International Journal of Learning, Teaching and Educational Research
Vol. 23, No. 9, pp. 343-360, September 2024
https://doi.org/10.26803/ijlter.23.9.18
Received Jun 25, 2024; Revised Sep 13, 2024; Accepted Sep 21, 2024

Improving Elementary Students' Computational Thinking Skills through an Educational Robot Intervention: A Quasi-Experimental Study

Siou-Yu Yang
Sing-An Elementary School
Chiayi, Taiwan

Yi-Chun Lin
R.O.C. Military Academy
Kaohsiung, Taiwan

Yen-Ting Lin[*]
National Pingtung University
Pingtung, Taiwan

Abstract. Globally, many educational institutions recognize the importance of computational thinking and have begun incorporating it into primary education. Educators cultivate students' computational thinking skills through block-based programming; however, a lack of digital learning tools that offer real interaction may negatively impact students' computational thinking learning performances. This study proposes a cost-effective, block-based, programmable computational thinking educational robot developed using the open-source Arduino platform, combined with Android application development. This robot is specifically designed for use in elementary computational thinking education. To assess the impact of the proposed approach on elementary students' learning achievement, motivation, and attitudes towards computational thinking education, a quasi-experimental design with control and experimental groups was implemented in the computational thinking curriculum at an elementary school. The experiment was conducted over a period of three weeks and involved two classes of students and one teacher. The control group engaged in computational thinking learning activities using computers, while the experimental group completed computational thinking learning activities using the computational thinking educational robot and application developed in this study. Data were collected through prior knowledge tests of computational thinking, learning motivation and attitude questionnaires,

[*] Corresponding author: *Yen-Ting Lin, ricky014@gmail.com*

and computational thinking achievement tests completed by the students. The results indicate that the experimental group outperformed the control group in learning achievement, motivation, and attitudes, demonstrating that physical interaction in learning can effectively enhance learning performances.

Keywords: computational thinking; educational robots; learning achievement; learning motivation; quality education

1. Introduction

In recent years, computational thinking (CT) has gained prominence across various disciplines. Globally, many educational institutions recognize the importance of CT and have begun incorporating it into primary education (Valentina et al., 2022). The aim of integrating CT at this level is to develop students' logical thinking, systemic thinking, and problem-solving skills. According to Chao (2016), CT is promoted as a critical skill for addressing real-world challenges

To foster students' CT skills, programming is commonly used by educators (Tikva & Tambouris, 2021). However, programming languages can be difficult and complex, particularly for younger students. Consequently, numerous visual programming tools, such as Scratch Code.org platform, Kodu and App Inventor, have been developed to assist in teaching and learning CT (Zhao et al., 2022). These tools simplify the programming process by using a matrix of programmable bricks, allowing students to focus on practical application without needing to master complex coding languages. Despite these tools, students may struggle to complete tasks if they lack sufficient CT skills and guidance (Jiang & Li, 2021).

To address this issue, several applications (apps) have been designed to enhance students' CT abilities, including Minecraft on the Code.org platform and Kodu. These apps use a digital game-based approach, guiding students through virtual roles and tasks using programmable brick matrices. However, literature indicates that purely digital learning games lack real-world interaction, which can negatively impact students' CT learning outcomes (Barz et al., 2024).

Students often find it challenging to learn programming in a programming development environment composed of abstract concepts. However, concretizing these concepts can increase their engagement and stimulate their learning motivation (Karaahmetoğlu & Korkmaz, 2019). Educational robots, through physical interaction, can effectively assist K-12 students in developing CT skills. Compared to purely digital programming, using physical robots in learning activities can enhance peer interaction. During these interactions, students actively participate in collaborative processes and willingly communicate programming decisions, fostering a positive learning environment. This suggests a gap in learning in which students lack physical contexts to apply their CT skills. In response to these challenges, some studies have implemented physical robots, such as Lego EV3 and mBot, in elementary education (Kalaitzidou & Pachidis, 2023). The use of the Lego Mindstorms EV3 robot kit for programming instruction has significantly increased students' positive attitudes toward programming (Yıldız & Seferoğlu, 2021). Additionally, studies have shown that educational robot technology is more effective in elementary schools than at other educational levels (Zhang et al., 2021). However, these robots are often expensive, have lengthy repair times, and their parts are not readily available, making them impractical for regular educational use.

Considering these issues, this study proposes a cost-effective, block-based, programmable CT educational robot developed using the open-source Arduino platform combined with Android application development. This robot is specifically designed for use in elementary CT education, with the aim of enhancing students' learning motivation, attitudes, and achievement through a hybrid of physical and digital educational robots. To evaluate the effectiveness of this approach, an experiment was conducted to address the following research questions:

- Do students who learn about computational thinking with the hybrid physical and digital educational robot demonstrate better learning achievements compared to those learning through the pure digital learning approach?
- Do students who learn about computational thinking with the hybrid physical and digital educational robot show higher learning motivation than those using the pure digital learning approach?
- Do students who learn about computational thinking with the hybrid physical and digital educational robot exhibit a more positive learning attitude than those using the pure digital learning approach?

2. Literature Review

2.1 Computational Thinking

The term CT was introduced by Jeannette M. Wing in March 2006 (Lodi & Martini, 2021) and it involves the following four steps: problem decomposition, pattern recognition, abstraction, and algorithm design. Problem decomposition involves breaking down complex problems into smaller, more manageable parts. Pattern recognition entails observing and identifying similarities within these smaller problems and organizing them. Abstraction involves highlighting the important parts of these problems while ignoring the irrelevant details. Algorithm design refers to developing methods to solve problems (Kramer, 2007). Education in CT often relies on programming, as it directly trains the cognitive tasks needed for CT and serves as a medium to demonstrate CT skills (Wong & Cheung, 2020). Thus, teaching CT generally involves programming courses, allowing students to practice CT skills, such as abstraction, flow control, pattern recognition, and debugging, through coding. Many studies have explored how programming education can enhance students' CT skills, focusing on guiding students to use CT to think and solve problems, rather than on the intricacies of programming itself (Fang et al., 2022).

In this context, recent research has developed CT education strategies. In 2019, Samar and Taima designed a curriculum using Scratch and Java programming to integrate key CT concepts. Through guided tasks and a progressively challenging curriculum, students subtly learn the core concepts of CT. This approach emphasizes that regardless of the tools used, it is possible to teach students CT, allowing other educators to apply these principles in their teaching (Samar & Taima, 2019). This method lets students learn CT through programming, with the advantage that they can understand CT concepts during the programming process. However, a disadvantage is that students may not apply CT concepts to the everyday problems they encounter.

Wong and Jiang (2018) observed the relationship between CT and programming in a course for upper elementary students. Their study showed that programming significantly enhanced students' CT skills, especially in problem analysis (Wong & Jiang, 2018). This method assesses students' achievement of CT skills at various stages of learning programming, with the advantage of precisely understanding which concepts students have mastered. However, the disadvantage is that students might only learn the same concepts repetitively without applying them to everyday situations. Sun and Liu (2023) enhanced students' CT abilities through a 13-week game-based Python programming activity. They found that students might need time to adapt to Python programming initially but significant improvements were noted in their algorithmic and construction skills even without programming activities.

However, programming languages often become a barrier to learning, especially for elementary students, significantly impacting their learning outcomes (Kite et al., 2021). Thus, elementary teachers often use block-based programming tools, such as Scratch and CodeMonkey, for instruction (Stewart & Baek, 2023). Although these tools lower the starting point for learning CT, studies have shown that students struggle with understanding the more abstract concepts in programming when using virtual tools (Çetin & Türkan, 2022). Further research has indicated that using physical teaching tools can effectively enhance learning outcomes when students are learning abstract concepts (Dağ et al., 2023). Consequently, numerous CT educational robots have been developed in recent years, allowing teachers and students to engage with robots through block-based programming tools, thereby facilitating the teaching of CT. The following sections introduce CT educational robots.

2.2 Educational Robots for Computational Thinking

In recent years, as technology has advanced, many manufacturers have begun producing robots for educational purposes. These robots, known as educational robots, are primarily used in classroom instruction, extracurricular clubs, robot camps, and competitions (Noh & Lee, 2020). Currently, there are many robots on the market designed for educational use such as LEGO EV3, Softbank Pepper, ASUS Zenbo, and Makeblock mBot. These robots are mainly used to cultivate students' understanding of CT and their programming skills.

Educational robots can provide practical and tangible learning experiences in teaching CT. Students translate abstract concepts into concrete actions and commands through programming, which helps them understand abstract concepts and logical thinking (Qu & Fok, 2022). Educational robots serve as interactive tools; during the interaction between students and robots, students can pose various questions and challenges and try different solutions, thus stimulating their creativity and imagination and fostering their logical thinking and problem-solving abilities (Noh & Lee, 2020). Additionally, robots can provide immediate feedback and corrections. By observing the robots' performance, students can promptly understand their mistakes and shortcomings, thereby adjusting and improving their solutions to enhance their problem-solving skills. Ou Yang et al. (2023) summarized the advantages of educational robots in teaching CT, including fostering higher-order thinking skills, enhancing CT skills, boosting creativity and problem-solving abilities, and increasing student engagement and interest.

Educational robots are increasingly used as tools to attract students to learn computer programming. When integrated with CT, they can help develop higher-order thinking skills (Jawawi et al., 2022). Shen et al. (2022) designed an educational robot curriculum for fifth-grade elementary students to participate in problem-solving algorithm design. Through experimental research, it was found that students' CT abilities in programming significantly improved, and it also aided their reasoning abilities in everyday life (Shen et al., 2022). Educational robots use a visual programming environment, allowing students to write programs in a relatively simple and fun way, which can effectively enhance their learning motivation and outcomes and promote the development of their CT skills (Chevalier et al., 2020). Concretizing abstract concepts can increase student engagement and stimulate their learning motivation (Karaahmetoğlu et al., 2019). Cervera et al. (2020) used educational robots in hands-on experimental classes for second to fourth graders. The study indicated that guiding students in using educational robots helped them understand programming logic and CT skills and promoted cooperation among peers.

Educational robots are well-suited for cultivating students' CT skills (Ching & Hsu, 2024). Students actively use body language, such as gestures, movements, and facial expressions, to reason, communicate their predictions, mimic robot actions, or make transitions between abstract and concrete concepts (Hsu et al., 2018). This is a form of embodied cognition. Embodied cognition is a cognitive theory that suggests most cognitive processes, whether in humans or other organisms, are shaped by sensory experiences involving the whole body. This concept extends to embodied learning, which is based on the idea that students' cognitive experiences, perceptions, and knowledge are formed through activities involving their bodies in relation to their environment (Kopcha et al., 2021). Embodied learning activities help reduce the cognitive load on students when transitioning between programming and representations during robot tasks (Moore et al., 2020), further promoting the acquisition of abstract conceptual knowledge. Ching and Hsu (2024) encourage future research to explore the effective design and integration of embodied learning activities to support and enhance the development of CT through educational robot technology.

However, the educational robots currently available on the market are relatively expensive due to their programmable capabilities, making it difficult to make them widely available in elementary education so that every student can use them. Therefore, this study proposes the development of an affordable block-based programmable CT educational robot using the open-source Arduino platform combined with Android app development. This robot is applied in elementary CT education to enhance student learning performances. By using an application to control the physical educational robot, students can understand how virtual code drives the physical robot and completes learning tasks.

3. Computational Thinking Educational Robot

The CT educational robot developed in this study consists of three main components, namely an Arduino-based educational robot, an Android-based CT application, and a paper-based task map. The educational robot is equipped with Bluetooth wireless communication components that transmit and receive messages to and from the Android-based CT application. The Android-based application includes modules for connection, programming blocks, and task management, allowing students to assemble programming blocks, which are then converted into commands that control the robot's motors for movement. The robot is also equipped with a color sensor at the bottom, which identifies its position on the task map and sends this information back to the Arduino microcontroller. The Arduino then communicates this data to the Android application, verifying whether the robot has correctly stopped at the designated spot on the map and providing feedback to the operator. The system architecture is illustrated in Figure 1.

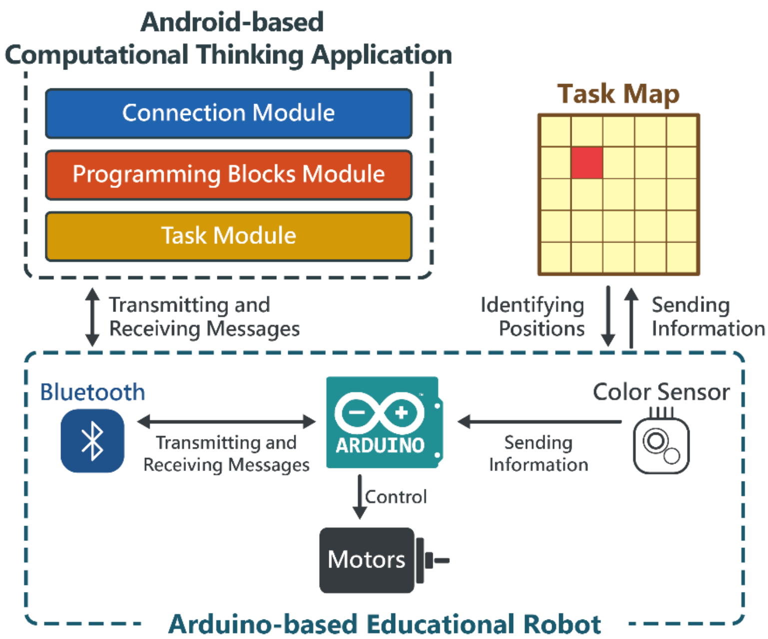


Figure 1: Computational thinking educational robot structure

The designed educational robot is a car-type robot equipped with two wheels, allowing easy movement. It is controlled by an L298N DC motor driver module, which enables precise positioning on the designated spots on the task map. A TCS3200 color sensor on the bottom of the robot identifies whether its position on the task map meets the requirements of the problem. The robot is equipped with Bluetooth wireless communication components to transmit and receive messages from smart devices, controlled by the CT application to move the robot. The physical model of the CT educational robot developed in this study is shown in Figure 2.

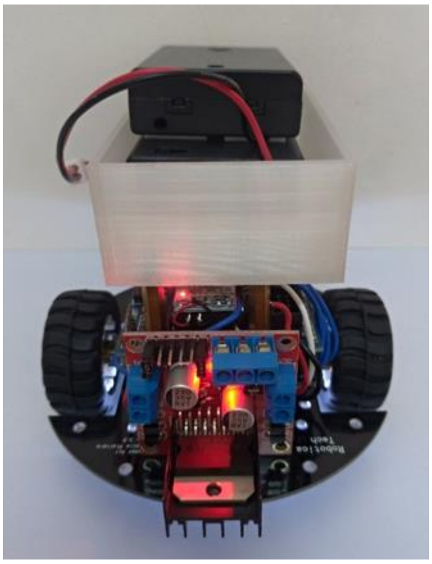


Figure 2: Computational thinking educational robot

The CT application developed in this study includes a programming block module and a task module. The task module provides problems that students solve using the block module. The problems in the task module are designed based on the Minecraft Adventurer challenges from the Hour of Code event on the Code.org website. Five problems are designed, ranging from basic linear tasks to complex looping tasks. Table 1 presents a comparison between Minecraft Adventurer and the problems designed in this study. The programming block module includes five types of programming functions: forward, backward, left turn, right turn, and loop. Students use these programming blocks to control the educational robot to move and meet the requirements of the task module. The CT app developed in this study is shown in Figure 3. The task map designed in this study provides an environment for the educational robot to navigate. It includes a black block at one location, and students must use hints from the task module and manipulate the programming block module to guide the educational robot to the correct position. The task map is shown in Figure 4.

Table 1: Comparison of Minecraft Adventurer and computational thinking educational robot tasks

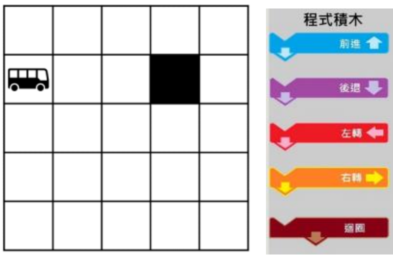
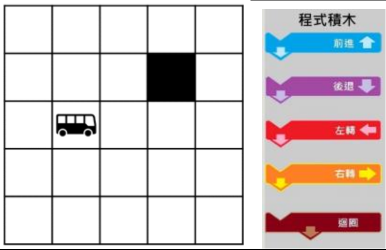
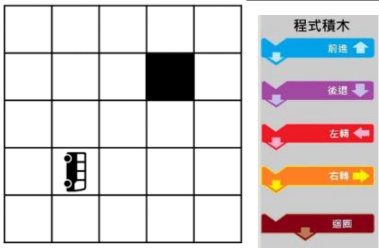
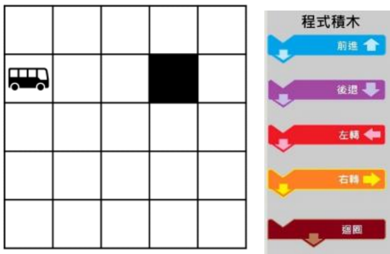
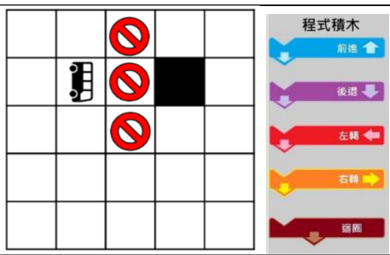
<table>
<tr><th>Minecraft Adventurer</th><th>Computational Thinking Educational Robot</th></tr>
<tr><td>Task 1: Add a second “move forward” command to reach the position of the sheep.</td><td>Task 1: Add three “forward” blocks to move the car to the black area.</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>Task 2: Time to shear the sheep! Use the “shear” command to collect wool from the sheep.</td><td>Task 2: Add a “forward” block and a “turn” block to move the car to the black area.</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>Task 3: We need to finish building the house before the sun sets, and it requires a lot of wood. Please cut down all three trees.</td><td>Task 3: Add forward blocks and a turn block to move the car to the black area.</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>Task 4: Place “place” and “move forward” commands inside a repeat loop to start building the first part of the house.</td><td>Task 4: Add a “forward” block and a “loop” block to move the car to the black area.</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>Task 5: Build the rest of the house using any materials you like. The repeat loop command will come in handy.</td><td>Task 5: Add a “forward” block, a “turn” block, and a “loop” block to move the car to the black area.</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

Figure 3: Screenshot of the computational thinking app

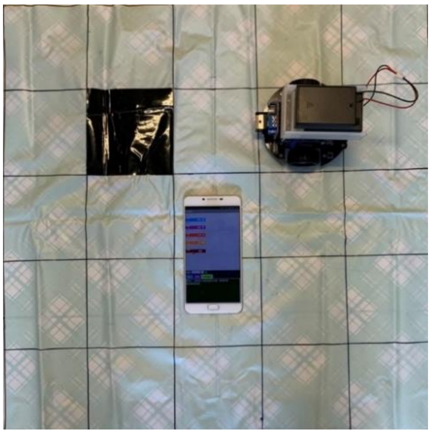


Figure 4: Complete set of the computational thinking educational robot accessories

4. Research Methodology

4.1 Research Design

To evaluate the effect of the CT educational robot developed in this study on elementary students' learning achievement, motivation, and attitudes towards CT education, a quasi-experimental design was utilized, specifically adopting a non-equivalent control group design. This sub-type of quasi-experimental design involves the use of both control and experimental groups but does not require random assignment, allowing for comparisons between groups that are naturally occurring or pre-existing in the educational setting. The sample was selected using convenience sampling, in which the subjects easily accessible to the researchers were chosen. The advantage of convenience sampling is that it allows for easy access to samples and is relatively low in cost (Mweshi & Sakyi, 2020).

The study was conducted over a period of three weeks, with three sessions (120 minutes each), involving 44 students (25 boys and 19 girls) with an average age of 9.7 years. The experimental group, consisting of 22 students, used the CT educational robot developed in this study, while the control group of 22 students engaged in CT learning activities using Minecraft Adventurer on Code.org. Both groups were taught by the same teacher to ensure consistency in instruction.

4.2 Research Tools

To evaluate learning performances, data were collected and analyzed using a prior knowledge test for CT knowledge, a CT achievement test, a learning motivation questionnaire, and a learning attitude questionnaire. All questionnaires and tests consisted of closed-ended items. The pre-test was designed to evaluate students' knowledge level and perceptions with regard to CT before participating in the experiment, while the post-test was designed to evaluate their learning outcomes and perceptions with regard to CT after the instructional intervention.

Both the prior knowledge test and the learning achievement test used question items from the Bebras International Computational Thinking Challenge (Dagiene & Dolgopolovas, 2022), comprising five questions with a total score of 100 points. The learning achievement test consisted of five practical tasks, with a maximum score of 100 points. Both the prior knowledge test and learning achievement test were evaluated by three experts: two elementary school computer science teachers with over 10 years of experience and one professor with over 10 years of experience in computer science education, ensuring expert validity. The learning motivation questionnaire was based on the intrinsic motivation dimension of the Motivated Strategies for Learning Questionnaire (MSLQ) (Pintrich & de Groot, 1990), measuring students' perceived importance and interest in the course's educational activities. This questionnaire contained nine items using a 7-point Likert scale and has been widely used in numerous studies (Wei et al., 2016). The reliability (Cronbach's alpha) of the learning motivation questionnaire in this study was 0.933. Cronbach's alpha is a commonly used measure of test reliability and a method to assess internal consistency (Tavakol & Dennick, 2011). Higher internal consistency reliability indicates that the items effectively measure the same construct. The learning attitude questionnaire was based on the scale developed by Hwang and Chang (2011), assessing students' attitudes towards learning. It consisted of seven items on a 5-point Likert scale, also widely used in research (Lin, 2016, 2023), with a reliability (Cronbach's alpha) of 0.933 in this study.

4.3 Experimental Procedure

The experiment was conducted in a regular elementary school computer classroom over three weeks, with each session being 40 minutes per week. In the first week, the teacher explained the learning content and procedure to the students participating in the experiment. Then, 10 minutes were spent explaining the learning activities, followed by 20 minutes to complete the CT prior knowledge test, and the final 10 minutes for filling out the learning motivation and learning attitude questionnaires. In the second week, the experimental group operated the CT educational robot developed in this study, while the control group used Minecraft Adventurer on code.org for their CT learning, while the teacher presented to assist both groups. The CT learning activities required both groups to use their respective learning tools to manipulate programming blocks to complete learning tasks. There were five tasks in total, ranging from basic linear tasks to complex loop tasks. The experimental group had to use programming blocks to control the CT robot to move and complete the task requirements. In the third week, both groups took the CT achievement test and again filled out the learning motivation and learning attitude questionnaires. The experimental procedure is shown in Figure 5.

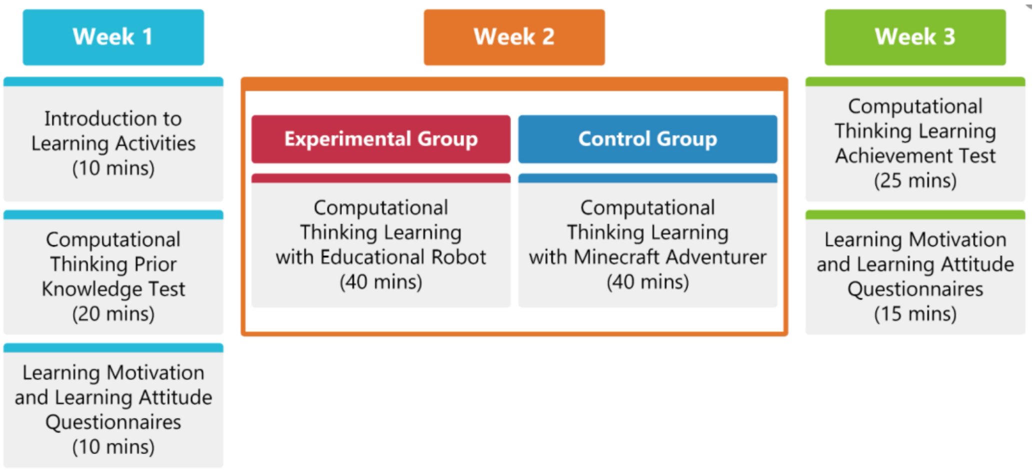


Figure 5: Experimental procedure

4.4 Data Collection and Analysis

The data collected in this study included pre-tests and post-tests of the CT prior knowledge test, the learning motivation and attitude questionnaires, and the CT achievement test. After organizing and cataloging the collected questionnaires and test data, statistical analysis was performed using statistical software. The data analysis techniques included descriptive statistics, ANCOVA, and independent sample *t*-tests to address the research questions and achieve the study's objectives.

5. Results

Based on the data collected from the quasi-experimental design described previously, the results of this study were analyzed and discussed from three aspects: student learning achievement, learning motivation, and learning attitudes.

5.1 Analysis of Computational Thinking Learning Achievement

To assess if there were any initial differences in CT prior knowledge between the experimental and control groups, an independent sample *t*-test was conducted on the pre-test scores of both groups. According to Table 2, the experimental group had an average score of 30.00, while the control group scored an average of 32.73. The variance *Levene* test showed no significant difference ($F = 1.196$, $p = 0.280 > 0.05$), suggesting equal variances between the groups. The *t*-test did not show significant differences ($t = -0.449$, $p = 0.656 > 0.05$), indicating no significant pre-test score differences between the groups.

Table 2: The independent sample t-test result of the prior knowledge for the two groups

Group	N	Mean	SD	*Levene*'s test		*t*-test	
				F	*p*	*t*	*p*
Experimental group	22	30.00	17.18	1.196	0.280	-0.449	0.656
Control group	22	32.73	22.72				

To evaluate the impact of the strategies proposed in this study on students' CT, a one-way ANCOVA was used to analyze the results of the learning achievement test, controlling for the effect of prior knowledge. Learning achievement test scores were set as the dependent variable, and prior knowledge test results as the covariate. Table 3 shows the one-way ANCOVA results for learning achievement scores between the groups, with the experimental group averaging 36.36, with a standard deviation of 21.94, and the control group averaging 24.55, with a standard deviation of 24.64. The analysis indicated significant differences between the groups ($F(1, 41) = 4.264$, $p = 0.045 < 0.05$), with the experimental group outperforming the control group. This result demonstrates that students using the CT educational robot developed in this study achieved better outcomes than those using Minecraft Adventurer on Code.org.

Table 3: The ANCOVA result of the learning achievement for the two groups

Group	N	Mean	SD	*F*	*p*
Experimental group	22	36.36	21.94	4.264	0.045*
Control group	22	24.55	24.64		

*$p < 0.05$

5.2 Analysis of Learning Motivation

To understand any pre-experimental differences in learning motivation between the groups, an independent sample *t*-test was conducted on the pre-test scores. According to Table 4, the experimental group had an average score of 5.31, and the control group 5.64. The variance *Levene* test showed no significant difference ($F = 0.049$, $p = 0.826 > 0.05$), suggesting equal variances between the groups. The *t*-test showed no significant differences ($t = -1.128$, $p = 0.266 > 0.05$), indicating no significant pre-experimental differences in learning motivation.

Table 4: The independent sample *t*-test result of the learning motivation for the two groups

Group	N	Mean	SD	*Levene*'s test		*t*-test	
				F	*p*	*t*	*p*
Experimental group	22	5.31	1.001	0.049	0.826	-1.128	0.266
Control group	22	5.64	0.928				

A one-way ANCOVA was performed on the post-test learning motivation scores to understand the impact of the different learning methods on students' motivation. As shown in Table 5, learning motivation post-test scores were set as the dependent variable, with pre-test scores as the covariate. The experimental group averaged 5.68 with a standard deviation of 1.03, while the control group averaged 5.39 with a standard deviation of 1.09. The analysis showed significant differences between the groups ($F = 4.263$, $p = 0.045 < 0.05$), with the experimental group scoring higher, indicating a positive impact on learning motivation from using the CT educational robot compared to the software-based learning.

Table 5: The ANCOVA result of the learning motivation for the two groups

Group	N	Mean	SD	*F*	*p*
Experimental group	22	5.68	1.03	4.263	0.045*
Control group	22	5.39	1.09		

*$p < 0.05$

5.3 Analysis of Learning Attitudes

To determine any pre-experimental differences in learning attitudes, an independent sample *t*-test was conducted on the pre-test scores. From Table 6, the experimental group averaged 3.80, and the control group 3.86. The variance *Levene* test showed no significant differences (F =1.851, p = 0.181 > 0.05), suggesting equal variances. The t-test showed no significant differences (t = -0.345, p = 0.732 > 0.05), indicating no significant pre-experimental differences in learning attitudes.

Table 6: The independent sample *t*-test result of the learning attitude for the two groups

Group	N	Mean	SD	*Levene*'s test		*t*-test	
				F	*p*	*t*	*p*
Experimental group	22	3.80	0.54	1.851	0.181	-0.345	0.732
Control group	22	3.86	0.70				

A one-way ANCOVA was then conducted on the post-test learning attitudes scores to understand if the different learning methods impacted students' attitudes significantly. As shown in Table 7, learning attitudes post-test scores were set as the dependent variable, with pre-test scores as the covariate. The experimental group averaged 3.98 with a standard deviation of 0.59, and the control group 3.70 with a standard deviation of 0.67. The results showed significant differences (F = 5.043, p = 0.025 < 0.05), with the experimental group scoring higher, indicating that the CT educational robot positively influenced students' learning attitudes compared to the software-based learning method.

Table 7: The ANCOVA result of the learning attitude for the two groups

Group	N	Mean	SD	*F*	*p*
Experimental group	22	3.98	0.59	5.403	0.025*
Control group	22	3.70	0.67		

*$p < 0.05$

6. Discussion

Using the CT educational robot and application developed in this study, the experimental group showed significant improvement in learning achievement, motivation, and attitude. These findings concur those of Zhang et al. (2021), who noted that interaction with physical objects during learning can significantly enhance students' learning effectiveness and motivation. Similarly, a study by Chiazzese et al. (2019) highlighted that using educational robots can promote the early development of CT, enabling students to apply learned concepts to solve real-life problems. This aligns with John Dewey's advocacy for "Learning by doing", which emphasizes experiential learning through context, reflection, learning, and knowledge, encouraging students to immerse themselves, explore, and discover the relationships among various elements within a context, leading to deeper understanding and insights. Additionally, the active engagement and continuous attempts during operation allow students to discover connections and achieve learning objectives (Amri et al., 2022).

The results of this study indicate that the control group students' post-test scores in learning motivation, effectiveness, and attitudes were lower than their pre-test scores. The control group used a digital game-based approach with a web application for CT learning activities. This might be related to findings by Li and Lee (2024), who discovered that students' learning motivation varies with their experiences using computers. Another possible explanation by Lai et al. (2023) suggests that while computer software can assist in learning, the key to enhancing motivation and outcomes is appropriate guidance from educators, ensuring meaningful interactions that can significantly improve learning effectiveness, motivation, and attitudes (Alam, 2022; Dita et al., 2021). Furthermore, Yu et al. (2021) analyzed that learning motivation has a direct correlation with learning outcomes, showing that both positive and negative impacts can arise from changes in learning motivation.

7. Conclusion

In summary, this study developed an educational robot to engage students in learning CT. The results of this study indicate that students who used the CT educational robot developed in this study achieved better learning achievements, motivation, and attitudes in elementary school education. Controlling the physical educational robot through an application helps students concretize abstract concepts, understand the relationship between code and the robot's movement, and further acquire abstract conceptual knowledge. Additionally, the self-assembled CT educational robot developed in this study was more cost-effective than commercially available educational robots.

However, the limitations encountered included insufficient experimental duration and a lack of diverse operational tasks. The curriculum design did not fully match the students' cognitive development, suggesting a need for continuous content updates and extended time with the CT educational robot. The intervention experiment in this study was conducted over a period of only three weeks, approximately 120 minutes in total, which may be insufficient. Future studies are recommended to extend the experimental period to 16 to 20 weeks to cover all three stages. Additionally, future plans should include different grade levels and design curricula that match students' cognitive abilities, as students' cognitive load directly affects their learning performance (Munandar et al., 2022; Zhampeissova et al., 2020).

It is hoped that further research can collect more samples for analysis and provide a multifaceted analysis of various student behaviors, such as high and low motivation and achievements, to explore the effect of the proposed approach on different types of students. Post-experiment analysis of various student aspects, such as those with high or low motivation and learning achievements, is also being planned.

8 Acknowledgements

This research was funded by National Science and Technology Council, Taiwan, R.O.C., grant numbers NSTC 111-2410-H-153-012, NSTC 112-2410-H-153-014, NSTC 110-2511-H-153-002-MY3, and NSTC 113-2410-H-153-018-MY2. This research was also funded by the Ministry of Education, Taiwan, R.O.C. under grants MOE PEE1121175 and MOE PED1136623.

9. References

Alam, A. (2022). Social robots in education for long-term human–robot interaction: Socially supportive behaviour of robotic tutor for creating robo-tangible learning environment in a guided discovery learning interaction. *ECS Transactions*, *107*(1), Article 12389. https://doi.org/10.1149/10701.12389ecst

Amri, S., Budiyanto, C., Fenyvesi, K., Yuana, R., & Widiastuti, I. (2022). Educational robotics: Evaluating the role of computational thinking in attaining 21st century skills. *Open Education Studies*, *4*, 322–338. https://doi.org/10.1515/edu-2022-0174

Barz, N., Benick, M., Dörrenbächer-Ulrich, L., & Perels, F. (2024). The effect of digital game-based learning interventions on cognitive, metacognitive, and affective-motivational learning outcomes in school: A meta-analysis. *Review of Educational Research*, *94*, 193–227. https://doi.org/10.3102/00346543231167795

Cervera, N., Diago, P. D., Orcos, L., & Yáñez, D. F. (2020). The acquisition of computational thinking through mentoring: An exploratory study. *Education Sciences*, *10*(8), Article 202. https://doi.org/10.3390/educsci10080202

Çetin, H., & Türkan, A. (2022). The effect of augmented reality based applications on achievement and attitude towards science course in distance education process. *Education and Information Technologies*, *27*, 1397–1415. https://doi.org/10.1007/s10639-021-10625-w

Chao, P.-Y. (2016). Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers & Education, 95*, 202–215. https://doi.org/10.1016/j.compedu.2016.01.010

Chevalier, M., Giang, C., Piatti, A., & Mondada, F. (2020). Fostering computational thinking through educational robotics: A model for creative computational problem solving. *International Journal of STEM Education*, *7*, Article 39. https://doi.org/10.1186/s40594-020-00238-z

Chiazzese, G., Arrigo, M., Chifari, A., Lonati, V., & Tosto, C. (2019). Educational robotics in primary school: Measuring the development of computational thinking skills with the Bebras tasks. *Informatics*, *6*(4), Article 43. https://doi.org/10.3390/informatics6040043

Ching, Y. H., & Hsu, Y. C. (2024). Educational robotics for developing computational thinking in young learners: A systematic review. *TechTrends*, *68*, 423–434. https://doi.org/10.1007/s11528-023-00841-1

Dağ, F., Şumuer, E., & Durdu, L. (2023). The effect of an unplugged coding course on primary school students' improvement in their computational thinking skills. *Journal of Computer Assisted Learning*, *39*(6), 1902–1918. https://doi.org/10.1111/jcal.12850

Dagiene, V., & Dolgopolovas, V. (2022). Short tasks for scaffolding computational thinking by the global Bebras challenge. *Mathematics*, *10*(17), Article 3194. https://doi.org/10.3390/math10173194

Dita, P. P. S., Murtono, Utomo, S., & Sekar, D. A. (2021). Implementation of problem based learning (PBL) on interactive learning media. *Journal of Technology and Humanities*, *2*(2), 24–30. https://doi.org/10.53797/jthkkss.v2i2.4.2021

Fang, J. W., Shao, D., Hwang, G. J., & Chang, S. C. (2022). From critique to computational thinking: A peer-assessment-supported problem identification, flow definition, coding, and testing approach for computer programming instruction. *Journal of Educational Computing Research*, *60*(5), 1301–1324. https://doi.org/10.1177/07356331211060470

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers and Education*, *126*, 296–310. https://doi.org/10.1016/j.compedu.2018.07.004

Hwang, G.-J., & Chang, H.-F. (2011). A formative assessment-based mobile learning approach to improving the learning attitudes and achievements of students. *Computers & Education*, *56*(4), 1023–1031. https://doi.org/10.1016/j.compedu.2010.12.002

Jawawi, D., Jamal, N., Abdul Halim, S., Sa'adon, N., Mamat, R., Isa, M., Mohamad, R., & Abdull Hamed, H. (2022). Nurturing secondary school student computational thinking through educational robotics. *International Journal of Emerging Technologies in Learning*, *17*(3), 117–128. https://doi.org/10.3991/ijet.v17i03.27311

Jiang, B., & Li, Z. (2021). Effect of Scratch on computational thinking skills of Chinese primary school students. *Journal of Computers in Education*, *8*, 505–525. https://doi.org/10.1007/s40692-021-00190-z

Kalaitzidou, M., & Pachidis, T. P. (2023). Recent robots in STEAM education. *Education Sciences*, *13*(3), Article 272. https://doi.org/10.3390/educsci13030272

Karaahmetoğlu, K., & Korkmaz, Ö. (2019). The effect of project-based arduino educational robot applications on students' computational thinking skills and their perception of basic STEM skill levels. *Participatory Educational Research*, *6*(2), 1–14. https://doi.org/10.17275/per.19.8.6.2

Kite, V., Park, S., & Wiebe, E. (2021). The code-centric nature of computational thinking education: A review of trends and issues in computational thinking education research. *SAGE Open*, *11*(2). https://doi.org/10.1177/21582440211016418

Kopcha, T. J., Ocak, C., & Qian, Y. (2021). Analyzing children's computational thinking through embodied interaction with technology: A multimodal perspective. *Educational Technology Research and Development*, *69*(4), 1987–2012. https://doi.org/10.1007/s11423-020-09832-y

Kramer, J. (2007). Is abstraction the key to computing? *Communications of the ACM*, *50*(4), 36–42. https://doi.org/10.1145/1232743.1232745

Lai, C.-F., Jeng, Y.-L., & Huang, S.-B. (2023). Improving programming skills: The use of learning style theory and the instant response supplement tool. *Library Hi Tech*, *41*(4), 1084–1098. https://doi.org/10.1108/LHT-03-2020-0059

Li, X., & Lee, J. J. (2024). Failure stories and surprising findings: Learning from when VR did not work. In G. Li, J. Filipe, A. Ghosh, & Z. Xu (Eds.), *Communications in computer and information science* (Vol. 1904, pp. 22–36). Springer. https://doi.org/10.1007/978-3-031-47328-9_2

Lin, Y. T. (2016). When mobile technology meets traditional classroom learning environment: How does it improve students' learning performances? In K. Wallace (Ed.), *Learning environments: Emerging theories, applications and future directions* (Chapter 8). Nova Science Publishers, Inc.

Lin, Y. T. (2023). Learning performances towards the book roll e-book system for flipped classrooms in software engineering education. *Educational Technology & Society, 26*(3), 190–202. https://psycnet.apa.org/record/2024-07480-013

Lodi, M., & Martini, S. (2021). Computational thinking, between Papert and Wing. *Science & Education*, *30*, 883–908. https://doi.org/10.1007/s11191-021-00202-5

Moore, T. J., Brophy, S. P., Tank, K. M., Lopez, R. D., Johnston, A. C., Hynes, M. M., & Gajdzik, E. (2020). Multiple representations in computational thinking tasks: A clinical study of second-grade students. *Journal of Science Education and Technology, 29*(1), 19–34. https://doi.org/10.1007/s10956-020-09812-0

Munandar, R. R., Rostikawati, T., & Susanto, L. H. (2022). Cognitive load lecture on diversity and classification of vertebrates through Moodle-based online learning. *Journal of Educational Sciences*, *6*(1), 107–115. https://doi.org/10.31258/jes.6.1.p.107-115

Mweshi, G. K., & Sakyi, K. (2020). Application of sampling methods for the research design. *Archives of Business Research, 8*(11), 180–193. https://doi.org/10.14738/abr.811.9042

Noh, J., & Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research and Development*, *68*, 463–484. https://doi.org/10.1007/s11423-019-09708-w

Ou Yang, F. C., Lai, H. M., & Wang, Y. W. (2023). Effect of augmented reality-based virtual educational robotics on programming students' enjoyment of learning, computational thinking skills, and academic achievement. *Computers & Education*, *195*, Article 104721. https://doi.org/10.1016/j.compedu.2022.104721

Pintrich, P. R., & de Groot, E. V. (1990). Motivational and self-regulated learning components of classroom academic performance. *Journal of Educational Psychology*, *82*(1), 33–40. https://doi.org/10.1037/0022-0663.82.1.33

Qu, J. R., & Fok, P. K. (2022). Cultivating students' computational thinking through student–robot interactions in robotics education. *International Journal of Technology and Design Education*, *32*, 1983–2002. https://doi.org/10.1007/s10798-021-09677-3

Samar, S., & Taima, S. (2019). *Computational thinking education: Who let the dog out?* [Conference session]. 2019 International Conference on Computational Science and Computational Intelligence (CSCI), April 20, 2020, Las Vegas, NV, USA. IEEE. https://doi.org/10.1109/CSCI49370.2019.00150

Shen, J., Chen, G., Barth-Cohen, L., Jiang, S., & Eltoukhy, M. (2022). Connecting computational thinking in everyday reasoning and programming for elementary school students. *Journal of Research on Technology in Education, 54*, 205–225. https://doi.org/10.1080/15391523.2020.1834474

Stewart, W., & Baek, K. (2023). Analyzing computational thinking studies in Scratch programming: A review of elementary education literature. *International Journal of Computer Science Education in Schools*, *6*(1), 35–58. https://doi.org/10.21585/ijcses.v6i1.156

Sun, L., & Liu, J. (2023). Effects of gamified Python programming on primary school students' computational thinking skills: A differential analysis of gender. *Journal of Educational Computing Research*, *62*(3), 846–874. https://doi.org/10.1177/07356331231225269

Tavakol, M., & Dennick, R. (2011). Making sense of Cronbach's alpha. *International Journal of Medical Education, 2*, 53–55. https://doi.org/10.5116/ijme.4dfb.8dfd

Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature review. *Computers & Education*, *162*, Article 104083. https://doi.org/10.1016/j.compedu.2020.104083

Valentina, D., Tatjana, J., Gabrielė, S., & Anita, J. (2022). Teaching computational thinking in primary schools: Worldwide trends and teachers' attitudes. *Computer Science and Information Systems*, *19*(1), 1–24. https://doi.org/10.2298/CSIS201215033D

Wei, C.-W., Lin, Y.-C., & Lin, Y.-T. (2016). An interactive diagnosis approach for supporting clinical nursing courses. *Interactive Learning Environments*, *24*(8), 1795−1811. https://doi.org/10.1080/10494820.2015.1057741

Wong, G. K. W., & Jiang, S. (2018). *Computational thinking education for children: Algorithmic thinking and debugging* [Conference session]. 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), January 17, 2019, Wollongong, NSW, Australia. IEEE. https://doi.org/10.1109/TALE.2018.8615232

Wong, G. K.-W., & Cheung, H.-Y. (2020). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments*, *28*(4), 438–450. https://doi.org/10.1080/10494820.2018.1534245

Yıldız, T., & Seferoğlu, S. S. (2021). The effect of robotic programming on coding attitude and computational thinking skills toward self-efficacy perception. *Journal of Learning and Teaching in Digital Age, 6*(2), 101–116. https://files.eric.ed.gov/fulltext/EJ1308356.pdf

Yu, Z., Gao, M., & Wang, L. (2021). The effect of educational games on learning outcomes, student motivation, engagement and satisfaction. *Journal of Educational Computing Research*, *59*, 522–546. https://doi.org/10.1177/0735633120969214

Zhampeissova, K., Gura, A., Vanina, E., & Egorova, Z. (2020). Academic performance and cognitive load in mobile learning. *International Journal of Interactive Mobile Technologies*, *14*(21), 78–91. https://doi.org/10.3991/IJIM.V14I21.18439

Zhang, Y., Luo, R., Zhu, Y., & Yin, Y. (2021). Educational robots improve K-12 students' computational thinking and STEM attitudes: Systematic review. *Journal of Educational Computing Research, 59*(7), 1450–1481. https://doi.org/10.1177/0735633121994070

Zhao, L., Liu, X., Wang, C., & Su, Y. (2022). Effect of different mind mapping approaches on primary school students' computational thinking skills during visual programming learning. *Computers & Education*, *181*, Article 104445. https://doi.org/10.1016/j.compedu.2022.104445