings (NCTM, 2014). Similarly, the CSTA framework includes practices such as collaborating and communicating around computing, to create and refine computational artifacts, and to gradually develop abstractions and an understanding of computational thinking as a problem-solving process (2017). The addition of STEM practices in the standards documents raises the stakes significantly for elementary teachers, who often teach most, if not all, of the core subjects in self-contained classrooms. Despite the updates to standards, teachers still must have the knowledge and training to implement them with fidelity in their classrooms, and comprehensive training programs are not a norm in most schools.

Creating Accessible Opportunities for All Learners

A major focus of the current standards-based reforms is not only to promote teaching of rigorous content, but also to provide equitable access to that content for all learners. Within the disciplines of computer science education and mathematics education alike, the use of hands on, visual approaches to teaching are promoted as a way to make the abstract more concrete for learners (CSTA, 2017; NCTM, 2014). Research has shown that using a gradual shift from concrete-to-representational-to-abstract representations (CRA) in teaching can help students shift from surface learning of mathematical content to deep learning that is focused on recognizing relationships among ideas and embodies the Common Core Mathematical Practices (National Governors Association Center for Best Practice, 2010; Van de Walle, Karp, Bay-Williams, 2012). CRA is a high-impact teaching strategy for all students, including those who struggle to learn mathematics (Hattie, 2017).

At the elementary level, connections between computational thinking and mathematics are present, in everything from basic counting (sorting) and addition (algorithms) to word problems (decomposition and abstraction). Both rely on logic and algorithmic thinking, and can be introduced using teaching approaches that help to increase the visual-spatial reasoning of students (Newcombe, 2013; Wai, Lubinksi, & Benbow, 2009). It is unlikely that students will transfer the computational thinking skills learned in elementary mathematics into computer science unless those connections are made explicit, however. As general educators, elementary teachers typically do not have sufficient training in computer science to effectively illustrate these connections for students. The development of the Bricklayer ecosystem is one attempt to address this area of deficit, by providing resources and training modules for teachers, and creating lesson plans and activities that make the connections between mathematics and computational thinking concepts and practices visible.

Bricklayer

Bricklayer (Winter, 2015; Winter, Love, & Corritore, 2016) is a collection of interactive web applications, downloadable software, YouTube videos, and reading material that facilitates a thoughtful and systematic exploration of construction techniques underlying 2- and 3-dimensional block-based visual art. Such exploration provides opportunities for students to exercise and develop spatial reasoning abilities as well as mathematical and computational thinking skills (Zinshteyn, 2017). At the heart of the Bricklayer ecosystem are two software libraries (the Bricklayer library and the Bricklayer-lite library) that provide graphical primitives for populating discrete virtual canvases (or 3D spaces) with blocks of various colors. The term 'artifact' refers to the visual art that can be created using these libraries.

Bricklayer program development is done using a downloadable integrated development environment called the BricklayerIDE. Within the BricklayerIDE, programs are written in the general-purpose functional programming language SML, with graphical and tool integration capabilities provided by the Bricklayer library. Thus, Bricklayer provides users with an authentic programming environment in which mathematical and computational skills can be developed. The output of Bricklayer programs are files which, through system- level scripts, are seamlessly integrated with third-party tools such as: LEGO[R] Digital Designer (LDD), LDraw, Minecraft, and STL viewers such as 3D Builder. The collage shown in Figure 1 highlights some of Bricklayer's artifact construction potential.

In addition to the Bricklayer graphics library (implemented in SML), the Bricklayer ecosystem also provides a second graphics library (implemented in javascript), with reduced functionality, called Bricklayer-lite. Bricklayer-lite is a visual programming language whose programs have a syntax consisting of assembled puzzle pieces which are similar in appearance to Scratch programs. Bricklayer-lite programs can be developed and executed through a web browser using a Google Blockly-based IDE, which we (also) refer to as Bricklayer-lite. A noteworthy capability of Bricklayer-lite is that, in addition to producing an artifact, the execution of a Bricklayer-lite program will produce a well-formed and well-formatted Bricklayer (ASCII) program text which can be executed (outside of the bricklayer-lite framework) using the BricklayerIDE. A significant portion of the Bricklayer ecosystem has been developed to help novices, especially primary school children. Bricklayer is open-source, freely-available, and can be found at: http://bricklayer.org.

Teaching Computational Thinking Through Mathematics

To date, Bricklayer has been introduced as a mathematics enrichment session in over 70 elementary schools, primarily working with gifted and talented students in upper elementary grades. Although Bricklayer has near alignment with a range of upper-elementary mathematics topics, including coordinate graphing and algebraic functions, the team has found it challenging to provide opportunities to deliver Bricklayer curriculum at a larger scale within local school districts. In an effort to address concerns of curricular alignment, the Bricklayer team have begun to develop lessons that are directly aligned to the Common Core State Standards for Mathematics (National Governors Association Center for Best Practice, 2010), to help teachers more clearly see the connections between computational thinking, coding, and the elementary mathematics they are tasked with teaching.

Bricklayer facilitates the use of the CRA instructional approach to help teachers engage students in the exploration of new concepts using concrete materials such as LEGO blocks, then help students make connections between the concrete models and semi-concrete representations such as pictures and diagrams (both paper and electronic), and finally, to explicitly connect these visual representations to mathematical abstractions, in which only numbers and mathematical symbols are used. This emphasis on the visual is also found in research on spatial ability. It is widely recognized that spatial ability plays a critical role in student achievement across the STEM disciplines (National Research Council, 2006; Wai, Lubinksi, & Benbow, 2009), and another study found that training in the visual arts is a significant predictor of abilities relating to the manipulation of artifacts in three-dimensional space (Walker, et al., 2011). Furthermore, spatial abilities are not static, that is, they can be developed through practice, especially through the visual arts (Newcombe, 2013; Sorby, 2009). Bricklayer explicitly incorporates the visual arts by engaging students in problem-solving activities that require them to visualize, and write code to generate, virtual two- and three-dimensional objects that can be recreated from physical LEGO bricks.

An Example of Bricklayer in the Elementary Mathematics Classroom

As the Bricklayer project has progressed, a perceived need from participants arose for the development of exemplar lessons that model the clear alignment between the mathematics and computer science standards and practices. The research team found that teachers were often hesitant to develop their own lessons, and at times to find direct connections to the standards. One of the current goals of the Bricklayer project is to begin developing lessons that tie directly to the mathematics and computer science standards, that teachers can use as anchor examples from which to build additional resources. One example of this alignment and planning that the Bricklayer team is currently examining involves a Grade 3 lesson that was developed to model the mathematical shift in thinking that occurs as students move from additive to multiplicative reasoning about the area of rectangles. The lesson was engineered to specifically address the standards and practices shown in Table 1.

Table 1

Correlation Between Gr 3 Bricklayer Lesson and Mathematics/Computer Science Standards & Practices

Common Core Mathematics Standards	Common Core Mathematical Practices	Computer Science Practices
CCSSM.3.OA.9 Identify arithmetic patterns, explain them using properties of operations	MP.2. Reason abstractly & quantitatively	CS.4 Developing & using abstractions
CCSSM.3.MD.6 Measure areas by counting unit squares;	MP.4 Model with mathematics	CS.5 Creating computational artifacts
CCSSM.3.MD.7 Relate area to the operations of multiplication & addition	MP.5 Use appropriate tools strategically	CS.6 Testing & refining computational artifacts

MP.7. Look for and make use of structure

Initially, students are given a task to complete with physical LEGO blocks, “Build a red shape that is 2 units wide by 3 units tall, then a black shape next to it of the same size. Repeat this pattern two times.” Students are encouraged to build using any LEGO sizes available, including 1x1 bricks, 1x2 bricks, etc. (Figure 1).

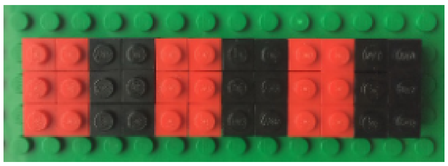

Figure 1. Concrete representation of the red and black repeating artifact.

Artifacts are then transferred to graph paper for record keeping, which the teacher can then use to model the grid system students will later use to code the artifacts in BricklayerLite. As such, there are some specifications as to how to represent the artifact on the graph paper that need to be introduced, beginning with Row 0 and Column 0 in the lower left corner of the graph paper (Figure 2).

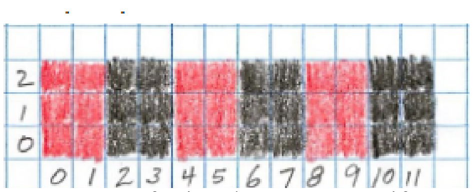

Figure 2. Transferring the concrete artifact to a representational model.

Once students have developed representational models, the teacher can facilitate a discussion about the mathematics involved when using all 1 by 1 bricks, for example, versus using all 2 by 3 bricks. Through discussion, students begin to recognize that there are multiple ways to represent the artifact that are mathematically equivalent, and build a bridge between additive and multiplicative reasoning using area models (Hulbert, et al., 2017).
A shift toward modeling using abstract numbers and symbols can then occur, as seen in Table 2.

Table 2

Connecting Additive and Multiplicative Solutions Using Abstract Notation

Type of Lego Used	Additive Notation	Multiplicative Notation
1 x 1 Bricks	1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 18 + 18 *1x1 bricks* = 36 total dots	6(1 x 1) + 6(1 x 1) + 6(1 x 1) + 6(1 x 1) + 6(1 x 1) + 6(1 x 1) = 18 + 18 *1x1 bricks* (dots) or 36 total dots
1 x 2 Bricks	2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 = 9 + 9 *1x2 bricks* = 18 dots + 18 dots, 36 total dots	3(1 x 2) + 3(1 x 2) + 3(1 x 2) + 3(1 x 2) + 3(1 x 2) + 3(1 x 2) = 9 + 9 *1x2 bricks* (2 dots each) = 9(2 dots) + 9(2 dots) = 18 dots + 18 dots, 36 total dots
2 x 3 Bricks	6 + 6 + 6 + 6 + 6 + 6 = 3 + 3 *2x3 bricks* = 18 dots + 18 dots, 36 total dots	1(2 x 3) + 1(2 x 3) + 1(2 x 3) + 1(2 x 3) + 1(2 x 3) + 1(2 x 3) = 3 + 3 *2x3 bricks* (6 dots each) = 18 dots + 18 dots, 36 dots

Along with bringing up the connection between additive and multiplicative reasoning, there is an important connection to be made to mathematical and computational structure, and how the use of this structure can simplify artifact creation when transferring the graph paper representations to BricklayerLite. To help students begin to reason about predicting where they will place the lower left corner of each red brick in the artifact, additional modeling of the mathematics may occur (Table 3).

Table 3

Abstraction for Placement of a Red Block: (Block # -1) x 4

Red Rectangle Number	Leftmost horizontal coordinate		
1	0	0 x 4	(1 - 1) x 4
2	4	1 x 4	(2 - 1) x 4
3	8	2 x 4	(3 - 1) x 4
Computational Thinking Phase →	*Decomposition→*	*Pattern Recognition (Basic and Advanced)*	

Students can then begin to transfer this thinking to the creation of virtual artifacts immediately (Figure 3).

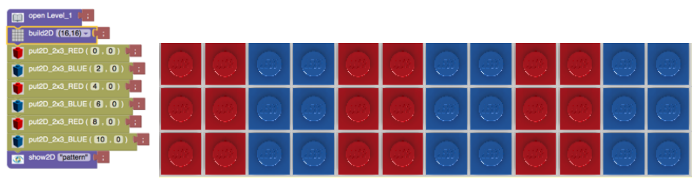

Figure 3. BricklayerLite production of the artifact.

Although one mathematical goal for this anchor lesson is to help students move from additive to multiplicative reasoning by creating area models, another goal is to help them see mathematical patterns and structure, and to use that structure to solve problems. This connects directly to the computational thinking practices of creating and refining artifacts, and developing and using abstractions, as it relates to thinking about these mathematical patterns (CSTA, 2017). By providing lessons that help teachers to see the explicit connections that can be made between mathematics and computational thinking, research team aims to help teachers become more adept at seeing and making these connections in future lessons they develop themselves.

Discussion

In recent years, the call to infuse computer science throughout K-12 education in response to an increasingly technology-centric world has gained momentum. Standards for K-12 computer science have been developed, and several efforts in the computer science community have striven to find ways to implement computer science programs at the elementary level (Schanzer, et al, 2015). Due to the complex nature of computer science and computational thinking, it can be challenging for general classroom teachers to integrate these concepts into their daily instruction. Bricklayer provides one potential avenue to help teachers do so, by utilizing visual-spatial approaches to connect directly to mathematical content and practices. There is still much work to be done in order to develop a comprehensive approach to integrating computer science seamlessly into elementary mathematics and science instruction. This paper offers an argument as to why this approach has the potential to be successful, as well as one example of how to fuse mathematics and computational thinking in ways that are developmentally appropriate, and directly tie to grade level standards. Additional work and research still needs to happen in order to fully understand the potential impacts that such programs can have, both on the development of students' computational thinking as well as their mathematical reasoning skills.

References

Computer Science Teachers Association (2017). *CSTA K-12 Computer Science Standards, Revised 2017.* Retrieved from https://www.csteachers.org/page/standards

Google & Gallup (2015). *Searching for computer science: Across barriers in U.S. K-12 education.* Retrieved from http://g.co/cseduresearch

Hattie, J. (2017). *Visible learning for mathematics: What works best to optimize student learning.* Thousand Oaks, CA: Corwin.

Hulbert, E., Petit, M., Ebby, C., Cunningham, E.P., & Laird, R. (2017). *A focus on multiplication and division: Bringing research to the classroom*. New York, NY: Routledge.

National Commission on Excellence in Education (1983). *A nation at risk: The imperative for educational reform. A report to the nation and the secretary of education, United States Department of Education.* Washington D.C.: The Commission: [Supt. of Docs., U.S. G.P.O distributor].

National Council of Teachers of Mathematics (1989). *Curriculum and evaluation standards for school mathematics*. Reston, VA: National Council of Teachers of Mathematics.

National Council of Teachers of Mathematics (2000). *Principles and standards for school mathematics*. Reston, VA: National Council of Teachers of Mathematics.

National Council of Teachers of Mathematics (2014). *Principles to actions: Ensuring mathematical success for all.* Reston, VA: National Council of Teachers of Mathematics.

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

National Research Council (2006). *Learning to think spatially: GIS as a support system in the K-12 curriculum.* Washington D.C.: The National Academies Press.

Newcombe, N.S. (2013). Seeing relationships: Using spatial thinking to teach science, mathematics, and social studies. *American Educator, 37*(1), 26-31.

NGSS Lead States (2013). *Next Generation Science Standards: For States, By States.* Washington D.C.

Schanzer, E., Fisler, K., Krishnamurthi, S., & Felleisen, M. (2015). Transferring Skills at Solving Word Problems from Computing to Algebra Through Bootstrap. Proceedings of the 46th ACM Technical Symposium on Computer Science Education -SIGCSE '15, 616–621.

Sorby, S.A. (2009). Educational research in developing 3-D spatial skills for engineering students. *International Journal of Science Education, 31*(3), 459-480.

Van de Walle, J., Karp, K., & Bay-Williams, J. (2012). *Elementary and middle school mathematics: Teaching developmentally* (8th ed.). New York, NY: Pearson.

Wai, J., Lubinski, D., & Benbow, C.P. (2009). Spatial ability for STEM domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of Educational Psychology, 101*(4), 817-835.

Walker, C.M., Winner, E., Hetland, L., Simmons, S., & Goldsmith, L. (2011). Visual thinking: Art students have an advantage in geometric reasoning. *Creative Education, 2*(1), 22-26.

Winter, V. (2015). The world needs more computer science! What to do? In D. Conway, S.A. Hillen, M. Landis, M.T. Schlegelmilch, & P. Wolcott (Eds.), *Digital media, tools, and approaches in teaching and their added value* (pp. 119-141). Germany: Waxmann Verlag GmbH.

Winter, V., Love, B., & Corritore, C. (2018). The art of the wunderlich cube and the development of spatial abilities. *International Journal of Child-Computer Interaction, 18,* 1-7.

Zinshteyn, M. (2017, August 1). Cal state drops intermediate algebra as a requirement to take some college-level math courses. *EduSource.* Retrieved from https://edsource.org/2017/cal-state-drops-intermediate-algebra-requirement-allows-other-math-courses/585595