2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #33567

Computational Thinking in First-Grade Students Using a Computational Device (Work in Progress)

Ms. Barbara Fagundes, Purdue University

Barbara Fagundes is a first-year Ph.D. student in the Engineering Education Department at Purdue University. Her doctoral research interests involve the representation of women in the STEM field, k-12 STEM curriculum, and computational thinking.

Nrupaja Bhide, Purdue University

Nrupaja is a PhD student at the School of Engineering Education at Purdue University. She is interested in pre-college engineering education, and improving access to STEM education. She previously worked on low-cost makerspace programs in rural India with BAIF Development and Research Foundation.

Prof. Tamara J. Moore, Purdue University

Tamara J. Moore, Ph.D., is a Professor in the School of Engineering Education and the Interim Director of the INSPIRE Research Institute for Precollege Engineering at Purdue University. Dr. Moore's research is centered on the integration of STEM concepts in K-12 and postsecondary classrooms in order to help students make connections among the STEM disciplines and achieve deep understanding. Her work focuses on defining STEM integration and investigating its power for student learning.

Kristina Maruyama Tank, Iowa State University of Science and Technology

Kristina M. Tank is an Associate Professor of Science Education in the School of Education at Iowa State University. She currently teaches undergraduate courses in science education for elementary education majors. As a former elementary teacher, her research and teaching interests are centered around improving elementary students' science and engineering learning and increasing teachers' use of effective STEM instruction in the elementary grades. With the increased emphasis on improved teaching and learning of STEM disciplines in K-12 classrooms, Tank examines how to better support and prepare pre-service and in-service teachers to meet the challenge of integrating STEM+C disciplines in a manner that supports teaching and learning across multiple disciplines.

Computational Thinking in First-Grade Students Using a Computational Device (Work-in-progress)

Introduction

In recent years, there has been a growing emphasis on the importance of integrating and implementing STEM and computational thinking topics across K-12 settings, and considerable attention is being given to essential concepts related to computer science subjects [1]. As coding and software development is a part of full STEM education, there is an increased interest in implementing computational thinking and problem-solving skills in early education. Therefore, more studies have been done in the past years to promote computer science skills in the initial school years. Nowadays, computational thinking has been widely recognized as a fundamental skill to be used by everyone in the world by the middle of the 21st Century. Computational thinking is also considered crucial for developing engineering habits of minds and solving engineering problems [2]. When students work on coding, they can learn how to design a computer program while developing their computational thinking skills [3]. Computational thinking (CT) includes the thought processes involved in formulating problems, solving problems, building systems, and human behavior through the lens of computer science concepts [4]. However, little is known about how and to what extent children acquire CT skills [5]. Therefore, in this study, we implemented coding activities using a robot game called Code n Go Robot Mouse with first-grade students. With this in-progress study, we aim to answer the research question: *What evidence of CT competencies do students demonstrate when engaging in activities with the Code n Go Robot Mouse game?*

A theoretical basis for the study

Computational thinking (CT) is a "range of mental tools" [6, p. 33] that can be used for designing systems and solving problems in the 21st Century. Systems thinking and problem-solving represent the overlap between CT and engineering design [7-9]. However, Shute et al. [7] point out that unlike engineering, CT goes beyond physical constraints and can help us understand complex phenomena through simulations and modelling. Different interpretations of the term CT have resulted in many skills being brought together under this umbrella [5],[10]–[11]. In an attempt to provide a succinct and precise definition of CT, Aho [12, p. 834] describes it as the "thought processes involved in formulating problems, so that their solutions can be represented as computational steps and algorithms." Thus, CT skills are not limited to computer science; CT goes beyond that to encompass different ways of systematic thinking and problem-solving.

When CT is defined as a set of problem-solving skills, Wing [6] argues that it is a fundamental skill for everyone. She calls explicitly for introducing pre-college students to CT and including it in their basic education along with literacy and numeracy [6]. Since Wing's article in 2006, there has been considerable work on introducing CT in the K-12 context. Researchers, educators, and computer industry experts have come together to explore the "pedagogical aspects of computational thinking" and its implications for students [5 p. 39]. However, most of the work is focused on "syntactical issues" of defining what CT is, and now there is a need to explore the barriers that children face in learning CT, including how the barriers are defined and what can be done to address them [5]. In this study, we attempt to understand how students navigate computational thinking processes, especially in the earlier grades. Analysis of activities and interactions when doing CT tasks is an under-investigated area [5]. Therefore, in this study, we use naturalistic inquiry, which can help us capture - through natural language and everyday settings - what people know, perceive and understand [13]. This will be done by observing how the students respond to the various challenges presented to them through the Code n Go Robot Mouse game.

The Code n Go Robot Mouse game has the basic features of computer programming, such as planning & algorithms, coding, debugging, etc., Moore et al [14]. This activity was used by blinded [14] to understand how students engage with multiple representations when working on tasks that require CT. Educational robotics - learning by programming a robot - is a useful tool for learning CT skills and gives students the added benefit of interacting with a concrete object to construct their knowledge [15]. After reviewing the current research on the use of robotics for learning CT, Ioannou, and Makridou [15, p. 2541] emphasize that it is essential to understand "the processes and conditions" under which students display CT competencies. Previous studies often lack details about the learning activities, resources, and environments, making it harder to replicate the studies and expand the research on CT in classrooms [15]. This WIP study attempts to address this gap by providing the processes and findings of how the Code n Go Robot Mouse game can be used to observe CT competencies in first-grade students.

Methods

This qualitative study aims to explore first-grade students' computational thinking behaviors during a plugged activity. We refer to these activities as plugged activities, since a computational device, in this case the robot mouse, is used for all the study tasks. We conducted the research using a naturalistic inquiry approach where the researchers were present in the classroom, observing students in their natural learning environment while capturing their engagement in CT competencies. The data were collected at a midwestern elementary school in an urban area where 74% of the students are eligible for free lunch and have the following population demographics: White–38.3%, Black/African, American–24%, Hispanic–34.4%, and Multi-racial–2.7%.

Figure 1. Code & Go® Robot Mouse game
Adapted from learningresources.com

Figure 2. Robot mouse
Adapted from learningresources.com

We conducted this preliminary study to better understand students' CT behaviors during these class activities with the Code n Go Robot Mouse game (*Figure 1*), which we will refer to as the robot mouse game for the rest of this paper. The robot mouse game includes a mouse (*Figure 2*) with colorful buttons on its back which can be used for coding. The game also consists of a coding activity set, as well as maze grids, maze walls, tunnels, coding cards, activity cards (maps), and a cheese wedge. The Code and Go Robot Mouse Activity Set was developed by Learning Resources. The robot mouse is programmable and its nose lights up when it touches the cheese wedge. The maze grids are made to be interlocked into different configurations. Each grid has enough size for one forward movement of the mouse and the code cards have direction arrows to help students to create their code as a sequence of actions (directions) for the mouse.

The current study was conducted before the COVID-19 with first-grade students and organized to be administered over two days with morning activities and afternoon activities. For this WIP we are looking only at morning and afternoon activities from one day. In the morning, the researchers and the teacher taught the students some basic concepts about coding and presented the robot mouse to help them get familiarized. There were 18 first-grade students, 12 girls, and 6 boys participating in the study. We divided the students into six groups (2-4 students for each group–these were the students' regular table seats) where every group had one set of the robot mouse game. They were encouraged to work together as a team in the coding activity with the robot mouse game. Each group had around 60 minutes to work, and they were initially told to build the map using the pieces provided and follow the map given to them. Then, the students created their algorithm to help the robot mouse reach the cheese overcoming obstacles on the way. They used the map they built for this activity and other materials such as coding cards that helped them make a coding plan before actually pushing the buttons on the robot mouse.

Every time they achieved the goal, they moved to the next map, adding up some difficulty. The students completed the coding tasks mainly on their own with minimum intervention from the researchers and the teacher. The facilitators walked around to assist the students when they did not quite understand the task or were stuck.

The teacher and two of the researchers implemented the activities with the students, and everyone from the research team observed students' engagement, either in person or through video. We strategically set up the main camera to capture the whole classroom environment and cameras from iPads at each table to capture students' interaction during the coding task. We also conducted task-based interviews with the students, one of which is a “representations task” where students had to do the coding using only the maps and without physically having the robot mouse. Another set of task-based interviews was an "applying" task where students tried to use a given code to determine where the mouse would end up.

The research team observed and recorded all the groups during the class activities. However, in this in-progress research, two researchers watched the videos focused only on two groups of students. The data from these two groups were coded using a priori coding with the CT competencies: algorithms and procedures, abstraction, data analysis, data collection, data representation, automation, pattern recognition, and decomposition [16]. The two researchers coded the videos following the CT competencies, then compared their codes across the data, coming to a consensus on any discrepancies between them in the coding. In future work on the remainder of the data, we will use our codebook from this work-in-progress to do interrater reliability to code separately in the future.

Preliminary findings

This study aimed to identify and analyze CT competencies in children during a plugged activity with the robot mouse game. We would like to point out that these CT competencies were developed for K-12 contexts. Since we are using it in the K-2 context, we don't expect that K-2 students demonstrate all these competencies mentioned above. Indeed, we are considering the trajectories through which students can move forward to achieve these competencies. Our preliminary findings indicate that children can engage in coding activities by applying CT competencies. For the data we analyzed, students mainly demonstrated two CT competencies: Algorithm & Procedures and Debugging/Troubleshooting.

Algorithms & Procedures

This competency looks at how students engage with an ordered set of instructions. The competency is further divided into 5 sub-levels: Following, Identifying, Applying, Creating, and Automating. The two groups that we analyzed for this WIP study were prone to skip the planning phase with the coding cards and directly push the buttons into the robot mouse to start coding. For example, one student looked at the maze and then started pushing the buttons to determine how the mouse should move while reciting the directions "straight, straight, straight, turn." As we move forward, we can get a better idea of whether this behavior of skipping the planning was typical for the whole class. An interesting observation about the Algorithm and Procedure competency was that students were often demonstrating competencies of Creating and Applying algorithms simultaneously. This is seen in the previous example where the student was planning and pushing the buttons simultaneously. There were very few occasions when they sequentially did the process of first creating an algorithm (what steps the mouse needs to take to get to the cheese) and then applying it (input the steps in the robot mouse). For example, one student said, "let's lay them [coding cards] out… let's lay them out so that we know where to go." The students then put the directions in the mouse but stopped using the cards during the next try. So, the students were not consistent in following the coding process first and then applying the code. In one part of the activity, the students were only given the map and the coding cards but not the mouse. During this part, both groups used the cards to create an algorithm first, and they were only given the mouse to input their instructions after they had the algorithm. So even when students didn't consistently use the competency of Creating algorithms, they could - and did - demonstrate this competency when it was presented as a specific challenge.

Debugging and Troubleshooting

When the students were presented with the robot mouse game, they did a short, guided activity to learn what debugging is and how it should be done in this game. Because skipping the planning stage was common, when students needed to debug their code, they mostly relied on memory to remember which steps were wrong and how they should be corrected, making comments like "oh he [robot mouse] has to go forward, forward, and then turn that way." It is likely that when the students use their memory to debug, they are mentally processing the information and making sense of the problem. If the mouse didn't go in the direction they planned, they kept trying different combinations of steps, saying, "Let's try it again" or "Let me try that now." However, debugging was often more sophisticated when there was an intervention from the instructor/facilitator, who prompted students to use the coding cards. For example, even when the facilitators encouraged students to use the coding cards to design their algorithm, students consistently erased all their code and started over when their code didn't work as expected. But, with support from the facilitator, they looked for the part of the code that might be wrong to fix that without erasing all the code. For example, on one table, the students used the cards but immediately discarded them after putting in the instructions. After the facilitators' suggestion to only remove the code, the student said, "so that [debugging with cards] can help us build it [the correct path]." Even though the students were not using the coding cards at times, they demonstrated the CT competency Debugging/Troubleshooting. They did it with enthusiasm, exclaiming, "oh no, we were so close" or "it didn't work, let me try."

Other CT competencies

One other competency that was observed was related to patterning. Some students could identify patterns in their code as they moved through the game levels, which means they were engaging in pattern recognition. For example, when working on the course, the students stated they had seen the same sequence of code - patterns - from the game's previous level. Thus, they could easily code the current level in the course using prior level patterns.

Conclusion

In this in-progress research, we explored first-grade students' CT behaviors during a coding activity with a robot mouse game. Our study attempts to contribute to the existing literature by providing details about the learning activities and resources used for understanding CT competencies in young students. We intently implemented this project to examine how these competencies appear in young students. The preliminary results show that students from the two groups we analyzed demonstrated Algorithms & Procedure and Debugging/Troubleshooting competencies during the robot mouse game. There are two possible directions for future exploration. We want to see which other competencies students demonstrate and also whether the prevalence of these two competencies is seen for the entire class. The instance of patterning can also be explored to understand more about how students demonstrate this competency. In continued and future research, we will explore if this is true for all students in the class and whether this is due to the specific nature of the assigned challenges or because the students find it easy to engage with these CT competencies over others. In addition to these CT competencies, we have also seen some teamwork behaviors that we might want to explore in a future study about teamwork and collaboration in students at a young age.

References

[1] M. Guzdial, “Education: Paving the way for computational thinking,” *Communications of the ACM*, vol. 51, no. 8, pp. 25–27, Aug. 2008, doi: 10.1145/1378704.1378713.
[2] E. M. Haluschak, K. Maruyama, and K. M. Tank, “Actions and Approaches of Teachers Communicating Computational Thinking and Engineering Design to First Grade Students (RTP).” in 126th Annual Conference and Exposition ASEE, 2019.
[3] J. Moon, J. Do, D. Lee, and G. W. Choi, "A conceptual framework for teaching computational thinking in personalized OERs," Smart Learning Environments, vol. 7, no. 1, Dec. 2020, doi: 10.1186/s40561-019-0108-z.
[4] J. M. Wing, "Computational thinking benefits society," 40th Anniversary Blog of Social Issues in Computing 2014, vol. 26. 2014, Accessed: Mar. 07, 2021. [Online]. Available: http://socialissues.cs.toronto.edu/index.html%3Fp=279.html.
[5] S. Grover and R. Pea, "Computational Thinking in K-12: A Review of the State of the Field," Educational Researcher, vol. 42, no. 1. pp. 38–43, Jan. 2013, doi: 10.3102/0013189X12463051.
[6] J. M. Wing, "Computational thinking," Communications of the ACM, vol. 49, no. 3. Association for Computing Machinery, pp. 33–35, Mar. 01, 2006, doi: 10.1145/1118178.1118215.
[7] V. J. Shute, C. Sun, and J. Asbell-Clarke, “Demystifying computational thinking,” 2017, doi: 10.1016/j.edurev.2017.09.003.
[8] J. M. Wing, “Computational thinking and thinking about computing,” *Trans. R. Soc. A*, vol. 366, pp. 3717–3725, 2008, doi: 10.1098/rsta.2008.0118.
[9] H. Ehsan, A. P. Rehmat and M. E. Cardella, “Computational thinking embedded in engineering design: capturing computational thinking of children in an informal engineering design activity,” *International Journal of Technology and Design Education*, 2020, doi: 10.1007/s10798-020-09562-5.
[10] National Research Council, Report of a Workshop on the Scope and Nature of Computational
Thinking. Washington, DC: The National Academies Press, 2010.
[11] D. Weintrop et al., "Defining Computational Thinking for Mathematics and Science Classrooms," Journal of Science Education and Technology, vol. 25, pp. 127–147, 2015, doi: 10.1007/s10956-015-9581-5.
[12] A. v Aho, "Computation and Computational Thinking," The Computer Journal, vol. 55, no. 7, 2012, doi: 10.1093/comjnl/bxs074.
[13] M. Quinn Patton, “Qualitative methods and approaches: What are they?,” *New Directions for Institutional Research*, vol. 1982, no. 34, pp. 3–15, 1982, Accessed: Apr. 13, 2021. [Online].
[14] T. J. Moore et al., "Multiple representations in computational thinking tasks: A clinical study of second-grade students," Journal of Science Education and Technology, vol. 29, pp. 19–34, 2020, doi: 10.1007/s10956-020-09812-0.
[15] A. Ioannou and E. Makridou, "Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work," Education and Information Technologies, vol. 23, no. 6, pp. 2531–2544, Nov. 2018, doi: 10.1007/s10639-018-9729-z.
[16] M. M. Hynes et al., "Inspiring young children to engage in computational thinking in and out of school," 126th Annual Conference and Exposition ASEE, 2019.