Understanding students' pre-existing computational thinking skills and its relationship with their block programming performance

Shenghua Zha, Ph.D.,
College of Education and Professional Studies, University of South Alabama, Mobile, AL 36608

Joe'l Billingsley, Ph.D.,
College of Education and Professional Studies, University of South Alabama, Mobile, AL 36608

Index Words:
computational thinking, block programming, prior experience, learner analysis

Introduction

Computational thinking (CT) skills have been specified in many state standards of computer science (CS) education (*Promote computer science*, 2019). They are a set of skills utilizing abstraction, algorithmic thinking, decomposition, pattern generalization, and evaluation to solve problems (Selby & Woollard, 2013). Educators and researchers have proposed that CT is not only used in CS-related fields but also frequently applied in other areas even in our daily lives (Barr et al., 2011; Wing, 2006, 2011).

As a result, investigations have been conducted in recent years regarding CT skill development without computers. Researchers deployed non-programming activities in K-12 classrooms (Ballard & Haroldson, 2021; Lambert & Guiffre, 2009; Thomas et al., 2019). Those activities include games, simulations, picture book reading, or story creation. Results of those studies showed a significant improvement of students' CS/CT skills, especially their basic coding skills, such as the definition of variables and use of simple loops and conditionals (Ballard & Haroldson, 2021; Grover et al., 2019; Lambert & Guiffre, 2009).

These findings not only supported the efficacy of non-programming activities in improving young students' CT skills but also encouraged us to further explore young students' development of CT skills prior to coding or robotics programs.

This study was a part of the learner analysis in the design of a robotics program. In this study, we examined the CT skills of middle school students from an underrepresented minority group. Our research questions were: 1) did the students' CT skills in solving gaming problems predict their performance of block programming? 2) What other factors, such as gender and prior coding or robotics learning experiences, contributed to students' performance of block programming?

Methods

Participants

Forty-eight students from three classes at a local middle school participated in this study. A convenient sampling method was used as the participating students were from the classes where the instructors collaborated with the authors in the robotics program. 42% were girls, and 56% were boys. 81% self-identified themselves as African Americans. 4% were Latinos and Caucasian, respectively. 67% were in Grade 8, and the rest were from Grade 7.

Instrument

We collected and analyzed data from a survey and a quiz, which were delivered at one time in a class session. The survey included questions asking students demographic information. The quiz included 16 questions from the CTt instrument validated by Román-González et al. (Román-González et al., 2016; Román-González et al., 2018). Among them, eight measured students' algorithmic thinking and pattern recognition skills in gaming problems, and 8 measured the same skills in block programming problems. Students had 40 minutes to complete the survey and quiz.

Analysis & Results

We conducted a multiple regression to identify the predictability of students' CT skills in solving non-programming problems, their prior coding and robotics learning experiences, and their gender on their CT skills in solving block programming problems.

The dependent variable was students' CT skills in solving block programming problems. It was the average of 8 quiz questions measuring CT skills in solving block programming programs. Students' CT skills in solving non-programming problems, their prior coding and robotics learning experiences, and their gender were the predictors. Among them, students' CT skills in solving non-programming problems were the average of 8 quiz questions measuring students' CT skills in solving non-programming problems. Students' prior coding and robotics experiences were dummy coded, with 1 meaning Yes and 0 meaning No. Students' gender was coded as 1 being boys and 2 being girls. Results of the descriptive statistics of these variables are shown in Table 1.

Table 1.
Descriptive Statistics

	Mean	*SD*	*N*
CT in Block Programming Problems	1.11	1.14	48
CT in Non-programming Problems	2.57	1.14	48
Prior Programming Experience	.17	.38	48
Prior Robotics Experience	.17	.38	48
Gender	.65	.48	48

Results of the multiple regression showed that students' CT skills in solving non-programming problems and their prior robotics learning experience significantly predicted and explained 44% variation of their CT skills in solving block programming problems ($F(4,43)=8.52$, $p<.01$).

$$CT_{block\text{-}programming}=.45*CT_{non\text{-}programming}+.91*Experience_{robotics}$$

We did not find a significant impact of students' prior programming experience and gender on their CT skills in solving block programming problems.

Discussions

Results of our study showed a significant relationship between students' CT skills in solving gaming and block programming problems. From the instructional design perspective, it supports the use of non-programming methods to improve students' CT skills. At the same time, it suggests that students may have developed some CT skills prior to any coding or robotics programs. Those skills may be developed from their interactions with real-world problems or in prior learning experiences. Therefore, it will benefit students' understanding of coding if we activate their prior knowledge and skills and use the real work examples or their prior knowledge to explain the coding algorithms.

We also found that students' prior robotics learning experience had a significant impact on their CT skills in solving block-programming experiences, but their prior programming learning experience did not. We think it may be because robotics provides tangible and visual aids for students at young ages to develop an abstract understanding of CT skills. However, studies with a large sample size are needed to verify this finding.

African American students have been reported as underrepresented in secondary CS education (Code.org, 2020). In addition, the gender disparity within the underrepresented minority group was found to aggravate this inequity (*Current perspectives and continuing challenges in cs education for u.S. K-12 schools 2020 report*, 2020). Results of our post-hoc analysis did not show a gender difference in students' CT skills in solving gaming problems when their prior robotics learning experience was taken into account. It could be because only boys were found to have prior robotics experiences. This finding endorses the results in the national report. In addition, it underscores the necessity to broaden participation for underrepresented minority students, especially the female students from that group. We suggest future studies consider the impact of students' prior experiences when examining the gender difference.

References

Ballard, E. D., & Haroldson, R. (2021). Analysis of computational thinking in children's literature for k-6 students: Literature as a non-programming unplugged resource. *Journal of Educational Computing Research*. https://doi.org/10.1177/07356331211004048

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology, 38*(5), 20-23.

Code.org, C., & ECEP Alliance. (2020). *2020 state of computer science education: Illuminating disparities*. https://advocacy.code.org/stateofcs

Current perspectives and continuing challenges in cs education for u.S. K-12 schools 2020 report. (2020). Gallup & Google. https://edu.google.com/latest-news/research/?modal_active=none

Grover, S., Jackiw, N., & Lundh, P. (2019). Concepts before coding: Non-programming interactives to advance learning of introductory programming concepts in middle school. *Computer Science Education, 29*(2-3), 106-135. https://doi.org/10.1080/08993408.2019.1568955

Lambert, L., & Guiffre, H. (2009). Computer science outreach in an elementary school. *Journal of Computing Sciences in Colleges, 24*(3), 118–124.

Promote computer science. (2019). https://code.org/promote

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2016). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior, 72*, 678-691. https://doi.org/10.1016/j.chb.2016.08.047

Román-González, M., Pérez-González, J.-C., Moreno-León, J., & Robles, G. (2018). Can computational talent be detected? Predictive validity of the computational thinking test. *International Journal of Child-Computer Interaction, 18*, 47-58. https://doi.org/10.1016/j.ijcci.2018.06.004

Selby, C. C., & Woollard, J. (2013, June). Computational thinking: The developing definition. 19th Annual Conference on Innovation and Technology in Computer Science Education, Canterbury, Great Britain.

Thomas, D. K., Milenkovic, L., & Marousky, A. (2019, February). Simulating a non-native invasion. *Science & Children*, 37-45.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35. https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2011). Research notebook: Computational thinking—what and why? *The Link Magazine*, *Spring*. https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Acknowledgement

This study is funded by the Gulf Coast RC&D Council, University of South Alabama Social Justice Research Initiative, and College of Education and Professional Studies at the University of South Alabama. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.