Learning Computer Programming in Context: Developing STEM-integrated Robotics Lesson Module for 5th Grade

Ikseon Choi
The University of Georgia
U.S.A.
ichoi@uga.edu

Theodore J. Kopcha
The University of Georgia
U.S.A.
tjkopcha@uga.edu

John Mativo
The University of Georgia
U.S.A.
jmativo@uga.edu

Roger Hill
The University of Georgia
U.S.A.
rbhill@uga.edu

Emily Hodge
The University of Georgia
U.S.A.
emhodge@uga.edu

Seungki Shin
The University of Georgia
U.S.A.
shin@uga.edu

Brian Way
The University of Georgia
U.S.A.
bway7@uga.edu

Jen McGregor
The University of Georgia
U.S.A.
jam7539@uga.edu

Sejin Kim
The University of Georgia
U.S.A.
sejink@uga.edu

Jasmine Choi
The University of Georgia
U.S.A.
jchoi816@uga.edu

Youngkwon Bae
Daegu National Univ. of Education
Republic of Korea
bae@dnue.ac.kr

Abstract: This paper introduces a novel STEM-integrated robotics curriculum that aligns with relevant curriculum standards in science and mathematics. Using a scenario-based learning environment, the curriculum engages students in applying the engineering design process, scientific inquiry, mathematical thinking, and computational thinking through robotics. Thus, computer programming is taught in a meaningful context where students must initialize and adapt a program to solve a given problem. This paper presents a theoretical framework for the STEM-integrated robotics curriculum, a process of developing the curriculum, and learning activities organized to promote intended thinking skills.

Introduction

The importance of exposing computer programming education to learners as early as possible is greater than it has ever been. Elementary school students tend to express their ideas directly and refine them with concrete outcomes when finding a solution to problems. Computer programming can supplement this direct expression of students' ideas and help them further refine concrete outcomes (An & Park, 2012; Flannery et al., 2013). Despite the importance of this early exposure, computer programming education has faced a lack of age-appropriate engaging curriculum (Esteves, Fonseca, Morgado, & Martins, 2011). Alternatively, utilizing storytelling, a method for using realistic stories for education, is a potential solution because it provides students with authentic contexts in the form of real-world problems for computer programming education (Carder, Willingham, & Bibb, 2001; Cooper, Dann, & Pausch, 2003; Kelleher, Pausch, & Kiesler, 2007).

In this project, we aim to do research on the development of a STEM-integrated (Science, Technology, Engineering, and Mathematics) Robotics Curriculum where 5th graders can meaningfully learn computer programming within an authentic context. We propose an innovative method of teaching and learning in order to engage students in advanced thinking skills, such as engineering design thinking, scientific inquiry, mathematical thinking, and computational thinking, while utilizing computer programming to accomplish a given mission.

A Framework for STEM-Integrated Robotics Curriculum

We developed a framework to guide the development of a STEM-integrated robotics curriculum. As shown in Figure 1, we incorporated the engineering design process as an overarching problem solving process. The

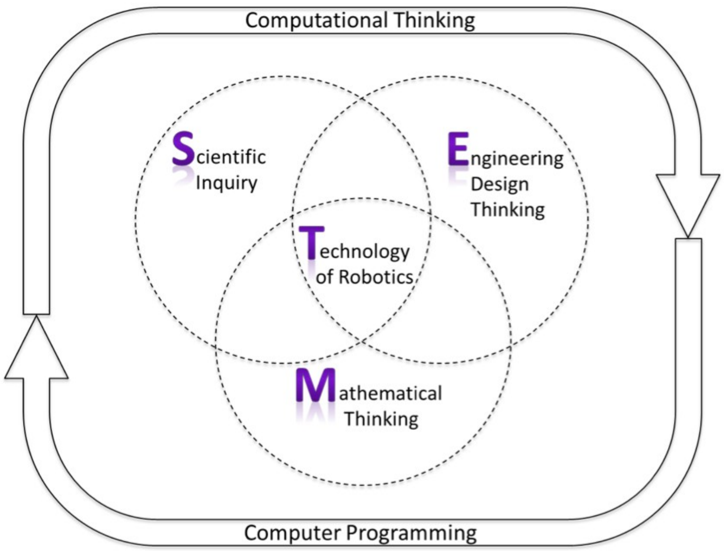

Figure 1: A Framework for STEM-integrated Robotics Curriculum

engineering design process is an iterative decision-making and problem solving process that can host strategies to promote students' creative thinking, solving of open-ended challenges, and computational thinking (Eide, Jenison, Mashaw, & Northup, 2002). Fifth grade science content was used as a context for the problem in which authentic learning would occur. Both the content and robotics tasks were supported with a scientific inquiry process that relied on mathematical computation and reasoning. In addition, we configured the framework to be a series of student learning processes that incorporate computer programming. In this way, the framework leverages further curriculum development where necessary STEM thinking – that is, engineering design thinking, science inquiry, mathematical thinking, and computational thinking – is seamlessly integrated during problem solving activities.

STEM-Integrated Curriculum Development Process

Our curriculum development followed the ADDIE (Analysis, Design, Development, Implementation, and Evaluation) process (Branch, 2009). In the Analysis phase, we analyzed the mathematics and science curriculum based on the Common Core Georgia Performance Standards (CCGPS) while developing our pedagogical framework from the literature. Then, we selected appropriate subject content to be incorporated into our curriculum. In the selection process, we considered the expected schedule of the math and science curriculum, so our curriculum could be delivered when students had appropriate prerequisites. In the Design phase, we composed student-centered, creative learning activities and lesson plans, taking into consideration the students' cognitive development stage based on the selected content. In the Development phase, we developed a student workbook, referred to as the *student design journal*, which combined and organized a series of learning activities. In the journal, students were prompted to reflect on their thinking at the end of every activity. In the Implementation phase, we delivered a teacher training workshop for twelve potential teachers who were planning to implement our curriculum, during which we received their feedback and suggestions. The iterative process of receiving feedback and revising the curriculum contributed to reducing the gap between theory and practice. The curriculum was then implemented into eight 5th grade classrooms across eight different schools. We are currently in the phase of Evaluation where we are analyzing students' learning outcomes and their experiences for further revision and recommendation.

Curriculum Analysis for 5th Grade

Science Curriculum. Based on the analysis of the current Georgia standards, CCGPS, for 5th grade science (See Table 1), we arranged a relevant science context based on spring semester curriculum, during which the pilot implementation was planned. This included the standard: Students will identify surface features of the Earth caused by constructive and destructive processes. The specific domains and clusters are indicated in Table 1 and are categorized as Earth Science (Georgia Department of Education, 2014a). With this in mind, we developed the context of our STEM curriculum around the real-world example of volcanoes.

Mathematics Curriculum. The domains and clusters of CCGPS 5th grade mathematics standards are shown in the following Table 2 (Georgia Department of Education, 2014b). Although we incorporated each domain in our instructional design, a specific focus was placed on (1) increasing understanding of and fluency with decimals and (2) graphing points on a coordinate plane to solve real world problems. The reason for this focus is that we expected a higher utilization of decimals and coordinate geometry when programming the robots to perform precise actions in the real world.

Domain	Cluster
Identify surface features caused by constructive processes.	• Deposition (Deltas, sand dunes, etc.) • Earthquakes • Volcanoes • Faults
Identify and find examples of surface features caused by destructive processes.	• Erosion (water—rivers and oceans, wind) • Weathering • Impact of organisms • Earthquake • Volcano
Relate the role of technology and human intervention in the control of constructive and destructive processes.	Examples include, but are not limited to • Seismological studies, • Flood control, (dams, levees, storm drain management, etc.) • Beach reclamation (Georgia coastal islands)

Table 1: Science Curriculum from CCGPS in Spring Semester

Domain	Cluster
Operations and Algebraic Thinking	• Write and interpret numerical expressions. • Analyze patterns and relationships.
Number and Operations in Base Ten	• Understand the place value system. • Perform operations with multi-digit whole numbers and with decimals to hundredths.
Number and Operations—Fractions	• Use equivalent fractions as a strategy to add and subtract fractions. • Apply and extend previous understandings of multiplication and division to multiply and divide fractions.
Measurement and Data	• Convert like measurement units within a given measurement system. • Represent and interpret data. • Geometric measurement: understand concepts of volume and relate volume to multiplication and to addition.
Geometry	• Graph points on the coordinate plane to solve real-world and mathematical problems. • Classify two-dimensional figures into categories based on their properties.

Table 2: Mathematics Curriculum from CCGPS for 5th Grade

<u>Learning Module Developed</u>

Engineering Design Process. Our STEM-integrated robotics curriculum was organized via the engineering design process. Thus, each lesson of the curriclum is arranged according to the different phases of the engineering design process, as shown in Figure 2. This flow allows the students to internalize a general problem-solving process.

Situated Cognition: Scenario-based Learning. Meaningful learning occurs when it is promoted in authentic contexts (Collins, Brown, & Holum, 1991). To create a meaningful learning context, students were given a scenario from the perspective of scientists and engineers, and were then tasked with working through their design journals as if they were real scientists and engineers. The following is an example of a scenario for one of lessons from our curriculum:

> *The scientists now know how to program the robot to perform a sequence of movements. It is time for them to investigate the volcanic terrain using their satellite image and to plan the exact movements of the robot. Scientists will use these plans to determine the movements that need to be programmed in order for the robot to navigate the terrain while collecting the three samples.*

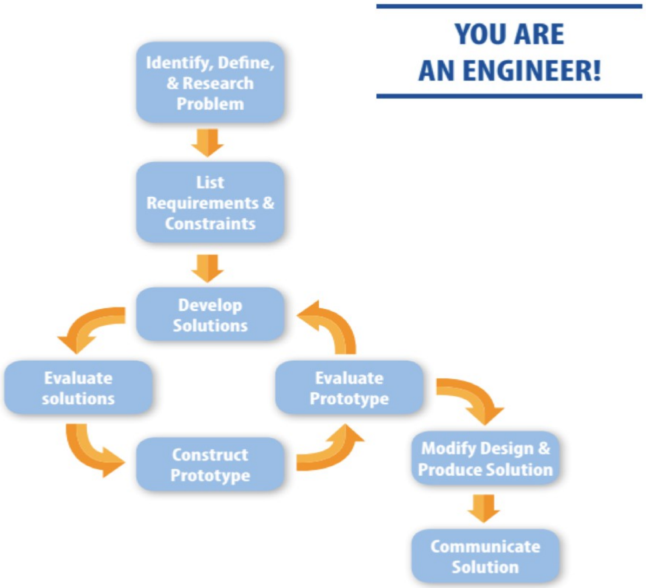

Figure 2: Status of Engineering Design Process

This kind of scenario was presented at the beginning of each lesson, and these smaller scenarios were all part of a larger context played out from lesson 1 through lesson 8.

Scientific Inquiry. Scientific inquiry is integrated throughout the student learning activities of the curriculum. As students begin programming, they must constantly observe the effectiveness of their program and draw conclusions about their program based on those observations. For example, in lesson 6, inquiry becomes more methodical as students must take measurements of the distance their robots travels at a certain speed. More concrete conclusions can be drawn with these sets of data and predictions can be made. Elsewhere students take on the role of engineers and utilize this data to make a more efficient program.

Mathematical Thinking. In order to solve the suggested problem, the lessons were composed such that the optimal solution could be found with mathematical thinking, mathematical problem solving strategies, and arithmetic. Through the lessons, students can explore the most efficient path for their robot on a coordinate grid and use a scientific inquiry process to analyze the speeds and distances associated with the movement of the robot. Thus students systematically analyze the movement of their robot in order to make precise calculations when predicting their proposed solution path.

Developing the Curriculum. Based on the aforementioned analysis of the existing curriculum, we developed an eight-lesson module that was designed for implementation at the 5th grade level. An overview of the series of eight lessons is presented in Table 3. The scenario was created to be a realistic experience encouraging the students to act as scientists through the exploration of a volcanic terrain, and act as engineers to complete the mission as efficiently as possible. The computer programing activities were carefully organized from lesson 3 to 6.

Lesson	Name	Driving Question
1	Danger Zone	How can scientists study dangerous environments?
2	Build a Bot	How do engineers build robots to accomplish specific tasks?
3	Primary Programming	How can the basic movements of a robot be controlled using simple programming commands?
4	Purposeful Programming	How can sequential movements of a robot be controlled using sequential programming commands?
5	Terrain Task	How can a robot be programmed to perform a specific task?
6	Prime Optimization	How can math be used to efficiently program a robot to perform a specific task?
7	Making Sense	How can sensors be used to program a robot to efficiently perform a specific task?
8	Share	How does the engineering design process help with problem solving?

Table 3: An Overview of Eight Robotics Lessons for 5th Grade

Computer Programming Activities in Context

We organized a series of sequential lessons where students learn computer programing according to the following four gradual stages. In the first stage, students are guided to understand a given problem situation, to articulate possible solutions, and plan for developing solutions. In the second stage, they learn the basic structure of computer programming with simple tasks. In the third stage, students begin to adapt the programming through problem solving activities of the various meaningful unit tasks. Finally, students try to solve the given problem or mission utilizing a computer program.

Out of the eight lessons developed for the STEM-integrated robotics curriculum, lessons 1 and 2 are dedicated to the first stage—understanding problems. Actual computer programming education spans lessons 3 to 6, which are organized according to all three of aforementioned stages. An evolution of the computer programming complexity can be seen as the students progress from lesson 3 through 6 (see Figure 3). In lesson 3, students first unpack basic programming tasks through acting out simple actions (i.e. walking forward, backward, left, right). They then must apply this thinking to actual simple programming tasks in lesson 4. These skills are further developed through the addition of meaningful tasks as students program to solve a problem in a scientific context (i.e. navigating volcanic terrain) in lesson 5. Finally, in lesson 6, students calculate speed and optimize the programmed solution by making more precise adjustments.

Initial Implementation and Evaluation

Development of the STEM-integrated robotics curriculum was followed by implementation in eight 5th grade classrooms throughout schools in southeastern United States. During implementation, data was collected to validate the effectiveness of the curriculum by measuring knowledge in STEM subject areas.

This curriculum is currently being validated through an empirical study. The examination of STEM attitudes and the degree of change in problem solving abilities based on the use of the engineering design process, scientific inquiry, and mathematical thinking are being analyzed through multiple data collection methods. These methods include the use of video recording for observing the change in students' inner perspectives, and analysis of artifacts measuring STEM knowledge through pre and post surveys, as well as students' workbooks.

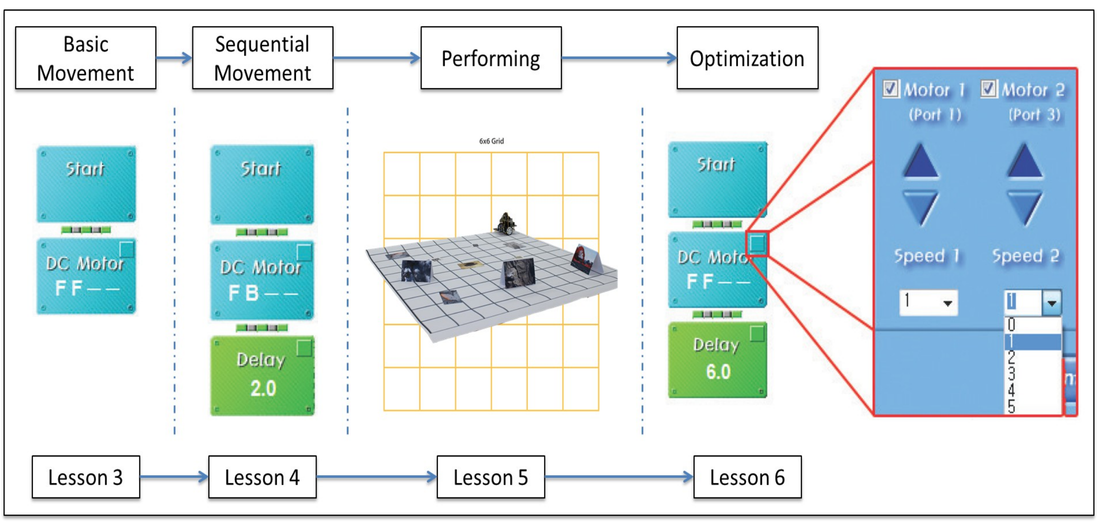

Figure 3: Programming Activities

Current data analysis of the 137 pre and post surveys administered indicates statistically significant differences. The participating students' mathematical thinking average out to 4.16 (*SD*=2.72) and 0.18 (*SD*=0.51) on the pretest, and 5.50 (*SD*=2.14) and 0.61 (*SD*=0.99) on the posttest. Compared to measurements taken before working with the curriculum, the students' performances after the curriculum indicates improvement. Further analysis of student observations and workbooks, along with feedback from the students and teachers, will help indicate the curriculum's effectiveness towards other STEM subject areas. This will, in turn, continuously enhance development of the curriculum through revisions, in order to maximize student engagement and mastery of computer programming.

Discussion and Next Steps

With the development of the framework for a STEM-integrated robotics curriculum, the next step will be to expand the module vertically and horizontally and develop a foundation for the adaptation of the curriculum model. Problem solving and programming skills will become the framework for how the curriculum will be designed for other grade levels. When the current fifth grade students move up to middle school, a high-level robotics curriculum is needed in order to ensure the continuity of the learning. Our team plans to develop modules for middle school incorporating specific 6th through 8th grade CCGPS standards that build on the computer programming knowledge gained by participating in the module developed in this study. In addition, with a focus on the robot's hands-on functions, the development of additional modules for lower elementary grade students who need concrete educational tools is required.

Conclusions

The aim of this project is to develop a STEM-integrated robotics curriculum for 5th grade through which computer programming can be taught in a meaningful way. Although integrating STEM subjects and multiple thinking skills into a classroom activity is not an easy task, through this project we successfully demonstrated how engineering design thinking, scientific inquiry, mathematical thinking, and computational thinking can be integrated seamlessly. Real-world computer programming requires logical, scientific, and creative thinking. This project presents the possible pedagogical strategies for promoting these types of thinking to solve complex problems. We found that the robots students built to solve a problem and the reflection journals they used during their journey are essential mediums for representing and revising their knowledge and thinking.

Acknowledgment

We acknowledge that part of this project was funded by the RoboRobo Co., Ltd through the Research for the Advancement of Innovative Learning (RAIL) at The University of Georgia.

References

An, J., & Park, N. (2012). The effect of EPL programming based on CPS model for enhancing elementary school students' creativity. *Embedded and multimedia computing technology and service*, 237-244.

Branch, R. M. (2009). *Instructional Design: The ADDIE Approach*: *Proceedings of the Second Sussex Conference*, *1977* (Vol. 722). Springer Science & Business Media.

Carder, L., Willingham, P., & Bibb, D. (2001). Case-based, problem-based learning: Information literacy for the real world. *Research Strategies, 18*(3), 181-190.

Collins, A., Brown, J. S., & Holum, A. (1991). Cognitive apprenticeship: Making thinking visible. *American Educator, 15*(3), 6-11, 38-46.

Cooper, S., Dann, W., & Pausch, R. (2003). Teaching objects-first in introductory computer science. *ACM SIGCSE Bulletin, 35*(1), 191-195.

Eide, A., Jenison, R., Mashaw, L., and Northup, L. (2002). *Introduction to engineering design and problem solving* (2nd ed.). New York, NY: McGraw-Hill.

Esteves, M., Fonseca, B., Morgado, L., & Martins, P. (2011). Improving teaching and learning of computer programming through the use of the Second Life virtual world. *British Journal of Educational Technology, 42*(4), 624-637.

Flannery, L. P., Silverman, B., Kazakoff, E. R., Bers, M. U., Bontá, P., & Resnick, M. (2013). Designing scratchJr: Support for early childhood learning through computer programming. In *Proceedings of the 12th International Conference on Interaction Design and Children* (pp. 1-10). ACM.

Georgia Department of Education. (2014a). *CCGPS Science Curriculum*. Retrieved from https://www.georgiastandards.org/Standards/Pages/BrowseStandards/ScienceStandardsK-5.aspx

Georgia Department of Education. (2014b). *CCGPS Mathematics Curriculum*. Retrieved from https://www.georgiastandards.org/Common-Core/Pages/Math-K-5.aspx

Kelleher, C., Pausch, R., & Kiesler, S. (2007). Storytelling alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1455-1464). ACM.