RESEARCH ARTICLE

Developing Computational Thinking in Early Childhood Education: Long-Term Impacts on CT Skills and Motivation Using the CAL Approach, ScratchJr, and Gamification

Alkinoos Ioannis Zourmpakis

Department of Special Education, University of Thessaly, Volos 38221, Greece

Check for updates

Correspondence to: Alkinoos Ioannis Zourmpakis, Department of Special Education, University of Thessaly, Volos 38221, Greece;
Email: alzourmpakis@uth.gr

Received: June 20, 2025;
Accepted: August 7, 2025;
Published: August 13, 2025.

Citation: Zourmpakis, A. I. (2025). Developing Computational Thinking in Early Childhood Education: Long-Term Impacts on CT Skills and Motivation Using the CAL Approach, ScratchJr, and Gamification. *Advances in Mobile Learning Educational Research*, 5(2), 1536-1547.
https://doi.org/10.25082/AMLER.2025.02.009

Abstract: Computational Thinking (CT) has been slowly being integrated into early education curricula as a critical skill for 21st-century learners. However, implementation of fully developed curricula devoted to CT development and the corresponding motivational aspects for young learners, particularly when using pedagogical strategies like gamification, are scarce, especially when it comes to their long-term effects. This study investigates the long-term impact of the "Coding as Another Language" (CAL) with ScratchJr and integrated gamification elements through the ClassDojo platform on the CT skills and motivation in early childhood education. In this study, we employed a quantitative, semi-experimental approach measuring CT skills utilizing a pre-test and post-test approach and a brief summative assessment test. Also, a motivational questionnaire was utilized post-intervention. The sample consisted of 12 second-grade students over an entire school year. The findings revealed a statistically significant improvement in students' CT development. Furthermore, students reported significant high levels of self-efficacy, grade, self-determination, and intrinsic motivation suggesting that the gamified, project-based approach successfully fostered sustained engagement and confidence in a collaborative environment. This research contributes valuable insights into the successful implementation of long-term, gamified coding programs for young children, demonstrating that such approaches can significantly enhance both cognitive skills and key motivational aspects.

Keywords: computational thinking, early childhood education, scratchJr, gamification, motivational aspects

1 Introduction

Computational Thinking (CT) is a fundamental problem-solving skill that is gradually considered an essential skill for students in the 21st century in order to be able to participate and navigate in an increasingly digital society (Zhang & Nouri, 2019; Wing, 2006; Guggemos, 2024; Ye et al., 2022). Recognized globally as a foundational core competence on par with reading, writing, and arithmetic, many countries have taken the initiative and integrated CT into their primary and secondary education curricula, such as Greece, Estonia, Australia, New Zealand, and the United States (Dufva & Dufva, 2016; Zhang & Nouri, 2019). Early engagement with CT practices, even in preschool, is vital and can have a significant impact on the development of basic skills in problem-solving (Kyza et al., 2021; Pila et al., 2019). What is more, CT skills can benefit students in equipping them and preparing them in STEM inquiry and problem-solving processes (Miller et al., 2020; NGSS, 2013), while at the same time reducing gender-based stereotypes in STEM fields (Bati, 2021; Videnovik et al., 2021).

CT has the potential to be effectively introduced as early as preschool and early elementary school, using developmentally appropriate tools (Kyza et al., 2021; Pila et al., 2019; Strawhacker et al., 2018). Especially for young learners, who may lack the developmental ability and readiness for the complexity of text-based programming, block-based environments are ideal (Bers & Resnick, 2015). ScratchJr is a tool that was created for this implicit purpose and age group. Unlike Scratch, which has literacy skills as a prerequisite, ScratchJr simplifies coding concepts, offering a low-barrier, visual programming environment that is aligned with their cognitive and linguistic abilities (Bers & Resnick, 2015; Sullivan et al., 2017), thus making it accessible to children as young as four years old (Bers, 2018; Papadakis, 2020).

Game-based learning and gamification can have a significant positive impact in education settings and be used as an effective pedagogical strategy (Papadakis & Kalogiannakis, 2024). According to studies, CT development can also benefit from gamified strategies, especially for younger learners, as they can leverage the motivational dynamics of gameplay, such as challenge, feedback, and interactivity, to foster engagement and increase students's motivation and sustained interest in problem-solving activities (Kazimoglu et al., 2012; Yunus & Zaibon, 2021; Carocca et al., 2024). This is a vital aspect in early childhood settings, where play is a natural medium for learning. However, few studies have explored the impact of game-based learning and gamification on CT development (Ma et al., 2023).

Moreover, even though there seems to be a growing interest in CT development, there is a noted scarcity of long-term research studies on the development of computational thinking and motivational development, particularly in early learners (Kyza et al., 2021). Consequently, this study investigates the development of CT skills and then motivational aspects of a small group of early education students (ages 7–8) using ScratchJr and following the Coding as Another Language (CAL) pedagogical approach and curriculum (Bers, 2019; Bers et al., 2023), while utilizing simple gamification practices to motivate students.

2 Literature review

2.1 Computational Thinking in Early Education

As the preschool years mark a transitional phase where children shift from concrete to abstract thinking, integrating computational thinking (CT) into early childhood education can be quite beneficial (Flannery & Bers, 2013; Piaget, 1973). As studies show, the development of vital life skills, such as problem-solving, planning, and decision-making, can be boosted by the early exposure to CT, even in early education (Bers & Horn, 2010; Bati, 2021). In addition, other important skills of that educational setting, *i.e.*, early literacy, numeracy, *etc.*, can also be enhanced (Sullivan & Strawhacker, 2021). However, CT can benefit not only technical skills but also socio-emotional and cognitive development, like peer collaboration, creativity, and persistence (Strawhacker et al., 2018; Papadakis & Kalogiannakis, 2024).

What is more, early exposure to CT development can also improve the learning of other important learning subjects, like STEM learning. Although few studies have researched CT development and STEM learning (Li et al., 2020), the Next Generation Science Standards (NGSS) has highlighted this strong connection between scientific inquiry and CT for more than a decade (NGSS Lead States, 2013). More precisely, CT development can improve children's capacity for scientific reasoning, such as formulating hypotheses, designing experiments, analyzing data, modeling phenomena, interpreting results, and solving complex problems (Basu et al., 2016; Wilensky, 2014; Tariq et al., 2024), and make them more eager to engage in STEM learning (Li & Oon, 2024).

The approach to teaching CT and the activities utilized are similarly of great importance and need to be carefully considered. There is a tendency in some pedagogical approaches to view coding as a new kind of core literacy, like reading and writing, that allows people to express themselves and communicate ideas (Bers, 2018; Vee, 2017). The primary focus is not to merely learn some technical skills but to be able to fluently express their ideas through producing projects (Relkin et al., 2021). Furthermore, even though CT activities in early education more commonly include unplugged activities (*e.g.*, physical games) and digital tools (*e.g.*, ScratchJr), digital environments tend to provide more authentic coding experiences (Sullivan & Bers, 2019). Game-based learning and gamification strategies, which align with some tools playful design, can further encourage experimentation and collaboration (Bers, 2012). Despite evidence that these methods may enhance CT and problem-solving skills (Cakir et al., 2021; Katchapakirin et al., 2022), research is scarce.

2.2 ScratchJr and computational thinking

Utilizing appropriate programming environments is essential for effectively incorporating CT into early education (Louka, 2023). ScratchJr stands out as a widely used, free visual programming platform designed specifically for young children aged 5-8 to begin learning the foundational concepts of CT and coding (Bers & Resnick, 2015; Louka, 2023).

In contrast to text-based languages, ScratchJr's graphical and block-based interface enables young learners to construct interactive stories and games by assembling blocks of code (Flannery & Bers, 2013). Basically, it is a "coding playground," aimed at experiential learning, creativity, and collaboration, while minimizing cognitive load, as it is aligned with the developmental appropriateness of young children (Bers, 2012; Papadakis, 2024). In this way young children are able to explore key programming concepts like sequencing, loops, and conditionals in a creative way and with ease (Bers, 2012; Papadakis, 2024).

Even though there is evidence in literature that supports ScratchJr's potential in early CT skill development (Louka, Kalogiannakis & Papadakis, 2024; Papadakis, 2024), some issues remain. Younger children, for example, may find flow control blocks difficult, while older ones can handle more complex programming (Strawhacker et al., 2018). Nonetheless, ScratchJr's movement-based programming seems to attract children's attention and motivate them to engage with it (Papadakis et al., 2024). This could be due to ScratchJr's open-ended, "low floor, high ceiling" design, which makes it accessible to beginners while still offering opportunities for advanced exploration (Bers & Resnick, 2015; Bers et al., 2022).

2.3 The Coding as Another Language (CAL) Approach

Effective CT education requires thoughtful instructional design (Liu et al., 2024). Coding as Another Language (CAL) curriculum is one such pedagogical approach that uses ScratchJr to merge computer science and playfulness with early literacy instruction (Bers, 2018). CAL's central idea is similar to the current trends of pedagogy around CT, where coding is a modern literacy and a way of communication and self-expression (Vee, 2017; Bers, 2019).

To be more specific, CAL is affected and is rooted in constructionism and the positive technological development (PTD) framework (Bers, 2012), emphasizing self-expression and creativity through coding (Bers, 2019). CAL follows a structured sequence of interdisciplinary lessons that include both "plugged" on-screen coding exercises and "unplugged" activities that teach computational concepts without the use of a device (Flannery et al., 2013; Bers et al., 2022). The approach works on the assumption that learning to code is a skill that requires explicit instruction rather than occurring in a natural way (Govender et al., 2014; Ribaux, 2024). Though there is evidence that the CAL approach is effective at linking coding and literacy, thereby nurturing both technical and socio-emotional development (Unahalekhaka & Bers, 2022), more research is required.

3 Methodology

This quantitative, semi-experimental study took place at the Experimental Primary School in Heraklion, Crete, Greece. Its purpose was to evaluate the development of second-grade students' Computational Thinking (CT) skills and motivation through their participation in an extracurricular teaching class for excellence, creativity, and innovation, an optional program offered beyond the standard school schedule which followed and implemented the "Coding as Another Language" (CAL) program. These classes are mainly carried out to support students' interests and talents across diverse areas of knowledge and skill. As such, the study employed a convenience sample consisting of 12 second-grade students who voluntarily joined the program. The sample was equally divided by male and female students.

The CAL curriculum, created by Professor Marina Umaschi Bers and the DevTech research group, consists of 24 lessons. The lessons were held weekly from October to June during which students worked with ScratchJr using either tablets or computers based on their preference. To further increase engagement and motivation, gamified elements were embedded into the learning process before the start of the teaching lessons. To be more specific, PBL gamification elements were utilized through the use of the ClassDojo platform, a gamification platform. That specific platform was chosen due to its ability to not only enhance learners' motivation and engagement, but also supporting classroom behavior management (Benzizoune, 2024). To be precise, students accumulated points or gotten badges when fulfilling objectives, cooperating or assisting each other. The points could be used for various rewards.

Two assessment tools were used to evaluate CT skill development. Initially, the students completed the TechCheck test before the first teaching lesson and again after the last one. This test included 15 multiple-choice questions that covered six CT domains, was suitable for young learners and it required no prior coding experience (Bers, 2018). In addition, at the end of the program in June, students also completed the "Show What You Know" (SWYK) assessment, a brief summative evaluation tool developed by the CAL research team (Bers et al., 2023). The SWYK assessment test had 10 multiple-choice questions, which students were rated. The DevTech team provided the instruments after having completed a comprehensive training and verification process to make sure they could be utilized correctly.

Furthermore, students completed another questionnaire at the end of the learning intervention, to measure their motivation. The questionnaire used Likert-scale questions and was based on the works of Glynn et al. (2009, 2011) and Salta & Koulougliotis (2015), and was adapted to CT learning. The questionnaire was reviewed and refined with the help of experts in the field to ensure content validity and contextual relevance. It included Likert-scale questions aimed at measuring students' motivational aspects, such as intrinsic motivation, self-efficacy, self-determination, grade motivation and career motivation. It should be noted that questions of each category were not grouped together but are separated.

This study followed all formal procedures, obtaining necessary approvals and ethical clearance from the Ethics Committee of the Department of Preschool Education at the University of Crete. Data analysis was performed using SPSS statistical software In addition, to strengthen the validity and suitability of the analyzed data and to ensure research integrity, we sought the assistance of two academic experts in computational thinking development in early education, who reviewed the data and results independently (Petousi & Sifaki, 2020).

4 Results

This study aimed to investigate how the use of the CAL program with ScratchJr and the utilization of a gamification application for a long-term period could affect students CT development, motivational aspects, and engagement. Table 1 shows the results of two CT measurement tests. Specifically, regarding the TechCheck test, students had a mean of 8.42 before the start of the learning intervention (SD = 2.678), whereas after its end in June, they displayed a mean of 12.50 (SD = 1.784). Based on the analysis, there was a statistically significant difference between the pre-test and post-test (M. diff = 4.08, SD = 2.151), showcasing a considerable enhancement of students' CT development. This can also be supported by the results of the SWYK test that also took place at the end of the CAL program. The results were quite positive, as in the 10-question test they had an average mean of 8.83 with a very low standard deviation (SD = 0.835), indicating not only that these students achieved high levels of CT development but also that they did so collectively, with very little variation in performance among them.

Table 1 Data Analysis of the CT measurement Tests

Types of Test	Mean	SD	Mean Difference	SD	Sig. Difference
Pre-test TechCheck	8.42	2.678	4.08	2.151	0.000
Post-test TechCheck	12.50	1.784			
SWYK	8.83	0.835	-	-	-

Regarding students' motivational aspects, the Science Motivation Questionnaire II (SMQ-II) (Glynn, 2011) was used, containing 25 items that are divided across five motivational dimensions, *i.e.*, intrinsic motivation, self-efficacy, self-determination, grade motivation, and career motivation. Regarding intrinsic motivation (Table 2), *i.e.*, enjoyment and inherent satisfaction derived from learning, students showed a significant boost as they showed to be very interested (Q3) (M = 4.08) and curious (Q4) (M = 4.17) in learning using ScratchJr, while also enjoying programming creatively (Q5) (M = 4.5) and having fun (Q2) (M = 4.17). There was a slight drop in the perceived usefulness and applicability of their learning (Q1) (M = 3.75), *i.e.*, a connection between their learning and its application in the real world. This is evident from the fact that nearly 40% neither agree nor disagree. However, the average mean is still quite high and above average, with none of the students disagreeing or absolutely disagreeing with that statement.

Table 3 presents the motivational levels regarding students' desire to achieve high marks or academic success (grade motivation). They generally considered the CAL program activities, with (Q9) (M = 4.08) or without (Q7) (M = 4.58) the use of the ScratchJr application, considerably important, and they were very eager to create challenging stories (Q8) (M = 3.92) and complete the project assignments (Q10) (M = 4.42). However, students were not entirely competitive with each other, as only 50% agreed or strongly agreed with wanting to do better than the others (Q6) (M = 3.08).

The 3rd category was self-efficacy, namely the belief that the students have of their own ability to effectively learn, understand, and be successful in their ScratchJr work and projects. Based on Table 4, students were quite confident of their knowledge of the command blocks (Q15) (M = 4.08), their ability to create games and stories in ScratchJr (Q14) (M = 3.92), and the projects assigned to them during the CAL program (Q13) (M = 4.17). Additionally, they were

Table 2 Intrinsic Motivation

Category	Questions	Absolutely Disagree (%)	Disagree (%)	Neither Agree nor Disagree (%)	Agree (%)	Strongly Agree (%)	Mean Average	Std Deviation
Intrinsic motivation	1. What I learn in ScratchJr helps me understand things in the world around me.	0	0	41.7	41.7	16.6	3.75	0.754
	2. It's fun to learn how to give commands using the ScratchJr blocks.	8.3	0	8.3	33.4	50.0	4.17	1.193
	3. Creating my own stories in ScratchJr makes learning more interesting.	8.3	8.3	16.7	0	66.7	4.08	1.443
	4. I'm curious to discover what new things I can do with ScratchJr.	8.3	8.3	0	25.0	58.4	4.17	1.337
	5.I enjoy learning how to program in this creative way.	0	8.3	0	25.0	66.7	4.5	0.905

Table 3 Grade Motivation

Category	Questions	Absolutely Disagree (%)	Disagree (%)	Neither Agree nor Disagree (%)	Agree (%)	Strongly Agree (%)	Mean Average	Std Deviation
Grade Motivation	6. I like that the stories or games I make in ScratchJr are better than those of other kids.	25	8.3	16.7	33.3	16.7	3.08	1.505
	7. It is important to me to do well in the CAL program activities	0	0	8.3	25	66.7	4.58	0.669
	8. I really want to create the most difficult and beautiful stories in ScratchJr.	0	25.0	0	33.3	41.7	3.92	1.24
	9. It is important that I can do well in the exercises with ScratchJr.	0	8.3	16.7	33.3	41.7	4.08	0.996
	10. It's important to me to successfully complete the projects assigned in the CAL program.	0	8.3	8.3	16.7	66.7	4.42	0.996

extremely confident that they had mastered ScratchJr and could create whatever they imagined (Q12) (M = 4.33), with all of them (100%) agreeing or strongly agreeing. Moreover, they cared dearly about not only creating something but also making work in a correct way (Q11) (M = 4.58).

As indicated in Table 5, the following category is related to students' self-regulation, initiative, and the use of strategies to learn, *i.e.*, their self-determination. The data reveal that students studied and were inclined to try different ways to learn (Q20) (M = 4.00) or solve an issue in ScratchJr (Q16) (M = 4.25). They also tended to spend time playing and creating in ScratchJr (Q17) (M = 4.08), while the vast majority of them agreed or strongly agreed (91.7%) to put a lot of effort into creating and making sure that their projects work (Q18) (M = 4.42). However, students didn't always plan before they started programming, as only half of them admitted to mostly doing it (Q19) (M = 3.75).

The last category is career motivation, which is related to the extent to which students are motivated to learn due to the potential relevance they believe it could have in their future or career. Based on Table 6, students did seem to believe that what they learned from the CAL program (Q22) (M = 3.92) and programming in ScratchJr (Q24) (M = 3.92) could potentially help them in the future. They were a bit less sure of how their creation, ScratchJr (Q21) (M = 3.75), will assist them in school, as only 58.3% of them believed that it could. Similarly, only half of them (50%) were almost certain that the way of thinking that they learned in ScratchJr could also be useful outside of programming (Q25) (M = 3.67). However, the majority of them were also unsure if they would follow a career in programming (Q23) (M = 3.17).

Table 4 Self-Efficacy

Category	Questions	Absolutely Disagree (%)	Disagree (%)	Neither Agree nor Disagree (%)	Agree (%)	Strongly Agree (%)	Mean Average	Std Deviation
Self-efficacy	11. It matters to me to see that my creations in ScratchJr work correctly.	0	0	8.3	25.0	66.7	4.58	0.669
	12. I believe I have learned well how to use ScratchJr to create whatever I imagine.	0	0	0	66.7	33.3	4.33	0.492
	13. I am confident that I can do well in the projects we make in the program.	0	8.3	16.7	25.0	50.0	4.17	1.03
	14. I believe I can become very good at creating stories and games in ScratchJr.	0	16.7	16.7	25.0	41.6	3.92	1.165
	15. I'm confident that I can understand how the command blocks in ScratchJr work.	0	0	25.0	41.7	33.3	4.08	0.793

Table 5 Self-Determination

Category	Questions	Absolutely Disagree (%)	Disagree (%)	Neither Agree nor Disagree (%)	Agree (%)	Strongly Agree (%)	Mean Average	Std Deviation
Self-Determination	16. I use different ways to solve a problem when I create something in ScratchJr.	0	8.3	8.3	33.4	50.0	4.25	0.965
	17. I spend enough time playing and creating in ScratchJr.	0	8.3	25.0	16.7	50.0	4.08	1.084
	18. I try very hard to make my creations correctly in ScratchJr	8.3	0	0	25.0	66.7	4.42	1.165
	19. I think carefully about my plan before I start connecting the programming blocks.	0	16.7	33.3	8.3	41.7	3.75	1.215
	20. I study and try out different things in ScratchJr to learn more.	0	8.3	25.0	25.0	41.7	4.00	1.044

Table 6 Career Motivation

Category	Questions	Absolutely Disagree (%)	Disagree (%)	Neither Agree nor Disagree (%)	Agree (%)	Strongly Agree (%)	Mean Average	Std Deviation
Career Motivation	21. The ability to make things in ScratchJr will help me at school in the future.	0	8.3	33.4	33.3	25.0	3.75	0.965
	22. The knowledge I gain from the CAL program will be useful as I grow up.	0	16.7	8.3	41.7	33.3	3.92	1.084
	23. Maybe when I grow up, I'll do something similar to what we do in ScratchJr.	0	16.7	58.3	16.7	8.3	3.17	0.835
	24. I believe that understanding programming through ScratchJr will help me later on.	0	0	33.3	41,7	25.0	3.92	0.793
	25. The way of thinking I learn from ScratchJr helps me solve other problems too, outside the computer.	0	8.3	41.7	25.0	25.0	3.67	0.985

5 Discussion

This study aimed to determine the long-term effects on students CT development and motivational aspects after following the Coding as Another Language (CAL) program, which made use of the ScratchJr app, and integrating and using a gamification platform for the duration of the program. Based on this study's findings, it is evident that the intervention was highly successful in improving students CT skills (Yang et al., 2023; Papadakis 2020, 2024; Louka & Papadakis, 2024), as well as enhancing their motivation (Unahalekhaka & Bers, 2021; Yang et al., 2023).

Regarding students CT skills development, there is a clear significant increase in the mean scores of the TechCheck test (pre-test to post-test), indicating the program's effectiveness in CT development. This is also confirmed by the considerably high scores on the final SWYK test, which, along with the very low standard deviation, suggests that the learning gains were consistent across the entire class. As such, block-based programming environments, like ScratchJr, can be valuable tools in introducing fundamental CT concepts, such as sequencing, loops, problem-solving, and conditionals, to young learners (Totan & Korucu, 2023). Additionally, a gamified project-based approach that merges computer science and playfulness with early literacy instruction and where students could "tinker" and experiment proved to have the capacity to assist students in developing robust problem-solving skills and a deeper understanding of computational concepts (Alotaibi, 2024; Videnovik et al., 2023). Moreover, the CAL program seems to be able to support learners with varying initial skill levels (Bers, 2019; Yang et al., 2023). Consequently, it is evident that the CAL program and ScratchJr can create an accessible learning environment where all students could develop the CT skills consistently (Bers et al., 2023; Yang et al., 2023).

Regarding students' motivational aspects, students exhibited in general very high levels of motivation in most categories. To be more specific, students exhibited very high levels of self-efficacy, grade, self-determination, and intrinsic motivation. They displayed a considerable interest, curiosity, confidence, and enjoyment while programming, believing confidently that they could express all their ideas in ScratchJr (Yang & Bers, 2024; Yang et al., 2023). Enjoyment and confidence are vital aspects of an effective and engaging learning program, as they usually can lead to higher engagement levels and less perseverance through challenges and difficulties (Papadakis et al., 2022; Song, 2024; Zourmpakis et al., 2024). Moreover, students were eager to complete their projects and valued the learning activities (Zourmpakis et al., 2023a; Zhao & Tu, 2024). They demonstrated a willingness to put forth significant effort and explore different solutions to problems, indicating a high degree of engagement and personal investment (Johansen et al., 2023; Zhao & Tu, 2024). However, it should be noted that students were not highly competitive with each other. This indicates that the program fostered a more collaborative or individually focused mindset rather than a competitive one. This is quite interesting, as learning environments that utilize gamification have sometimes been found to increase competitiveness and thus lower motivation (Sánchez-Martín, et al., 2017; Korkmaz & Öztürk, 2020), especially in the long run (Papadakis & Kalogiannakis, 2024). Additionally, students didn't often plan before programming. This finding could be due to the constructivist learning approach that was followed along with ScratchJr, as students were driven to an experimental, problem-solving, and iterative approach that tested their ability to test their ideas and identify solutions to their problems, a valuable skillset (Heliawati et al., 2021; Resnick & Rusk, 2020; Maida et al., 2023).

Unlike the other types of motivation, the career motivation had more moderate results, which could be expected in young children (Cahill & Furey, 2017). Although students understood the value of the knowledge and skills they acquired in the program, they were not sure how they could be used or transferred in other contexts. Consequently, they were unable to see how some skills, like problem-solving, could have value in other subjects, like science education (Hurt et al., 2023; Saidin et al., 2021). This suggests that while the program was successful in the development of CT skills, teachers should also focus on helping students build bridges and transfer their skills from programming concepts and problem-solving to other subjects that could benefit (Li & Oon, 2024; Ye et al., 2022), such as STEM education (Hurt et al., 2023).

6 Limitations

The present study contains some certain limitations. The small sample size and its geographical limitation, namely a school from an urban area, limit the ability to generalize these results. Furthermore, due to the small sample size, even if the sample were evenly divided by male and female students, it couldn't produce sufficient data. Moreover, socio-cultural difference was not taken into account. However, this study was meant to act also as a pilot in order to understand the long-term impact of a carefully planned pedagogical program combined with ScratchJr and gamification regarding CT development and students' motivational aspects. On the other hand, the use of classroom observations or student interviews could have further validated the data or even provide new insights and deeper understanding of the observed effects (Bostic et al., 2021; Powney & Watts, 2018) Although the assessment was focused on CT skills within programming, it should be noted that the transferability of CT skills, such as problem-solving, in other contexts, like STEM education (Hurt et al., 2023), is not certain and should be researched further into the future.

7 Conclusions

The present study offers vital insights and showcases that a long-term, gamified pedagogical approach that is rooted in constructionism and focuses on self-expression, creativity, and problem-solving through coding, while utilizing ScratchJr and gamification, can not only significantly enhance early learners computational thinking (CT) skills but also enhance key motivational factors, such as intrinsic motivation, self-efficacy, grade motivation, self-determination, and, to a certain degree, career motivation too. During the lesson, students felt confident and supported, enjoyed the creative process, and were motivated to engage deeply with the block-based programming, highlighting its value in early education.

Based on the finding of the study, the successful integration of gamification emerged as a valuable pedagogical strategy, promoting long-term engagement without hindering the cooperative aspect of the program, as some studies have shown (Papadakis et al., 2024). This finding reinforces the idea that well-designed and thoughtful gamified experiences can sustain student interest over time, even in cooperative learning environments (Zourmpakis, et al., 2022; Zourmpakis et al., 2023b). However, it would be interesting to investigate if environments that also focus on individual students needs and adapt learning, such as adaptive gamification environments (Papadakis et al., 2023; Zourmpakis et al., 2024), could yield even higher results, both in CT development and motivational aspects. Moreover, future studies could also consider the implementation of learning analytics the CT development or AI-driven feedback, both of which could build upon adaptive learning and adaptive gamification (Sarıyalçınkaya et al., 2021; Ugras et al., 2024; Vashishth et al., 2024).

Furthermore, this research study highlights the importance of viewing CT skills not only as a competency for programming and computer science but also as a transferable set of problem-solving approaches that could be integrated and used in other contexts as well. Educators and curriculum designers should be encouraged to link CT learning with other contexts and support them in applying these strategies, such as abstraction, decomposition, pattern recognition, and algorithmic thinking, the basics of CT development, across various subjects, especially in STEM education, where such skills can enhance inquiry and analytical thinking (Hurt et al., 2023; Li & Oon, 2024; Ye et al., 2022).

However, the role of the educator is critical in realizing these benefits and changes. The successful implementation of programs like CAL and the implementation of gamification strategies that can lead to CT development and students' high levels of motivation depend heavily on strong teacher training and professional development, especially in the case of transferring CT skills across disciplines. Teachers will require not only technical competence and skills but also pedagogical content knowledge (Papadakis et al., 2024; Zourmpakis et al., 2023) that will allow them to effectively teach CT and integrate it meaningfully into other subject areas (Hurt et al., 2023; Li & Oon, 2024; Ye et al., 2022). Consequently, supporting and educating teachers with proper teacher training programs will be essential towards making CT a foundational lifelong skill for all learners.

While the study displays some promising results, its limitations restrict the generalization of it and underscore the need for further research. Future research should aim to validate these findings with larger and more diverse student populations. Similar long-term studies should also examine how CT skills can transfer entirely or in specific tasks in STEM subjects and if the negative aspects of gamification that have been demonstrated in literature regarding lower motivation in long-term use are linked with the implementation of different pedagogical strategies.

Conflicts of Interest

The author declares no conflict of interest.

References

Alotaibi, M. S. (2024). Game-based learning in early childhood education: a systematic review and meta-analysis. Frontiers in Psychology, 15. https://doi.org/10.3389/fpsyg.2024.1307881

Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. Research and Practice in Technology Enhanced Learning, 11(1). https://doi.org/10.1186/s41039-016-0036-2

Bati, K. (2021). A systematic literature review regarding computational thinking and programming in early childhood education. Education and Information Technologies, 27(2), 2059–2082. https://doi.org/10.1007/s10639-021-10700-2

Bati, K., & İkbal Yetişir, M. (2021). Examination of Turkish Middle School STEM Teachers' Knowledge about Computational Thinking and Views Regarding Information and Communications Technology. Computers in the Schools, 38(1), 57–73. https://doi.org/10.1080/07380569.2021.1882206

Benzizoune, O. (2024). Enhancing classroom management and student engagement: The role of Class-Dojo and gamification in education. Journal of English Language Teaching and Applied Linguistics, 6(3), 106-114. https://doi.org/10.32996/jeltal

Bers, M. U. (2012). Designing Digital Experiences for Positive Youth DevelopmentFrom Playpen to Playground. https://doi.org/10.1093/acprof:oso/9780199757022.001.0001

Bers, M. U. (2018). Coding and Computational Thinking in Early Childhood: The Impact of ScratchJr in Europe. European Journal of STEM Education, 3(3). https://doi.org/10.20897/ejsteme/3868

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. https://doi.org/10.1007/s40692-019-00147-3

Bers, M. U., & Horn, M. S. (2009). Tangible programming in early childhood: revisiting developmental assumptions through new technologies: Childhood in a digital world. In High-tech tots: Childhood in a digital world. Information Age Publishing.

Bers, M. U., & Resnick, M. (2015). The official ScratchJr book: Help your kids learn to code. No Starch Press.

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/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). https://doi.org/10.1145/3507951.3519285

Bostic, J., Lesseig, K., Sherman, M., & Boston, M. (2019). Classroom observation and mathematics education research. Journal of Mathematics Teacher Education, 24(1), 5–31. https://doi.org/10.1007/s10857-019-09445-0

Cahill, M., & Furey, E. (2017). The Early Years-Career Development for Young Children: A Guide for Educators. CERIC. https://cica.org.au

Çakır, N. A., Çakır, M. P., & Lee, F. J. (2021). We game on skyscrapers: the effects of an equity-informed game design workshop on students' computational thinking skills and perceptions of computer science. Educational Technology Research and Development, 69(5), 2683–2703. https://doi.org/10.1007/s11423-021-10031-6

Carocca, F., Blake-West, J., & Bers, M. (2024). Localizing the Coding as another Language: ScratchJr Curriculum Through the Culture Based Model Framework. Proceedings of the 18th International Conference of the Learning Sciences - ICLS 2024, 2139–2140. https://doi.org/10.22318/icls2024.110257

Dufva, T., & Dufva, M. (2016). Metaphors of code—Structuring and broadening the discussion on teaching children to code. Thinking Skills and Creativity, 22, 97–110. https://doi.org/10.1016/j.tsc.2016.09.004

Flannery, L. P., & Bers, M. U. (2013). Let's Dance the "Robot Hokey-Pokey!" Journal of Research on Technology in Education, 46(1), 81–101. https://doi.org/10.1080/15391523.2013.10782614

Glynn, S. M., Brickman, P., Armstrong, N., & Taasoobshirazi, G. (2011). Science motivation questionnaire II: Validation with science majors and nonscience majors. Journal of Research in Science Teaching, 48(10), 1159–1176. Portico. https://doi.org/10.1002/tea.20442

Glynn, S. M., Taasoobshirazi, G., & Brickman, P. (2008). Science Motivation Questionnaire: Construct validation with nonscience majors. Journal of Research in Science Teaching, 46(2), 127–146. Portico. https://doi.org/10.1002/tea.20267

Govender, I., Govender, D. W., Havenga, M., Mentz, E., Breed, B., Dignum, F., & Dignum, V. (2014). Increasing self-efficacy in learning to program: exploring the benefits of explicit instruction for problem solving. The Journal for Transdisciplinary Research in Southern Africa, 10(1). https://doi.org/10.4102/td.v10i1.19

Guggemos, J. (2024). On the Predictors of Computational Thinking and Its Relationship with Artificial Intelligence. Artificial Intelligence for Supporting Human Cognition and Exploratory Learning in the Digital Age, 179–201. https://doi.org/10.1007/978-3-031-66462-5_10

Heliawati, L., Afakillah, I. I., & Pursitasari, I. D. (2021). Creative Problem-Solving Learning through Open-Ended Experiment for Students' Understanding and Scientific Work Using Online Learning. International Journal of Instruction, 14(4), 321–336. https://doi.org/10.29333/iji.2021.14419a

Hurt, T., Greenwald, E., Allan, S., Cannady, M. A., Krakowski, A., Brodsky, L., Collins, M. A., Montgomery, R., & Dorph, R. (2023). The computational thinking for science (CT-S) framework: operationalizing CT-S for K–12 science education researchers and educators. International Journal of STEM Education, 10(1). https://doi.org/10.1186/s40594-022-00391-7

Johansen, M. O., Eliassen, S., & Jeno, L. M. (2023). "Why is this relevant for me?": increasing content relevance enhances student motivation and vitality. Frontiers in Psychology, 14. https://doi.org/10.3389/fpsyg.2023.1184804

Katchapakirin, K., Anutariya, C., & Supnithi, T. (2022). ScratchThAI: A conversation-based learning support framework for computational thinking development. Education and Information Technologies, 27(6), 8533–8560. https://doi.org/10.1007/s10639-021-10870-z

Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2012). Learning Programming at the Computational Thinking Level via Digital Game-Play. Procedia Computer Science, 9, 522–531. https://doi.org/10.1016/j.procs.2012.04.056

Konstantina, L., Papadakis, S., & Kalogiannakis, M. (2025). Computational Thinking Using ScratchJr. – A Case Study. Futureproofing Engineering Education for Global Responsibility, 363–374. https://doi.org/10.1007/978-3-031-83523-0_34

Kyza, E. A., Georgiou, Y., Agesilaou, A., & Souropetsis, M. (2021). A Cross-Sectional Study Investigating Primary School Children's Coding Practices and Computational Thinking Using ScratchJr. Journal of Educational Computing Research, 60(1), 220–257. https://doi.org/10.1177/07356331211027387

Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). On Computational Thinking and STEM Education. Journal for STEM Education Research, 3(2), 147–166. https://doi.org/10.1007/s41979-020-00044-w

Li, Z., & Oon, P. T. (2024). The transfer effect of computational thinking (CT)-STEM: a systematic literature review and meta-analysis. International Journal of STEM Education, 11(1). https://doi.org/10.1186/s40594-024-00498-z

Liu, Z., Gearty, Z., Richard, E., Orrill, C. H., Kayumova, S., & Balasubramanian, R. (2024). Bringing computational thinking into classrooms: a systematic review on supporting teachers in integrating computational thinking into K-12 classrooms. International Journal of STEM Education, 11(1). https://doi.org/10.1186/s40594-024-00510-6

Louka, K. (2022). Programming environments for the development of CT in preschool education: A systematic literature review. Advances in Mobile Learning Educational Research, 3(1), 525–540. https://doi.org/10.25082/amler.2023.01.001

Louka, K., & Papadakis, S. (2024). Enhancing computational thinking in early childhood education through ScratchJr integration. Heliyon, 10(10), e30482. https://doi.org/10.1016/j.heliyon.2024.e30482

Ma, J., Zhang, Y., Zhu, Z., Zhao, S., & Wang, Q. (2023). Game-Based Learning for Students' Computational Thinking: A Meta-Analysis. Journal of Educational Computing Research, 61(7), 1430–1463. https://doi.org/10.1177/07356331231178948

Maida, S., Kashmala, P., Awais, H., & Khaldoon, K. (2023). Project-based Iterative Teaching Model for Introductory Programming Course. Nile Journal of Communication and Computer Science, 5(1), 10–41. https://doi.org/10.21608/njccs.2023.321167

Miller, E. C., Severance, S., & Krajcik, J. (2020). Connecting Computational Thinking and Science in a US Elementary Classroom. Integrated Approaches to STEM Education, 185–204. https://doi.org/10.1007/978-3-030-52229-2_11

NGSS Lead States. (2013). Next generation science standards: For states, by states. National Academies Press. https://epsc.wustl.edu

Öztürk, Ç., & Korkmaz, Ö. (2019). The Effect of Gamification Activities on Students' Academic Achievements in Social Studies Course, Attitudes towards the Course and Cooperative Learning Skills. Participatory Educational Research, 7(1), 1–15. https://doi.org/10.17275/per.20.1.7.1

Papadakis, S. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age. Mobile Learning Applications in Early Childhood Education, 101–121.
https://doi.org/10.4018/978-1-7998-1486-3.ch006

Papadakis, S. (2022). Apps to Promote Computational Thinking and Coding Skills to Young Age Children: A Pedagogical Challenge for the 21st Century Learners. Educational Process International Journal, 11(1).
https://doi.org/10.22521/edupij.2022.111.1

Papadakis, S., Zourmpakis, A.-I., & Kalogiannakis, M. (2023). Analyzing the Impact of a Gamification Approach on Primary Students' Motivation and Learning in Science Education. Learning in the Age of Digital and Green Transition, 701–711.
https://doi.org/10.1007/978-3-031-26876-2_66

Papadakis, S., Zourmpakis, A., Kasotaki, S., & Kalogiannakis, M. (2024). Teachers' Perspectives on Integrating Adaptive Gamification Applications into Science Teaching, Journal of Electrical Systems, 20(11s), 2593-2600.
https://doi.org/10.52783/jes.7917

Piaget, J. (1973). The Child and Reality: Problems of Genetic Psychology. (A. Rosin, Trans.). New York: Grossman.

Pila, S., Aladé, F., Sheehan, K. J., Lauricella, A. R., & Wartella, E. A. (2019). Learning to code via tablet applications: An evaluation of Daisy the Dinosaur and Kodable as learning tools for young children. Computers & Education, 128, 52–62.
https://doi.org/10.1016/j.compedu.2018.09.006

Powney, J., & Watts, M. (2018). Interviewing in Educational Research. Routledge.
https://doi.org/10.4324/9780429503740

Relkin, E., de Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. Computers & Education, 169, 104222.
https://doi.org/10.1016/j.compedu.2021.104222

Resnick, M., & Rusk, N. (2020). Coding at a crossroads. Communications of the ACM, 63(11), 120–127.
https://doi.org/10.1145/3375546

Ribaux, J. (2024). Code Comprehension for Novices with Explicit Instruction. Proceedings of the 2024 ACM Conference on International Computing Education Research - Volume 2, 571–573.
https://doi.org/10.1145/3632621.3671419

Saidin, N. D., Khalid, F., Martin, R., Kuppusamy, Y., & Munusamy, N. A. (2021). Benefits and Challenges of Applying Computational Thinking in Education. International Journal of Information and Education Technology, 11(5), 248–254.
https://doi.org/10.18178/ijiet.2021.11.5.1519

Salta, K., & Koulougliotis, D. (2015). Assessing motivation to learn chemistry: adaptation and validation of Science Motivation Questionnaire II with Greek secondary school students. Chemistry Education Research and Practice, 16(2), 237–250.
https://doi.org/10.1039/c4rp00196f

Sánchez-Martín, J., Cañada-Cañada, F., & Dávila-Acedo, M. A. (2017). Just a game? Gamifying a general science class at university. Thinking Skills and Creativity, 26, 51–59.
https://doi.org/10.1016/j.tsc.2017.05.003

Sarıyalçınkaya, A. D., Karal, H., Altinay, F., & Altinay, Z. (2021). Reflections on Adaptive Learning Analytics. Advancing the Power of Learning Analytics and Big Data in Education, 61–84.
https://doi.org/10.4018/978-1-7998-7103-3.ch003

Stamatios, P. (2022). Can Preschoolers Learn Computational Thinking and Coding Skills with ScratchJr? A Systematic Literature Review. International Journal of Educational Reform, 33(1), 28–61.
https://doi.org/10.1177/10567879221076077

Strawhacker, A., Lee, M., & Bers, M. U. (2017). Teaching tools, teachers' rules: exploring the impact of teaching styles on young children's programming knowledge in ScratchJr. International Journal of Technology and Design Education, 28(2), 347–376.
https://doi.org/10.1007/s10798-017-9400-9

Sullivan, A., & Umashi Bers, M. (2019). Computer Science Education in Early Childhood: The Case of ScratchJr. Journal of Information Technology Education: Innovations in Practice, 18, 113–138.
https://doi.org/10.28945/4437

Sullivan, A., & Strawhacker, A. (2021). Screen-Free STEAM: Low-Cost and Hands-on Approaches to Teaching Coding and Engineering to Young Children. Embedding STEAM in Early Childhood Education and Care, 87–113.
https://doi.org/10.1007/978-3-030-65624-9_5

Tariq, R., Aponte Babines, B. M., Ramirez, J., Alvarez-Icaza, I., & Naseer, F. (2025). Computational thinking in STEM education: current state-of-the-art and future research directions. Frontiers in Computer Science, 6.
https://doi.org/10.3389/fcomp.2024.1480404

Totan, H. N., & Korucu, A. T. (2023). The Effect of Block Based Coding Education on the Students' Attitudes about the Secondary School Students' Computational Learning Skills and Coding Learning: Blocky Sample. Participatory Educational Research, 10(1), 443–461.
https://doi.org/10.17275/per.23.24.10.1

Uğraş, H., Uğraş, M., Papadakis, S., & Kalogiannakis, M. (2024). Innovative Early Childhood STEM Education with ChatGPT: Teacher Perspectives. Technology, Knowledge and Learning, 30(2), 809–831.
https://doi.org/10.1007/s10758-024-09804-8

Unahalekhaka, A., & Bers, M. U. (2021). Taking coding home: analysis of ScratchJr usage in home and school settings. Educational Technology Research and Development, 69(3), 1579–1598. https://doi.org/10.1007/s11423-021-10011-w

Unahalekhaka, A., & Bers, M. U. (2022). Evaluating young children's creative coding: rubric development and testing for ScratchJr projects. Education and Information Technologies, 27(5), 6577–6597. https://doi.org/10.1007/s10639-021-10873-w

Vashishth, T. K., Sharma, V., Sharma, K. K., Kumar, B., Panwar, R., & Chaudhary, S. (2024). AI-Driven Learning Analytics for Personalized Feedback and Assessment in Higher Education. Using Traditional Design Methods to Enhance AI-Driven Decision Making, 206–230. https://doi.org/10.4018/979-8-3693-0639-0.ch009

Vee, A. (2017). Coding literacy: How computer programming is changing writing. Mit Press. https://doi.org/10.7551/mitpress/10655.001.0001

Videnovik, M., Vlahu-Gjorgievska, E., & Trajkovik, V. (2020). To code or not to code: Introducing coding in primary schools. Computer Applications in Engineering Education, 29(5), 1132–1145. Portico. https://doi.org/10.1002/cae.22369

Videnovik, M., Vold, T., Kiønig, L., Madevska Bogdanova, A., & Trajkovik, V. (2023). Game-based learning in computer science education: a scoping literature review. International Journal of STEM Education, 10(1). https://doi.org/10.1186/s40594-023-00447-2

Wilensky, U. (2014). Computational thinking through modeling and simulation. white paper, Summit on Future Directions in Computer Education, Orlando, FL.

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

Yang, Z., & Bers, M. (2023). Examining Gender Difference in the Use of ScratchJr in a Programming Curriculum for First Graders. Computer Science Education, 34(4), 864–885. https://doi.org/10.1080/08993408.2023.2224135

Yang, Z., Shaffer, P. M., Hagan, C., Dubash, P., & Bers, M. (2023). Impact Study of the Coding as Another Language Curriculum: Study B. Grantee Submission.

Ye, J., Lai, X., & Wong, G. K. (2022). The transfer effects of computational thinking: A systematic review with meta-analysis and qualitative synthesis. Journal of Computer Assisted Learning, 38(6), 1620–1638. Portico. https://doi.org/10.1111/jcal.12723

Zaibon, S. B., & Yunus, E. (2022). The Effectiveness of Game-Based Learning Application Integrated with Computational Thinking Concept for Improving Student's Problem-Solving Skills. E-Infrastructure and e-Services for Developing Countries, 429–442. https://doi.org/10.1007/978-3-031-06374-9_28

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. Computers & Education, 141, 103607. https://doi.org/10.1016/j.compedu.2019.103607

Zhao, Z., & Tu, C. (2024). A Study on the Relationship between Perseverance and Learning Satisfaction among Students. International Journal of Instructional Cases, 8(1), 135-150.

Zourmpakis, A.-I., Kalogiannakis, M., & Papadakis, S. (2024). The Effects of Adaptive Gamification in Science Learning: A Comparison Between Traditional Inquiry-Based Learning and Gender Differences. Computers, 13(12), 324. https://doi.org/10.3390/computers13120324

Zourmpakis, A. I., Papadakis, S., & Kalogiannakis, M. (2022). Education of preschool and elementary teachers on the use of adaptive gamification in science education. International Journal of Technology Enhanced Learning, 14(1), 1. https://doi.org/10.1504/ijtel.2022.120556

Zourmpakis, A.-I., Kalogiannakis, M., & Papadakis, S. (2023). Adaptive Gamification in Science Education: An Analysis of the Impact of Implementation and Adapted Game Elements on Students' Motivation. Computers, 12(7), 143. https://doi.org/10.3390/computers12070143

Zourmpakis, A.-I., Kalogiannakis, M., & Papadakis, S. (2023). A Review of the Literature for Designing and Developing a Framework for Adaptive Gamification in Physics Education. The International Handbook of Physics Education Research: Teaching Physics, 5-1-5–26. https://doi.org/10.1063/9780735425712_005