Volume 3
Issue 1 *Examining Block Coding: Practical Guidance for Educators and Teacher Educators*

Article 1

2020

Preservice Teacher Perceptions of Coding in Literacy Instruction

Denise K. Frazier
Indiana University East

Anastasia M. Trekles

Recommended Citation

Frazier, D. K., & Trekles, A. M. (2020). Preservice Teacher Perceptions of Coding in Literacy Instruction. *Journal of Computer Science Integration, 3* (1), 1-34. https://doi.org/10.26716/jcsi.2020.03.1.1

Preservice Teacher Perceptions of Coding in Literacy Instruction

Abstract

Coding is a language with many similarities to what is traditionally thought of as literacy. Preservice teachers are familiar with literacy instruction, but were not exposed to computer science during their K-12 education nor in their teacher education course work. Yet, they are responsible for preparing children for future careers, including the growing field of computer science, which should be integrated as early as possible into the general education curriculum to build awareness, interest, and ultimately, skills. In this study, preservice teachers in a K-6 reading interventions class were trained in Scratch and provided a template to use with children struggling in various aspects of literacy. This article examines how preservice teachers perceive the relationship between coding and literacy through the theoretical framework of gaming, and whether they would include coding in literacy instruction. Results indicate preservice teachers do not feel confident enough in their teaching abilities to feel comfortable integrating coding into literacy instruction. Lack of prior knowledge and time constraints contributed to those that chose not to participate. Success occurred as Scratch was found to be motivating and individualized when using self-selected pictures and voice to connect to the written word, supporting children's literacy learning.

Keywords

coding, gaming theory, literacy, preservice teachers, elementary

DOI

10.26716/jcsi.2020.03.1.1

Corresponding Author

Denise K. Frazier, Indiana University East, School of Education
2325 Chester Blvd., Richmond, IN 47374
dkfrazie@iu.edu

Preservice Teacher Perceptions of Coding in Literacy Instruction

There are currently 500,000 open computing positions in the United States, yet America's education system does not provide widespread access to computer science, indicated by the fact that only 35% of high schools teach computer science (Computer Science Teachers Association, 2018). Very few states have adopted strategic plans, standards, and certification programs related to computer science (CS), yet Indiana is one state that has. Although the state has a clear pathway to certify CS teachers, only *one* CS teacher graduated in Indiana in 2016, making it difficult to prepare children to fill the current (2017) 4,701 CS job openings in Indiana (Computer Science Teachers Association, 2018). Although lacking a plethora of certified CS teachers, children can still learn about coding from general education teachers as there are parallels between reading and writing code and reading and writing text (Vee, 2017).

Coding is a language (Baker-Doyle, 2018) that requires sequence to work, just like math requires an order of operations and literature follows plot lines. Programmers write using symbols that are designed to be read, then executed by a computer (Vee, 2017). Relatedly, letters are symbols that must be in a specific order to form words, and words are put in logical order to create comprehensible sentences. Similar to traditional writing, programming is a method to organize information (Vee, 2017). Children need to crack the code to understand computer programming languages, just as they need to decipher the alphabetic code to read (Eulenberg, 1982; Gee, 2013; Hutchison, Nadolny, & Estapa, 2015; Thompson et al., 2018), as comprehension of computer programming languages occur in the same part of the brain as natural language (Portnoff, 2018) . Computers make many think only of STEM skills, but in essence, they can be better described as embodiments of language, communication, and the entirety of the human condition (Eulenberg, 1982; Portnoff, 2018). Processing like a computer requires computational thinking, which embodies a vast skillset that includes problem-solving and idea formation (Wing, 2006), but also encompasses reading, building, and writing (Vee, 2017).

The English language follows generalizations (Gribbin, 1996). For example, the “i before e except after c” notion applies to words such as “believe” and “receive;” but not to “weigh” and “species.” Sight words (recognized at a glance) and high-frequency words (appear most often in text), are categorized as phonetically decodable or include irregular spellings and are generally memorized (Farrell, Hunter, & Osenga, 2019). Kindergarten and first grade students spend much time learning sight words to improve reading fluency, which in turn supports comprehension and understanding of text, and allows students to begin to shift toward the integration of written, oral, and visual communication (Hagge, 2017). In the same way, students of programming must learn basic commands and the structure of the coding language in order to improve fluency and build more advanced programs (Codeacademy, n.d.).

This investigative case study examined the interplay of literacy development and coding via one-on-one interactions between preservice teachers and elementary students. Preservice teachers tutored children in a literacy interventions course who encountered literacy challenges, such as sight word acquisition, and made determinations about the integration of coding activities within their instruction. This study was guided by the following research questions:

1. How do preservice teachers perceive the relationship between coding and literacy, and specifically the teaching of sight word acquisition through the online coding environment Scratch?
2. How do preservice teachers perceive the relationship between gaming and literacy acquisition in elementary students?

DOI: 10.26716/jcsi.2020.03.1.1

3. What obstacles do preservice teachers encounter when making choices to include online game-based coding in literacy instruction?

Discovering a possible connection between coding and literacy could promote computer science or computational thinking in the classroom and motivate children who struggle in literacy. This study builds on previous research in game-based learning (Gee, 2008; 2013; Papert, 1980) and literacy and coding instruction (Kidd, et al., 2014; Thompson, et al., 2018; Vee, 2017). It may also provide teachers with additional tools to consider how and when to integrate coding skills within their elementary level literacy curriculum where computer science instruction may be new to teachers or not perceived as appropriate, practical, or worthwhile to incorporate in their classrooms.

Theoretical Framework

While the connection between learning outcomes and technology is often unclear in the literature, there is evidence that under the right conditions, technology can be useful in advancing learning when there is a clear and meaningful connection between the two (Bouygues, 2019). For example, if technology usage in reading instruction simply incorporates digital storybooks or online drill and practice, it can be detrimental to reading retention for younger children (Yienger, 2016). However, when technology is used purposefully within traditional literacy instruction, it can help students form new and meaningful connections, as demonstrated by the way video-based gaming engages children in reading and writing (Gee, 2008). In particular, game-like environments that integrate problem-solving, goal attainment, and other motivational elements can offer students high levels of learning attainment (National Academies of Sciences, Engineering, and Medicine, 2018).

Gee (2013) offered a framework in understanding how game environments, such as those found in online coding apps Code.org and Scratch (scratch.mit.edu), assist in learning noting that video gaming itself is a form of literacy. He referred to a “good” game environment as one that enables and reinforces learning, active exploration, and player autonomy, which a coding app such as Scratch allows due to the open-ended nature and full user control (Hagge, 2017; Kafai & Burke, 2014; Thompson, et al., 2018). Table 1 provides a brief summary of this framework, featuring those principles most pertinent to the current study.

Table 1

Gee’s (2013) Selected Principles of Effective Game-Based Learning Environments

Principle	**Description**
Co-design	Quality game-based learning involves the learner actively as an author, rather than a mere passive participant.
Customization	A good game allows players to adapt the environment to their needs.
Manipulation	Players can manipulate tools that enhance and extend their effectiveness.
Problem-Solving	Good game-based learning involves the solving of complex problems that allow multiple solutions.
“Sandboxing”	Good games allow players to experiment, fail, and try again with safety.
Pleasant frustration	Game-players learn best when they are at the edge of challenge, and when the environment is neither too easy nor too difficult.
Identity	Deep learning in a game-based environment requires that the player is committed and invested, which correlates to how well the player can see themselves as part of the story.
Situated Meaning	Meaning of words and concepts is situated within the game environment, but they are most powerful for the learner when they can be tied to world experience.

If it can be accepted that literacy refers to any process of decoding meaning through the use of symbols (Eulenberg, 1982; Gee, 2013; Hutchison, Nadolny, & Estapa, 2015; Thompson et al., 2018), then code, and therefore the games that are both created by it and teach its use, are also a form of literacy. Gee (2013) argued that when playing a game, learners must decode the design and produce new solutions, similar to the process of decoding letters and writing words. Further, when designed well, games include assessment and can create motivation for players to continue evolving their knowledge and skills by offering well-ordered problems that offer challenges that are difficult but attainable, allowing players to practice at higher and higher levels of expertise as their skills grow (Gee, 2008). Once one set of skills is mastered, another, slightly more advanced cycle of practice and attainment is presented. In this way, game-based learning and literacy are closely linked, as students learn to decode and construct letters, then words, then sentences, and so on.

Furthermore, Gee (2013), as well as Prensky (2006), discussed the nature of games as a form of storytelling, which in turn builds literacy skills. The player is also an author, creating her story as she manipulates and explores the game world. Learning and navigating any computer-mediated game environment also typically requires the player to take on a persona, and to explore the ideas of character, narrative, and genre, which Gee (2013) called the Identity Principle. In addition, the concept of the Situated Meaning Principle allows players to explore the meanings of symbols, words, artifacts, and texts as they are situated within the virtual environment. Conventional items that the player recognizes from the real world may take on new meanings and contexts in the game world, expanding understanding of what is not only possible, but also what is *impossible.* As Prensky (2006) noted, there is a great deal of power in a game's ability to expand the imagination, and as such, storytelling capability and potential.

It is with this game-based learning framework provided by Gee (2013) and Prensky (2006) that the current study is informed. Coding is offered as both a complement to and a component of literacy development, through the lens of a game-like online environment (the coding app Scratch) that offers continuously increasing challenges to learners.

Review of Literature

The International Literacy Association developed a Literacy Glossary (2020) to define terms as a shared vocabulary for the profession as both language and literacy are evolving. Literacy is defined as:

> The ability to identify, understand, interpret, create, compute, and communicate using visual, audible, and digital materials across disciplines and in any context. Over time, literacy has been applied to a wide range of activities and appears as computer literacy, math literacy, or dietary literacy; in such contexts, it refers to basic knowledge of rather than to anything specific to reading and writing (section L).

This definition does not directly address the National Reading Panel's (2020) five key concepts of reading instruction (phonemic awareness, phonics, fluency, vocabulary, and comprehension) or writing, but does reference basic knowledge of digital materials and computer literacy. The definition of literacy relates to digital technology as it is ever changing due to the pace in which new technologies emerge (Eulenberg, 1982; Hutchison et al., 2016). Related to computer literacy, the programming language LOGO for children was introduced in the late 1960's, yet coding in schools has only recently made a comeback (Bers, 2018) and the practices of reading and writing human languages directly compares to programming (Vee, 2017). Programmers who write code produce step-by-step directions for a computer to complete a task (Hutchison et al., 2016; Vee, 2017). Coding and literacy share similarities as both use a symbol system (programming language and natural written language) and technology (computer and paper/pencil) to communicate through a sharable product (Bers, 2018). However, the computer literacies involve added layers to account for new possibilities (Warschauer, 2004). The National Reading Panel (2020) concluded that the availability of computer technology research related to reading indicates it may help children read. With the growing number of jobs in computer science, a need for educating children in coding has surfaced. There is a call to include coding in elementary classrooms as younger children are more open to embrace new concepts, find it motivating, and develop life-long skills (Randles, 2020).

Coding is something that most preservice teachers were not exposed to during their K-12 education as it was not part of school curriculum and still is lacking in many states. As of 2018, only six states had strategic plans for K-12 computer science, including Arkansas, Hawaii, New Hampshire, North Carolina, Rhode Island, and Wyoming (Computer Science Teachers Association, 2018). More states (twenty-two) have adopted K-12 computer science standards (Computer Science Teachers Association, 2018), and fifteen have implemented requirements that students must graduate high school with at least one computer science credit. While thirty-three states and Washington, D.C. offer computer science teacher certification options, it is clear that more work is needed in order for state legislation, standards, and teacher preparation programs to fully address the issue of educating teachers to understand computer science and implement it in their classrooms (Computer Science Teachers Association, 2018). One start would be to incorporate computational thinking. Wing (2006) argued computational thinking is a fundamental skill that all children should engage in as it promotes analytical thinking and problem solving. This is needed to meet computer science standards which are meant to be inquiry-based and hands-on related to both concepts and practices as specifically named in

Indiana's standards (Indiana Department of Education, 2017). However, just the word "coding" may intimidate many elementary education majors as thoughts turn to challenging programming languages. Yet, coding refers to a type of logical thinking, involving problem solving, sequencing, and planning which can be done prior to actually writing computer code. Understanding the vocabulary of computer programming is a necessary part of computational literacy to think and read like programmers (Grafwallner, 2018), and viewing the standards puts it into perspective. For example, one Indiana K-2 coding standard (Indiana Department of Education, 2017) under the heading "data and information" states:

> **K-2.DI.1** Use technology resources to solve age-appropriate problems and communicate thoughts, ideas, or stories in a step-by-step manner.

This is done often in K-2 classrooms through story retell and writing. Enabling children to see sequence helps to identify relationships between the spoken and written word and to make predictions of letter sound relationship and word order (Kidd, et al., 2014). By explicitly teaching it as coding, this allows children to see the relationship to reading and writing, and builds on their prior knowledge. Similarly, the Indiana CS standards for grades 3-5 (2017) are developmentally appropriate and cross-curricular. For example, an "impact and culture" standard states:

> **3-5.IC.3** Evaluate the accuracy, relevance, appropriateness, comprehensiveness, and biases that occur in electronic information sources.

This standard relates to critical literacy theory (Behrman, 2006) as students navigate text to better understand the world by developing literacy skills to identify biases in language and critically think about societal issues (Baker-Doyle, 2018; DeVries, 2015). This also connects to media literacy defined as understanding, interpreting, and critiquing media, as in the case of fake news (Common Sense Education, 2017); however, it also entails creative and social expression, and other technical skills (Peppler, Santo, Gresalfi, & Tekinbas, 2014). These connections have become more apparent through the Hour of Code.

The Hour of Code was started as a one-hour introduction to computer science with the goal of showing that it is fun and creative (Hour of Code, 2019). The website offers tutorials and guides to make planning easy. For instance, following a unit on story writing and the Hour of Code, one English teacher combined these by having her students compare and contrast novels to computer games and saw many similarities such as how games are often constructed from stories (Bradley, 2017). Coding stories have been found to be an effective way to introduce students to coding (Burke, O'Byrne, & Kafai, 2016). Next, they brainstormed computer game design connected to their written works and developed games through Scratch (Bradley, 2017). This type of design thinking posits the learner as an active constructor of knowledge (Papert, 1980; Peppler, Santo, Gresalfi, & Tekinbas, 2014; Warschauer, 2004). This lesson incorporated higher-order thinking skills as students created written stories, applied them to gaming, then generated a product through Scratch, the highest level of Bloom's Taxonomy (Krathwohl, 2002).

Computer science learning activities are often structured like games allowing students to think logically through steps within highly motivational, problem-based environments that integrate a variety of skills including media and text literacies, in order to come up with viable solutions and meet the goals set forth within the game (Hagge, 2017; Hutchison et al., 2015). Well-designed games include problem-solving skills and player feedback to support learning (Gee, 2013). The app Scratch Jr. and the website Scratch teach coding literacy through games and interactive stories. It is meant to introduce coding to users that lack experience as they snap together visual programming blocks with a mouse, similar to building with LEGOs (Hagge, 2017). Scratch Jr. is a free app for young children (ages 5-7) to program interactive stories and games (MIT Media Lab, n.d.-a).

Traditional print literacy skills support children's ability to create with Scratch as the commands require users to read and write (Peppler & Warschauer, 2011). The string of commands in Scratch must be meaningfully put together in grammatical combinations (Peppler & Warschauer, 2011). Just like children learn to read in grades K-3 and read to learn in grades 4-6 (Chall, 1983), with Scratch Jr., children are not just learning to code but are coding to learn (MIT Media Lab, n.d.-a). Related to the stages of writing (Calkins, 1986), coding is a process to construct a narrative (Burke, et al., 2016). Following this principle, coding can support other academic learning skills, and as children code, they create and express themselves, developing sequencing skills as well as the ability to read and write, including sight words.

However, any game to promote literacy must be paired with carefully planned and well-designed lessons to meet learning objectives (Hutchison et al., 2016). Scratch is recommended for ages 8-16 and is an online community where students can create their own interactive stories, games, and interactions and share them with people around the world (MIT Media Lab, n.d.-b). Scratch 3.0 was released on January 2, 2019, allowing projects and play on tablets as well as computers. A unique aspect of Scratch is that all projects must be "remixable," meaning users can make copies of someone else's project and add their own ideas, resulting in a remix with the thinking that collaboration creates often improved and interesting projects (Baker-Doyle, 2018; MIT Media Lab, n.d.-a; Peppler, Santo, Gresalfi, & Tekinbas, 2014). This allows students to move beyond receivers of knowledge to producers of knowledge, a higher-order thinking skill (Kafai & Burke, 2014; Krathwohl, 2002; Lee, 2011). Additionally, deeper literacy practices are experienced. Hagge (2017) identified five Scratch activities promoting different types of literacy, such as how visual literacy is deepened as children read graphical forms of instructions in projects, and how socio-emotional literacy is strengthened while collaborating with Scratch members to create a digital story.

Methods

The study used an investigative case study approach using qualitative data from open-ended written responses, observations, and focus groups. This method allowed common themes to emerge in support of the central research questions (Yin, 2009).

Setting

Western Elementary School (WES – a pseudonym) is a Title 1, K-6 public school located in the rural Midwest. It has shown steady growth over the last 10 years serving 484 students during the 2018-19 year. The community is primarily White (89%), with an 11% (5% Hispanic, 1% Black, 5% two or more races) minority rate and a 36% free/reduced meal rate. Participating preservice teachers (PSTs) attend a nearby public, four-year, master's-granting regional university located within five miles of WES and attracting students from the local community as well as others within a three-county radius.

Participants and Course

Elementary education and early childhood majors take the course Literacy Interventions the semester prior to student teaching to learn how to assess and address the literacy needs of elementary children. This is an intense, field-based course where PSTs have a heavy, yet purposeful, workload to prepare them for student teaching and working with readers who struggle. PSTs tend to leave the course confident and viewing themselves as a professional. This is a strengths-based class meaning even though each child has a literacy need, individualized instruction is planned to be motivating and build on student strengths. Tutoring occurs two times per week in 30-minute sessions and texts are chosen based on student interest. The first three weeks of the course take place on campus and is intense as PSTs learn how to assess children's reading, writing, and spelling; and to plan instruction using research-based literacy strategies. Some of the course requirements include lesson planning, writing initial and final student reports, and planning hands-on, engaging instruction to meet individual student needs.

During the fall 2018 semester, 31 PSTs were enrolled in the course tutoring 62 elementary children at WES. The students were recommended by their teachers to participate in tutoring and the principal worked with the university professor on scheduling. Teachers look at children's test scores and classroom performance to determine eligibility. Data was derived from the following sources: NWEA Measures of Academic Progress (given 3 times per year, all grade levels); ISTEP (Indiana's annual summative assessment for grades 3-6); IREAD-3 (Indiana Reading Evaluation and Determination summative assessment given annually to 3rd grade children to measure foundational reading skills); and *Star Reading* (to determine reading level, given annually to 3rd-6th grade children). Tutoring needs varied by child and grade level but generally relate to phonemic awareness, phonics, fluency, vocabulary, comprehension and/or writing. Children already identified as Title 1 are not eligible for tutoring. The children were pulled out of their classroom for one-on-one tutoring which took place in the hallway, library, or cafeteria. Each PST tutored either a kindergarten or first grade student; and a third or fourth grade student, each for 30 minutes for a total of one hour per week per student. Western Elementary School provided a classroom, computer, and projector for the course instructor to conduct class in for one hour per week. This is advantageous as tutoring struggles and successes can be addressed and shared immediately following tutoring. The spring 2019 semester was similar with 17 enrolled PSTs tutoring 34 children. This course has been at WES since 2015 and a strong school/university partnership has been established. Teachers value this course as indicated by their willingness to allow children to leave their classrooms for tutoring and they know a literacy professor is supervising instruction. PSTs assess their tutees at the beginning and end of the semester and write detailed reports on their findings that are shared with the teachers. Additionally, PSTs provide teachers with copies of their lesson plans and anecdotal notes each time they tutor.

Coding

The research study was co-planned by the course instructor, who specializes in reading education; and an instructional design professor, an expert in technology. Both were interested in examining potential connections between coding and literacy. The WES principal endorsed the study and IRB approval was obtained. The purpose of the study was to examine PSTs knowledge and perceptions of coding and literacy, as well as whether Scratch supports children struggling in literacy. During the fall 2018 semester, the researchers presented information regarding the coding study during class time at WES, explaining that programming is a language and coding commands need to be put in a certain order to get a desired result; similarly of course, letters need to be put in a precise sequence to spell a word, and syntax is necessary for sentence meaning. The PSTs were familiar with information related to reading. For example, all had to do a running record and analyze the results which included examining whether the child used semantics, syntax or visual cues when deciphering words. PSTs had already started tutoring, and since all were working with a kindergarten or first grade student, sight word knowledge was a frequent part of their lessons. WES lacked technology having only one computer lab for their K-6 school, so the thought was that the children might be motivated to learn with the novelty of technology. Additionally, PSTs had the workload of planning four lessons per week for tutoring, so the researchers thought it would be appealing to include Scratch for the word study portion of the required lesson plan to lessen the time it took to plan instruction. The PSTs were told that the researchers had developed a sight word remix (template) through Scratch that they could incorporate in lesson planning (Appendix A). The remix was a sight word game that included a cat (a "sprite") and preprogrammed sight words. Children would hear a word, then click on the matching written word (from three choices). The provided remix could be edited by the PST and child to make it personal and motivating. For example, the child's voice could be recorded saying the sight words and pictures that the child related to the word could be added from either the Internet or photos the tutor and tutee took together to personalize the learning and motivate the student.

To participate in the study, assessment needed to indicate the child had a literacy need related to sight words. However, any PST could attend the Scratch training on campus, even if they did not participate in the study. The PST time requirement for study participation included Scratch training at the beginning of the semester and answering questions to gather prior knowledge from the fall 2018 semester (Appendix B), pre-study questions related to expectations during the spring 2019 semester (Appendix C), and focus group participation (Appendix D) at the end of the semester; less than a two hour time commitment. Additionally, the researchers would observe each PST using Scratch with their student. There was not a set number of observations as the focus of tutoring needed to be on the child's needs, not the research study. Therefore, data was collected from the PST and two researchers.

Based on PST focus group feedback at the end of the fall 2018 semester, minor adjustments to the study were made for the spring 2019 semester. For example, an assignment was added to the syllabus to include more specific training on coding and Scratch during the on-campus portion of the class prior to going to WES. No additional time outside of class was required for PSTs to participate in the 2019 study and all received Scratch coding training, although WES children still first needed to be identified as requiring sight word tutoring to participate in the study.

Data Collection and Analysis Procedures

As indicated previously, data was collected from PSTs before, during, and after they worked with identified students on sight word acquisition through questionnaires, observation, and focus groups. The two researchers participated in observations and focus groups at two different times, as well as worked separately on coding the responses to the short answer item to add inter-rater reliability to the identification of the categories. Open coding was applied to allow for emergent patterns and themes, which were then categorized (Creswell, 2013). Data analysis resulted in a number of themes, including confidence in prior knowledge and skills, the need for additional training in both coding and literacy teaching strategies, time commitments, concerns over university obligations, and interest in technology.

Results

Due to a number of factors, as described further in the focus group results, only one PST, referred to by the pseudonym Allison, completed full participation in the study. The PST was observed with her student on two separate occasions with each researcher independently, completed questionnaires, and discussed her experiences with the researchers after concluding her time with her assigned student. While the study did not yield more individual participants when it came to employing Scratch, it did allow the researchers to explore the obstacles that all PSTs in the study group faced in terms of incorporating a tool like Scratch into their literacy instruction, which in turn helped to re-shape the research questions and the lens through which the case study was viewed.

Scratch in Sight Word Acquisition

Allison used the Scratch remix provided by the researchers with the kindergarten child she tutored. Allison's student was struggling with learning sight words and he was so quiet that she had trouble communicating with him to determine his specific needs. By incorporating technology, she hoped it would motivate him and strengthen their relationship. She also chose to participate in the study as her own kindergarten son was struggling in school and was motivated by technology. She hoped that what she learned in the study could be used with her child and in her future classroom.

Allison said the addition of Scratch was the thing her student was the most engaged with throughout the semester due to his interest in Minecraft and technology. "It was the first time he smiled," she noted in her post-study interview. What helped the most for him in learning sight words was allowing him to choose pictures to go with the sight words. For example, while the researcher was observing, the child chose a picture to go with the word given. He was quite talkative with Allison during this activity. She reported later that he really enjoyed having choice in the pictures he matched with the words Allison chose for him and was overall satisfied with Scratch and learning. Her son also benefitted from and enjoyed coding so much that he was receiving Botley the Coding Robot for Christmas.

There were some struggles. For example, Allison felt like she did not have enough time to use Scratch adequately with her student, and suggested that Scratch be taught to all PSTs in the class before tutoring. In this way, PSTs would be more prepared to use it even if the child they tutored was not struggling in the area of sight word acquisition. Allison prepared teacher recommendations and an end-of-tutoring letter to his family where she suggested continuation of Scratch coding to learn sight words. She also struggled as the student viewed the time as "play" and did not appear to immediately understand the correlation between coding and learning.

Limited Interest and PST Concerns

Although only Allison used Scratch with her student, two other PSTs, Molly and Richard (pseudonyms), came to the off-campus Scratch training, showed interest in using it during the fall 2018 semester, and participated in the post-semester focus group. Molly said she wanted to participate in the study because she thinks "coding is cool." However, as a senior she felt her schedule was too full to include the time needed to educate herself beyond the Scratch training to be truly comfortable in using the tool. She felt that coding should have been offered in her technology class so she could use it in her teaching methods classes, such as this one. Yet, she took the technology course early in her college career and wished she would have taken it later as she had forgotten the content since she was not able to use it with children. She felt Scratch could be used to promote literacy in other ways, especially writing. For example, if punctuation is omitted in writing, there are problems, just like if you skip something in coding, the program will not work.

Similarly, Richard indicated that he had taken the technology class several semesters prior, and had not spent time practicing many skills learned there since. He also noted that he felt pressure to spend his limited time on other classes and in preparation for student teaching the following semester, as university requirements had changed and he felt he needed more time to adequately prepare to meet the challenge. However, he also indicated that despite these concerns, he would have gladly worked with his student using Scratch had his student seemed interested in technology. In his case, he found that non-digital routes seemed to work better in tutoring sessions, and after reflection on the student's needs and interests, did not feel that introducing coding would benefit the student.

Focus Group Results

The majority of the PSTs who did not participate in the study said they did not have time. They did not feel they could attend the additional training or take the time to better learn coding. Most felt overwhelmed as it was the semester prior to student teaching and there were many changes happening at the university, including the addition of the EdTPA external evaluation, set to occur during the student teaching semester. Many indicated feeling so overwhelmed with learning how to help children struggling in literacy with research-based reading strategies that they simply did not want to take on something else. In addition, several were simply not interested in technology, or indicated being intimidated by the idea of integrating such an advanced technology concept into their teaching. Because most of these students had taken their technology course more than one year ago, their confidence was low and many shared concerns about their ability to learn Scratch, much less use it with a student. Some PSTs noted that they did not want to "waste time" when tutoring, fumbling through an unfamiliar program, and several even indicated that they were afraid to look "dumb" in front of their student.

Discussion

As of this writing, the International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) have collaborated on a draft of standards for computer science educators (ISTE, 2019). In addition, ISTE has published competencies for Computational Thinking (2018) related to problem-solving using the power of computing. In both cases, these sets of standards recognize that learners use logical processes to decode, analyze, and create new information. While the understanding of computer-based technical functions is important, it is just as important for educators to develop effective teaching strategies that implement real-world problem-solving that require creative solutions. Understanding the use of digital devices and software is only one piece of the puzzle, just as understanding how to create words out of letters is only one part of literacy development (Grafwallner, 2018; Hutchison et al., 2015). Once teachers are able to see the relationships between computer science and reading and writing literacy, they are able to more readily develop lessons that take advantage of game-based and other digital environments. However, without training in these areas, teachers may struggle to help students see the connections, as indicated by the PSTs in the current study. The study includes implications that preservice teacher training requires more attention to computational thinking and computer science instruction as part of the essential skills they are taught before they enter the classroom.

Due to the shortage of computer science educators, teacher education programs need to embed computer science curriculum into their programs so that general education teachers are prepared to weave CS content into their future classrooms. Simply adding the vocabulary “computational thinking” in subject areas such as math and literacy support CS instruction as children learn the terminology and see it used in multiple contexts. Additionally, weaving coding in as writing instruction is a way to incorporate it as a new form of composition (Burke, O’Byrne, & Kafai, 2017). Finally, computational participation extends it even further to include the social aspect of learning when interacting virtually with others, and improving and changing others’ content through remixing and co-creating (Burke, O’Byrne, & Kafai, 2016; Gee, 2013; Papert, 1980; Scratch 2011; Warschauer, 2004).

During the study, it was indicated by participant Allison that her kindergarten student did not immediately see the connection between reading and Scratch, though she as the teacher readily did. She noticed in particular how the student was able to create his own story, as well as new ways of understanding the onscreen symbols as words – in other words, she was able to observe directly the Situated Meaning and Identity Principles as described by Gee (2013). Tutoring happened two times per week (one hour weekly) during one semester and much literacy content needed to be covered in the short time period. Perhaps with more time and emphasis placed on coding as a language, both student and teacher would see this connection.

Peppler & Warschauer (2011) found that an 8-year-old built reading and writing skills over a 2 ½ year time period by coding with Scratch. This child first showed fast technology learning with Bryce5, a challenging software program to master, but requiring little to no reading. Her success with this program gleaned confidence in programming and led to discovering Scratch. Exploring the program over time, the child found success in the paint editor and audio recording. Scratch uses block code so she learned by combining blocks of similar shape and color, but then she started to show interest in incorporating letters, specifically by identifying specific letters on the keyboard to elicit movement from her Scratch "sprite." She was the first in her community center to stack code to make objects move which demonstrated a "link between traditional print literacy and Scratch," serving as motivation to learn more about gaming and pushing her to engage in more print reading (Peppler & Warschauer, 2011, p. 29). Each Scratch block has an action written on it, so in order to choose, one must read. The child in Peppler & Warschauer's study (2011) had time to explore Scratch over time which increased her interest in the program and motivated her to read to write the block program to create the game. For the first time, she was reading and writing texts as she saw a relationship between programming literacies and decoding print text. Over time, she became aware of how language operates by making "connections between the Scratch programming language and her spoken (and increasingly written) English language" (Peppler & Warschauer, 2011, p. 32). In this study, Allison, like many members of the post-study focus group, indicated that learning a language takes time and includes similar skills, whether that language is English, Spanish, or computer code. Pseudocode is embedded text that only the programmer, not the computer, can read, helping the programmer organize code into steps or parts (Baker-Doyle, 2018). Pseudocode serves as a frame in coding, similar to planning writing, and helps situate meaning for the programmer (Gee, 2013). Children who learn to pseudocode in programming can apply the same concept to writing English or another language as they participate in metacognition (Baker-Doyle, 2018). Furthermore, just as in writing, style matters in coding and through peer craft, coders get feedback from others as they improve their work. Similarly, peer editors do the same thing when engaging in writing.

It is also worth noting that children today have unlimited access to digital content and must engage in content creation on a continual basis to decipher meaning. In support of literacy and coding, educator Mark Davis (2018), stated, "My students practiced decoding through the process of coding, learned syntax as a new vocabulary, and became fluent in a global language of programming" (para. 6). He continued by noting the increase in motivation as interdisciplinary connections are made. This relates directly to the framework offered by Gee (2013) – when game-players are introduced to new challenges that are just on the edge of difficulty and therefore able to be attained through practice, they are motivated to continue to reach for their goals as they will eventually succeed. When students experience success in coding at one level, they are encouraged to continue growing their skills and gaining new ones in order to accomplish more advanced tasks. This was the case in Peppler & Warschauer's (2011) study. In research in other areas related to learning and motivation, these tenets continue to hold (National Academies of Sciences, Engineering, and Medicine, 2018). In most learning endeavors, challenge that is manageable and not overly frustrating encourages learners to persist and see the value in what they are learning more readily. The study shows that educators can use Scratch as a differentiation tool, as in this case to work with readers who struggle. Multimodal computer programming whether through LOGO (Papert, 1980) or Scratch might aid in students gaining basic literacy skills (Peppler & Warschauer, 2011). Scratch also challenges high-ability learners (Hagge, 2017). This is a way to differentiate beyond ability grouping, as coding allows challenge with text complexity beyond the typical. More time needs to be allotted for programming language learning to support reading and writing skills at the elementary level, instead of test preparation, and Scratch can be integrated in a variety of subject areas (Peppler & Warschauer, 2011).

As noted by Darling-Hammond, et al. (2009), teachers need a great deal of time and practice in implementing new strategies and gaining the skills and confidence needed to use them effectively. As one of the primary concerns of PSTs who did not participate in the study, time continues to be an important element of ensuring that teachers are adequately prepared with the knowledge, tools, and confidence to implement computer science instruction in the classroom. Until computer science standards are more fully integrated into state graduation, teacher education programs, and teacher certification requirements, new teachers entering the profession will unlikely be fully prepared to address coding with their students. They may, indeed, not even see the value in doing so. This study shows that there is, however, value in exploring the interplay between different types of literacies, including fundamental reading and writing skills, through the lens of technology and computer gaming. Yet, educators privilege spoken words and written text over other types of communication (Peppler & Warschauer, 2011). Coding apps such as Scratch, Code.org, and many others implement game-based features in order to provide learners with motivational challenges that continually stretch their abilities while offering them new and creative ways to explore decoding (reading) and encoding (writing). When incorporating the computational participation piece, coding prepares students to "communicate, socialize, and engage in literacy practices needed in the future careers as global citizens" (Burke, O'Byrne, & Kafai, 2017).

Limitations and Further Considerations

A limitation to this study is that, as an investigative case study, it is limited to one Midwestern region, one school, and one university, which means a small sample size and limited ability to generalize to other settings (Yin, 2009). This site also reflected a relatively low rate of socioeconomic and ethnic diversity, and only one PST used coding with a student. Although this limited the data relating coding to sight word acquisition, it did develop further questions as to why PSTs avoided coding, and provided insight into additional research that may be done regarding PST confidence and ability to use technology with literacy teaching strategies. As a follow-up in the spring 2019 semester, more PSTs (4/17) were interested in including coding with their students, yet due to uncontrollable weather issues during this semester, tutoring time to include coding was reduced dramatically.

Although the children in this study were struggling in some aspect of literacy, none had been diagnosed with dyslexia. Dyslexia presents itself as problems with word reading/decoding and spelling/encoding, issues that some of the children were experiencing. Thompson et al. (2018) found coding, in addition to direct and explicit instruction, helped students with their reading and spelling issues. Future research may wish to consider dyslexia and other print and reading disabilities with regard to differences in response to coding as a teaching intervention.

The use of coding goes beyond sight word acquisition. Future research could include multiple aspects of literacy with different children. For example, Scratch supports writing and collaboration when designing a game as others remix the game and leave comments for possible improvements, providing for the same gaming literacies that Gee (2013) discusses in his framework of understanding how video games support and enhance learning. Children also explore new career opportunities as game designers, programmers, or writers as they strengthen communication skills and visual and disciplinary literacy. This study has a strong relationship to the limited body of research currently available, and provides further insight into coding and its relationship to literacy. More research, especially qualitative studies within social contexts, is needed in how technology is actually used in school populations (Warschauer, 2004).

https://inspire.redlands.edu/jcsi/vol3/iss1/1
DOI: 10.26716/jcsi.2020.03.1.1

References

Baker-Doyle, K. (2018). I, pseudocoder: Reflections of a literacy teacher-educator on teaching coding as a critical literacy. *Contemporary Issues in Technology and Teacher Education, 18*(2), 255-270.

Behrman, E. (2006). Teaching about language, power, and text: A review of classroom practices that support critical literacy. *Journal of Adolescent and Adult Literacy, 49*(6), 490-498. https://doi.org/10.1598/jaal.49.6.4

Bers, M.U. in Lynch, M. (2018, January 29). Coding as a literacy for the 21st century. *EducationWeek's blogs*. https://blogs.edweek.org/edweek/education_futures/2018/01/coding_as_a_literacy_for_the_21st_century.html

Bouygues, H.L. (2019). Does educational technology help students learn? An analysis of the connection between digital devices and learning. Paris, France: Reboot Foundation. https://reboot-foundation.org/does-educational-technology-help-students-learn/

Bradley, L. (22 June 2017). Coding in English class? Yes! And in your Class too! *KQED Education.* https://doi.org/10.1093/oso/9780190908683.003.0007

Burke Q., O'Byrne, I., & Kafai, Y. (2016). Computational participation: Understanding coding as an extension of literacy instruction. *Journal of Adolescent and Adult Literacy, 59*(4), 371-375. https://doi.org/10.1002/jaal.496

Calkins, L. (1986). *The art of teaching writing*. Portsmouth, NH: Heinemann.

Chall, J. (1983). *Stages of reading development.* New York, NY: McGraw-Hill.

Codeacademy (*n.d.) List of commands line commands.* https://www.codecademy.com/articles/command-line-commands

Common Sense Education (2017) *Topic backgrounder: News & media.* https://d1e2bohyu2u2w9.cloudfront.net/education/sites/default/files/tlr-asset/topicbackgrounder-newsmedialit_2017_final.pdf

Computer Science Teachers Association (2018). State of computer science education: Policy and implementation. https://code.org/files/2018_state_of_cs.pdf

Creswell, J. W. (2013). *Qualitative inquiry and research design: Choosing among five approaches.* Thousand Oaks, CA: Sage.

Darling-Hammond, L., Wei, R.C., Andree, A., Richardson, N., & Orphanos, S. (2009). *Professional learning in the learning profession: A status report on teacher development in the United States and abroad.* Dallas, TX: National Staff Development Council.

Davis, M. (13 February 2018). Using the language of code to empower learning. *Literacy Daily*. https://www.literacyworldwide.org/blog/literacy-daily/2018/02/13/using-the-language-of-code-to-empower-learning

DeVries, B. (2015). *Literacy assessment & intervention for classroom teachers*. Scottsdale, AZ: Holcomb Hathaway, Publishers, Inc.

Eulenberg, J.B. (1982). Computer literacy is literacy. *Journal of Special Education Technology, 5*(4), 3-4.

Farrell, L., Hunter, M., & Osenga, T. (2019). A new model for teaching high-frequency words. *Reading Rockets*. https://www.readingrockets.org/article/new-model-teaching-high-frequency-words

Gee, J.P. (2008, August 12). Big thinkers: James Paul Gee on grading with games. *Edutopia,* https://www.edutopia.org/video/big-thinkers-james-paul-gee-grading-games

Gee, J.P. (2013). *Good video games and good learning. Collected essays on video games, learning, and literacy* (2nd ed). New York, NY: Peter Lang. https://doi.org/10.19173/irrodl.v8i3.498

Grafwallner, P. (9 January 2018). Encoding literacy in computer science: An instructional coach shares how she and a fellow teacher embedded literacy lessons in his computer science class. *Edutopia.* https://www.edutopia.org/article/encoding-literacy-computer-science

Gribbin, B. (1996). The role of generalization in studying grammar and usage. *The English Journal*, *85*(7), 55-58. https://doi.org/10.2307/820507

Hagge, J. (2017). Scratching beyond the surface of literacy: Programming for early adolescent gifted students. *Gifted Child Today, 40*(3), 154-162. https://doi.org/10.1177/1076217517707233

Hour of Code (2019). *What is the hour of code?* https://hourofcode.com/us

Hutchison, A., Nadolny, L. & Estapa, A. (2016). Using coding apps to support literacy instruction and develop coding literacy. *The Reading Teacher*, *69*(5), 493-503. https://doi.org/10.1002/trtr.1440

Indiana Department of Education (2011). IREAD-3. https://www.doe.in.gov/assessment/iread-3

Indiana Department of Education (2017). *Science & Computer Science Standards.* https://www.doe.in.gov/standards/science-computer-science

International Literacy Association (2020). *Literacy Glossary.* https://www.literacyworldwide.org/get-resources/literacy-glossary?Site=ILA

International Society for Technology in Education (2019). ISTE Standards for Computer Science Educators (draft). https://www.iste.org/standards/for-computer-science-educators

International Society for Technology in Education (2018). ISTE Standards for Computational Thinking. https://www.iste.org/standards/computational-thinking

Kafai, Y. & Burke, Q. (2014). *Connected code: Why children need to learn programming.* Cambridge, MA: MIT Press. https://doi.org/10.7551/mitpress/9992.001.0001

Kidd, J.K., Pasnak, R., Gadzichowski, K. M., Gallington, D.A., McKnight, P., Boyer, C.E. & Carlson, A. (2014). Instructing first-grade children on patterning improves reading and mathematics. *Early Education and Development, 3,* 134-151. https://doi.org/10.1080/10409289.2013.794448

Krathwohl, D.R. (2001). A revision of bloom's taxonomy: An overview. *Theory Into Practice, 41*(4), 212-218. https://doi.org/10.1207/s15430421tip4104_2

Lee, Y. (2011). Scratch: Multimedia programming environment for gifted learners. *Gifted Child Today, 34*(2), 26-31. https://doi.org/10.1177/107621751103400208

MIT Media Lab (n.d.-a). *About Scratch Jr.* https://www.scratchjr.org/about/info

MIT Media Lab (n.d.-b). *About Scratch.* https://scratch.mit.edu/about

National Academies of Sciences, Engineering, and Medicine (2018). *How people learn II: Learners, contexts, and cultures.* Washington, DC: The National Academies Press. https://doi.org/10.17226/24783

NWEA (2020). *NWEA website.* https://www.nwea.org/

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

Peppler, K., Santo, R., Gresalfi, M., & Tekinbas, K.S. (2014). *Script changers: Digital storytelling with Scratch.* Cambridge, MA: The MIT Press. https://doi.org/10.7551/mitpress/9695.001.0001

Peppler, K. & Warschauer, M. (2011). Uncovering literacies, disrupting stereotypes: Examining the (dis)abilities of a child learning to computer program and read. *International Journal of Learning and Media*, *3*(3), 15-41. https://doi.org/10.1162/ijlm_a_00073

Portnoff, S. (2018). The introductory computer programming course is first and foremost a language course. *ACM Inroads, 9*(2), 34-52. https://doi.org/10.1145/3152433

Prensky, M. (2006). *Don't bother me, Mom, I'm learning! How computer and video games are preparing your kids for 21st century success and how you can help.* St. Paul, MN: Paragon House.

Randles, J.R. (2020, February 6). 6 reasons for coding in K-5 classroom. *ISTE blog.* https://www.iste.org/explore/In-the-classroom/6-reasons-for-coding-in-K-5-classrooms

Thompson, R., Tanimoto, S., Lyman, R., Geselowitz, K., Begay, K., Nielsen, K., …Berninger, V. (2018). Effective instruction for persisting dyslexia in upper grades: Adding hope stories and computer coding to explicit literacy instruction. *Education and Information Technologies, 23,* 1043-1068. https://doi.org/10.1007/s10639-017-9647-5

Vee, A. (2017). *Coding literacy: How computer programming is changing writing.* Cambridge, MA: The MIT Press.

Warschauer, M. (2004). *Technology and Social Inclusion: Rethinking the Digital Divide.* Cambridge, MA: The MIT Press. https://doi.org/10.1108/00220410610653352

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3). http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf

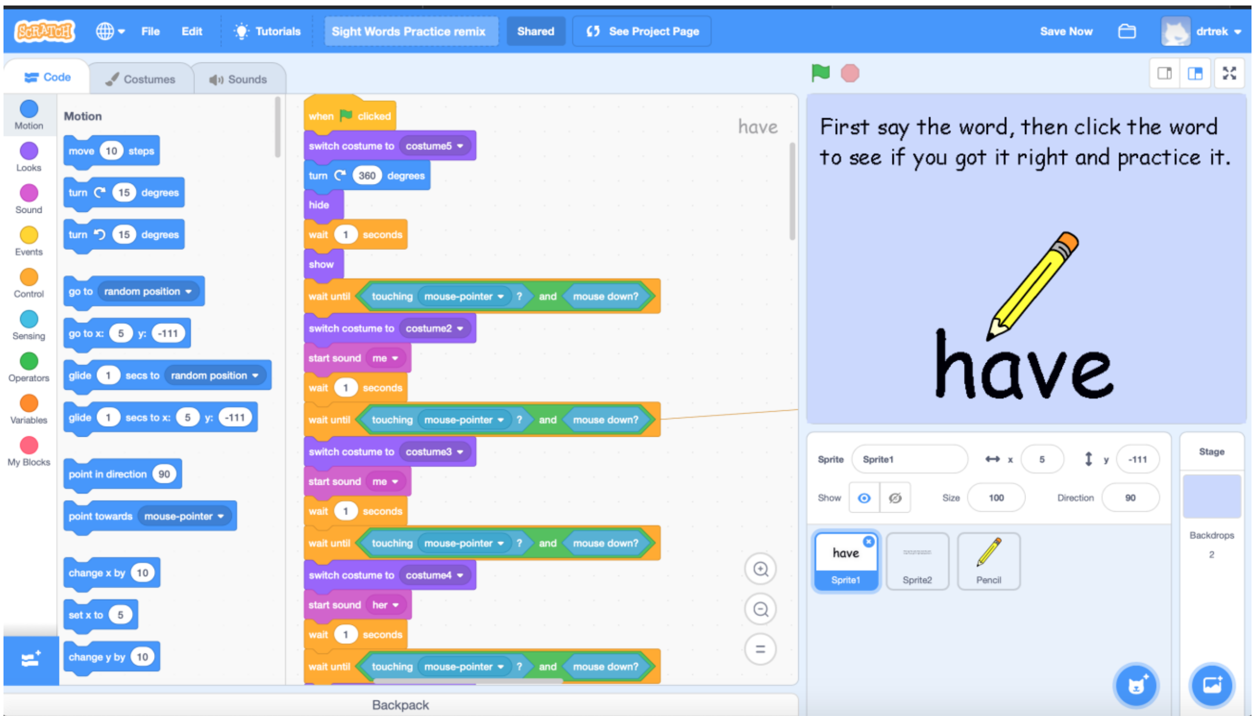
Yienger, M.E. (2016). Too much tech harms reading retention in young children. *Inquiries, 8*(3). http://www.inquiriesjournal.com/articles/1374/too-much-tech-harms-reading-retention-in-young-children

Yin, R.K. (2009). *Case study research: Design and methods.* Thousand Oaks, CA: SAGE Publications.

DOI: 10.26716/jcsi.2020.03.1.1

Appendix A

Screen shot of Scratch remix used in study



Appendix B

Participant Prior Knowledge Assessment

Thank you for consenting to participate in this study. This study is about understanding how preservice teachers see coding instruction, specifically through the Scratch application, as related to sight word acquisition and literacy skills development. To give us a better understanding of your prior knowledge, please respond to the following prompts:

- Tell us about your prior knowledge or experience with coding. For example, have you had any computer science classes where you have learned about or used coding? Even if you haven’t had experience coding yourself, what do you know about coding? Where is coding used?
- Share your experience and what you know about word study strategies. Word study is phonics, spelling, and vocabulary instruction. What strategies do you know about or have you used with children to help them with word study?
- Explain any knowledge you have about the relationship between coding and word study and/or reading.
- Please share your experience or knowledge of the Scratch website.

Appendix C

Initial Study Participant Questionnaire

Name:____

Pre-Study Questions

1. Please tell us why you chose to participate in this study. For example, was it due to the needs of your student(s)? Are you interested in learning in general/coding/technology?

2. Describe, in general, the reading needs of your student who you believe may benefit from using coding.

3. Tell us what you know (if anything) about coding.

4. Please describe what you know about word study strategies.

5. What are your perceptions about the connections (or lack of connections) between coding and reading?

6. What, if any, expectations do you have after participating in this study?

Appendix D

Focus Group Questions

1. Why did you choose to participate or not in the study this semester?

2. What would be your suggestions for increasing participation next semester?

3. What are your ideas for using coding in literacy?

4. What apprehension do you have related to coding?