Spring 2020

Coding in the Classroom

Sierra J. Fresh

Coding in the Classroom

By
Sierra J. Fresh

An Honors Thesis Submitted in Partial Fulfillment of the
Requirements for Graduation from the
Western Oregon University Honors Program

Dr. Rachel Harrington, Education and Leadership
Dr. Breeann Flesch, Computer Science
Thesis Advisors

Dr. Gavin Keulks
Honors Program Director

June 2020

Acknowledgements

Graduating college has been one of the more challenging, yet rewarding things I have done. I have learned so much and grown in so many ways during my time at WOU. I truly could not have done this without the many people who were by my side the whole time. First, I am so thankful for my family, who have supported me in countless ways through these four years. Without your loving arms to return to when I was uncertain, overwhelmed, or celebrating, this experience would not have been as meaningful to me.

I also want to thank Madi, who has been more like a sister than a roommate to me. As I look back on my college experience, you have been one of the most steady and supportive people in my corner. We have gone through so much together, and I appreciate you. Also, thanks for creating the basis for this project with me as my co-teacher during summer school!

To the people who have come into my life, or chosen to stay in it this last year, I appreciate you and I thank you. This has been a hard year, but such a rewarding one as I became more of myself. Thank you to my roommates, to my girlfriend, and to my family for loving me unconditionally.

Finally, I could not have made my way through college without the support of my professors in the Education Department. Thank you to Rachel Harrington for supporting me in this project, Kristen Pratt for guiding me through my student

teaching and ESOL endorsement, and to my many other education professors who have helped me learn how to be a culturally responsive educator who advocates for students. I can't wait to go into the field and follow in your footsteps.

Table of Contents

Abstract

The 21st century has been characterized by rapid growth in technology and computer science. With this shift, computer science curricula have not always been introduced in the classroom at a similar pace (Yadav, Hong, & Stephenson, 2016). With the overwhelming amount of curriculum that needs to be taught and lack of resources at the district level, adding non-required curricula such as computer science and coding to the schedule can be difficult. There simply is not enough time in the day to teach it all. This project bridges the gap between computer science curriculum and the classroom, by showing how it overlaps with the Oregon Common Core State Standards.

I have created a unit that teaches coding to fifth graders. It includes specific learning objectives aligned to the Common Core State Standards for Mathematics. These lessons teach coding using a scaffolded approach based on Bloom's Taxonomy. This project provides teachers with a small unit for their fifth-grade class that will teach coding and that aligns with the standards that are required to be taught, which brings the computer science and elementary mathematics education worlds together. The existing resources to teach coding lack alignment to Oregon Common Core State Standards, so the goal of this project was to add alignment in order to make teaching computer science curriculum accessible for the classroom teacher.

Statement of the Problem

A basic level of computer competency is required for practically anything in our technological world (Pierce, 2013). For many years, schools have offered computer literacy courses that teach things like keyboarding or basic computer processing skills. Though these are beneficial, they do not go beyond the simple operation of computer technologies. As we become more reliant on technology, the abilities to create with technology and understand the inner-workings of technology are of growing importance as the job market produces more and more of these jobs (Kazakoff, Sullivan & Bers, 2013). This kind of creating with technology is a huge part of what is done when coding. Computer coding also teaches sequential thinking and computational thinking (Kazakoff, Sullivan & Bers, 2013), which are skills that are valuable across subject areas and throughout life. Sequential thinking focuses on the step-by-step processes that are required in many areas of life. They are used in things from telling a story to cooking a meal. Computational thinking allows students to begin to think the way a computer does - with methodical steps which are powerful for problem solving. It also helps people break large problems into smaller ones that are more approachable. Since coding education can help teach ways of thinking that are valuable in many spheres of life, educators are beginning to realize that teaching computer science curricula geared toward coding and programming might be an important addition to classroom curriculum.

Despite the growing awareness of how technological understanding is a vital skill for students, we have not seen educators begin to involve a programming-focused, computer science curricula in the classroom at significant rates (Yadav, Hong, & Stephenson, 2016). The sheer quantity of content that is presented in any given grade makes it difficult to add new material, and begs the question of whether or not “topics remain in the curriculum because of tradition or, more important, whether [these topics] are necessary in promoting students’ readiness for college, careers, and life” (National Council of Teachers of Mathematics, 2014, pg. 73). In the educational culture of today, subjects that are tested are given more resources and time than non-tested subjects. Since coding education is not listed in any academic standards, it is very difficult to incorporate it into the school day. With this disparity between the importance of teaching computer science, and the lack of time and resources to do it, it seemed clear that there was a need for a computer science curriculum that is cross-curricular and ready to teach. A curriculum like this had the potential to be closely tied to Common Core State Standards for Mathematics (CCSS-M) and accessible for. This is what I hoped to create with this project.

Currently, three out of five schools in the U.S. do not offer any computing courses that include programming or coding (Google Inc. & Gallup Inc, 2016). While computer science professionals have been creating many resources to bring children into the world of coding, many of these resources lack alignment to the CCSS-M. I wanted my project to address this problem of alignment between the available coding resources and the CCSS-M that schools need to teach. Furthermore, for schools lacking computing courses, a very common barrier is a lack of teacher confidence in delivering computer science instruction (Microsoft Philanthropies, 2018). With coding resources which are strongly connected to the CCSS-M, I hope that teaching programming will become approachable for teachers with little experience of their own. I believe that with the clearly defined lesson plans and aligned standards that I created teachers will gain confidence in their ability to teach coding to their students.

With my project, I also hope to make programming accessible for students that would not otherwise be involved with a coding curriculum. In many schools, coding curriculum falls into after school "Coding Clubs" (Firth, 2014). These can present accessibility issues for many students, as they require an adult to arrange alternative transportation for the student up after the club is over. This can be difficult for working parents who rely on bus transportation for their children. Furthermore, the nature of coding requires technology, which can be an added expense for lower income families. Bringing coding into the general education classroom allows students who might not have the chance to explore it the option.

Bringing coding into the classroom also creates opportunities for encouraging women to be involved in STEM. Women are avid consumers of technologies but are notably underrepresented in technological fields. This inequity is problematic because the computing and technological industry are fast-growing and high-paying (National Center for Women & Information Technology, 2012). Bringing computer science education, specifically coding, into the classroom may help to create a pipeline of female coders. Overall, with this project I hope to create easy access to coding curriculum for elementary educators, that is cross-curricular and accessible to a diverse range of students.

Review of Literature

Coding education is a relatively new sphere of education. Because of this, there is not an overwhelming amount of research available to provide us with best practices. While the canon of research is not vast, there is some information available. In the following sections I will attempt to synthesize the research that exists in order to set up our later discussion of how to implement coding education in a 5th grade classroom.

Benefits and Drawbacks of Technology and Young Children

The children of today are growing up in a world where technology is practically an extension of their own body. This fact alone creates a distinction between them and their parents, who grew up without the saturation of technology that we are seeing in today's world. This concept can be defined by the idea of Digital Natives and Digital Immigrants. Digital Natives are those who were born after the invention of digital technologies, and who would be more comfortable accessing them. Digital Immigrants are those who were born before the invention of digital technologies, and who had to learn how to use them after growing up without them. These differing generations understandably have different relationships to technology because they grew up in very different technological worlds (Chaves, Hamilton Viana, Maia Filho, Osterne Nonato, &

Melo, Armando Sérgio Emerenciano de., 2016). For this reason, there are many differing opinions about the impact technology has on young children, both inside and outside of the classroom. In many cases, it can be jarring for parents to see their children learning through the use of technology, especially because it is so different from how they themselves learned in their K-12 schooling. Manches and Plowman (2017) argue that there is no question that there are benefits to technology use and computing education in schools, however, the body of evidence is not yet vast enough to inform the best practices in this type of education. For this reason, in their article, they suggest that the education community wait to implement computing education in our schools until more research is available to inform our practices. However, following the concept of Digital Natives vs. Digital Immigrants, Chaves, Hamilton Viana, Maia Filho, Osterne Nonato, & Melo, Armando Sérgio Emerenciano de. (2016), would argue that while we certainly need to evaluate best practices for teaching Digital Natives with technology, we should not avoid it. It is a necessary and important aspect of modern education.

Furthermore, there has been a call to pause the efforts to create coding programs based on the reality that our schools are filled with extremely diverse students, who will grow up to do a variety of things. Firth (2014) suggests that not every child will grow up to become a computer programmer, so should all students learn to code? While certain experts such as Manches and Plowman call for a slight pause in the initiation of computing education programs (such as programs to teach Microsoft Office and typing skills) in schools, others along call for an increase in this type of education, and even more focused programs to teach computing skills such as computer programming. Resnick (2014), who is the director of the lifelong kindergarten group at the Massachusetts Institute of Technology's Media Lab, suggests that programs that teach Microsoft Office, for example, are no longer enough for the students of today. The author, along with Firth (2014), argue that while it is true that not every student will grow up to become a computer programmer, the ways of thinking that are learned through coding are very valuable skills for this growing generation. According to Resnick (cited in Firth, 2014), coding has the potential to give you a new way of thinking about yourself and the world around you. Coding education, which helps students understand technology and how it interacts with the world, can have a big impact on the way they understand they utilize and employ their technological resources in their lives and careers. Furthermore, according to Firth (2014), there is evidence that coding can improve abstract thinking and problem-solving abilities. In Douglas Clements' research in the 1980s, we see evidence that using the programming language Logo increases young children's ability to think abstractly (Firth, 2014). The development of abstract thinking is one of the main goals in the

CCSS-M. These standards, throughout every subject, promote the idea that helping students learn how to think, persevere, and problem solve is one of the most important things education can do. When young children are given an education using technology, specifically programming, these ways of thinking are promoted.

There is some research on how coding education affects older children, specifically in those middle school. We know that the middle school years can be pivotal in a child's sense of self, according to child psychologist (Erikson, 1959). During the middle school years, children are beginning to struggle with finding their place among their peers. According to Pierce (2013), coding education with middle schoolers can lead to improved self-esteem, classroom engagement and participation of all abilities. This study is based on anecdotal evidence of middle school children actually going through a programming curriculum. The author found that integration of students with diverse backgrounds is extremely important in the education system, and can be achieved in part with a coding education.

Best Practices in Coding Education

Within education, well researched and understood best practices guide pedagogy in order to deliver content in the most effective way possible. This is no different with coding education. While the amount of research on best practices available in this specific niche of education is not as large as in other areas, there are studies that can help guide pedagogy. In the United Kingdom, coding education has been implemented for many years (Firth, 2014). While it would be difficult to simply transplant their curriculum to the United States, we can use it to guide our development of pedagogy and curriculum around coding education. Firth (2014) describes an after-school coding program in which students ages nine and ten learn to program in many different coding languages. This program is being implemented across all of England, but will no longer just be an after-school club, it will be a subject taught in schools during the day. According to Bers (2014), one of the interviewees in the article, learning how to program is a new type of literacy. This specific program is training teachers in the best practices they believe are important. Above all, they want to support students as they design, create and express themselves through coding (Firth, 2014). With this in mind, they try to encourage kids to struggle to solve problems in their programs by themselves instead of immediately calling for help. The author suggests that this helps to teach children the valuable skills of creative thinking and problem solving. According to another similar program to teach programming, encouraging students to work together to solve the problems they run into can also be valuable in teaching collaboration (Pierce, 2013). The author argues that students who have solved common problems can support their peers. For these two already-established programs, best practices that promote creativity, perseverance, and collaboration should be encouraged.

There are some specific ways these authors suggest promoting creativity, perseverance, and collaboration. One of the most important ones is allowing students to have freedom. Both Firth (2014) and Pierce (2013) stress the importance of allowing students to create their own projects rather than the teacher assigning them. This can help create buy-in that would be much more difficult if the teacher provided the project. According to these authors, when students feel ownership over what they are doing, they are more willing to persist through the difficulties and struggles. Pierce (2013) describes encouraging students to connect their projects to the things they are learning in their content classes. This can help them develop even more understanding of the content curriculum as they use and apply it in interesting and unique ways.

Another best practice in coding education that has been promoted is to teach through a lens of constructionism. The term constructionism, invented by Seymour Papert, means that constructing something externally (a product) helps to build the knowledge internally, thus taking that knowledge from being purely abstract to concrete (Papert, 1993). With coding, Pierce (2013) explains that this means creating opportunities for students to practice. When they enter a code and then see the robot move, constructionism helps them to grasp whatever concept they are working on in a deeper way. Garov and Tabakova-Komsalova (2017) explain that ten and 11 year olds should learn programming through starting with small tasks and building up to larger tasks once they fully understand the smaller ones. The way they describe these small stepping stones to coding fits with Papert's idea of constructionism, as creating a product helps solidify the smaller, but very important ideas of coding (such as sequential thinking and cause and effect), in order to move on to the larger tasks.

Why Teach Coding?

While coding education is still relatively new and has not been researched to the extent of other fields of education, there has been research on the ways of thinking that coding develops and why those are beneficial for young children. Coding can help develop ways of thinking that can benefit children later in their lives, such as teaching computational thinking, abstract thinking, and sequential thinking (Firth, 2014). These ways of thinking are important in scientific fields, and in computer science specifically. According to this research and others that we will look at, we should teach coding education.

In the educational world of today, we are preparing children for jobs that might not even exist yet (Kazakoff, Sullivan & Bers, 2013). The rate of job growth and creation is increasing quickly, which means that the job market students are living in today will be vastly different from that of the future. According to the authors, with this in mind, we need to be proactive in the ways that we are teaching and preparing students for their futures. We cannot sit back and prepare them for jobs that will be obsolete by the time they graduate. However, we run into the problem that we do not actually know the types of jobs that will exist in 20 years. Thus, we must prepare them for their futures by helping them learn to think in the ways that the tech industry requires.

Thinking sequentially and computationally are ways of thinking that are extremely important to this field of study. According to Firth (2014), training students to think in these ways gives them critical problem-solving skills. This goal fits in very well with the goals of our public education system, where we teach things such as Standards for Mathematical Practice, which aim to teach similar forms of thinking to those taught in coding education. These standards focus on skills such as perseverance, critical thinking, and being precise (National Governors Association Center for Best Practices & Council of Chief State School Officers, 2010). The Standards for Mathematical Practice are newer to teachers than content standards, which in some cases can make teachers less confident to teach them. Coding is a great way to incorporate these Standards for Mathematical Practices.

Another type of thinking that is taught through coding education is sequential thinking. Sequential thinking refers to the step by step thinking required to solve problems. This kind of thinking helps students develop the ability to think beyond their current step and plan for what will happen next (Manches & Plowman, 2017). According to the research done by Kazakoff, Sullivan and Bers (2013), after only one week of coding education, sequencing ability is shown to go up. The study was performed in an Early Childhood classroom, with Kindergarteners learning how to code, however many of the principles can be abstracted to apply to older students. While there is a lot of development that takes place between years 5 and 10 (ages of typical Kindergarten and 5th grade students), classrooms in which every student is at the same level simply do not exist. Students are diverse, and so are their independent skill sets. With this in mind, the research that sequencing ability goes up through coding education can be applied to an upper elementary classroom in order to point out the potential benefit to introducing this type of curriculum.

Sequencing is a skill that is used in many realms of education. For example, in many of the Common Core State Standards for Literacy, sequencing is a skill that should be taught. Standard W.5.3, which is a 5th grade writing standard, states “Write narratives to develop real or imagined experiences or events using effective technique, descriptive details, and clear event sequences” (National Governors Association Center for Best Practices & Council of Chief State School Officers, 2010). This standard asks 5th grade students to demonstrate their ability to sequence events in a clear way. This skill is one that needs to be developed. Through the research done by Kazakoff, Sullivan and Bers (2013), it seems clear that coding would be able to at least help students become more successful at this.

Overall, according to Pierce (2013), being able to program and code is important to being considered “fluent” in today’s world. We teach students to read because this ability creates many other opportunities and is necessary for being able to interact with the world. In the same way, learning to code is important for students to become “fluent” in the world we live in. The ways that coding helps you think about things differently and allows you to be creative make it a beneficial addition to the fast-moving world of today. There is benefit in learning how to think abstractly, computationally, and sequentially, not only because these forms of thinking are used in many technological fields, but because they are used in many areas of life. Students should be equipped to think in ways that are required in different realms of society, and coding is one very effective way that children can be taught to do this.

Methods

Coding in a “Free Choice Learning” Environment

During the summer of 2018, I had the opportunity to teach coding in a summer camp for students involved in a regional Dual Language Program. The camp’s goal was to help students maintain and practice their Spanish over the summer while taking part in fun and interesting activities. Students were able to choose what “club” they wanted to be part of. Options that were available in the camp ranged from theater, leadership, cooking, to cinematography. Along with my teaching partner, I taught a four-week coding camp in Spanish. The lessons that we taught in our coding club were created by us for this free-choice learning environment, which is an environment in which students are able to choose what they want to learn about. In this environment, our goal was to inspire students to find a love for the Spanish language through the use of technology, specifically coding.

In a free choice learning environment, I discovered that students were excited to learn and participate because they found enjoyment in what it was that they were learning. Engagement in education is important because without students focused on what they are doing, it is very difficult to actually teach anything. Technology is naturally engaging for students of today’s world. They grow up learning how to use it in various ways, so when it comes to learning how to control and create it, they are generally very interested. I wanted to take this engagement that comes from when students learn something that they are interested in, and use it to teach the standards that we have to teach as educators, in a classroom setting.

For this project, I used the lessons we created and the things I learned in order to create a coding unit that can be used in a more structured learning environment. The free choice learning environment that I developed this unit in was very fun and allowed students to play around with the things that can be done through coding, but I believed there was a lot of potential to connect this learning to the traditional classroom setting.

One area where my unit diverged from the free choice environment was through connecting it to CCSS-M. Given that it was developed in an environment that was aiming to help students have fun, we did not have standards that we were connected to. We had objectives every day, but those were based on the goal we had for students relating to coding. Through looking carefully at the CCSS-M content standards and Standards for Mathematical Practice, I created new lessons and adapted my existing lessons to align to them.

Furthermore, in connecting coding, a non-tested and non-required subject, with mathematics, a frequently-tested and required subject, teachers will be able to teach two subjects at once. In the short school day and school year that we have, it is important to teach as effectively as possible in order to accomplish the many things that need to be done during the year. Teaching math through teaching coding is one way to capitalize on time in the classroom, while teaching students valuable skills and ways of thinking that can help them in their futures.

Lesson Plan Template:

In order to create these lessons, I used a basic lesson plan template. This template is created with the *gradual release of responsibility* in mind. This lesson style demonstrates a teacher slowly releasing independence of a new task to students so that they might have the most success possible when asked to do the task on their own (Pearson & Gallagher, 1983). In Figure 1 below, lesson plan template is in black and I have included written explanations directly in each section in a different color.

<table>
<tr><td colspan="4">Lesson Title/Description: The name of the lesson and a brief description of the lesson so that someone can understand roughly what the lesson is about from reading just one section.</td></tr>
<tr><td>Standards (content and SMP):

Standards that will be worked on in the lesson.

• Common Core State Standards (Mathematics Standards)

• Standards for Mathematical Practice</td><td colspan="2">Central Focus:

The central focus is the persisting principle or idea that students should learn from the entire unit. This central focus is what I would hope students will remember if they only remember one thing about the unit.</td><td>Learning Targets:

Learning targets vary every day and are the thing that the specific lesson focuses on teaching.</td></tr>
<tr><td colspan="4">How have you addressed the needs of diverse learners ? (Ex: IEPs, 504s, linguistic & cultural diversity, students without prerequisite knowledge, etc.)

This can vary class by class. In a classroom setting, this is where you think about students in your class who have varying and diverse needs.

I will add supports that can help a variety of students, but in a more general sense than I would if I knew the specific group of students that I will be teaching.</td></tr>
<tr><td colspan="4">Materials/Equipment/Supplies/Technology/Preparation:

All of the materials that will be needed to teach the lesson will be listed here.</td></tr>
<tr><td colspan="2">Procedure: Teacher Does……….</td><td colspan="2">Procedure: Students Do……..</td></tr>
</table>

Time How long will this part of the lesson take?	**Motivation/Hook:** What will the TEACHER do to get students interested in the lesson?	**Motivation/Hook:** What will the STUDENTS do?
How long will this part of the lesson take?	**Teaching/Group Application/Independent Application:** Each one of these will be included in every lesson. This is where the Gradual Release of Responsibility takes place. First, how will the TEACHER teach the material? Second, how will the TEACHER support students practicing in a group setting? Finally, how will the TEACHER release responsibility and support as students practice individually?	**Teaching/Group Application/Independent Application:** What will STUDENTS do as the teacher is teaching? How will the STUDENTS practice as a group? Will it be full class? Small groups? How will the STUDENTS practice and show understanding on their own?
How long will this part of the lesson take?	**Closure:** What will the TEACHER do to wrap up the lesson?	**Closure:** What will the STUDENTS do to wrap up the lesson?

Figure 1. Annotated Lesson Plan Template.

Pilot Testing at a Local Elementary School's Math Night:

During the Spring of 2019, I took part in a local elementary school's Math Night. This was an opportunity to pilot test the first lesson focused on developing sequential thinking. This Math Night setting is not 100% consistent with the environment that I am planning my lessons for, but it gave me the opportunity to see where my lesson works well and in what areas I should change it to be more effective at teaching the learning targets.

The lesson was called "Code Your Partner" and it allowed students to take on the role of either "Programmer" or "Robot". In this lesson, the Programmer's job is to direct the Robot through a large floor maze and around obstacles. The goal is to teach the Programmer that it is important to be extremely specific and concise when programming their robot, or they run the risk of creating a faulty code and the robot not completing the task of finishing the floor maze.

Through the experience of piloting my lesson at Jefferson's Math Night, I realized a few things about this lesson that were extremely successful. First, it is very engaging for students. They enjoyed taking on the role of either "Programmer" or "Robot" and using their thinking skills creatively to solve a problem. I found that some students wanted to stay and participate in this activity for an extended period of time. This lesson also has a lot of opportunities for extensions, which makes it appropriate for a variety of skill levels. For use with students who are a bit older or more advanced at thinking sequentially, you can include obstacles that have students use commands such as “step over” or “go under”. These allow for a more advanced level of thinking. This lesson also has potential to be turned into a game, which could allow students to use a deeper level of thinking as they are thinking strategically as well as sequentially.

There were also a few ways that I would change this lesson based on this pilot test. Because this lesson uses a floor maze, it takes up a lot of space, and only allows for one pair of students to actively participate at a time. It would be useful to create something for the other students to be doing while they are waiting, especially in a full-class setting. Down-time can lead to off-task behaviors, which take away from the potential learning that is taking place. Students could do the maze from other directions as one way to include more students. They also could be in charge of moving obstacles around as they get more advanced in their sequential thinking skills.

While there were some areas of improvement that I noted as a result of this pilot test, I received great feedback from community members. The Corvallis High School Robotics club gave me the feedback that they really enjoyed the lesson and found it to be very unique. They asked me for a copy of the lesson plan to use in their community outreach events. I was excited to give them a PDF copy of this lesson as it was on the Math Night, though I will be updating it to include the changes I have talked about as well as to make it more appropriate for a classroom learning environment. Overall, this experience has given me the information that these lessons are not only engaging, but have the ability to teach the thinking skills that I am aiming to teach.

Results

Lesson Plans

Lesson Title/Description: LESSON 1: Code your partner		
Standards: **5.G.A.1:** Use a pair of perpendicular number lines, called axes, to define a coordinate system, with the intersection of the lines (the origin) arranged to coincide with the 0 on each line and a given point in the plane located by using an ordered pair of numbers, called its coordinates. Understand that the first number indicates how far to travel from the origin in the direction of one axis, and the second number indicates how far to travel in the direction of the second axis, with the convention that the names of the two axes and the coordinates correspond (e.g., *x*-axis and *x*-coordinate, *y*-axis and *y*-coordinate).	**Central Focus:** Programmers use precise language and sequential directions to code.	**Learning Targets:** Using a floor-sized coordinate plane, students will be able to give specific and precise directions to guide their blindfolded partner through a maze without running their partner into anything.

<table>
<tr><td colspan="2">SMP 6: Attend to precision.</td><td></td><td></td></tr>
<tr><td colspan="4">How have you addressed the needs of diverse learners? (Ex: IEPs, 504s, linguistic & cultural diversity, students without prerequisite knowledge, etc.)

I will provide sentence frames to help emergent bilinguals.

Sentence frames will include:

Take __________ steps forward/back.
Turn to the right/left.
Take __________ steps over __________.</td></tr>
<tr><td colspan="4">Materials/Equipment/Supplies/Technology/Preparation:

-anchor chart that defines sequential thinking and precision
-sentence frames
-floor grid and maze
-obstacles for the maze
-blindfold</td></tr>
<tr><td colspan="2">Procedure: Teacher Does……….</td><td colspan="2">Procedure: Students Do……..</td></tr>
<tr><td>Time

5 min</td><td>Motivation/Hook:

Teacher will talk to the students about sequential thinking and being precise, using the anchor chart.
Sequential thinking is the ability to think beyond what is happening in the present.
Being precise refers to giving directions specifically and exactly.</td><td colspan="2">Motivation/Hook:

Students will listen to the anchor chart explanation of what sequential thinking and being precise mean.</td></tr>
</table>

<table>
<tr><td></td><td>The teacher will write both of these definitions on an anchor chart and talk about them with the class.</td><td></td></tr>
<tr><td>

7 min</td><td>Teaching:

The teacher will explain that for this activity, half of the class is going to transform into robots and the other half will be programmers, and that they will code each other.

The teacher will start by demonstrating the activity as the robot, and asking the students to be the programmers.

The teacher will follow the directions given by the students <u>exactly.</u>
For example, if the students say, “go forward”, the teacher must walk forward indefinitely, until they run into something or the programmer says “stop”.</td><td>Teaching:

The students will listen to the explanations given by the teacher. They will code the teacher, learning that they need to be extremely precise.</td></tr>
</table>

 1 min **5 min** **5 min**	**Group Application:** The teacher will choose partner groups, and ask the students to decide on who will be the robot first and who will be the programmer. They will switch eventually, so this doesn't matter very much. The teacher will ask students to practice guiding their partners through mini mazes, with the robot partner having a blindfold at the discretion of the teacher. ● The teacher will remind students that they need to use very precise language and sequential thinking skills. After 5 minutes, partners switch roles and practice again. The teacher should rotate around the room and make sure students are giving precise directions. Listen for students giving vague directions like "walk forward" instead of identifying a number of steps. The teacher can prompt the learner with questions like "how many steps would you like your robot to take?" or "Do	**Group Application:** Students will practice coding each other, switching roles after 5 minutes. Then, they will work in the larger floor mazes and practice coding each other. They will practice using very precise language, and can reference sentence frames if needed.

	you think that direction is precise enough for your robot?"	
10 min 10 min	**Independent Application:** Once 5 minutes is finished, direct half of the class to one of two larger floor mazes, and have the other half go to the other one. Describe that this is the "challenge maze" where programmers will really test their skills. Have programmers code their robots through the larger floor maze. This maze will serve as a sort of "challenge" after the practice maze. Then, have them switch to the other maze that is set up and switch which partner is the robot. Once the students have coded for 10 more minutes, bring them back together.	**Independent Application:** Programmers direct robots through floor maze. Students switch roles and the new programmers direct the new robots through the other floor maze.
5 min	**Closure:** The teacher will bring the students back together and ask	**Closure:** Students will volunteer to do the maze in front of the class.

	some groups to present to the class, if they want. They will talk about the important things they learned, such as precision and sequences. The teacher can ask some guiding questions such as: ● What mistakes did you make? ● How did you fix those mistakes? ● What did you change the second time you did it? ● Did you develop any strategies to help you? The teacher will close the lesson by describing that the big idea is that programming depends on being precise and using our sequential thinking skills to know what we want to do a few steps in the future.	They will participate in a discussion about the important things they learned.

Rubric for Lesson 1

Criteria	Meets	Does not meet
Directions given by the student are precise and specific. The directions the student gives should		

be clear and without room for misinterpretation. Ex: "Walk forward" would not be specific or precise but "Walk forward three steps" would be specific and precise.		

Lesson Title/Description: LESSON 2: Code.org Course F		
Standards: SMP 1: Make sense of problems and persevere in solving them. SMP 2: Reason abstractly and quantitatively. SMP 6: Attend to Precision.	**Central Focus:** Programmers use precise language and sequential directions to code.	**Learning Targets:** Using Course F, Lesson 2 on Code.org, students will be able to use precise coding language (blocks) in order to advance an angry bird through a maze.
How have you addressed the needs of diverse learners ? (Ex: IEPs, 504s, linguistic & cultural diversity, students without prerequisite knowledge, etc.) ● The 2017 version of Course F is available in 6 different languages for emergent bilinguals ● There is an offline coding activity that can be utilized if students need to be more physically involved to learn		
Materials/Equipment/Supplies/Technology/Preparation:		

<table>
<tr><td colspan="3">

- Technology access (chrome books, computer labs, etc)
- Code.org Course F, Lesson 2
    - The teacher can set up a classroom on Code.org that allows them to track student progress in real time. This can help manage off task behavior as well as give feedback as students are practicing. In order to do this, the teacher should make a teacher account. Once they have done this, they can add or remove parts of the lesson that they want to in order to fit it to their own students. The following lesson plan will use all of Course F, Lesson 2.

</td></tr>
<tr><td colspan="2">Procedure: Teacher Does……….</td><td>Procedure: Students Do……..</td></tr>
<tr><td>Time

5 min</td><td>Motivation/Hook:

Teacher passes out technology

Teacher guides students to access their classroom on Code.org

Teacher references the learning target for the day and helps students get started.

Teacher asks students if they know what angry birds are. The teacher should try to create excitement by explaining that today the students will be coding their own angry birds game!</td><td>Motivation/Hook:

Students access Code.org through the technology provided and find their own classroom.</td></tr>
</table>

3 min	**Teaching:** Utilizing the video provided at the beginning of the lesson, the teacher will help students understand how to use Code.org.	**Teaching:** Students will watch the video as a class. They may ask questions or refer back to it as they wish.
5 min	**Group Application:** Teacher guides students through the first segment of the lesson. Teacher should model reading the instructions at the top and talk through how to drag blocks. Take student input as far as how many blocks to use. **It is okay if students are wrong, this is a learning experience and you can try again. **	**Group Application:** Students watch the teacher model and participate by suggesting how many blocks should be used to guide the angry bird through the maze.
15 min	**Independent Practice:** Teacher tasks students with finishing the rest of the lesson. Teacher tells them they will have 15 minutes to get as far as they can and if they finish early, they can continue with Lesson 3.	**Independent Practice:** Students work through the lesson independently or in groups depending on technology availability. Students ask for support as they need it.

	Teacher walks around to offer support and manage technology use. Teacher can ask guiding questions as they check in with students: ● What have you tried so far? ● What has worked? What hasn't? ● Can you explain to me how you know to do ______?	
3 min	**Closure:** Teacher references back to the learning target and reminds students that this is what they worked on today. Teacher references the Standard for Mathematical Practice and asks students questions: ● What examples of attending to precision did you see in our community today? ● How did you make sense of problems and persevere in solving them?	**Closure:** Students participate in reflection.

	• What did we do in this activity to reason abstractly and quantitatively?	

Rubric for Lesson 2

Criteria	Meets	Does not meet
Student successfully advances the angry bird through the mazes using the blocks of code. **The number of mazes that need to be completed to "meet" would be up to the individual teacher's discretion**		

Lesson Title/Description: LESSON 3A: Spheros		
Standards (content and SMP): SMP 1: Make sense of problems and persevere in solving them. SMP 2: Reason abstractly and quantitatively.	**Central Focus:** Programmers use precise language and sequential directions to code.	**Learning Targets:** Using software that allows students to create their own code, students will create a series of code that uses

SMP 6: Attend to Precision.		precise language and sequential directions in order to accomplish a task.

How have you addressed the needs of diverse learners ? (Ex: IEPs, 504s, linguistic & cultural diversity, students without prerequisite knowledge, etc.)

- Spheros allow students to code in a hands-on way. This is good for students who learn through physically doing something.

Materials/Equipment/Supplies/Technology/Preparation:

- Spheros: a class set can be purchased on Sphero.com
  - Many districts have access to Spheros or similar technology.
- iPads
- Large paper to make maze
- Writing utensil

Procedure: Teacher Does……….		**Procedure: Students Do……..**
Time **3 min**	**Motivation/Hook:** Teacher refers to the objective for the day to help students get ready to learn. Teacher explains that today the students will be making mazes and coding a robot to navigate through the maze. Teacher will show students the robot and give a quick demonstration of how to move the robot. This is just to get the students interested.	**Motivation/Hook:** Students will listen to the task for the day and watch the demonstration.
7 min	**Teaching:**	**Teaching:**

	The teacher will ask students to move closer to where they are standing. **Students can come to a carpet if there is one available or just stand in a circle.** The teacher will use a piece of butcher paper about 2 ft by 2 ft to draw a maze with the students. **Make it clear that the maze should have between 4 and 7 turns in it. The more turns that are present, the harder the maze is to program. ** The teacher will demonstrate how to program the robot to navigate the maze, making sure to call student attention to the important functions of the software. **Such as how to move forward, turn, stop, etc.** The teacher will program their robot through the maze in front of the students. **You might not be able to program it all, but at least make sure students see each important function.** Run the program so students can see what they will be doing. Set up classroom expectations about how to use the Spheros. These should be things like treating them	Students will watch teacher example and ask questions as they have them.

	respectfully and using them for the task at hand.	
30 min	**Group Application/Independent Application:** Depending on the make-up of the class and technology available, group students into pairs or have students work individually. Pass out paper and writing utensils. Ask students to draw their maze, with between 4 and 7 turns. Once students have drawn their maze and you approve it, give them a Sphero and iPad. Instruct them to begin programming their Sphero. Remind them of the expectations for using the Spheros.	**Group Application/Independent Application:** Students draw their maze on the paper provided, with between 4 and 7 turns. Students program their Spheros, asking for support as needed.
10 min	**Closure:** This closure can happen right away or at another time. Students will take turns all presenting their program to the class.	**Closure:** Students present their program to the class and watch other students present as well.

	The teacher will watch to make sure that students have programmed the robot to move through the maze. The robot might not make it perfectly through the maze but it should stay within the maze for the most part. The teacher will look over the code that students created in order to assess their use of precise language and how they attended to precision.	

Lesson Title/Description: LESSON 3B: Scratch		
Standards (content and SMP): SMP 1: Make sense of problems and persevere in solving them. SMP 2: Reason abstractly and quantitatively. SMP 6: Attend to Precision.	**Central Focus:** Programmers use precise language and sequential directions to code.	**Learning Targets:** Using software that allows students to create their own code, students will create a series of code that uses precise language and sequential directions in order to advance a robot through a maze.
How have you addressed the needs of diverse learners ? (Ex: IEPs, 504s, linguistic & cultural diversity, students without prerequisite knowledge, etc.)		

Materials/Equipment/Supplies/Technology/Preparation: ● Technology to access Scratch.mit.edu		
Procedure: Teacher Does……….		**Procedure: Students Do……..**
Time **3 min**	**Motivation/Hook:** Teacher will refer to the objective of the day. Teacher explains that they will be using Scratch to create their own code. Teacher will show students the tutorial video that is available on Scratch's website in order to help them understand the software a little bit more.	**Motivation/Hook:** Students will listen to the task and watch the videos.
7 min	**Teaching:** Teacher projects Scratch onto the board so that students can see what the teacher does on their computer. Teacher shows students how to use the paint function on backgrounds to create a maze. Students should create a maze that has between 4 and 7 turns. The teacher will create their own maze and then demonstrate how	**Teaching:** Students will watch the teacher's example and ask questions as they have them.

	they program their sprite to navigate the maze. Teacher passes out technology and helps students log into scratch.mit.edu to create their own project.	
30 min	**Group Application/Independent Application:** Teacher asks students to create their maze in Scratch in partners or by themselves, depending on what the teacher decides. Ask students to program their sprite to go through the maze, using between 4 and 7 turns. Students can play around with the other functions of the software if they would like.	**Group Application/Independent Application:** Students create their maze and program their sprite. They can ask the teacher for support if needed.
10 min	**Closure:** This closure can happen right away or at another time. Students will take turns all presenting their program to the class.	**Closure:** Students present their program to the class and watch other students present as well.

	The teacher will watch to make sure that students have programmed the robot to move through the maze. The robot might not make it perfectly through the maze but it should stay within the maze for the most part. The teacher will look over the code that students created in order to assess their use of precise language and how they attended to precision. The teacher can ask students questions such as: ● Where did you use sequential directions in your code? ● How did you attend to precision?	

Rubric for Lesson 3A/B Summative Assessment:

For this assessment, the teacher will make use of the product created during lesson three. This product serves as a summative assessment for the entire unit. They will watch the robot or sprite move through the code as well as look at the code in order to assess using the following rubric:

Criteria	Meets	Does not meet
Maze includes between 4-6 turns and the robot		

or sprite makes it through them with few (1-2) deviances from the maze.		
Code makes use of precise language. Ex: includes a number of seconds or a number of steps the robot/sprite should move, or includes a stop function.		
Code uses clear sequential directions in order to advance the robot or sprite through the maze.		

Discussion

Analysis of Benefits and Drawbacks

Incorporating a coding curriculum in the classroom is very beneficial to students. This type of curriculum can enhance their quality of education as the skills learned in this area can transfer to other parts of education. These are also skills that will help prepare students as they enter the career field and begin jobs that we have not yet begun to imagine.

While there are benefits to a coding education being included in a public education classroom, I recognize that there may also be some drawbacks. Namely, the time it takes to roll out this type of curriculum is more than is often available in the busy days of a public school. With the many standards and goals that teachers are attempting to meet, it can feel tedious to include something that is not directly drawn from standards. I would argue, however, that the research previously discussed about the value of coding and how it relates to other subject areas, makes it worth including in the classroom. Furthermore, when done thoughtfully we can meet content standards through coding education. If we are able to teach both the required standards and these important skills at the same time, coding education would be a valuable addition to our curriculum.

Another drawback that can be somewhat difficult to overcome is funding. Coding education, by its nature, requires some form of technology. In lower income districts and schools like the one I am currently student teaching in, gaining access to this technology can be difficult. While this project includes one offline coding activity, the majority of them require at least access to the internet through some type of computer. In Oregon, however, internet access and computer technology is available in every district state-wide. This makes this unit possible without requiring extra materials or resources. Spheros, which I believe are a very useful tool in teaching coding education, are expensive and can be seen as frivolous when there are other expenses that money should go toward. For this reason, I included an alternate activity, which still requires a computer, but is more accessible to more people.

Overall, while there are both benefits and drawbacks to including a coding education in the classroom, I believe that the benefits outweigh the drawbacks. Coding education teaches students skills such as sequential thinking, computational thinking and abstract thinking. These skills are beneficial in mathematical practices, literature, science, as well as areas of everyday life. A well constructed coding education that ties back to the CCSS-M and still teaches these skills is a valuable addition to the classroom.

Limitations

One of the obvious limitations of this project is the grade level. In order to create this project, I focused specifically on crafting lessons that could be used in

5th grade, which is at the top of most elementary schools. I did this because I am hoping to teach upper-elementary myself, and thus am more familiar with the things that older primary aged students are capable of, and what is developmentally appropriate. With the limited scope of the lessons I created, there would need to be some adjustments if used in much older or younger classrooms, or even with 5th grade students with different levels of computer proficiency. This would be something that the teacher could easily adapt based on their own knowledge of what their students need. One benefit of creating this unit at a 5th grade level is that it can easily be adjusted to meet the needs of younger grades in Elementary Schools or older students in Middle Schools. It is in the middle, which makes adjustments much easier in both directions.

My Own Personal Plans for Using it in my Classroom

While I do not currently have a classroom of my own where I have the ability to change the curriculum that is taught, I do plan to use this in my own future classroom. During the 2020-2021 school year I will be teaching in a 5th grade classroom in a Dual Language Program. I look forward to adapting these lessons to fit the needs of my specific students. I have created lesson plans that are open to adaptation purposefully to allow for the changes that may be necessary.

I can see incorporating this type of coding curriculum as a once or twice a week, 30 minute learning block. During this time, we will engage with technology in different ways in order to learn the coding skills that will be useful to students later in their lives. As I mature as a teacher, I am sure I will continue adapting and changing the coding education that I have planned thus far. I am aware that in my current position as a Teacher Candidate I am learning a lot about pedagogy and developing my own beliefs. These will continue changing and I am excited to see how coding education fits into these beliefs in future years.

Accessing Materials and Why I Chose Certain Platforms

Most of the platforms that I have chosen to include in these lessons are available for free online to teachers and students. I purposefully included these kinds of resources because I wanted to limit the barriers to this type of education that might present themselves to lower income students. Coding education can cost a lot of money upfront, as a district or school would need to purchase technology which can be expensive. Because of this, coding education has the potential to favor higher-income students and districts, which would increase the privilege gap. Choosing free platforms helps to increase accessibility to coding education, which will be very useful in our students' futures.

The platforms that I have chosen to incorporate include Code.org, Scratch.mit.edu and Spheros. The latter platform is the only one that costs additional money, and I have created an alternative assignment on Scratch to help mitigate this inaccessibility. Code.org is a great platform that allows students to begin coding through programming video games that they are more than likely already familiar with. Some of these include Angry Birds and Minecraft. This source teaches students how to code gradually, which is very important in education. Scratch.mit.edu is a bit more open-ended of a resource. Students have the ability to create their own projects or to complete already organized projects. This source mirrors the immediate feedback that Spheros provide, which makes it a good free alternative. Spheros are small robotic balls that can be programmed using an app. I like this software because students get instant feedback regarding the effectiveness of their code. If, for example, students are attempting to make the robot go forward but they run the code and it goes backward, they instantly know they need to go back and reformat their code.

These resources are available free (except for Spheros) on the internet at the links provided. Many districts have access to a set of Spheros, so it might be possible to use a set if requested in advance. On all of the platforms listed, teachers can receive a tutorial of how the software works through accessing the website.

References

Chaves, Hamilton Viana, Maia Filho, Osterne Nonato, & Melo, Armando Sérgio Emerenciano de. (2016). EDUCATION IN TIMES NET GENERATION: HOW DIGITAL IMMIGRANTS CAN TEACH DIGITAL NATIVES. *Holos., 32*(2), 347-356.

Erikson, E.H. (1959). Identity and the life cycle: selected papers. International Universities Press.

Firth, N. (2014). Code generation. *New Scientist*, *223*(2985), 38–41. https://doi.org/10.1016/S0262-4079(14)61724-3

Garov, K. A., & Tabakova-Komsalova, V. V. (2017). Learning Content of Educational Tasks in Computer Programming Training for 10-11 Year Old Children. *TEM Journal*, *6*(4), 847–854. https://doi.org/10.18421/TEM64-26

Google Inc. & Gallup Inc. (2016.) Trends in the State of Computer Science in U.S. K-12 Schools. Retrieved from www.goo.gl/j291E0

Hour of Code. (2018). Code.org. Retrieved from https://hourofcode.com

Kazakoff, E., Sullivan, A., & Bers, M. (2013). The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. *Early Childhood Education Journal*, *41*(4), 245–255. https://doi.org/10.1007/s10643-012-0554-5

Manches, A., & Plowman, L. (2017). Computing education in children's early years: A call for debate. *British Journal of Educational Technology*, *48*(1), 191–201. https://doi.org/10.1111/bjet.12355

Microsoft Philanthropies. (2018). Computer Science Professional Development Guide. National Center for Women & Informational Technologies. (2012). Girls in IT: The Facts.

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

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

Papert, S. (1993). *The children's machine : Rethinking school in the age of the computer*. New York: BasicBooks.

Pearson, P. D. and M. C. Gallagher, "The Instruction of Reading Comprehension," *Contemporary Educational Psychology*, 8, 1983, pp. 317-344.

Pierce, M. (2013). Coding for Middle Schoolers. *T H E Journal*, *40*(5), 20–23. Retrieved from http://ezproxy.wou.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=94425381&site=ehost-live

Scratch. (2003). Lifelong Kindergarten Group at the MIT Media Lab. Retrieved from https://scratch.mit.edu.

Sphero. (2018). Sphero Inc [Mobile Application Software]. Retrieved from https://edu.sphero.com

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms. *TechTrends: Linking Research & Practice to Improve Learning*, *60*(6), 565–568. https://ezproxy.wou.edu:4285/10.1007/s11528-016-0087-7