Basic Code Understanding Challenges for Elementary School Children

Olivia M. Nche
School of Computing
Clemson University
oncheey@g.clemson.edu

Raquel Boulware
Business Development
Claflin University
rboulware@claflin.edu

S. Megan Che
College of Education
Clemson University
sche@clemson.edu

Eileen T. Kraemer
School of Computing
Clemson University
etkraem@clemson.edu

Murali Sitaraman
School of Computing
Clemson University
msitara@clemson.edu

Victor B. Zordan
School of Computing
Clemson University
vbz@clemson.edu

***Abstract*— We describe a research study aimed at understanding the basic code reasoning challenges of elementary school children. The research targets third to fifth grade African American students as a step towards making computer science accessible to all children. The study was conducted in a summer camp with 40 students and replicated with 20 new students the following summer. Also participating in the second summer camp were 19 returning students. For data collection, the study uses code-tracing activities involving concepts such as variables, assignments, operators, and sequencing. Performance data is automatically collected in the background as children engage in the activities incorporated in a video game and also through think-aloud sessions. Results include common code understanding challenges for all children.**

Keywords*— Broadening participation, K-12 education, coding, game, summer camp, coding concepts

I. Introduction and Related Work

Several efforts have been directed towards strengthening the computing pipeline, yet under-representation remains a challenge. As a result, endeavors to include traditionally underrepresented groups play a critical role in broadening participation [1]. In this research, we target African American children not only to include them in the quest to gain computing knowledge, but also to understand how they reason about fundamental CS concepts and what their challenges are. Findings from prior studies indicate that most beginning students struggle with understanding fundamental CS concepts [2]. It has been argued that if students cannot read and understand small fragments of code, it is highly unlikely that they can write comparable pieces of code [3]. Furthermore, if students are grounded in fundamental concepts, they are more likely to succeed even if computing paradigms change [4]. In this study, we explore challenges related to fundamental coding concepts and tracing snippets of pseudo-code.

Several studies target broadening participation [1] with the goal of including as many students as possible in computing education. These endeavors range from direct interventions in classrooms to extracurricular settings like summer camps, workshops and roadshows. Studies show that summer camps can be beneficial in engaging underrepresented groups [5]. Kumar [3] conducted a study to investigate whether solving code tracing problems could help improve students' code writing skills. He observed a statistically significant improvement in code writing skills when the students first engaged in code tracing problems. Another study found that practicing line-by-line tracing helped improved tracing skills for students [6]. We incorporate code tracing activities in a game, because games are fun and engaging, and have been shown to be beneficial in learning [7].

II. Research Experiment Summary

In this study, we aim to answer the following question: What are the basic code understanding obstacles that elementary students face?

The study was first done in the summer of 2018 with about 40 students from the lowcountry of coastal South Carolina. The participants were about evenly distributed between males and females. The students were divided into two groups that met twice a week for two hours for a total of four contact hours per week per cohort. In the summer of 2019, another set of 39 students registered to attend the summer camp of whom 19 were returning students. We divided the entire population into two cohorts with all returning students in one group and the new students in another. This was done to help us tailor our curriculum accordingly.

This curriculum was tailored around a detailed taxonomy which we conceived [8], based on the work of Rich, et al., which presents K-8 learning trajectories [9]. The taxonomy identifies individual coding concepts and is rationalized as follows. Code containing a single assignment statement is

978-1-5386-5541-2/18/$31.00 ©2020 IEEE

introduced first. Then the concept of sequencing of assignment statements is presented. After that, arithmetic operators and Boolean operators are introduced.

This study was facilitated by a custom built video game which we designed [10]. Following the lessons in class, students played this video game. There were a total of three modules for both the curriculum and the game. Module 1 was dedicated to assignments, sequence, variables (types) and operators. Module 2 focused on conditionals, and module 3 on loops. This study reports results based on the data from the first module only. Each module in the game contained 5 levels and each level had 5 questions that focused on the specific concepts. The questions were designed to focus on the specific concept that each level targets. The questions per level were variations the same concept.

The think-alouds were different from standard think-alouds in that the students were prompted to explain the reasoning behind their answer choices as they played a version of the game. This game version had only five questions which combined multiple concepts.

III. RESULTS

A. Quantitative analysis of basic coding challenges

Analysis of the game data revealed different mean scores for different concepts. Sequencing of assignments and operators were the concepts with the lowest mean scores.

B. Qualitative analysis of basic coding challenges

The think-aloud studies also revealed that students struggled with operators and sequence of assignments. We illustrate their reasoning with examples:

Correct thought process (about a sequence of assignments). Amongst the students who answered the sequence question correctly, some explained that "the second statement is the only (one) to consider." This shows that these students have grasped the concept of order to some extent. Others explained that "when you put a value in, if you put a different one in, the first one is gone."

Incorrect thought process. Some students who did not understand sequencing of assignments solved a nonexistent problem, for example, by assuming that the "assignment statements are cumulative." For the question shown below, a student just added or subtracted everything in both assignments in some order, and picked an answer.

book_box $\longleftarrow 6 + 2$;

book_box $\longleftarrow 5 - 1$;

How many books are in the book_box?

A. 8 B. 5 C. 6 D. 4 E. Don't know

Another student used the process of elimination of choices and mentioned "round." When a wrong answer choice which a student expected was not available, the student decided to find a choice that was close to their expected answer.

Correct thought process, wrong answer. Some students thought about the problem correctly, stating that the second sequence statement is the correct one to focus but choose a wrong answer. The problem had to do with arithmetic mistakes, but not with understanding a sequence of assignments. For example, for the above question, they picked 6 as an answer choice because they evaluated 5 - 1 incorrectly as 5 + 1. We noticed this trend in later conditional questions where the student evaluates the condition correctly but pick the wrong answer because of arithmetic mistakes.

Finally, in the think-alouds, students also had difficulty with relational operators and Boolean operators.

IV. CONCLUSIONS AND FUTURE DIRECTIONS

This research focused on understanding basic code reasoning challenges for elementary African American school children by engaging them in fundamental concepts in a lecture format and providing an avenue for practice and assessment via a video game. A key goal in the study was to pinpoint potential challenges that children might face. We found that students struggled with sequencing of assignments and all operators. A continuation of this work will be to carry out similar studies to pinpoint challenges that middle and high school students might face with computing concepts.

ACKNOWLEDGMENTS

This work was funded in part by a grant from Boeing SC and by NSF grant 1738760.

REFERENCES

[1] J. Margolis, J. J. Ryoo, C. D. Sandoval, C. Lee, J. Goode, and G. Chapman, "Beyond access: Broadening participation in high school computer science," *ACM Inroads*, vol. 3, no. 4, pp. 72–78, 2012.

[2] M. Piteira and C. Costa, "Learning computer programming: study of difficulties in learning programming," in *Proceedings of the 2013 International Conference on Information Systems and Design of Communication*, pp. 75–80, ACM, 2013.

[3] A. N. Kumar, "A study of the influence of code-tracing problems on code-writing skills," in *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*, pp. 183–188, ACM, 2013.

[4] A. Schwill *et al.*, "Fundamental ideas of computer science," *Bulletin-European Association for Theoretical Computer Science*, vol. 53, pp. 274–274, 1994.

[5] K. Sung, K. Gourd, A. McMahon, K. Mansoor, and R. Gaggero, "A collaborative course for learning how to teach summer java coding camps," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pp. 515–520, ACM, 2018.

[6] S. Erete, N. Pinkard, C. K. Martin, and J. Sandherr, "Exploring the use of interactive narratives to engage inner-city girls in computational activities," in *2016 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, pp. 1–4, IEEE, 2016.

[7] D. R. Michael and S. L. Chen, *Serious games: Games that educate, train, and inform*. Muska & Lipman/Premier-Trade, 2005.

[8] O. M. Nche, E. Colbert-Busch, M. Sitaraman, and V. B. Zordan, "Presenting cs concepts through multiple representations to engage african-american elementary school children,"

[9] K. M. Rich, C. Strickland, T. A. Binkowski, C. Moran, and D. Franklin, "K-8 learning trajectories derived from research literature: Sequence, repetition, conditionals," in *Proceedings of the 2017 ACM conference on international computing education research*, pp. 182–190, ACM, 2017.

[10] O. M. Nche, J. Welter, S. M. Che, E. T. Kraemer, M. Sitaraman, and V. B. Zordan, "Codetracesure: Combining gaming, cs concepts, and pedagogy," in *RESPECT 2019*, IEEE, 2019.