Computing for the Critters: Exploring Computational Thinking of Children in an Informal Learning Setting

Hoda Ehsan
Purdue University
School of *Engineering Education*
West Lafayette, IN, USA
hehsan@purdue.edu

Carson Ohland
Purdue University
School of Aeronautics and Astronautics Engineering
West Lafayette, IN, USA
cohland@purdue.edu

Tikyna Dandridge
Purdue University
School of *Engineering Education*
West Lafayette, IN, USA
tdandrid@purdue.edu

Monica Cardella
Purdue University
School of *Engineering Education*
West Lafayette, IN, USA
cardella@purdue.edu

***Abstract*— This Work-in-Progress study focuses on computational thinking (CT) in children. Women, minorities, and persons with disabilities are currently underrepresented in STEM education and careers. However, providing appropriate early CT learning opportunities can potentially help learners develop both interest and self-efficacy in computer science and STEM fields.**

Our research aims to characterize computational thinking and engineering thinking for K-2 aged children. In this paper, we report on our findings of children exploring a recently installed exhibit at a local Midwestern science center. The exhibit was designed to help children learn about engineering careers and CT, and provides different learning activities to elicit CT in children from diverse learning groups. The exhibit provides information about different engineering disciplines/fields, and examples of engineers from diverse groups (i.e. it includes images of people of different races, ethnicities, gender, and ability).

In this study, we audio- and video-recorded families while they explored the exhibit. They then participated in interviews and completed surveys. The video data provided evidence that children can engage in a range of CT competencies in the computer-supported coding game. Our findings can help us further consider how families can support children as they learn CT skills in out-of-school environments.

Keywords—Computational Thinking, Out-of-School, Engineering Education

I. Introduction & Background

Consistent with efforts to increase engagement in STEM education and the world-of-practice, many have called for increasing diversity in STEM education [1], [2]. In a recent report, the National Science Board [2] highlights the importance of preparing skilled individuals for the STEM workforce who will become future innovators. It also emphasizes the notion of equity and access in STEM education, meaning that all children, regardless of their gender, background and abilities, should be provided with appropriate STEM learning experiences that they relate to, engage in and benefit from.

Providing appropriate and diverse learning opportunities will invite individuals with different backgrounds, ethnicities, gender identities and abilities to the participate in STEM fields. According to the National Science Foundation [10] people with disabilities, minorities and women are underrepresented in STEM education and STEM-related workforce. Therefore, creating opportunities for individuals from these underrepresented groups is necessary.

A. Computational Thinking & Children

In the past decade, academic discourse on computational thinking (CT) in pre-college research has increased [3] and K-12 education standards have considered CT as a subject in K-12 education (NGSS) [4]. In 2006, CT was described as a set of thinking skills that all students needed to solve problems across multiple fields [5]. It is not only computer scientists' ways of thinking, but a logical problem-solving process which can be relevant to all disciplines. In other words, "Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent" (p.1) [6]. CT involves multiple competencies such as abstraction, pattern recognition, decomposing problems and developing algorithms and these competencies enable people to solve complex problems.

To prepare children to live in a technology- and computing-oriented world, we must provide opportunities for children to develop CT abilities. Beyond computer science careers, learning CT can prepare children for a variety of STEM-related careers, as there is overlap between CT and engineering, mathematics, and algorithmic thinking, and CT can help with everyday problem solving [7]. Therefore, it is critical that we provide opportunities for children to solve problems computationally and we expose children to tasks that require CT and computational tools.

Jeannette Wing first highlighted the importance of teaching computational thinking to children in 2006 [5], and since then, many researchers, educators and policy makers have considered computational thinking for K-12 children as an application of the science of learning [19]. Much of the recent work on K-12 education has focused on definitional issues and tools that promote computational thinking [20] which also resulted into designing grade- and age-appropriate curriculum and standards like the Next Generation Science Standards [21]. Hence, Grover and Pea [20] state that there exists a large gap that highlights the need for empirical inquiries.

On the other hand, the empirical studies that have investigated computational thinking in K-12-aged children mostly have focused on 3rd grade and above [16]. For example, Bruke [17] used a programming language with middle school students in an afternoon school program. The finding suggested that children were able to use computational skills during programming and helped them in learning content of different subjects. In a recent study, Weindtrop and his colleagues [18] aimed to bring computer science to elementary classrooms and investigated children's engagement in programming and computational thinking practices. However, their focus was mainly on 3rd-6th grade students. Therefore, characterizing computational thinking of younger children is necessary, and this the aim of this study.

B. *Learning in Informal Environments*

Learning is not limited to school settings, but constantly happens in a variety of contexts. According to the NRC report Learning Science in Informal Environments: People, Places, and Pursuits [8], there are four major categories of learning environments: everyday activities, programs, designed settings and in-school. The authors stated that people spend more than 80% of their time in out-of-school settings. Therefore, it can be argued that most learning happens in out-of- school settings. While learners in informal learning settings might not be aware of the learning that is happening, the learning that takes place in informal settings is as rich and influential as learning in classrooms. These experiences grab learners' attention and interest, support their direct experiences and conceptual understanding, and motivate learners to investigate the broader educational landscape [8]. Moreover, Miaoulis [9] posits that exposing children to engineering and technology-based activities is important as it promotes problems solving and prepares children to solve their everyday problems which mostly happen in out-of-school settings.

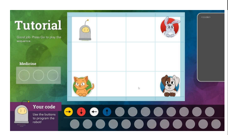
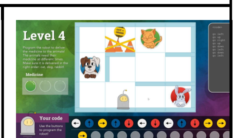
C. *Purpose of the Study*

In our larger STEM+C project, we aim to provide opportunities for children from diverse groups to engage in engineering and computational thinking in both formal and informal settings. With regards to informal learning opportunities, we have designed and installed an exhibit in a local science center that engages children in CT and engineering and it is designed to be inviting to all children. One purpose of this project is to investigate what CT looks like when enacted by K-2 aged children in informal learning environments like museums and science centers. In this Work-in-Progress paper, we aim to *present examples of CT exhibited by K-2 aged children during the computer-based coding game portion of the exhibit*.

II. Overview of the Exhibit

The science center exhibit, *Computing for the Critters*, was designed to encourage and introduce computational thinking and engineering to children and their families. The exhibit consists of five sections: An introduction to computational thinking and the activity instructions, a physical maze in which children can climb, a station to plan and test routes through the maze, panels with detailed information about different branches of engineering, and an interactive coding video game. Table 1 illustrates photos of the Interactive Coding game platform.

Table 1. Interactive Coding Game platform

<table>
<tr><th colspan="3">Interactive Coding Game</th></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
<tr><td colspan="3">
</td></tr>
</table>

The portion of the exhibit that we focus on in this paper is an interactive multilevel computer-based coding game. The purpose of the game, like the maze before it, is to deliver medicine to all three of the animals in an animal hospital. In the game, however, attendees are prompted to code a robot to deliver the medicine for them instead of moving through a physical space. Families can enter a series of movement (up, down, left, right) instructions for the robot, and can then press the “GO” button to see the robot follow the input path. Each of level consists of a different maze which the robot must navigate to deliver to all three animals in a single execution of the code. The robot must sometimes reach the animals in a specific order if the animals need to receive the medicine at particular times. If a level is successfully completed but a more efficient path is possible, the player is given the option to improve their solution to use fewer moves.

III. Methods

A. Participants

This study was conducted at a science center in the Midwest, as part of an ongoing investigation that is part of a larger study that examines children’s computational thinking in both formal and informal settings. After the research team gained ethical approval from the IRB office at the researchers’ institution, families were invited to participate in the study through two methods. The first method was contacting parents of children from a previous study that explored computational thinking in the classroom. The second method for recruitment was approaching families already visiting the science center who had K-2 aged children

For this Work-in-Progress paper, we use a Case Study approach, and we report preliminary findings from five of the families that participated in this study. The families are diverse in terms of their composition (i.e. the number of children, gender of children and parents, and parents’ age range) and parents’ prior engineering experiences. For example, one family included a father and a daughter, whereas another family who participated in the study consisted of two parents and one seven-year-old son and three older siblings. One mother stated that the family participated in several engineering and computer science activities before; however, for some families this was their first experience.

B. Research Design

To capture the nature of computational thinking exhibited by families when visiting and interacting with the Coding for the Critters exhibit, we have used a qualitative case study methodology. According to educational researchers, the case study methodology is an empirical form of inquiry that examines a phenomenon within a bounded system or a case [11-13]. Researchers who employ the case study methodology can use as few as an individual case and provide in-depth and descriptive analysis that interprets critical events. [14]. In this study, each K-2-aged child represents a case and the phenomenon under investigation is child’s engagement in computational thinking.

Audio and video recordings were collected of the families interacting at the exhibit so that interactions and conversations could be examined at a later time. After playing at the exhibit, parents and children participated in an interview about their experience with our exhibit and prior interacting with this exhibit and prior activities that may have been similar. Finally, parents completed a survey to provide information about their engineering background and their perceptions of engineering. For this Work-in-Progress, we are only focusing on the analysis from audio- and video-recording data of families’ interactions with a multilevel computer-based coding game.

C. Analysis

To analyze the video and audio-recordings from each family, we used a coding scheme based the computational thinking framework that our research team has created previously [22-23]. Table 2 includes the computational thinking framework.

In this study, we have utilized video analysis as an approach to make meaning of children’s actions and dialogue. We have followed the analytical model by Powell, Francisco, and Maher [15]. This model suggests seven non-linear phases to analyze videos including reviewing the video data, describing the video data, identifying critical events, transcribing, coding, constructing a storyline, and composing a narrative.

To determine which actions in the video represented a CT competency, we watched videos collectively to discuss how a CT code was depicted by the children and parents. We then coded each video individually and met to compare and reconcile any disagreements. Coding disagreements were resolved by re-watching the video segment where the code was applied and negotiating which CT code was most appropriate. This process was carried out until interrater agreement was achieved.

Table 2. Definitions of Computational Thinking Competencies

CT Competency	Definition
Abstraction	Identifying and utilizing the structure of concepts/main ideas
Algorithms and Procedures	Following, identifying, using, and creating an ordered set of instructions (i.e., through selection, iteration and recursion)
Automation	Assigning an appropriate set of tasks to be done repetitively by computers
Data Collection	Gathering information pertinent to solve a problem
Data Analysis	Making sense of data by identifying trends
Data Representation	Organizing and depicting data in appropriate ways to demonstrate relationships among data points via representations such as graphs, charts, words or images
Debugging/ Troubleshooting	Identifying and addressing problems that inhibit progress toward task completion
Problem Decomposition	Breaking down data, processes or problems into smaller and more manageable components to solve a problem
Parallelization	Simultaneously processing smaller tasks to more efficiently reach a goal

Simulations	Developing a more or representations to imitate natural and artificial processes
Pattern Recognition	Observing patterns, trends, and regularities in data (Google's definition)

IV. FINDINGS

In this Work-in-Progress paper, we are only reporting on the preliminary findings of analyzing the engagement of five families when playing the multilevel computer-based coding game. From our analysis, our preliminary findings revealed that families could engage in computational thinking while playing a multilevel computer-based coding game. Although children demonstrated that they could engage in CT independently while playing, parents were able to provide complementary CT competencies that helped children progress through each level when challenges would arise for children. As children advanced to higher levels, the role of parents were likely to transition from being a supporter to a coordinator of activity. Overall, collaboration between children and parents during the game resulted in the following CT competencies: Abstraction, Algorithm and Procedure, Data Collection, Debugging and Troubleshooting, Pattern Recognition, Problem Decomposition and Simulation. Below we will describe what each competency looks like when enacted by children in the context of the computer-based coding game.

Abstraction involves identifying and utilizing the main ideas or the structure of concepts in order to simplify the problem. In other words, abstraction is when we filter out some details or characteristics to focus on those main and necessary details that help us transfer information in a context that a problem has occurred. In the context of the coding game, children were able to abstract the arrows on the game buttons and recognize that they hold the meaning of the move direction. They also engaged in abstraction as they saw shapes of different obstacles like traffic cone or walls and realized that the robot is not able to move through those blocks that have obstacles.

Algorithm and Procedure involves following, identifying, using and creating a set of sequenced instructions to solve a problem. In the context of this coding game, children engaged in this competency very often as they were coding the robot to deliver medicine to certain animals. They created their own sets of instructions or followed other family members' instructions for the robot to move. The instructions were a set of arrows that were put in a sequential series.

Data Collection involves gathering necessary information to solve a problem. In the context of this coding game, children were engaged in collecting data by reading the task prompts and discussing how they want to help the robot deliver medicine to the animals. Another example of Data Collection was when a child would count the steps in a few different paths in order to identify the most efficient path to code for the robot.

Debugging and Troubleshooting involves identifying and addressing errors within the task. In the context of the computer-based game, children were able to debug the sets of instruction they created for the robots. They were able to identify which codes (arrows) were not in the right place that caused the robot to move on a wrong path. They were then able to address the error by either recording from the beginning or fixing the parts they saw errors.

Pattern Recognition involves observing patterns in small/decomposed pieces of the tasks or data that helps solve the more complex problems. In the context of the computer-based coding game, children were able to observe how a chunk of codes (arrows) would work together when being run, and then they would use that chunk of codes as one code for next problems.

Problem Decomposition involves breaking down the problem into smaller and more manageable components to solve the problems. In the context of the computer-based coding game, problem decomposition happened when children read the task prompts about the order of animals that should get their medicine. Children divide the task, by thinking and talking about to which animal they have to deliver the medicine first, and then planning for how to code the robot to get to the animals in the requested order.

Simulation involves developing a model or a representation to imitate natural or artificial process. In the context of the computer-based coding game, children used their fingers to imitate the movement of robot. They used their fingers to test the algorithm they created for the robot to deliver medicine to animals.

V. CONCLUSION

For this Work-in-Progress paper, our goal was to investigate what CT can look like when enacted by K-2 aged children as they interacted with a computer-based coding game as part of a science center exhibit. To answer this question, we analyzed data collected from five families who visited this exhibit at their local science center. The preliminary findings highlighted that children are able to engage in a variety of CT competencies. These examples shed light on what computational thinking of K-2 aged children looks like.

We also found that children's engagement in CT differed with and without adults' help. As children progress through the game, the type of support they needed changed and as a result, parents' roles changed as well. Therefore, future research is needed to investigate parents and siblings' roles and how they influence children's engagement in CT, and also the strategies they need to use to promote CT in K-2 aged children.

ACKNOWLEDGMENT

The material is based upon work supported by the National Science Foundation under Grant DRL- 1543175. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The Computing for the Critters exhibit was developed in partnership with Purdue's Exhibit Design Center.

We would like to thank Kiri Northam, Jeff Rollins and Ariel Murawski for their work and their creativity in creating the exhibit.

REFERENCES

[1] National Academy of Engineering, "*The engineer of 2020: visions of engineering in the new century*." Washington, DC: National Academies Press Washington, DC, 2004.

[2] National Science Board, *Preparing the Next Generation of STEM Innovators: Identifying and Developing the Nation's Human Capital*. 2010.

[3] B. C. Czerkawski and E. W. Lyman, "Exploring issues about computational thinking in higher education," *TechTrends*, vol. 59, no. 2, pp. 57–65, 2015.

[4] National Research Council, "*Next generation science standards: For states, by states*," The National Academies Press, Washington, DC, 2013.

[5] J. M. Wing, "Computational Thinking," *Commun. Assoc. Comput. Mach.*, 2006.

[6] J. Cuny, L. Snyder, and J. M. Wing, "Demystifying computational thinking for non-computer scientists," *Unpubl. Manuscr. progress, Ref. http//www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf*, 2010.

[7] V. Barr, C. Stephenson, and B. V. Barr, "Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?," *ACM Inroads*, 2011.

[8] National Research Council, "Learning science in informal environments: People, places, and pursuits," National Academies Press, 2009.

[9] I. Miaoulis, "K-12 Engineering–the missing core discipline," in *Holistic engineering education*, Springer, 2010, pp. 37–51.

[10] National Science Foundation, National Center for Science and Engineering Statistics, "*Women, minorities, and persons with disabilities in science and engineering.*" Special Report NSF, 15-311. Arlington, VA, 2015

[11] R. K. Yin, *Case study research: Design and methods.* Thousand Oaks, CA: Sage, 2009.

[12] S. B. Merriam and E. J. Tisdell, *Qualitative research: A guide to design and implementation.* San Francisco, CA: Jossey-Bass, 2016.

[13] R. E. Stake, *The art of case study research.* Thousand Oaks, CA: Sage, 1995.

[14] B. Flyvbjerg, "Five misunderstandings about case study research," in *Qualitative Inquiry*, vol. 12, no. 2, pp. 219-245, 2006.

[15] A. B. Powell, J. M. Francisco, and C. A. Maher, "An analytical model for studying the development of learners' mathematical ideas and reasoning using videotape data," *J. Mathematical Behavior,* vol. 22, no. 4, pp. 405-435, 2003.

[16] Lye, Sze Yee, and Joyce Hwee Ling Koh. "Review on teaching and learning of computational thinking through programming: What is next for K-12?." *Computers in Human Behavior, 41* (2014): 51-61.

[17] Q. Burke. "The markings of a new pencil: Introducing programming-as-writing in the middle school classroom." *Journal of Media Literacy Education* 4, no. 2 (2012): 121-135.

[18] D. Weintrop, A. Hansen, D. Harlow and D. Franklin, "Bringing computer science into elementary school classrooms." Presented at American Educational Research Association, 2018.

[19] National Research Council, "Committee for Workshops on Computational Thinking: Report of a workshop of pedagogical aspects of computational thinking," National Academies Press, 2011.

[20] S. Grover and R. Pea. "Computational thinking in K–12: A review of the state of the field." *Educational Researcher* 42, no. 1 (2013): 38-43.

[21] NGSS Lead States.. Next Generation Science Standards: For states, by states. National Academies Press. 2013

[22] H. Ehsan, and M. E. Cardella, "Capturing the Computational Thinking of Families with Young Children in Out-of-School Environments," in *124th ASEE Annual Conference and Exposition,* 2017.

[23] A. Dasgupta, A. Rynearson, Ş. Purzer, H. Ehsan, and M. Cardella, "Computational thinking in Kindergarten: Evidence from student artifacts". in *124th ASEE Annual Conference and Exposition,* 2017.