Journal of Mathematics Education
December 2016, Vol. 9, No. 2, pp. 119-131

Programming Multimedia Stories in Scratch to Integrate Computational Thinking and Writing with Elementary Students

Shaunna Smith
Texas State University, U.S.A.
Lauren E. Burrow
Stephen F. Austin University, U.S.A.

This article provides a commentary on how elementary-aged children's uses of Scratch and Scratch Jr. as a multimedia storytelling tool provided a unique opportunity for connecting computational thinking, mathematics, and writing. These areas are not often thought of as harmonious, yet the authors provide examples of how this type of multimedia creation uses a variety of mathematical and verbal literacy processes that simultaneously link to Papert's concept of the personally meaningful creation of digital artifacts and Brennan and Resnick's (2012) computational thinking framework. Using standards for computer science, mathematics, and writing, the authors suggest ways that these activities can be integrated into the classroom to support creative digital artifacts and measureable learning objectives.

Key Words: Multimedia stories, computational thinking, computational writing.

As teacher educators we are constantly seeking out examples that will demonstrate to our pre-/in-service teachers the ideals of interdisciplinary pedagogy. Often times we find those examples in our own homes; for as professors who are also parents of young children, we are constantly met with instances that demonstrate our children's natural tendencies to practically and effortlessly implement interdisciplinary tasks to achieve personally meaningful goals. We have learned to embrace those moments where our professional and personal lives intersect and to allow our own children to teach us how to be better teachers for our pre-/in-service teachers so that they can also be better teachers for the young students in their classrooms. For example, we have learned to move past simple parental pride/wonder/fascination/awe and actively acknowledge and document when our children utilize interdisciplinary connections between computation thinking, mathematics, and writing while playing with visually-based programming tools to create their own multimedia stories at home.

In Dr. Smith's case, tracking her daughter's use of *Scratch, (a free visual-based programming language inspired by Logo - https://scratch.mit.edu)* to create multimedia performances provided her in-service teachers with authentic examples of the harmonious intersection of math and literacy. And in Dr. Burrow's case, reflecting on her son's awe-inspiring ability to approach new stories created in a virtual world, provided her pre-service teachers with authentic examples of young students' tenacious problem-solving mentality that interweaves computational thinking with practical logic models suited for mathematics. It has been fascinating to listen to both of our children as they describe the math and literacy that they negotiate while producing meaningful multimedia and it has helped us better explain that process to our pre-/in-service teachers. This article provides a commentary that will explore the intersectionality of computer science, mathematics, and writing by highlighting applicable learning theories, connecting relevant standards, and providing professor-parent examples of how these practices can be integrated into the elementary classroom.

Theoretical Framework and Related Literature

Constructionism asserts the belief that hands-on design activities can provide personally meaningful contexts for "learning by making" because the learner builds their own knowledge during the process of creation and has ownership over the creative process and the products they create (Papert & Harel, 1991, p. 10). In these types of design-based learning environments, individuals learn while engaging in the iterative design of creative artifacts, which involves development, building, evaluation, and recurring reflection (Bekker, Bakker, Douma, van der Poel, & Scheltenaar, 2015). These learning processes are most prevalent in creative design contexts; however, they are similar to the iterative writing process used in language arts and the mathematical processes that are used to solve problems, which will be discussed in the following section.

Computational Thinking

The process of computational thinking parallels that of writing, and math. Wing (2006) describes computational thinking in a process-based manner as a series of formalized analytical thinking processes, which can be applied to digital and non-digital contexts. Wing describes computational thinking as having the following characteristics:

(1) being able to conceptualize in abstract ways - not simply code or program a computer
(2) forming a fundamental basis of understanding - not a rote memorized answer
(3) being in tune with human thinking - not thinking like a computer
(4) combining mathematical and engineering thinking - not something separate or uncomplimentary
(5) focusing on ideas - not simply artifacts

(6) applicable to everyone - not just computer scientists.

The Computer Science Teacher Association's (2011) K-12 standards state that computational thinking "can be used across all disciplines to solve problems, design systems, create new knowledge, and improve understanding of the power and limitations of computing in the modern age (p. 9)." At the elementary level (CSTA's Level 1), this includes a focus on computational thinking strategies and introduction of basic technical skills. Computational thinking is thought to include the creative design of a digital artifact that the individual has ownership of. Barba (2016) describes it as, "what we can do while interacting with computers, as extensions of our mind, to create and discover." The term *computational thinking* became mainstream in education with the ideas of Papert as he intertwined the learning power of engaging in personally meaningful creation with the use of computer-aided design tools. Beginning with his co-invention of the Logo programming language designed for children in the 1960s, Papert's (1980) theoretical view has inspired a unique world of children's computational creativity and new approaches to teaching and learning (Kafai, Peppler, & Chapman, 2009; Resnick, Maloney, Monroy-Hernández, Rusk, Eastmond, Brennan, & Kafai, 2009). Brennan and Resnick's (2012) computational thinking framework bridges both Papert's (1980/1991) design-based learning view of creating personally meaningful digital artifacts and Wing's (2006) view of examining analytical thinking processes that are used to solve problems. Brennan and Resnick's computational thinking framework involves three key dimensions: (1) knowing certain *computational concepts* (i.e. sequence, loops, parallelism, events, conditionals, operators, data), (2) being able to employ those concepts using *computational practices* (i.e. experimenting and iterating, testing and debugging, reusing and mixing, abstracting and modularizing), and (3) developing new *computational perspectives*, an awareness of self, others, and world (i.e. expressing, connecting, and questioning). Through using this framework, teachers can engage students in computational thinking strategies that are useful in supporting learning in a variety of contexts.

Connecting to Standards and Measurable Objectives

The procedures involved in computational thinking have similar processes as those used in mathematics and writing. Children naturally tap into the skills and subjects interchangeably in order to maneuver, respond, create, design, and re-create in multiple languages (i.e. logic, numerical expression, and traditional verbal words). While children are creating digital multimedia artifacts, they are using a variety of creative approaches and cognitive processes that link to content standards and measurable objectives. In the context of using visual-based programming languages, particularly Scratch, numerous researchers have explored these concepts with children, including elementary children (ranging in ages 4 to 11) and secondary children (ranging in ages 12-18). These research efforts indicate clear

connections to computational thinking in the context of media literacy and writing standards within language arts Common Core State Standards (CCSS), as well as mathematical processes and practice standards advocated by the National Council of Teachers of Mathematics (NCTM).

Connecting to CCSS for writing.

The CCSS for English Language Arts Writing standards asks students in K-12 to master a range of skills and applications related to the writing experience. With each advancing grade level, students are expected to "demonstrate increasing sophistication in all aspects of language use" while maintaining skills learned in previous grade levels (Common Core State Standards, 2016). Students' advancement of writing involves consideration of multiple text types and purposes; production and distribution of writing; research to build and present knowledge; and an engagement in a range of writing styles. Interactions with visual-based programming allows 21st century opportunities for students to develop skills in all of these writing areas (ref. Table 1 and 2).

In a unique theoretical article connecting the similarities between mathematical and verbal processes in the context of traditional coding techniques, Hoat (1987) notes, "both programming and expository writing are based on the ability to recognize that a complex whole is composed of manageable parts" (p.93), because complementary thought processes are used to write a computer program or write a set of verbal instructions to explain something. Similarly, Resnick, et al. (2009) indicate that even when using visual-based programming languages, such as Scratch, programming becomes an extension of writing because successful programs connect ideas and procedures in a logical manner. Atkinson (2014) takes it a step further by stating that programming and coding are in fact new literacies that we must teach our students in order to be fluent in the digital world. This can be seen in recent books about learning how to write programs - or "code"- that are written in a whimsical narrative style using the metaphor of adventure and quests to explain computational thinking and computer science concepts (e.g. Bueno, 2015; Kubica, 2012; Liukas, 2015; Yang & Holmes, 2015).

Through a series of ethnographic case studies of children between ages 8-18, Peppler and Kafai (2007) found that informal integration of Scratch within afterschool *computer clubhouse* contexts encourages reflection within the writing production process while participants create unique and personally meaningful projects that align with media literacy. Burke and Kafai (2012) detailed how a writer's workshop model can seamlessly integrate narrative digital storytelling using Scratch. However, they note the mathematics and computer science connections must be explicitly addressed in order to make an impact (i.e. participants should be required to incorporate specific variables or required to create a non-linear narrative otherwise they might only use a small number of variables and/or mathematical processes to create their digital

story). These computational thinking strategies can easily be incorporated into the writing process by challenging students to engage in multimedia negotiations as they brainstorm, draft, revise, and test their immersive story with their classroom peers.

Connecting to NCTM standards for mathematical content and process. The NCTM standards are subdivided into content standards (i.e. number and operations, algebra, geometry, measurement, data analysis and probability), and process standards (i.e. problem solving, reasoning and proof, communication, connections, and representations). All of which can be addressed through engagement with visual-based programming. Fessakis, Gouli, & Mavroudi (2013) conducted a formal classroom intervention through an exploratory case study of ten children aged 5-6 who used Utah State University's National Library of Virtual Manipulatives (2012) visual programming environments, called "Ladybug Leaf" and "Ladybug Maze." They found positive impacts on the participants' development of mathematical concepts, problem-solving, and social skills. Though the participants were not creating their own artifact, the explicit mathematics concepts took center stage as they applied numeracy and angles to successfully solve the puzzles.

Ke (2014) used mixed methods (i.e. observation, interview, and pre/post Attitudes towards Math Inventory instrument) to conduct a six-week design-based learning experiment with 64 students aged 12-14. Participants were randomly assigned to create ten small groups with diverse gender and mathematical abilities based on school records. The study found that formal integration of Scratch, to support the experience-driven game design processes and development, positively impacted mathematical thinking, abstract reasoning, algebraic procedures, programming skills, dispositions toward mathematics, and ability to craft story. In a two-year case study that used design-based research strategies, including mixed methods with a quasi-experimental design (i.e. pre/post-test Visual Blocks Creative Computing Test) and qualitative techniques (i.e. questionnaire, structured observation), Sáez-López, Román-González, and Vázquez-Cano (2016) used Brennan and Resnick's (2012) computational thinking framework to create an intervention for 107 elementary-aged children. They found that formal classroom integration of hands-on creation with Scratch visual-based programming language into visual arts coursework not only positively impacted students' ability to learn programming concepts, logic, and computational practices, but also positively impacted their self-efficacy toward computer science content and careers.

Through a look at relevant literature, we have provided examples of research that show engagement in multimedia creation with visual-based programming languages can have a positive impact on computational thinking, language arts, and mathematics skills. Table 1 shows the

connections between Brennan & Resnick's (2012) computational concepts, the CCSS standards, and the NCTM mathematics content standards.

Table 1

Connections between Brennan & Resnick's (2012) computational thinking (CT) framework, language arts standards (CCSS) and mathematics standards (NCTM)—Part 1

Brennan & Resnick's (2012) Computational Thinking (CT) Framework		**CCSS Writing Standards CCSS.ELA-LITERACY (3rd Grade)**	**NCTM Mathematics Content & Process Standards (3rd-5th Grade)**
Computational Concepts	**Sequence:** identifying a series of steps for a task	**W.3.3.C**: Use temporal words to signal event order	**Content:** Number and operations
	Loops: running the same sequence multiple times	**W.3.5**: Planning, revising, editing	**Content:** Formalizing patterns, functions, and generalizations
	Parallelism: making things happen at the same time	**W.3.3.B**: Develop experiences, events, and characters	**Content:** Developing meanings of operations and measurement
	Events: one thing causing another thing to happen	**W.3.3.A**: Organize event sequence that unfolds naturally	**Content:** Algebraic reasoning and functions
	Conditionals: making decisions based on conditions	**W.3.5**: Planning, revising, editing	**Content:** Data analysis and probability
	Operators: support for mathematical and logical expressions	**W.3.4**: Produce writing developed and organized to appropriate task and purpose	**Content:** Number and operations, computing fluently
	Data: storing, retrieving, and updating values	**W.3.8**: Recall and gather information	**Content:** Data analysis and probability

To further demonstrate the connections between processes, Table 2 shows the connections between Brennan & Resnick's (2012) computational practices and computational perspectives, the CCSS standards, and the NCTM mathematics process standards.

Table 2

Connections between Brennan & Resnick's (2012) computational thinking (CT) framework, language arts standards (CCSS) and mathematics standards (NCTM)—Part 2

Brennan & Resnick's (2012) Computational Thinking (CT) Framework		**CCSS Writing Standards CCSS.ELA-LITERACY (3rd Grade)**	**NCTM Mathematics Content & Process Standards (3rd-5th Grade)**
Computational Practices	**Experimenting & Iterating:** developing a little bit, then trying it out, then developing more	**W.3.10**: Write routinely over extended time frames (research, reflection, revision)	**Process:** Reasoning and proof, Problem solving
	Testing & Debugging: making sure things work, finding and solving problems	**W.3.5**: Planning, revising, editing	**Process:** Problem solving
	Reusing & Remixing: making something by building on existing projects or ideas	**W.3.8**: Recall and gather information	**Process:** Communication, Connections
	Abstracting & Modularizing: exploring connections between the whole and the parts	**W.3.2.C:** Use linking words to connect ideas within categories	**Process:** Representations, Reasoning and proof, Connections
Computational Perspectives	**Expressing:** realizing that computation is a medium of creation	**W.3.3.B**: Develop experiences, events, and characters	**Process:** Representations
	Connecting: recognizing the power of creating with and for others	**W.3.6**: Use technology to publish writing and to interact and collaborate with others	**Process:** Connections
	Questioning: feeling empowered to ask questions about the world	**W.3.7**: Conduct short research projects that build knowledge	**Process:** Communication

In Action: Creating Multimedia Using Visual-Based Programming to Integrate Computational Thinking, Mathematics, and Writing

We will now provide two case study examples of how our own young children used visual-based programming to create multimedia stories, including connections to standards. Both examples will highlight the use of *Scratch Jr.* (https://www.scratchjr.org), which is a developmentally appropriate version of *Scratch* for use with younger children (available as a free download for iOS and Android tablet devices).

Case Study 1: 7 year-old uses Scratch Jr. to Create "Fairy Tale Feature"

On a rainy weekend Dr. Smith came to the kitchen table and found her daughter using the family's iPad. She asked her daughter what she was doing to which her daughter replied, "I'm playing a cool math game. It's letting me make a movie" Intrigued, Dr. Smith looked closer to discover she was using the Scratch Jr. app, which had recently been downloaded for free. Interested in the fact that her daughter identified it as a "math game," Dr. Smith asked her what type of math she was doing. She responded that she was making a fairy tale story and she had to measure the distances she wanted the characters to move during each scene (CT: Sequence, Operators; NCTM: Number operations, computing fluently). When asked if she had any trouble getting them to move, her daughter responded that it was easy to make them move where she wanted them to but "then the bottom screen got covered with the blocks (of code) so I figured out how to make it more simple with repeats" (CT: Loops; NCTM: Formalizing patterns, functions, and generalizations). Dr. Smith was immediately impressed with her daughter's recognition of needing to simplify the coding process. Seemingly out of nowhere, her daughter excitedly said, "Oh, I figured out how to make them talk to each other" as she indicated the event functions within the app. Dr. Smith watched as her daughter experimented with recording her own silly voices and figured out how to use those recordings to enable her characters to have conversations (CT: Events, Parallelism; NCTM: Algebraic reasoning and functions, Developing meanings of operations and measurement). After about one hour of experimenting and problem solving, her daughter had a completed Scratch Jr. multimedia story that had three scenes, 5 characters, and user interactivity (see Figure 1).

From a superficial glance, this can be viewed as just another playful exploration of multimedia creation. Some would even say that this strategy could be best used as a choice-based option during activity stations in the back of the room as a means of reinforcing concepts learned during formal instruction. However, from an instructional standpoint this case study can be used to show how hands-on creation that is student-driven can illuminate learning concepts using an authentic inquiry-based approach. For instance, instead of a teacher formally stating that loops should be used to make code

more efficient and to minimize redundancy, Dr. Smith's daughter discovered the concept during her own open-ended experimentation. Her acknowledgment that too many "blocks" on the screen made it difficult to add more code led her to create "repeats" using the loops function. Similarly, instead of a teacher formally stating that the "message tool" should be used to program interactions between the characters, Dr. Smith's daughter explored "parallelism" as she sought out ways to efficiently create synchronized conversations between the characters and transitions between the scenes. This is similar to approaches young writers must discover as they attempt to create dialogue within a story or transitions between events and scenes.

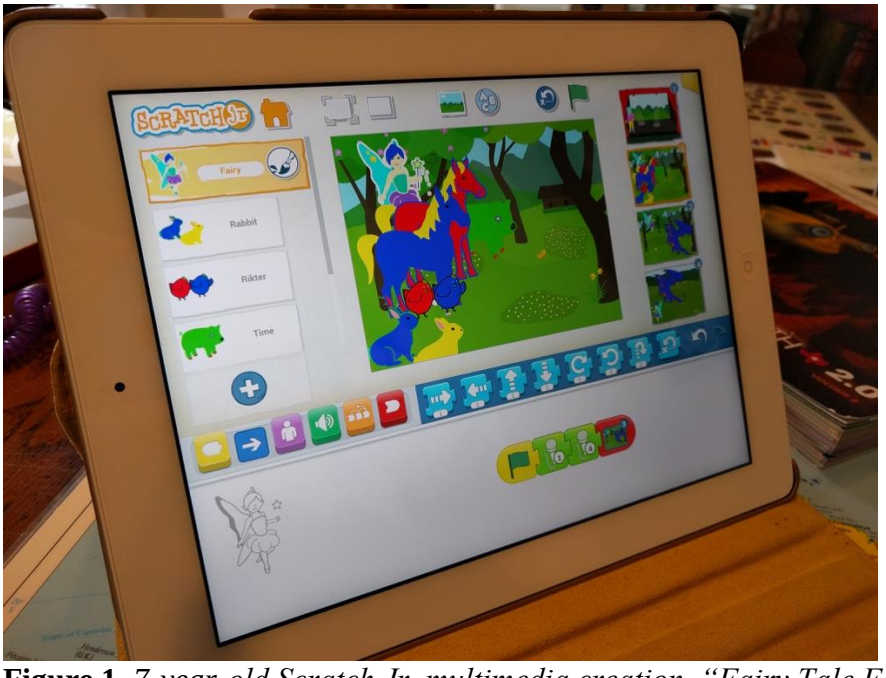

Figure 1. *7-year-old Scratch Jr. multimedia creation, "Fairy Tale Feature".*

Case Study 2: 5 year-old uses Scratch Jr. to create "a Moveable Comic Book World"

Dr. Burrow's tale begins the day after she finished reading aloud one of the books from the popular *Captain Underpants* series by Dav Pilkey to her middle son. The next day the newest title in the series had not been delivered yet, but her son was looking for a new comic book to read. Dr. Burrow suggested that they could write their own comic book story. As the son of an early literacy enthusiast, he was familiar with completing story dictations with her; however, he insisted "it won't be the same. In *Captain Underpants* the characters move [through a flip book style element]." Enter Dr. Burrow's iPad and the *Scratch Jr*. app which she had recently installed at Dr. Smith's suggestion. Dr. Burrow opened the app and explained to her son that with this program he could control the characters and make them really move. He excitedly asked, "How?!" To which she hesitantly responded, "Well, let's find

out.” As she timidly clicked on various buttons to test out features within the app, her lack of technological confidence quickly crept in. While Dr. Burrow was reticent to “experiment” for fear that she might delete some of their work, her young son unabashedly dragged and dropped with lightning speed; answering his own questions of “how do I make it?”; “what happens if I …”; and “why won’t it …” with discoveries of “oh! That’s how!” and “oh! Cool! Look at what I made him do!” Dr. Burrow's role of active co-creator quickly become one of proud observer.

Figure 2. *5-year-old Scratch Jr. multimedia creation, “A Moveable Comic Book World: A Wizard's Tale”.*

Within minutes her son's audible self-dialog changed to comments like, “I’ll just move you there! HA!HA! [insert devious comic book villain laugh] you can’t trick me!” Dr. Burrow soon realized he was no longer seeing the app as a tool with computational steps to be objectively analyzed and systematically ordered, but rather had blended the programming logic into his fantasy story world treating his decisions as a means to expand and advance his characters’ storylines. His “testing and debugging” actions within *Scratch Jr.* effortlessly demonstrated his ability to engage in the iterative writing process as he developed and strengthened his writing as needed by planning, revising, editing, rewriting, or trying a new approach (*CCSS.ELA-LITERACY.W.3-10.5*). Unlike Dr. Burrow's adult-approach that treated the app as a separate entity from the story, her son's child-approach of “leap

before you look" allowed him to problem solve *within* the technology and create in real-time, by testing and troubleshooting as he went. Figure 2 illustrates his focus on increasing the number of characters to add conflicts to his plot and using the coding workspace area as a trial-and-error board full of actions "waiting" to be used when his characters needed to summon "powers."

Comparison of Both Case Studies

From an aesthetics viewpoint, Dr. Burrow's son's approach and coding space can seem inelegant compared to Dr. Smith's daughter's tendency to simplify her workspace and her ability to adjust her planning to utilize discovered patterns as shortcuts. However, when considering this approach from an English Language Arts instructional viewpoint, one can recognize the essential first step of encouraging children to build interest in their topic and motivation towards their task through a method similar to "power writing." Teachers use "power writing" to encourage students to generate ideas, without limitation or pre-judgment, during a word writing "dump" for a sustained amount of time. Students then return later to review, edit, and revise their writing once they have enough content to actually work with. This process can be translated to mathematics instruction when students brainstorm possible solutions to unfamiliar problems and then later return to test their solutions.

Allowing children to spend enough work time in the computational process, like Dr. Burrow's son did, allows them to focus on "reasoning" and "problem solving" as it related to the subject matter/topic of focus. Once they have generated enough ideas and content their next steps should naturally manifest into "looping" actions, "debugging," "remixing," and "expression" as they continue to float between the exploration of connections between the whole and the parts in order to develop representations that are actually meaningful to them. Together, these two examples show how this type of multimedia creation can be used to teach computational thinking concepts in an inquiry-based approach that nurtures creativity and discovery.

Conclusion

This article provides a commentary on how elementary-aged children's uses of *Scratch* and *Scratch Jr.* as a multimedia storytelling tool for at-home playtime rituals of mathematizing and story crafting provided the authors with unique opportunities to consider the connections between computational thinking, mathematics, and writing. This has implications for ways in which teacher educators can encourage pre-/in-service teachers to integrate programming multimedia stories in Scratch and Scratch Jr. to connect computational thinking and writing with elementary students. While these areas are not often thought of as harmonious, the authors' unique positions as parent-professors afforded them the ability to provide authentic case study examples of how this type of multimedia creation uses a variety of

mathematical and verbal literacy processes that simultaneously link to Papert's concept of the personally meaningful creation of digital artifacts and Brennan and Resnick's (2012) computational thinking framework. In the end, if computational thinking is thought of as "a source of power to *do* something and figure things out, in a dance between the computer and our thoughts" (Barba, 2016)," useful examples can be found in non-traditional learning environments, such as the home.

References

Atkinson, T. (2014, January). Is coding the new literacy?. *International Reading Association - IRA website*. Retrieved from: http://www.reading.org/reading-today/digital/post/rty/2014/01/10/is-coding-the-new-literacy

Barba, L. (2016, March 8). Computational thinking: I do not think it means what you think it means. *Berkeley Institute for Data Science*. Retrieved from https://bids.berkeley.edu/news/computational-thinking-i-do-not-think-it-means-what-you-think-it-means

BBC. (2016). Introduction to computational thinking. *Bitesize Educational Resources.* Retreived from http://www.bbc.co.uk/education/topics/z7tp34j

Bueno, C. (2015). *Lauren ipsum: A story about computer science and other improbable things*. San Francisco, CA: No Starch Press.

Burke, Q., & Kafai, Y. (2012). The writers' workshop for youth programmers: Digital storytelling with scratch in middle school classrooms. *SIGCSE'12 - Proceedings Of The 43Rd ACM Technical Symposium On Computer Science Education*, 433-438.

Common Core State Standards Initiative (2016). English Language Arts: Writing Standards, K-12. Retrieved from http://www.corestandards.org/ELA-Literacy/W/introduction.

CSTA. (2011). *CSTA K-12 computer science standards.* Retrieved from https://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education, 63*, 87-97.

Hoat, N. (1987). Conquering the Myth: Expository Writing and Computer Programming. College Composition and Communication, 38(1), 93–95.

Kafai, Y. B., Peppler, K., Chapman, R. (Eds.) (2009). *The Computer Clubhouse: Creativity and Constructionism in Youth Communities.* New York, NY: Teachers College Press.

Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers in Education, 73*, 26-39.

Kubica, J. (2012). *Computational fairy tales.* CreateSpace Publishing.

Liukas, L. (2015). *Hello ruby: Adventures in coding.* NY: Feiwel and Friends.

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

Papert S. & Harel I. (1991) *Constructionism: Research reports and essays*. Norwood, NJ: Ablex.

Peppler, K., & Kafai, Y. (2007). From SuperGoo to Scratch: Exploring creative digital media production in informal learning. *Learning, Media And Technology*, *32*(2), 149-166.

Resnick, M., Rusk, N., & Cooke, S. (1999). The Computer Clubhouse: Technological Fluency in the Inner City. In D. A. Schon, B. Sanyal, W. J. Mitchell (Eds.), *High technology and low-income communities: Prospects for the positive use of advanced information technology* (pp. 263-285). Cambridge and London: MIT Press.

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.

Sáez-López, J., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using 'Scratch' in five schools. *Computers & Education*, *97*, 129-141.

Utah State University National Library of Virtual Manipulatives (2012). *Ladybug Leaf and Ladybug Maze*. Online java apps to teach mathematical concepts located at http://nlvm.usu.edu/en/nav/vlibrary.html

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

Yang, G. & Holmes, M. (2015). *Secret coders: Get with the program.* New York, NY: Roaring Book Press.

Authors:

Shaunna Smith, Ed.D.
Texas State University
Email: shaunna_smith@txstate.edu

Lauren E. Burrow, Ed.D.
Stephen F. Austin University
Email: burrowle@sfasu.edu