East Asian Math. J.
Vol. 36 (2020), No. 2, pp. 317-330
http://dx.doi.org/10.7858/eamj.2020.020

Incorporating Coding on Student Experience: Lessons Learned from an Action Research

Meghan Schultz · Jihwa Noh[1]

ABSTRACT. The purpose of this action research project was to study the effects of incorporating coding into the middle school math classroom affected student dispositions with math and their understanding of mathematical concepts. The project, involving a total of 107 US middle school students, used five data sources to examine these effects: a survey, a chart measuring student engagement, a pre- and post-assessment before and after the coding project, and teacher observation with reflection forms. After analyzing the data, it was found that incorporating coding into the middle school math classroom could have a positive impact on student math dispositions and their understanding of math concepts.

I. BACKGROUND

Computer Science opportunities in K-12 schools have increased significantly in recent years (Humble, 2018; Moreno-Leon, Robles, & Roman-Gonzalez, 2016). There are many advantages to teaching computer science to children, aside from the high demand for workers in computer programming related jobs and the 21st Century being heavily influenced by computing (Batista & Baptista, 2014; Sáez-López, Román-González, & Vázquez-Cano, 2016; Songy, 2017). Some of these skills include problem-solving, critical thinking, computational thinking, creativity, and

Received February 13, 2020; Accepted February 23, 2020.
2010 Mathematics Subject Classification: 97D40, 97M10
Key Words: coding, engagement, mathematics, Scratch, programming
1)Corresponding author

©2020 The Youngnam Mathematical Society
(pISSN , eISSN)

collaboration

Currently, computer science, and specifically, computer programming are offered as elective courses in many schools in the United States (Guzdial, 2016); however, it is recommended to give all students programming (coding) experiences within their content area classes (Batista, 2014; Calder, 2010; Sáez-López et al., 2016).

Although many studies report the positive impact coding has on student 21st Century skills, using coding as a vehicle to intentionally target and solidify math skills has yet to be determined. Without this kind of information, it is difficult for school districts to carve out time for coding during math class. If research showed that using coding to teach and learn math allows students a deeper understanding of the mathematical concepts, then schools will be more willing to adopt the idea of implementing coding experiences in the math curriculum. Therefore, there is a need to gather and present information about whether or not using coding in the math classroom has any effect on student content learning and this action research project was designed to address that need. Specifically, the research question posed is: In what ways does incorporating coding into middle school math classrooms affect students' dispositions with math and their understanding of mathematical concepts?

II. LITERATURE REVIEW

The following review of literature will focus on how incorporating coding in the school math classroom affects student dispositions (motivation, engagement), increases content knowledge, and nurtures learners' 21st Century skills. Lastly, there will be a brief review of action research, which is the research design employed by this project.

1. Motivation and Student Engagement

Students have positive attitudes and high motivation when visual coding languages are integrated into a core content area classroom (Sáez-López et al., 2016). When coding was initially taught in K-12 classrooms, Resnick et al. (2009) explained that three things made the implementation difficult; syntax errors with text-based coding, introducing coding through activities that were not intriguing to students, and the fact that coding was not taught in union with core classes (Batista & Baptista, 2014). Due to these problems, it has been suggested that visual programming language be used in schools (Sáez-López et al., 2016). Visual coding languages consist of preset blocks that can be dragged together based upon their shapes and desired outcomes. This block-based structure saves students a lot of frustration because they are not discouraged by syntax errors. The blocks are often organized or labeled by color and can be connected to other blocks similar to the way puzzle pieces connect. This can make learning a visual programming language intriguing to students since they are able to tinker with the blocks, just as they would do with physical building blocks, which allows creative expression to happen faster.

The Massachusetts Institute of Technology (MIT) Media Lab research group Life Long Kindergarten lead by Mitch Resnik, aimed to create a visual programming language, Scratch, with which students could learn coding through tinkering within a meaningful social setting (Batista & Baptista, 2014). Scratch allows students to drag and drop blocks and to create coding projects that interest them. Through trial and error, students are immediately able to see if their program works or not (Calder, 2010). Scratch allows for student choice and freedom as students are able to create many different types of projects. Students have the power to choose how they express their interests through coding (Resnick et al., 2009). Since Scratch is an engaging and intuitive programming language for students, it has proved to be an excellent platform for problem-solving through mathematical concepts (Kalelioglu & Gulbahar, 2014). Aspinall explained that Papert viewed math as a place to explore, make mistakes, and self-correct these mistakes (Aspinall, 2017). Coding allows students to do math in a way where they can see the connection to their own lives. Students expressed that this type of integration in the classroom motivates them to learn more as they enjoy the active approach it brings to learning.

2. Increased Content Knowledge

Little is known about the benefits of incorporating computer programming into core content area classrooms, specifically math, to enhance content understanding. Studies addressing this are mostly suggesting ways to focus on intended math outcomes when teaching coding. One way is by having students create a program that performs a certain algorithm. Calder's (2010) study had students create a program that generated simple addition equations. The children playing the game needed to match the solution to the equation to the corresponding number of aliens. In Wherf, Pearson, Shebab, and Tapia's (2015) study, students created animations for different polygons when learning geometry. Another way teachers can integrate coding into the math classroom is to allow students to play programs that are already created and developed. For example, Scratch has academic activities created by members of the Scratch Community like Triangle Classification, the Circle Theorem, and Prime Factorization (Batista & Baptista, 2014). When creating a Scratch project, students use measurements of geometry, length, and time, coordinates, angles, relational thinking with input, repeated procedures, and algorithmic problem solving. Since students can physically experiment with math, (i.e., angle size), they form a deeper understanding that would not be possible without using computing.

3. 21st Century Learning Skills

When students use coding in the math classroom they are learning essential skills for the 21st Century, such as how to problem solve, think computationally, and collaborate effectively (Resnick et al., 2009). Students are motivated to develop problems that interest them when they learn by doing. Visual programming languages encourage students to find solutions to their projects. (Kalelioglu & Gulbahar, 2014). When students are working to solve the problems they have chosen, they are mastering learning targets while solving real issues (Johnson et al., 2014). Students must use critical thinking and problem-solving skills to create a program that works since computer programs tell a computer exactly what to do (Sáez-López et al., 2016). Therefore, this process enhances problem-solving skills (Kalelioglu & Gulbahar, 2014).

Computational thinking is a fundamental set of skills that everyone must know to function well in society (Aspinall, 2017). Although computational thinking is generally connected to computer scientists, it is used by all people and in many disciplines (Yadav, Stephenson, & Hong, 2017). These ideas have had a lot of influence in scientific fields but also the artistic and humanities fields as well. Yadav et al. (2017) argued that computational thinking should be a requirement for all K-12 learners.

Collaboration is another 21st Century skill that is nurtured by implementing coding in core content area classrooms. Since computing focuses on creativity, choice, and finding solutions to problems, it is collaborative by nature (Israel et al., 2015). Coding is essentially another language, learning to code gives students another way to communicate their thoughts and ideas to others.

Although implementing coding experiences into content area classrooms is usually viewed as a positive experience, there has been opposition. Graves (1990) argued that math teachers are assigned to teach computer science or computer programming courses and are not informed enough to explain it at the level needed for college readiness. Often this occurs because there are not enough teacher training opportunities for these teachers to be informed about what students currently need to be successful (Graves, 1990). However, teacher professional development programs are continuing to be improved to include experiences around teaching computer science and computational thinking, unpacking standards, and using technology (Yadav et al., 2017).

4. Action Research

Since its inception in the work of Kurt Lewin in 1946, the meaning and purpose of action research has taken on many forms. In essence, action research is a form of problem solving: a problem is identified to work on, with the aim to improve or to solve it (Elliot, 1991). The researcher gathers information on the problem and tries out new procedures or makes some other change in practices to see if they result in a solution. This form of research has a history that is rooted in problem solving in social and organizational settings. An advantage of action research is that the project and how elaborate it is depends on who is conducting it and what they are doing. It is a customized methodology that can be kept quite simple or become very elaborate depending on those involved and how they want or need to proceed.

In educational settings there are a variety of action research methods available to use, depending on the group and the goals that are identified. When it involves a single teacher investigating an issue in his or her classroom, the teacher-researcher is inclined to learn more and be more willing to apply what he or she learns, when they do something themselves. So action research has a good chance of changing teacher practice. Like behavior modification, practice helps pave the way to a new method or assists in enabling the change that is desired (Reason & Bradbury, 2001).

III. METHODOLOGY

1. Theoretical Framework

The theory of constructionism guided this action research study. The theory of constructionism is based on building knowledge structures through student actions. This happens most frequently when students are constructing objects or ideas meaningful to them (Papert, 1980). This model explains how ideas get transformed when students are given the freedom to express their work by making their favorite representations. Futhermore, this theory suggests that through the process of self-correction (in coding or math), students develop new ways to solve problems and persevere through uncertainty and by letting students create artifacts around a concept, such as a computer program, they are able to develop and sharpen their ideas (Aspinall, 2017).

In this study, coding was integrated into a math classroom. The effect this integration had on student understanding of a math concept and their overall math experiences were examined. Based on the theory of constructionism, students might have a deeper understanding of the math concept after participating in the coding experience. Additionally, their math experience might be enhanced after creating computational artifacts.

2. Research Procedures

In this study, students created a coding project that calculated distance and midpoint depending on input from a user. For example, if the user typed in the points (1,2) and (−3,5), the program would take those points and calculate the midpoint and distance between them. To answer the research questions, this study utilized a mixed-methods design. The following instruments were administered; a pre-assessment that asked questions related to distance and midpoint, a post-assessment that asked the same questions from the pre-assessment, a rubric used for scoring the projects, and a chart to track time on task. The qualitative survey asked a series of questions and students responded using a likert scale. The qualitative data was a student survey describing student attitudes towards using coding in the math classroom.

The population for this action research study was seventh grade students enrolled in a mid-sized middle school in the Midwestern US (n=107). These seventh grade students were enrolled in Intermediate Algebra for both first and second semesters of the school year. Intermediate Algebra was a required high school math course offered at the middle school level. The sample features 54 females and 53 males and was representative of two-year accelerated math students in the middle school population.

The quantitative data collected before students created their coding project was a pre-assessment that was comprised of three questions on distance and midpoint, each worth one point. At the conclusion of the coding project, the same three questions were given to students as a post-assessment. When the students turned in their coding projects the teacher-researcher graded the projects based on a rubric that assessed proficiency towards key learner outcomes accuracy, application, coding efficiency, presentation, and creativity, and on task time in their project. Each of the learner outcomes was scored 0-3 points, for a total of 18 points. Additionally, while students worked on their projects, the teacher- researcher tracked the time on task of three students, chosen at random. The researchers observed the three students after a series of 15 minutes. At the end of the each 15 minutes the researchers recorded a plus sign if the student was on task and a minus sign if the student was not working. The plus and minus signs were recorded in a chart where each student had a column broken into 15 minute intervals. The qualitative data measure collected was a survey the students filled out on Google Forms. The survey was five questions and asked students to reflect on their project experience. This survey was given to gauge student attitudes towards using coding in the math classroom.

3. Analysis of Data

The student received a point for each of the three questions in both the pre-assessment and post-assessment only if the question was completely correct. The teacher-researcher recorded the scores from the pre-assessment and post-assessment in a spreadsheet. Then, the researcher compared the pre-assessment with post-assessment scores and identified if there was an increase in score, no change, or a decrease in score. Next, both of the researchers used at rubric to score the student Scratch projects. The researchers recorded the score of each student in a spreadsheet and created a graph that identified the breakdown of the scores. In order to analyze the data from the time-on-task chart, the teacher-researcher totaled the number of "+" signs within each student column and calculated a percentage that described what percentage of time the student was on task.

To analyze the qualitative data measure, the student survey, the researcher totaled the number of students who strongly agreed, agreed, were neutral, disagreed, or strongly disagreed with the five survey statements. The totals were then converted into percentages and put into a graph and were used to analyze themes related to student perceptions of the implementation of a coding project in the math classroom.

IV. FINDINGS

1. Student Math Dispositions

The first research question this study addressed was related to identifying prominent feelings and perceptions students had towards using coding in the math classroom. There were 101 students who tokk the survey. Student responses to the questions in the dispositions survey are presented in Table 1. Based on the survey findings it can be concluded that the majority of students enjoyed using Scratch in the math classroom to demonstrate their understanding of the distance and midpoint formula. In the same survey, students were asked if they would like to use Scratch again during the school year. Based on the findings it can be concluded that most students would like to use Scratch again during the school year. Additionally, students were asked if the topic of distance and midpoint became more meaningful to them after creating their coding project. Based on the findings it can be concluded that most students felt the concepts of distance and/or midpoint either became more meaningful or remained the same to them. Another question students were asked within this survey was whether they collaborated during the creation of their project. A majority of students

Survey Questions \ Student Responses	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I enjoy using Scratch in Math	38 (37.6%)	34 (33.7%)	23 (22.8%)	4 (4.0%)	2 (2.0%)
I would like to use Scratch again in another unit to demonstrate my understanding of a topic	45 (44.6%)	23 (22.8%)	26 (25.7%)	5 (5.0%)	2 (2.0%)
The topic became more meaningful after creating this project	17 (16.8%)	43 (42.6%)	34 (33.7%)	6 (6.9%)	0 (0.0%)
	Yes			No	
Were you able to collaborate with others during this project?	80 (79.2%)			21 (20.8%)	

<Table 1> Math Dispositions Survey Responses

collaborated during the project. Some students chose to work alone and others chose to listen to music while creating their project.

Another measure of student experience came from the time-on-task chart. The three students who were randomly selected for the study showed a high percentage of on task time when working on their projects. The on task time chart showed that two of the three selected students were on task for the entire 60 minutes of work time. The third student struggled to get started, but once the researcher helped the student start their program they were on task the rest of the project work time. This led to the student being on task 75% of the project work time.

2. Student Understanding of Mathematical Concepts

The second research question this study addressed was whether using coding in the math classroom affected student understanding of specific math concepts. There were 107 students who took the pre and post-assessment on midpoint and distance. Findings from analysis of these data points are explained in Table 2.

When the results from the pre to the post-assessment were analyzed, 39% of students grew in their understanding of midpoint and distance, as their score from the pre to the post-assessment increased. Of the 107 students, very few students' scores decreased when comparing their scores from their pre to their post-assessment (13%). On the initial pre-assessment of midpoint and distance, 40 students of the 107 showed mastery in their understanding of midpoint and distance as they scored 3/3 on the pre-assessment. Therefore, the teacher-researcher analyzed the scores of the students who did not score 100% on the pre-assessment (n=67). The findings demonstrates that of these students who did not score 100% on their pre-assessment, 62.7% increased in score from their pre to their post-assessment. A small number of students received the same score or decreased in score from the pre to the post-assessment (37.3%).

All Student Growth (n=107)			Low-End Student Growth (n=67)		
Growth	Remained Same	Decreased	Growth	Remained Same	Decreased
42(39.3%)	51(47.7%)	14(13.1%)	42(62.7%)	11(16.4%)	14(20.9%)

<Table 2> Pre to Post Assessment Growth

Additionally, the teacher-researcher used a project rubric to assess the coding projects and identify student mastery over key math and coding topics. Almost all students (95.3%) showed mastery over the assessed criteria (scored 18/18 on their coding project). One student scored a 16/18 on the project as his program did not run correctly. The five students who scored a zero did not turn it in their project.

Through triangulating the different data sources many themes were found. Overall, students enjoyed their experiences using code in the math classroom. Many shared that the math concepts of distance and midpoint became more meaningful to them when coding was integrated into their learning experience. Furthermore, the coding project allowed students to either increase in their understanding of distance and midpoint or remain the same after completing a coding project. Aside from the main math outcomes of understanding distance and midpoint, students were also able to demonstrate mastery over key learner outcomes such as, accuracy, application, coding efficiency, presentation, and creativity. Incorporating coding could be a way to keep students engaged in the math classroom, as students were on task the majority of the project work time.

V. NEXT STEPS

The purpose of this study was to understand ways in which incorporating coding into the middle school math classroom affected student experiences with math and their understanding of mathematical concepts. This paper examines how incorporating coding into middle school math classrooms affect student experiences with math and their understanding of mathematical concepts. The research question posed was: In what ways does incorporating coding into middle school math classrooms affect students' experiences with math and their understanding of mathematical concepts? Given the analysis of the data in this study, several conclusions can be drawn supporting the use of a visual programming language in the math classroom.

This study showed the incorporation of Scratch Programming in the math classroom led to math becoming more meaningful to students. Overall, the students enjoyed the experience and shared they would like to use Scratch again to demonstrate their understanding of math concepts. The use of code within their math classroom gave more meaning to their unit of study. Another conclusion that can be drawn is incorporating code into the math classroom can foster important 21st Century skills, such as collaboration and problem solving. Another conclusion that can be drawn is that incorporating code into the math classroom leads to high engagement. In the on-task-time survey, the students selected had high percentages of on-task-time.

When using code in the math classroom as a vehicle to help teach a specific math skill, student understanding of the math concept improves. In this study, the scores from student pre to post assessments either grew or remained the same. This demonstrated that overall, student understanding of midpoint and distance grew from completing a coding project within the math classroom. Very few students decreased in their score from the pre to the post-assessment.

Furthermore, by incorporating code in the math classroom students will get practice around 21^{st} Century learner outcomes such as accuracy, application, coding efficiency, and creativity. Therefore, allowing students to use coding in the classroom to demonstrate their current knowledge of a skill or continue to grow in their understanding of a math concept leads to academic growth.

Based on this study, the teacher-researcher plans to incorporate code into the math classroom multiple times each school year. The findings of this study may suggest that all math teachers aim to implement code into their math classroom to improve student experience and understanding of mathematical concepts. As long as the implementation of the coding project is aligned and centered around a key math concept, teachers are fostering crucial 21st Century Skills in students, all while improving understanding of key math concepts.

References

Aspinall, B. (2017). *Code breaker: increase creativity, remix assessment, and develop a class of coder ninjas!* San Diego, CA: Dave Burgess Consulting, Inc.

Batista, S., & Baptista, C. (2014). Learning object for linear systems: Scratch in mathematics. *International Journal on New Trends in Education and Their Implications, 5*(1), 71-81.

Calder, N. (2010). Using Scratch: An integrated problem solving approach to mathematical thinking. *Australian Primary Mathematics Classroom, 15*(4), 9-14.

Elliot, J. (1991). *Action research for educational change.* Milton Keynes and Philadelphia: Open University Press.

Graves, D. (1990). Computer science and math: Estranged partners in a changing world. *Education, 110*(4), 486-489.

Guzdial, M. (2016). Bringing computer science to U.S. schools, state by state. *Communications of the ACM, 59*(5), 24-25.

Humble, S. (2018). *Creating the coding generation in primary schools: A practical guide for cross-curricular teaching.* New York, NY: Routledge.

Israel, M., Wherfel, Q., Pearson, J., Shehab, S., & Tapia, T. (2015). Empowering K-12 students with disabilities to learn computational thinking and computer programming. *Teaching Exceptional Children, 48*(1), 45-53.

Johnson, L., Becker, A., Estrada, S., & Freeman, A. (2014). NMC Horizon Report: 2014 K-12 edition. *Austin, Texas: The New Media Consortium.* Retrieved from http://www.nmc.org/pdf/2014-nmc-horizon-repo rt-he-EN.pdf

Kalelioglu, F., & Gulbahar, Y. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education, 13*(1), 33-50.

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

Lewin, K. (1946). Action research and minority problems. *Journal of Social Issues, 2*, 34-46.

Lye, S., & Koh, J. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51-61.

Moreno-Leon, J., Robles, G., & Roman-Gonzalez, M. (2016). Code to learn: Where does it belong in the K-12 curriculum?. *Journal of Information Technology Education: Research, 15*, 283-303.

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

Reason, P., & Bradbury, H. (2001) *Handbook of action research*. London: Sage.

Resnick, M., M., Monroy-Hernández, J., Rusk, A., Eastmond, N., Brennan, E., & Silverman, B. (2009). Scratch: programming for all. *Communications of the ACM, 52*(11), 60-67.

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

Songy, S. (2018). How a Tough Challenge Taught My Students More about Coding Than I Ever Could. *The Education Digest, 83*(8), 47-50.

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM, 60*(4), 55-62.

Meghan Schultz
Jesup Middle Schools
Jesup, Iowa 50648 USA
E-mail address: Mjschu84@uni.edu

Noh, Jihwa
Department of Mathematics Education
Pusan National University
Pusan, 46241 Korea
E-mail address: nohjihwa@gmail.com