ASEE'S VIRTUAL CONFERENCE
At Home with Engineering Education
JUNE 22 - 26, 2020 #ASEEVC
Paper ID #30871

First-graders' Computational Thinking in Informal Learning Settings (Work in Progress)

Ms. Hoda Ehsan, Purdue University at West Lafayette

Hoda is a Ph.D. student in the School of Engineering Education, Purdue. She received her B.S. in mechanical engineering in Iran, and obtained her M.S. in Childhood Education and New York teaching certification from City College of New York (CUNY-CCNY). She is now a graduate research assistant on STEM+C project. Her research interests include designing informal setting for engineering learning, and promoting engineering thinking in differently abled students in informal and formal settings.

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 representation of woman in the STEM field, k-12 engineering education and computational thinking.

Prof. Tamara J Moore, Purdue University at West Lafayette

Tamara J. Moore, Ph.D., is a Professor in the School of Engineering Education and Director of STEM Integration in the INSPIRE Institute 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. Tamara Moore received an NSF Early CAREER award in 2010 and a Presidential Early Career Award for Scientists and Engineers (PECASE) in 2012.

Kristina Maruyama Tank, Iowa State University

Kristina M. Tank is an Assistant 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 disciplines in a manner that supports teaching and learning across multiple disciplines. More recently, her research has focused on using literacy to support scientific inquiry, engineering design, and STEM integration.

Dr. Monica E Cardella, Purdue University at West Lafayette

Monica E. Cardella is a Program Director with the Division of Research on Learning at the National Science Foundation. She is also a Professor of Engineering Education at Purdue University.

First-graders' Computational Thinking in Informal Learning Settings (Work in Progress)

Introduction

Recently computational thinking has emerged as a fundamental skill for pre-college students. One way of integrating this new skill into the curriculum is through integrated STEM education. The importance of STEM education as a driving force for economic stability and growth is unquestionable and has been a catalyst for change across the globe in recent years. Given the growth of technology and digital computers in the 21st century and the demands for professionals and engineers with computer science and problem-solving skills, computational thinking (CT) has gained attention in pre-college STEM education. Furthermore, Wing's influential 2006 article made the case that CT should be a skill that all students, including pre-college and non-computer science majors, should learn [1]. However, if CT is something that all students should learn then, as noted in [2], "to be useful a definition must ultimately be coupled with examples that demonstrate how computational thinking can be incorporated in the classroom" (p. 50). Therefore, in this study, we aim to characterize the computational thinking of first-grade students while participating in a field-trip with activities that integrate CT into engineering tasks. The research question for our work-in-progress study is: *What does children's engagement in computational thinking competencies look like when solving different engineering and computing problems?*

An Overview of the Empirical and Theoretical Basis for the Study

CT incorporates wide ideas and concepts as it is a method for solving problems and a specific way of thinking that connects the following concepts: logic, algorithms, decomposition, patterns, abstraction, and evaluation [3]. Wing's definition of CT [1] includes thinking recursively, by interpreting code and data, using abstraction and decomposition when working on large and complex task or designing a large complex system, and problem-solving that is not only beneficial to experts in programming, but equally valuable to those who are not using a computer. Furthermore, CT is "a multifaceted construct as it [is comprised] of several sub competencies such as problem decomposition, abstraction, debugging, and pattern recognition" [4]. Research on how children grasp CT concepts and how to integrate its concepts in education is still insufficient [5].

CT as a problem-solving approach has been defined as the necessary skills for everyday activities [1]. Lu and Fletcher [6] believe that if practicing CT starts from young ages, by the time children get to high school, CT will become the second nature for them. Several studies provide evidence that students as early as Kindergarten can engage in CT competencies – acquiring skills through different types of unplugged and plugged activities (e.g., [7][8][4][10]). Plugged activities refer to those which require using digital devices and unplugged activities can be done without any digital devices. The findings of our earlier works showed children's capability of engaging in competencies of computational thinking. For example, three studies conducted in informal learning settings provided evidence of K-2 aged children engaging in meaningful CT during an unplugged engineering design activity with and without adult supervision [4][10] and in a plugged activity translating between different computational representations while engaging with a computational cognitive device [9]. Additionally, we observed children's engagement in computational thinking competencies could be supported by adults' facilitation strategies [12][13].

For this work-in-progress paper, we are using a CT framework developed by the Purdue INSPIRE Research Institute for Pre-College Engineering [11]. The CT framework includes Abstraction, Algorithms and Procedures, Use of Data, Debugging/Troubleshooting, Problem Decomposition, Parallelization, Simulations & Patterning. A definition of each of the competencies of our CT framework will be included in the results section.

Methods

In this Work in Progress paper, we utilized an exploratory qualitative approach to capture young children's engagement in computational thinking competencies. We focused on video data we collected from 21 students from a first-grade class during a field trip to a small science center. The students were very diverse in terms of gender and race. During this field trip, students were divided into groups of three-four. Each group was assigned a facilitator who was also a researcher. The groups circulated around five different stations and interacted with different engineering activities. The facilitator explained the activities to the students and occasionally interacted with students and answered their questions. The field trip was 2-hour long and students spent 20-25 minutes on each activity. The activities were in a wide range of unplugged and plugged activities. Three of the activities are included in this study and are described below.

The Foos is a programming app designed for 4- to 9-year old children. In this game, children have to help inhabitants of the world of Foosville solve problems by telling them what to do using codes. On each level, the students have an object to find and navigate to reach it. The codes are direction arrows to reach the object. Early levels the goal is seen at the beginning of the level. As the game progresses, the goal is off screen and students must search for the object.

Coding for the Critters is an exhibit designed for to introduce CT to 5- to 7-year old children. The exhibit poses a story that a robot needs to deliver medicine to sick animals. The exhibit consists of five stations :
(1) a panel of introduction to CT and the activity instructions, (2) a physical maze in which children can climb in pretending to be robots, (3) a hands-on station to plan and test routes through the maze, (4) panels with detailed information about different branches of engineering, and (5) an interactive coding video game that a robot should be coded. In this field trip children only interacted with the station number 2 and 5.

Puppy Playground is an engineering design activity that 5- to 7-year old children are asked to design a puppy playground for Eva's puppy. They use the giant foam blocks in the science center to build their playground.

To analyze the video recordings, we utilized a coding scheme based on our CT framework. To gain a deep understanding of the competencies, we collectively watched several videos of K-2 aged children engaging in STEM+CT activities in different context and discussed enactment of CT competencies.

Findings

In this WIP paper, we are reporting on the preliminary findings of analyzing the engagement of 21 first-grade children in CT competencies during four unplugged and plugged activities. Our preliminary findings revealed that children could engage in CT competencies, but their engagement looked differently based on the activities. In Table1, we have presented examples of children's enactments in CT competencies for each activity. As obvious in the Table, we observed examples of children utilizing use of Data, debugging, Problem Decomposition and Patterning in all three of the activities, while Abstraction, Algorithm, Parallelization and Simulation only happened in two activities.

Conclusion

In this WIP paper, we shared multiple examples of first grade children engaging in CT competencies across different unplugged and plugged engineering activities. While these examples would not always mirror what these CT competencies look like when enacted by young children, we believe presenting a wide range of examples can broaden our understating of children's abilities to engage in computational thinking. Additionally, the findings highlighted that different unplugged and plugged activities can provide different CT opportunities for children. These activities were intentionally designed with components to promote CT development in children. Capturing these components can help educators, researchers and curriculum developers to provide and design effective CT opportunities for young children. Therefore, future research should involve an in-depth analysis of the activities to investigate ways these activities engage children in CT. In the next stages of this study, we will also focus on how children solve problems using combinations of CT competencies.

Table 1. Examples of Computational Thinking Competencies across All Activities

		Activities 1		
		Coding for the Critters	Puppy Playground	The Foos
CT Competencies	**Abstraction:** Identifying and utilizing the structure of concepts/main ideas (without a specific context) to generalize	We did not see any meaningful examples of Abstraction in this activity.	Student 1 arranges the shapes on their vertical side so that they are tall, but the student 2 keeps laying them down flat, on their horizontal. When the student 1 sees this, he says: "We can make it like 2D". Thus, student A start laying down the other shapes on the horizontal. It seems to be simulation, since the student is turning a physical prototype into a representation of it.	When playing the game multiple times, students begin to realize that the maze is the same each time. So, they apply the code (direction arrows) successfully for the entire level even when they cannot see part of the maze.
	Algorithm and Procedures: Following, identifying, applying, creating and automating an ordered set of instructions (i.e., through selections, iteration, and recursion)	The students are interacting with the coding game. One student enters some codes while using his fingers as he counts how many moves the robot has to take, then he hits Go. The code doesn't work and he says: my bad. The second student deletes all the codes that were written. Then he starts from the beginning and creates new codes.	We did not see any meaningful examples of algorithm in this activity.	The first-time students were presented with a goal for the level being on the left (rather than the right) caused difficulties for the students. Their algorithms began by focusing on left-to-right progressions. However, as they failed the level and attempted again, students tried new arrow directions and began to build successful algorithms.
	Use of Data: Combined the following codes into a larger code to encompass all use of data (including Data Collection, Data Analysis, Data Representation)	Facilitator reads the prompts on the panel. The students engage in the conversation below: Student #1: So, we need to get these balls to them [the animals]? Student #2: Look at the colors. They have different colors. We have to drop the balls in the tunnels for them.	The facilitator asks if it's a big dog or a little dog. One of the students replies: "It's a little dog, … because it's right there" while she points to the poster where there is a picture of the dog along with the problem statement.	The students look around by moving through the game scenario to locate their start point and where the target shapes are. They analyze the environment. They attempt to try out the sequence of arrows.
	Debugging/Troubleshooting : Identifying and/or	Student A deletes all the codes that his friend had created and applied previously which went wrong (the child indicated: "my bad"). He starts	Facilitator helps students identify what is the issue with one of the blocks (slide). Then, one of the students approached to	Student A is working on her first level of the game. She adds four right arrows, but there should also be jumps in there. Another student who observes what she

	addressing problems (errors, lack of efficiency/effectiveness) that inhibit progress toward task completion.	from the scratch and creates new codes.	try to figure out what the issue was. (debugging)	is doing tells her: “now you have to debug it” and then she corrects her codes.
	Problem Decomposition: Breaking down data, processes or problems into smaller and more manageable components to solve a problem	Teacher reads the instruction and tells the students try to get to the cat, to the dog and to the rabbit [with no specific order]. The student enters the codes to take her to the rabbit and says, “I’m trying to take it to the rabbit first”. She presses GO, and then sees the error OOPS. Then the other child starts from that point and creates and applies the codes that take them to the dog. She then creates a series of codes that takes them to the cat.	One of the students says that he is going to tart with a fence and he starts placing shapes in a fence before moving on to the next thing.	The student is having an issue with the algorithm. He says, "Let's see what happens" and tests a piece of the code. He says "now that's got a big problem...Oh, now I see". He, therefore, breaks down the problem, and tried just a piece of the code without worrying about the whole thing. He then focused on the other parts.
	Parallelization: **Simultaneously** and intentionally processing smaller tasks with different goals to more efficiently reach an overall goal.	Students decide that each one is responsible of feeding one animal. They go inside the maze at the same time and drop the balls in the tubes.	Student 1 focuses on the overall structure of the yard, starting by building walls. In meanwhile, student 2 focuses on providing the yard with a few toys for the puppy to play. Both students have a defined role: while one of them work on building the wall, the other one focus on providing some toys for the yard. These two different roles aim to achieve an overall goal which is build a puppy playground.	We did not see any meaningful examples of parallelization in this activity.
	Simulations: **Imitating** natural or artificial processes by using a developed model or representation.	During the coding game, the student enters some codes that can take the robot to the rabbit. Before she tests her codes, she uses her fingers and shows the path she is expecting her code to take the robot to.	We did not see any meaningful examples of simulation in this activity.	We did not see any meaningful examples of simulation in this activity.
	Patterning: Recognizing and identifying	During the maze activity, students recognize that the color pattern between elements of the problem	Time is over and the student tells the facilitator: “I was trying to make patterns”	Student 1 helps out Students 2 the student by telling her, “So now you have

	patterns, trends, and regularities in data (Google) or creating patterns	(I.e. the color of the tubes, balls and the animals). As a result, they plan to drop the same color ball in the same color tube for each animal.	while she places one of the shapes in the design structure.	to debug it and go like a pattern, run, jump, go, and then jump".

[1]Links to the activities which include pictures and descriptions will be provided when paper is unblinded.

References

[1] J. Wing, "Computational Thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–35, 2006.

[2] W. Sung, J. Ahn, and J. B. Black, "Introducing Computational Thinking to Young Learners: Practicing Computational Perspectives ThroughEmbodiment in Mathematics Education," *Technol. Knowl. Learn.*, vol. 22, no. 3, pp. 443–463, 2017.

[3] S. P. Jones, "Computing at school in the UK : from guerrilla to gorilla," *Commun. ACM*, no. April, pp. 1–13, 2013.

[4] A. Dasgupta, A. M. Rynearson, S. Purzer, H. Ehsan, and M. E. Cardella, "Computational Thinking in K-2 Classrooms: Evidence from Student Artifacts (Fundamental)," *2017 ASEE Annu. Conf. Expo.*, p. 22, 2017.

[5] C. Wilson and M. Guzdial, "How to make progress in computing education," *Commun. ACM*, vol. 53, no. 5, pp. 35–37, 2010.

[6] J. J. Lu and G. H. L. Fletcher, "Thinking about computational thinking," *SIGCSE Bull. Inroads*, vol. 41, no. 1, pp. 260–264, 2009.

[7] M. U. Bers, "Coding as another language: a pedagogical approach for teaching computer science in early childhood," *J. Comput. Educ. J. Comput. Educ.*, vol. 6, no. 4, pp. 499–528, 2019.

[8] H. Ehsan, T. M. Dandridge, I. H. Yeter, and M. E. Cardella, "K-2 students' computational thinking engagement in formal and informal learning settings: A case study (fundamental)," *ASEE Annu. Conf. Expo. Conf. Proc.*, vol. 2018-June, 2018.

[9] T. J. Moore *et al.*, "Multiple Representations in Computational Thinking Tasks: A Clinical Study of Second Grade Students," *J. Sci. Educ. Technol.*

[10] Ehsan, H., Rehmat, A. P., & Cardella, M. E. (2020). Computational thinking embedded in engineering design: capturing computational thinking of children in an informal engineering design activity. *International Journal of Technology and Design Education*, (0123456789). https://doi.org/10.1007/s10798-020-09562-5

[11] Hynes, M.M., Cardella, M.E., Moore, T.J., Brophy, S. P., Purzer, S., Tank, K.M., …Ehsan, H. (2019). Inspiring young children to engage in computational thinking in and out of school (research practice). *ASEE Annual Conference and Exposition, Conference Proceedings.*

[12] Ehsan, H., Rehmat, A. P., Osman, H., Ohland, C., Cardella, M. E., & Yeter, I. H. (2019). Examining the role of parents in promoting computational thinking in children: A case study on one homeschool family (fundamental). *ASEE Annual Conference and Exposition, Conference Proceedings*.

[13] Rehmat, A. P., Ehsan, H., & Cardella, M.E. (2020). Instructional strategies to promote computational thinking for young learners. *Journal of Digital Learning in Teacher Education, 36*(1), 46-62.