Enhancing Computer Science Education for K-2 Students: Insights from a Randomized Controlled Trial

Abstract

In this study, we conducted a randomized controlled trial to examine the effectiveness of a computer science (CS) curriculum, Coding as Another Language-ScratchJr (CAL-ScratchJr) on the programming and computational thinking skills for students from kindergarten to second grade classrooms. Using multilevel modeling regression analysis with a sample of 1057 students, we found that the CAL-ScratchJr curriculum was effective in improving students' programming skills but no significant differences were detected for students' computational thinking skills. These findings shed light on the educational efficacy of CAL-ScratchJr as a promising CS curriculum for young children. Implications on policy and future research were also discussed in the paper.

1. Introduction

In response to the demand for computer and information technology professionals in the digital economy (Wurman & Donovan, 2020), educators, parents, researchers, and policymakers have placed a greater emphasis on computer science (CS) education in the K-12 education system. Indeed, CS education can provide students with critical thinking, problem-solving, and creativity skills which are not only valuable in the tech industry but also in various other fields (Papert, 1990; Wing, 2006). The increasing availability of CS education in K-12 schools, provides students with the opportunity to explore potential career paths, prepare them for higher education, and equip them with the skills required to succeed in the 21st century.

However, most of the current CS programs are designed for older children, such as middle school students and above, disregarding the fact that early childhood is a critical period for children's development, as it is during this time that they acquire foundational learning skills, cognitive abilities, and social-emotional skills (Friedman-Krauss et al., 2021; Ruhm & Waldfogel, 2012). Research shows that well-designed early childhood educational programs and interventions can have both short-term and long-term positive impacts on children's social competence, intellectual and academic performance (Friedman-Krauss et al., 2021).

The K-12 Computer Science Framework (2016) recommends that by the end of second grade, students should have a basic understanding of key concepts such as software, algorithms, variables, control structures, and modularity, as well as skills such as debugging. However, the lack of high-quality and developmentally appropriate CS programs for early childhood education makes it challenging for teachers to effectively integrate CS into their existing curriculum, and for students to acquire the necessary CS and CT skills by the end of second grade. This in turn,

hinders their ability to be better prepared for more advanced CS curricula in later stages of their education.

"Coding as Another Language - ScratchJr" (CAL-ScratchJr) is a CS program for K-2 students developed by the DevTech Research Group at Boston College (Bers et al, 2023). This curriculum introduces the powerful ideas from CS integrated with conversations in literacy in a developmentally appropriate, structured, creative and playful way (Bers, 2019). The CAL-ScratchJr curriculum includes 24 lessons with unplugged and plugged activities using the most popular programming language ScratchJr among young children across the world (Unahalekhaka & Bers, 2021). Although early evidence indicates the promise of this curriculum (Bers et al., under review), there are no experimental research studies that systematically examine the educational efficacy of this program on the coding and CT skills for children from kindergarten to second grade.

To this end, the current study will fill the gaps in the literature by examining the effectiveness of the CAL-ScratchJr curriculum in children's programming and CT skills through a randomized controlled trial. The following research questions guided our study design and data analysis:

1. What is the impact of the CAL-ScratchJr curriculum on children's coding skills?
2. What is the impact of the CAL-ScratchJr curriculum on children's CT skills?

2. Methods

2.1 Study procedures

To evaluate the impact of the CAL-ScratchJr curriculum on students' coding skills and CT skills, a randomized controlled trial with a delayed treatment was conducted among 26 public

elementary schools in two school districts on the east coast of the United States. The CAL-ScratchJr curriculum was developed for kindergarten, first and second grade students so only these three grade levels were included in the study. At the beginning of the study, the participating schools followed a randomization process and were assigned into either the treatment group where the curriculum was implemented immediately or the control group where the curriculum was implemented in the next school year. During the intervention, students in the treatment group received CAL-ScratchJr curriculum instructions while the students in the control group were business-as-usual. All the teachers in the treatment group participated in a 4-hour professional development (PD) training hosted by an experienced trainer.

2.3 Participants

In total, the analysis sample included 861 students from site A and 196 students from site B. These included 237 students in kindergarten, 348 students in the first grade and 472 students in the second grade. Detailed student demographic information by the treatment and control groups can be found in Table 2.

2.4 Measures

2.4.1 Coding Stages Assessment

The Coding Stages Assessment (CSA) is a validated assessment (de Ruiter & Bers, 2021) to evaluate the coding skills using the ScratchJr programming language of children from kindergarten to lower elementary grades. During the assessment, the child is prompted with a series of questions and they either answer verbally or solve a coding task using ScratchJr. The split-half reliability for CSA is Guttman's Lambda 6 = .94 (de Ruiter & Bers, 2021).

2.4.2 TechCheck

As the measure of CT in this study, TechCheck has been validated and used among kindergarten, first, and second graders (Relkin et al., 2020). TechCheck assessment includes 15 multiple-choice questions and is known as an "unplugged" assessment because it tests computational thinking skills, but does not require technology or computer programming knowledge to complete. It has a reliability of $\alpha = 0.68$ (Relkin et al., 2020).

3. Results

Table 3 summarizes the descriptive statistics of the CSA and TechCheck pre and post scores for the students in the treatment and control groups.

Baseline equivalence was established by examining whether pre-test scores were significantly different between the treatment and control group controlling for the school level data. Our results showed that the CSA scores did not differ significantly between the treatment group and control group ($\beta = .81$, $SE = .59$, $p = .18$), although descriptive statistics indicate that the treatment group in general scored .3 higher than the control group. Similarly, we did not detect significant between group differences for the TechCheck pre-test scores ($\beta = .01$, $SE = .30$, $p = .98$).

As for the impact of the CAL curriculum, we conducted multilevel regression analysis. As shown in Table 4, whether the students received the CAL curriculum (CSA Model 1) significantly impacted their CSA post-test ($\beta = 4.68$, $SE = .68$, $p < .001$) controlling for their grade level and CSA pre-test scores. However, we did not find any significant effect of the intervention condition on students' TechCheck post-test scores (TC Model 1, $\beta = -.08$, $SE = .27$, $p = .78$).

4. Discussion

While educators and researchers are paying attention to CS education, most of the existing CS programs are designed for older children and teenagers and high quality CS curricula for young children is lacking (Bers, 2019). Our study is the first experimental study that provides evidence that the CAL-ScratchJr curriculum was effective in improving the programming skills for students from kindergarten to second grade. And the finding is consistent with the positive findings from the previous pilot studies on the CAL-ScratchJr related curriculum (Bers et al., 2022). It is worth noting that not all classrooms completed the whole curriculum due to the constraints posted by COVID-19, meaning that the effect sizes of the CAL-ScratchJr curriculum generated in the current study might be smaller than the real effect size if all 24 lessons in the curriculum are implemented. Future research is warranted to replicate the results and further investigate the effect sizes of the CAL-ScratchJr when all lessons are implemented.

In addition, a larger effect size on the CSA was found for students in the first and second grades compared with the kindergarteners. This is probably due to the fact that the CAL-ScratchJr curriculum is adaptive to grade levels, meaning that the curriculum for first and second grades is designed to be more challenging than that for the kindergarteners. For example, the most advanced ScratchJr skill introduced in the kindergarten curriculum is *adjusting parameters*, while in second grade, students also learned more advanced skills such as *parallel programming, next repeat loops and sending messages using multiple colors*. Another possible but less likely reason is that students in elementary school are more cognitively and mathematically ready to learn coding. Evidence suggests that early childhood is a key developmental period for children where their cognitive abilities develop at a tremendous speed (Campbell et al., 2001) and educational investment in early childhood education, particularly

pre-k and kindergarten programs, such as the Head Start, has the highest economic value compared with programs at higher grade levels (Garces et al., 2002). However, most of these studies only focused on language arts and cognitive capability benchmarks. Coding is more cognitively demanding compared to most of the early language arts instructions and it also relies on math skills such as counting, addition, subtraction and multiplications, which may not have been learned until elementary grades. Future studies need to be conducted to examine in further detail about how children in different grade levels respond to the curriculum in the classroom to better guide future curriculum design and policy.

On the other side, we did not find any significant differences in students' CT scores. This null finding could be due to several reasons. First, compared with coding and programming, CT is higher-level cognitive and meta-cognitive skills, which may take longer time with more exposure for the children to develop and grow (García Peñalvo et al., 2016). Although various aspects of CT such as abstract thinking, logical reasoning, and problem solving could be developed during learning programming, it does not necessarily mean that children being more proficient in coding can always be successfully transferred to significant improvements in their CT skills. The CAL-ScratchJr curriculum was designed to have 24 lessons, 45 minutes per lesson. However, this study was carried out during COVID-19 under many constraints and due to the tight schedule of some schools, not all participating teachers completed the 24 lessons using 45-min blocks. According to the lesson logs and interviews from the teachers, some of the classrooms, especially those from site B, only implemented less than half of the lessons before the post-tests. This level of intervention intensity may be enough to improve children's coding skills using the ScratchJr app, but may not be sufficient for the students to develop higher level cognitive skills that are related to CT.

The second explanation to the null finding is that the control group in our study was business-as-usual. This means that although these students did not receive the CAL-ScratchJr curriculum, they may have other unplugged CS curricula that were implemented regularly in their school and the amount of class time for either the CAL-ScratchJr curriculum or other CS instructions may be the same. Learning to code using ScratchJr is not the only approach where young children can develop their CT skills after all. It is possible that while the treatment group received the CAL-ScratchJr curriculum, students in the control group were exposed to other CS activities which were also helpful for developing their CT abilities.

6. Conclusion

The current study conducted a randomized controlled trial to examine the efficacy of a CS curriculum CAL-ScratchJr on the programming and CT skills for students in kindergarten, first and second grade classrooms. Our findings revealed that the CAL-ScratchJr curriculum was effective in improving children's programming skills using the ScratchJr coding language. However, we did not find a statistically significant difference in students' CT skills between the treatment and the control groups. These findings shed light on the exciting potential of the CAL-ScratchJr curriculum as an effective and developmental appropriate CS curriculum for children in kindergarten to second grade classrooms in improving their programming skills. Future research with larger sample sizes and higher fidelity of implementation is warranted to replicate the findings and provide more evidence regarding the impact of the CAL-ScratchJr curriculum on students' CT skills. Study designs where the control group receives a non CS related curriculum are also encouraged to better understand the efficacy of the current curriculum.

References

Bers, M. U., Blake-West , J., Kapoor, M. G., Levinson, T., Relkin, E., Unahalekhaka, A., & Yang, Z. (2023). Coding as another language: Research-based curriculum for early childhood computer science. *Early Childhood Research Quarterly*, 64, 394–404. https://doi.org/https://doi.org/10.1016/j.ecresq.2023.05.002

Bers, M. U., Govind, M., & Relkin, E. (2022). Coding as another language: computational thinking, robotics and literacy in first and second grade. In *Computational Thinking in PreK-5: Empirical Evidence for Integration and Future Directions* (pp. 30-38).

Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, *6*(4), 499-528.

Bers, M., Blake-West, J., Govind, M., Levinson, T., Relkin, E., Unahalekhaka, A., & Yang, Z. (under review). Coding as Another Language: Research-Based Curriculum for Early Childhood Computer Science. Early Childhood Research Quarterly.

Campbell, F. A., Pungello, E. P., Miller-Johnson, S., Burchinal, M., & Ramey, C. T. (2001). The development of cognitive and academic abilities: growth curves from an early childhood educational experiment. *Developmental psychology*, 37(2), 231.

de Ruiter, L. E. & Bers, M. U. (2021). The Coding Stages Assessment: development and validation of an instrument for assessing young children's proficiency in the ScratchJr programming language. Computer Science Education. DOI: 10.1080/08993408.2021.1956216

Friedman-Krauss, A. H., Barnett, W. S., Garver, K. A., Hodges, K. S., Weisenfeld, G. G. & Gardiner, B. A. (2021). The State of Preschool 2020: State Preschool Yearbook. New Brunswick, NJ: National Institute for Early Education Research.

Garces, E., Thomas, D., & Currie, J. (2002). Longer-term effects of Head Start. *American economic review*, *92*(4), 999-1012.

García Peñalvo, F. J., Reimann, D., Tuul, M., Rees, A., & Jormanainen, I. (2016). An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers.

K–12 Computer Science Framework. (2016). Retrieved from http://www.k12cs.org.

Papert, S. (1990). Children, computers and powerful ideas. *New York: Basic Books*, *10*, 1095592.

Relkin, E., de Ruiter, L., & Bers, M. U. (2020). TechCheck: Development and validation of an unplugged assessment of computational thinking in early childhood education. *Journal of Science Education and Technology*, *29*(4), 482-498.

Ruhm, C., & Waldfogel, J. (2012). Long-term effects of early childhood care and education. *Nordic Economic Policy Review*, *1*(1), 23-51.

Unahalekhaka, A., & Bers, M. U. (2021). Taking coding home: Analysis of ScratchJr usage in home and school settings. *Educational Technology Research and Development*. https://doi.org/10.1007/s11423-021-10011-w

Wing, J. M. (2006). "Computational thinking and thinking about computing". Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 364(1844), 1171–1178.

Wurman, Z. E., & Donovan, W. (2020). Breaking the Code: The State of Computer Science Education in America's Public Schools. White Paper No. 206. *Pioneer Institute for Public Policy Research*.

Table 1. Demographic information for the students in the treatment and control group.

	Treatment (543)		Control (508)	
	N	Percentage	N	Percentage
Site B	125	23.02%	66	12.99%
Site A	418	76.98%	442	87.01%
Gender (Female)	279	51.38%	264	51.97%
Ethnicity				
White	289	53.22%	313	61.61%
Hispanic	102	18.78%	77	15.16%
African American	37	6.81%	57	11.22%
Asian	29	5.34%	16	3.15%
Other	86	15.84%	45	8.86%
Grade Level				
Kindergarten	141	25.97%	96	18.90%
Grade 1	152	27.99%	195	38.39%
Grade 2	250	46.04%	217	42.72%
Disability	61	11.23%	63	12.40%
Limited English Proficiency	77	14.18%	50	9.84%
Low SES	130	23.94%	118	23.23%

Table 2. Descriptive statistics of the assessment scores.

	Treatment (549)				Control (508)			
	Mean	SD	Min	Max	Mean	SD	Min	Max
CSA Pre-test	3.77	2.89	0.00	26.00	3.51	2.19	0.00	13.90
CSA Post-test	11.42	6.57	1.10	39.00	5.58	3.28	0.00	26.00
CSA Gain	7.65	5.89	-5.80	32.10	2.07	2.66	-3.40	15.20
TechCheck Pre-test	7.29	2.52	1.00	15.00	7.32	2.41	0.00	14.00
TechCheck Post-test	8.70	2.68	1.00	15.00	8.71	2.51	2.00	15.00
TechCheck Gain	1.41	2.49	-5.00	7.00	1.38	2.40	-5.00	7.00

Table 3. Correlation matrix table

	CSA Pre-test	CSA Post-test	TC Pre-test	TC Post-test	Grade	Condition
CSA Pre-test	1.00					
CSA Post-test	0.44***	1.00				
TC Pre-test	0.24***	0.24***	1.00			
TC Post-test	0.23***	0.30***	0.54***	1.00		
Grade	0.44***	0.26***	0.03	0.04	1.00	
Condition	0.05	0.49***	-0.01	0.00	0.02	1.00

*Note: * $p < .05$, ** $p < .01$, *** $p < .001$.*

Table 4. Multilevel modeling regression analysis results for the CSA and TechCheck (TC) post-tests.

	CSA Unconditional Model		CSA Model 1		TC Unconditional Model		TC Model 1	
Fixed-effect								
	Coefficient	SE	Coefficient	SE	Coefficient	SE	Coefficient	SE
Intercept	8.57***	0.62	0.92	0.56	8.47***	0.18	4.45***	0.30
Grade			1.13***	0.21			0.13	0.09
Condition			4.68***	0.68			-0.08	0.27
CSA pre-test			0.93***	0.06			0.54***	0.03
Random-effects parameters								
	Variance	SE	Variance	SE	Variance	SE	Variance	SE
Intercept	8.59	2.72	2.11	0.95	0.57	0.23	0.28	0.13
Residual	26.61	1.17	19.68	0.87	6.22	0.27	4.60	0.20
ICC	0.24		0.10		0.08		0.06	
N	1051		1051		1051		1051	

*Note: * $p < .05$, ** $p < .01$, *** $p < .001$.*