Effect of unplugged and plugged coding activities on secondary school students' computational thinking skills

Tunahan Yılmaz [a] iD, Serkan İzmirli [b] * iD

[a] Ministry of National Education, Türkiye.
[b] Çanakkale Onsekiz Mart University, Türkiye.

Highlights	Abstract
• This research focused on the investigation of both the perception and performance of computation thinking skills. • The unplugged coding group demonstrated a notable improvement in both perception and performance of computational thinking skills in the post-test compared to the pre-test. • Within the group engaged in plugged coding, there was a notable improvement in computational thinking skills performance during the post-test compared to the pre-test, despite the absence of a significant shift in perception. • No significant difference was observed in terms of the perception of computational thinking skills between the plugged and unplugged coding groups, and the same was found in their performance as well. • Students' computation thinking perceptions highly aligned with their computational thinking performance.	The objective of this research was to investigate the influence of both unplugged and plugged coding activities on the computational thinking skills of secondary school students. Using an experimental design with a pretest-posttest control group, the study indicated that students enhanced their computational thinking skills through engagement in coding activities. Specifically, students in the experimental group, participating in unplugged coding activities (Tospaa unplugged coding activities), exhibited heightened perception and improved performance in computational thinking skills in the post-test compared to the pre-test. Conversely, the control group, involved in plugged coding activities (Scratch block-based coding), did not show a significant change in perception of computational thinking skills in the post-test compared to the pre-test. However, their performance in computational thinking skills improved significantly in the post-test compared to the pre-test. Furthermore, no significant differences were observed in terms of the perception of computational thinking skills between the unplugged coding group and the plugged coding group, and similarly, no significant differences were found in terms of the computational thinking skills performance between unplugged and plugged groups. In conclusion, the study also shows a high alignment between students' perceptions of computational thinking and their actual performance in computational thinking.

1. Introduction

Some competencies are expected from individuals in order to raise a society that produces and develops both itself and its country (Uçak & Erdem, 2020). In order for individuals to adapt to the century in which

they live, the International Society for Technology in Education (ISTE) has identified the competencies that individuals should acquire in the 21st century. The International Society for Technology in Education emphasizes that these competencies, comparable to literacy, should be acquired by all students. Among the expected proficiencies in every individual, in addition to skills such as self-learning, conscientious use of the digital environment according to rules, and collaborative work, is "computational thinking" (ISTE, 2011).

In accordance with Wing (2006), computational thinking (CT) encompasses problem-solving, system design, and comprehension of human behavior through the fundamental principles of computer science. ISTE (2011) defines CT as a problem-solving approach, emphasizing its significance as a fundamental skill not exclusive to computer scientists but essential for all individuals (Wing, 2006). Brennan and Resnick (2012) delve into the multifaceted nature of CT, identifying three dimensions: computational concepts, practices, and perspectives. Computational concepts refer to the concepts utilized by designers during programming (e.g. loops, sequences, conditionals), while computational practices focusing on the process of thinking are the practices that designers establish when working with concepts (e.g. debugging, reusing). Computational perspectives are the perspectives that designers create regarding themselves, others and the technological world. A change in the designer's perspective about CT occurs after participating in a CT skill development activity (e.g. Scratch). The International Society for Technology in Education (ISTE) stated that CT skills are important in 21st century skills and added computational thinking to its standards (ISTE, 2023). Code.org has added different activities to its website to provide individuals with CT skills. The Computer Science Teachers Association (CSTA) included concepts such as algorithm, problem solving, and programming, which are sub-dimensions of CT skills, in its standards. "Unplugged" and "plugged" coding activities are seen as important tools to help children develop CT skills (Fagerlund et al., 2021; Kafai & Burke, 2013; Sun et al., 2021; Zhao & Shute, 2020).

1.1. Plugged and Unplugged Coding

The concept of coding can be defined as writing code to program a computer (Zhang & Nouri, 2019). Coding is the ability of students to design and execute algorithmic processes. Put differently, it is the capacity of students to instruct the computer on how to address a problem (Popat & Starkey, 2019). In these definitions, it is stated that coding is done with a device such as a computer and a tablet. Therefore, coding done with a device such as a computer and a tablet can be called "plugged coding". Plugged coding can be carried out using text-based programming languages like Python or block-based coding tools such as Scratch. Block-based coding tools like Scratch resemble games more than traditional programming languages, and they can serve as an entry point to coding for young children (Geist, 2016). Block-based coding (e.g. Scratch), which is one of the plugged coding methods especially for students in middle school and lower age groups, facilitates coding teaching and makes it fun (Aytekin et al., 2018; Çatlak et al., 2015), increases self-confidence and curiosity (Totan, 2021), and improves problem solving and critical thinking skills (Yükseltürk & Altıok, 2015).

"Unplugged coding" is an activity that enables learning computer science concepts such as algorithm creation and looping with pencil, paper, cards, logic games or simple body movements (Sigayret et al., 2022). Children as young as 5 years of age can grasp coding by physically transforming programming into a system of continuously dropping blocks. Games like Robot Turtles, a board game, can offer an early introduction to coding logic for children at an early age, all without the necessity of using a computer (Geist, 2016). As an example of computer-free coding, websites such as https://www.csunplugged.org and https://tospaa.org have printable teaching materials for teaching computer science without using a computer with cards, puzzles and games. Unplugged coding is fun, exciting and helpful in teaching basic coding concepts for beginners (Kalelioğlu, 2017; Kırçalı, 2019). Physical games such as Robot Turtles and Tospaa.org, which do not require using a computer, are called unplugged coding. It can be stated that both block-based coding (Sctatch), one of the plugged coding methods, and unplugged coding (tospaa.org) are effective in teaching coding to children.

1.2. Plugged and Unplugged Coding for Computational Thinking

Within the literature, there are investigations that analyze the impact of plugged coding on CT perception or performance. These studies employ a single-group pretest-posttest experimental design, which is considered a less robust research methodology. In one of these studies, Ataman-Uslu et al. (2018) found that Scratch coding activities did not significantly contribute to the CT perceptions of 6th grade middle school students. Similarly, in another study, Aydoğdu (2020) concluded that block-based coding instruction with Scratch had no effect on university students' CT perceptions. In contrast, Pérez-Marín et al. (2020) found that teaching coding with Scratch significantly increased the CT performance of 4th, 5th, and 6th grade students. In studies on block-based coding (Scratch), it was observed that students' CT perceptions or performances either did not change or increased. In one of the studies on robotic activities, one of the plugged coding methods, Bal (2019) found that robotic coding activities with Ardunio improved middle school students' CT perceptions. Ramazanoğlu (2021) also found that robotic coding activities with mbot significantly increased the CT perceptions of 11th grade students. In one of the studies on text-based coding, which is one of the plugged coding methods, Alsancak-Sırakaya (2019) found that programming instruction using C# programming language significantly increased the CT perceptions of first-year university students. It can be stated that the effect of plugged coding on CT perception or performance cannot be fully revealed since there is no control group in the studies using one-group pretest-posttest experimental design, which is a weak design in the literature. In addition to single-group experimental studies, there are also studies with control groups in the field. In one of these studies, Piedade and Dorotea (2023) examined the influence of Scratch block-based coding activities on CT performance of 4th grade students. In the study, which employed a quasi-experimental design with a posttest control group, students who participated in the Scratch coding activity exhibited significantly higher CT performance than the group that did not receive Scratch or engage in any Computer Science activity. However, the absence of a pretest in this study and the fact that the control group did not partake in any Computer Science activity make it unclear whether the improvement in CT performance can be attributed specifically to the computer science course or the Scratch activities.

In the literature, there are studies exploring how unplugged coding influences perceptions or performance in computational thinking. There are studies using a single group pretest-posttest experimental design. In one of these studies, Delal and Oner (2020) concluded that the CT performance of 6th grade students who received unplugged coding training increased significantly. In a study by Threekunprapa and Yasri (2020), the CT performance of middle school students who performed unplugged coding activities increased significantly. Similarly, Dağ et al. (2023) found that CT performance of primary school 3rd and 4th graders increased. Tonbuloğlu and Tonbuloğlu (2019) examined the CT perceptions of 5th grade students who performed unplugged coding activities using a one-group pretest-posttest experimental design in the quantitative dimension of their mixed-method study. At the end of the study, they found that unplugged coding positively affected students' CT perceptions. In addition to the studies employing a one-group experimental design, Relkin et al. (2021) investigated the impact of unplugged coding training using a pre-assembled robot kit on the CT skills of 1st and 2nd-grade students. Using a quasi-experimental design with a control group, the experimental group underwent training with a robot kit, while the control group received instruction through traditional classroom activities without coding. The results showed that the CT performances of the group trained with the robot kit surpassed those of the group that did not receive coding training. Sun et al. (2021) explored the influence of unplugged coding activities on CT skills in their study, which utilized a quasi-experimental design and involved 7th-grade students. For this study, two experimental groups and one control group were established. The experimental groups were given unplugged activities and the control group was given basic computer knowledge without unplugged activities. According to the findings obtained, the CT performances of both experimental groups increased significantly compared to the control group. In the literature, previous studies showed that unplugged coding activities increase students' CT perceptions and performances.

There are a limited number of studies in the literature comparing the effects of plugged and unplugged coding activities on CT. In one of these studies, del Olmo-Muñoz et al. (2020) found that the CT

performance of 2nd grade students who received unplugged coding training was significantly higher than those who received plugged coding training. Similarly, in Sun and Liu's (2023) study with first grade students, the unplugged group showed better CT performance than the plugged group. Kırçali and Özdener (2023), in their study with 6th grade students, did not find a significant difference between the CT perceptions of the group receiving unplugged coding training and the groups receiving plugged coding training.

Upon reviewing the studies in the literature, it is seen that coding education contributes positively to students' CT skills (Erümit et al., 2020; Kaya et al., 2020; Pérez-Marín et al., 2020; Oluk et al., 2018; Zhaou & Shute, 2019). Nevertheless, to heighten the validity of studies in the literature, more research employing experimental designs is needed on teaching CT skills through coding (Ataman-Uslu et al., 2018; Zhang & Nouri, 2019). In addition, different coding activities and different methods were used in different studies (Erümit et al., 2020; Kaya, et al., 2020; del Olmo-Muñoz et al., 2020). It is necessary to compare how different programming languages and tools improve CT skills (Zhang & Nouri, 2019). In the literature, there is a limited number of studies comparing the effects of plugged and unplugged coding activities on CT skills. Moreover, it has also been observed that perception scales or performance tests were used to measure CT. In this context, this study examines the effects of plugged and unplugged coding instruction on both CT perception and CT performance. For unplugged coding, paper-based applications were made with outputs from Tospaa.org, while for plugged coding, the frequently used Scratch block-based coding tool was used.

1.3. Purpose and Research Questions

The main purpose of this study is to examine the effects of unplugged and plugged coding activities on 5th grade students' CT perceptions and performances. The following questions were addressed for the main purpose of the study:

1. What is the effect of unplugged coding (Tospaa) on improving students' **perception of CT skills**?
2. What is the effect of unplugged coding (Tospaa) on improving students' **CT skills performance**?
3. What is the effect of plugged coding (Scratch) on improving students' **perception of CT skills**?
4. What is the effect of plugged coding (Scratch) on improving students' **CT skills performance**?
5. Is there a significant difference between the **perception of CT skills** of the group trained with unplugged coding (Tospaa) and the group trained with plugged coding (Scratch)?
6. Is there a significant difference between the **CT skills performance** of the group trained with unplugged coding (Tospaa) and the group trained with plugged coding (Scratch)?

2. Methodology

2.1. Research Model

The research employed a quasi-experimental design involving a pretest-posttest control group. One of the two existing groups was randomly designated as the experimental group, while the other served as the control group. The study was carried out in two randomly selected classes from the 5th-grade level of a school, with one class assigned to the experimental group and the other to the control group. A detailed explanation of the experimental design is provided in Table 1.

Table 1.

Research design

Group	Pretest	Experimental Procedure	Posttest
Experimental (18 students)	Computational thinking skill levels scale Computational thinking test	Coding education with Tospaa unplugged coding game	Computational thinking skill levels scale Computational thinking test
Control (18 students)	Computational thinking skill levels scale Computational thinking test	Coding education with Scratch block-based coding tool	Computational thinking skill levels scale Computational thinking test

As indicated in Table 1, both the computational skill levels scale (assessing perception) and the computational thinking test (evaluating performance) were given as pretests to all groups before the study. Subsequently, the experimental group underwent training with the Tospaa unplugged coding game, while the control group received training with the Scratch block-based coding tool over a period of 5 weeks. Following the training, the computational thinking skills scale and computational thinking test were administered to all groups as posttests.

2.2. Participants

This study involved 36 students from the 5th grade attending a public school in Istanbul, Turkey. The experimental group comprised 18 students (7 girls and 11 boys), aged 10-12, who had not previously received training in Tospaa coding activities or the Scratch block-based coding tool. Similarly, the control group consisted of 18 students (7 girls and 11 boys) within the same age range who had not been exposed to Tospaa coding activities or the Scratch tool.

2.3. Data Collection Tools

In the study, students' CT skill levels scale and CT test were used.

2.3.1. Computational Thinking Skill Levels Scale

The assessment of students' computational thinking perception utilized the "CT Skill Levels Scale," originally developed by Korkmaz et al. (2017) for university students and later adapted for the secondary school students by Korkmaz et al. (2015). This scale, administered as both a pretest and posttest during the experimental process, follows a 5-point Likert type format and comprises 22 items. It encompasses five sub-factors: creativity, algorithmic thinking, collaboration, critical thinking, and problem-solving. The scale demonstrates a strong internal consistency with a coefficient of .81. A sample scale item is "*I cannot produce so many options while thinking of the possible solution ways regarding a problem.*" (Korkmaz et al., 2015). In the present study, the Cronbach's Alpha reliability coefficient for the scale was .92 for the experimental group and .80 for the control group.

2.3.2. Computational Thinking Test

The "Computational Thinking Test", initially developed in Spanish and validated for content by Roman-Gonzalez (2015), underwent a criterion validity study conducted by Román-González et al. (2017). It was subsequently adapted into Turkish by Çetin et al. (2020) and Turkish version was employed as both a pretest and posttest to assess students' computational thinking (CT) performance in this study. While the original test contained 28 items, the Turkish-adapted version consisted of 24 items after analysis. The CT performance test covers questions pertaining to various concepts: 3 questions for basic directions and sequences, 3 questions for loops-repeat until, 4 questions for loops-repeat times, 3 questions for if - simple conditional, 4 questions for if - else complex conditional, 3 questions for while conditional, and 4 questions for simple functions. The test is of moderate difficulty (item difficulty index: .53), and the average item discrimination index is .47. The KR20 internal consistency value for the test is .78 (Çetin et al., 2020). A sample question item was given in Figure 1. In the current study, the KR20 internal consistency coefficient was .74 for the experimental group and .62 for the control group.
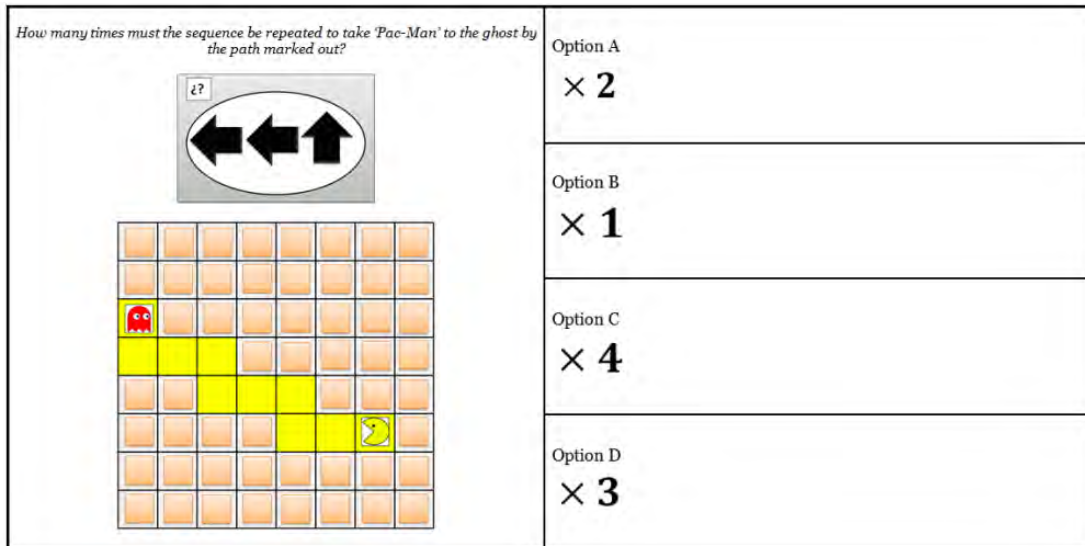
Fig. 1. A sample question in CT Performance test (Román-González et al., 2017) (Turkish version was used in this study)

2.4. Implementation Process

The study involved 5th-grade students in a public school, with two classes randomly chosen for participation. Subsequently, one of these classes was randomly assigned as the control group, and the other as the experimental group. The entire implementation spanned 7 weeks, equivalent to 14 class hours. In the initial week, both groups underwent pretests, involving the computational thinking skills perception scale and the computational thinking performance test. Following this, for a duration of five weeks, both the experimental and control groups received training on linear program writing, looping, and conditions (both simple and complex). The specific activities conducted each week in the experimental group were as follows:

- In the experimental group, the subject of that week was covered with the Tospaa unplugged coding game. Tospaa is a coding game made with various cards without using a computer, phone or tablet (Tospaa, 2023)
- One Tospaa game card was provided for each two students in the experimental group. However, each student was allowed to do the activity related to the topic using the cards under the guidance of the instructor. Tospaa unplugged coding game consists of 29 cards, a game board and various blocks (water, stone, flag). Each card has a different scenario. They were asked to create the scenarios on the cards by placing water, stones, target and other blocks on the game board. They were then asked to create algorithms to reach the goal on the card by placing the blocks on the side of the game board.
- For example, activities 8-16 on the Tospaa cards related to the topic of "looping" were carried out (Figure 2).

Fig. 2. Tospaa application on looping

The activities carried out each week in the control group were as follows.

- In the control group, the topic of that week was covered with Scratch unplugged coding game. Scratch is a block-based coding tool used to teach the basics of algorithm development and coding to students aged 8-16 (Scratch, 2023). This group was the control group since lessons were already routinely taught with a block-based coding program such as Scratch.

- In the control group, there was only one computer for every two students. However, Scratch applications were provided for each student.

- For example, the activities in Figure 3 were carried out with the Scratch block-based coding tool on the topic of "looping".
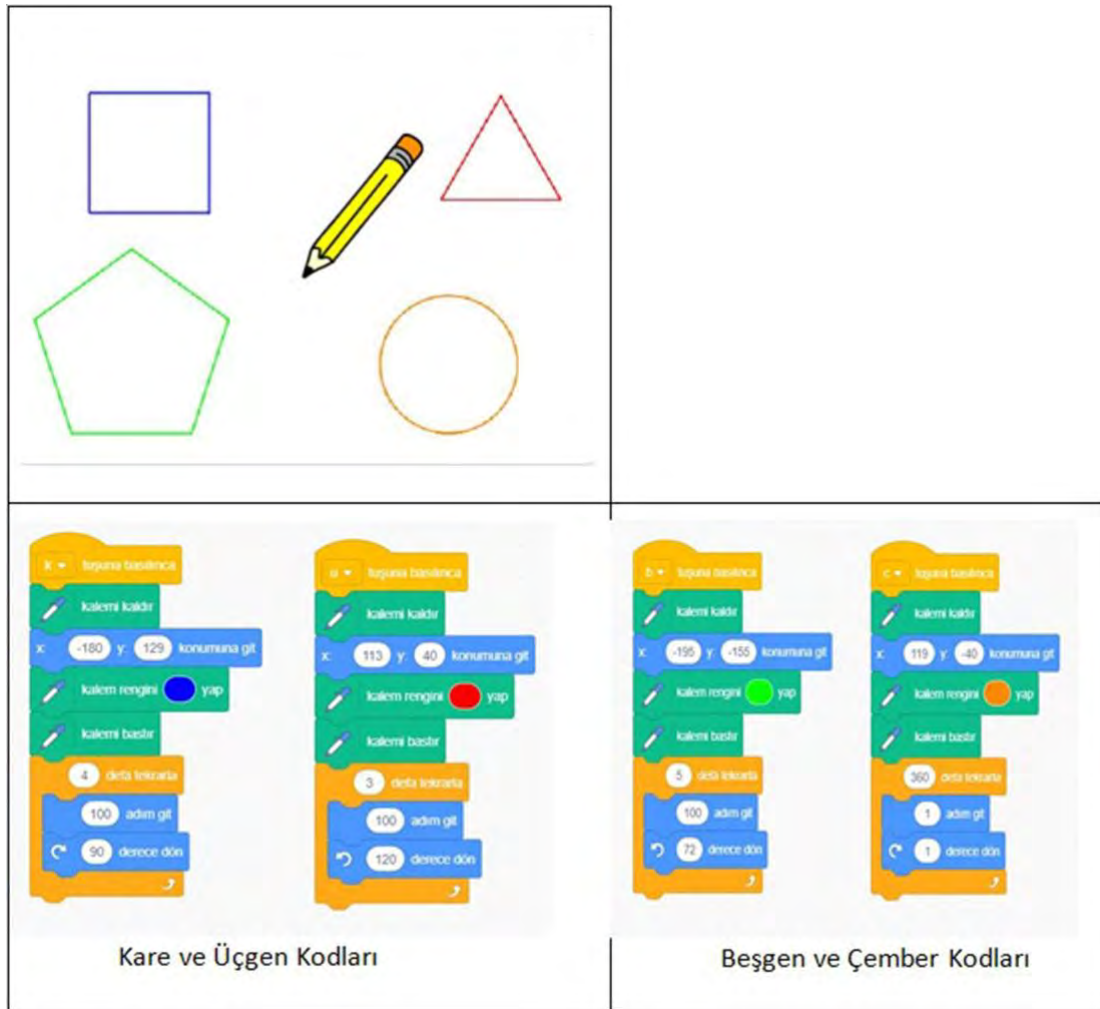
Fig. 3. Scratch application related to the topic of looping (Text on the left: "Square and Triangle Codes"; Text on the right: "Pentagon and Circle Codes")

After 5 weeks of training, the perception of computational thinking skills scale and the computational thinking performance test were administered to both groups as a posttest in the 7th week.

2.5. Data Analysis

The significance level for data analysis was set at .05. To assess the normal distribution of the data, both Skewness-Kurtosis and Shapiro-Wilk normality values were examined. Skewness kurtosis values falling within the range of -1 to +1 are indicative of a normal distribution (Hair et al., 2013). Additionally, for groups with 29 or fewer participants, the Shapiro-Wilk test was employed to assess normality, while the Kolmogorov-Smirnov test was used for groups with more than 29 individuals. The Cohen-d value was computed to determine the effect size in significant findings, with a value of .2 denoting a small effect size, .5 indicating a medium effect size, .8 signifying a large effect size, and values above 1 suggesting a very large effect size (Can, 2013).

Given that the prerequisites were satisfied for questions 1, 2, and 3 of the study, these were analyzed using the paired samples t-test. For the 4th question, where prerequisites were not met, the nonparametric Wilcoxon Signed Ranks Test was employed. Questions 5 and 6 were subjected to analysis using the independent samples t-test, a parametric test, as the prerequisites were met.

2.6. Ethical Procedures

The study received approval from the Çanakkale Onsekiz Mart University School of Graduate Studies Scientific Research Ethics Committee (Date: 17.03.2022, Number: 06/17). Permission for data collection was granted by the Istanbul Provincial Directorate of National Education. Furthermore, consent was obtained from the parents of participants in the study, given that their ages were below 18.

3. Findings

3.1. Perception of Computational Thinking Skills of the Group Receiving Unplugged Coding Training

In order to examine the difference in CT perceptions of the students in the experimental group who received Unplugged coding training, a paired-samples t-test analysis was used as one of the parametric tests since the data were normally distributed. According to the results of the analysis, the CT perceptions of the students in the experimental group who received unplugged coding training increased significantly from pretest ($\bar{x}$=68.78; SD=8.64) to posttest ($\bar{x}$=72.16; SD=10.69) ($t_{(17)}$=2.77; p=.013<.05). Cohen-d value was calculated for the effect size of the difference between the two groups. The calculated Cohen-d value was .92, which indicates a large effect size.

3.2. Computational Thinking Skills Performance of the Group Receiving Unplugged Coding Training

Since the data were normally distributed, a paired samples t-test analysis was used to examine the change in CT performance of the students in the experimental group who received computer coding training. According to the results of the analysis, the CT performance of the group receiving unplugged coding training increased significantly from pretest ($\bar{x}$=10.67; SD=2.47) to posttest ($\bar{x}$=14.50; SD=3.38) ($t_{(17)}$=5.26; p=.000<.001). Cohen-d was calculated for the effect size of the difference between the two groups. The calculated Cohen-d was 1.75, which indicates a very large effect.

3.3. Perception of Computational Thinking Skills of the Group Receiving Plugged Coding Training

To investigate the difference in the CT perceptions of the students in the control group who received plugged coding training with Scratch, a paired samples t-test analysis was performed since the data showed a normal distribution. According to the results of the analysis, the CT perception of the group trained with Scratch did not change from pretest ($\bar{x}$=69.65; SD=9.61) to posttest ($\bar{x}$=72.52; SD=9.58) ($t_{(17)}$=1.56; p=.136>.05).

3.4. Computational Thinking Skills Performance of the Group Receiving Plugged Coding Training

In order to examine the change in CT performance of the students in the control group who received plugged coding training with Scratch, Wilcoxon signed-rank test was used as a non-parametric test since the data did not show normal distribution (Table 2).

Table 2.

Computational thinking skills performance of the block-based coding training group - wilcoxon signed-rank test result

Pretest-Posttest	n	Mean Ranks	Sum of Ranks	z	p	Cohen-d
Negative Ranks	1	4.00	4.00	-3.07	.002*	1.02
Positive Ranks	13	7.77	101.00			
Ties	4					

*: p<.05

According to the results of the analysis, the CT performance of the group trained with Scratch increased significantly from pretest to posttest (z=-3.07, p<.05). Cohen-d was calculated for the effect size of the difference between the two groups. The calculated Cohen-d was 1.02, which indicates a very large effect.

3.5. Comparison of the Perceptions of the Group Receiving Unplugged Coding Training and the Group Receiving Plugged Coding Training on Computational Thinking Skills

In order to examine whether there was a difference between the CT perceptions of the students in the experimental and control groups, the pretest mean scores of the two groups were first compared. No significant difference was found between the CT perception pretest mean score ($\bar{x}$=69.65) of the control group and the CT perception pretest mean score ($\bar{x}$=68.78) of the experimental group (p=.777>.05). For this reason, posttests were compared with independent sample t-test to examine the difference between

the two groups. There was no significant difference between the CT perception of the group receiving unplugged coding training (x̄=72.16; SD=8.64) and the group receiving plugged coding training (x̄=72.52; SD=10.69) ($t_{(34)}$= -.105; p=.917>.05).

3.6. Comparison of Computational Thinking Skills Performance of the Group Receiving Unplugged Coding Training and the Group Receiving Plugged Coding Training

In order to examine whether there was a difference between the CT performances of the students in the experimental and control groups, the pretest mean scores of the two groups were first compared. No significant difference was found between the CT performance pretest mean score (x̄=10.17) of the control group and the CT performance pretest mean score (x̄=10.67) of the experimental group (p>.05). For this reason, in order to examine the difference between the two groups, the posttests were compared with independent sample t-test analysis. According to the results of the analysis, there was no significant difference between the CT performances of the group receiving unplugged coding training (x̄=14.50; SD=3.38) and the group receiving plugged coding training (x̄=12.67; SD=2.95) ($t_{(34)}$= 1.73; p=.092>.05).

4. Discussion and Conclusion

In this study, the CT perceptions of the students in the experimental group who received unplugged coding training with Tospaa increased significantly from pretest to posttest. In parallel with this finding regarding CT perception, in Tonbuloğlu and Tonbuloğlu's (2019) study, unplugged coding positively affected CT perceptions. In this study, the CT performance of the group that received unplugged coding training with Tospaa also increased significantly from pretest to posttest. These results are in line with the studies examining CT performance at different levels of education in the related literature. Relkin et al. (2021) with 1st and 2nd graders, Dağ et al. (2023) with 3rd graders, Delal and Oner (2020) with 6th graders, Sun et al. (2021) with 7th graders, and Threekunprapa and Yasri (2020) with middle school students found that unplugged coding training increased significantly CT performance.

In the present research, the CT perception of the group receiving plugged coding training with Scratch did not change from pretest to posttest. On the contrary, the CT performance of the group receiving plugged coding training with Scratch increased significantly from pretest to posttest. In the literature, it is seen that there is a similar trend that CT performance increased while CT perception did not change in general. Among the studies examining CT perception, Ataman-Uslu et al. (2018) with 6th grade students and Aydoğdu (2020) with university students found that Scratch did not change the CT perception. However, on the contrary, Dikkartın Övez and Acar (2022) found that the CT perceptions of 7th grade students who received Scratch coding training with goal-based scenario approach changed positively. The reason for the positive development of students' CT perceptions may be the goal-based scenario approach used. Among the studies examining CT performance in the literature, Pérez-Marín et al. (2020) with 4th, 5th, and 6th grade students and Piedade and Dorotea (2023) with 4th grade students found that teaching coding with Scratch significantly increased the CT performance. Contrary to the results of this study, different findings were found in the literature where robotic activities and text-based coding were used as plugged coding methods. Bal (2019) found that robotic coding activities improved CT perceptions of middle school students and Ramazanoğlu (2021) found that robotic coding activities improved CT perceptions of 11th grade students. In addition, Alsancak-Sırakaya (2019) found that teaching programming using C# programming language increased the CT perceptions of first-year university students.

In this study, there was no significant difference between the CT perceptions of 5th grade students who received unplugged coding training and those who received plugged coding training. No significant difference was also found between the CT performances of both groups. Similarly, there was no significant difference between the groups' CT perceptions in Kırçali and Özdener's (2023) study involving 6th graders. In contrast, del Olmo-Muñoz et al. (2020) with 2nd grade students and Sun and Liu (2023) with 1st grade students found that the unplugged group showed significantly better CT performance than the plugged group. The reason for this difference in findings may be the different age groups of the target group.

It is stated that there are some problems in the literature on the measurement of computational thinking skills. Some studies use performance tests that focus on problem solving to measure CT, while other studies use specific questions that measure each CT skill separately (Zhang & Nouri). In this study, CT skills were measured with both a performance test and a perception scale. It was also concluded that the results of the performance test and the perception scale were largely parallel to each other.

5. Limitations and Suggestions

This study examined the effects of plugged and unplugged coding on 5th grade students' CT perceptions and performance. According to the results of the study, teachers can benefit from unplugged coding (e.g. Tospaa) and plugged coding (e.g. Scratch) activities to improve students' CT skills. In the literature, there are many studies examining the effect of plugged-unplugged coding on CT perception or performance, which are designed with a one-group pretest-posttest model, which is a weak design (e.g., Alsancak-Sırakaya, et al. Alsancak-Sırakaya, 2019; Ataman-Uslu et al., 2018; Aydoğdu, 2020; Bal, 2019; Delal & Oner, 2020; Pérez-Marín et al., 2020; Ramazonoğlu, 2021; Threekunprapa & Yasri, 2020). Although this study tried to overcome this methodological limitation by using a pretest-posttest control group design, similar studies should be repeated in terms of generalizability of the research results. Tospaa.org activities were used for unplugged coding activities and Scratch block-based coding tool was used for plugged coding activities in this study. In future studies, different tools can be used for plugged coding activities (e.g. Blockly) and different activities can be used for unplugged coding activities (e.g. https://www.csunplugged.org/). In plugged coding activities for children, it is stated that the logic of coding cannot be fully conveyed with drag and drop structure using ready-made code blocks (Wohl et al., 2015). For this reason, it can be stated that it is important for children to first learn the logic of coding by using unplugged coding activities. In this context, in order to compare the effects of plugged and unplugged coding on CT from this point of view, one group can be first unplugged and then plugged and the other group can be given the opposite by using counterbalanced design, one of the experimental methods (e.g. Sun & Liu, 2023).

References

Alsancak-Sırakaya, D. (2019). Programlama öğretiminin bilgi işlemsel düşünme becerisine etkisi. *Türkiye Sosyal Araştırmalar Dergisi, 23*(2), 575-590. https://dergipark.org.tr/en/pub/tsadergisi/issue/47639/448409

Ataman-Uslu, N., Mumcu, F., & Eğin, F. (2018). Görsel programlama etkinliklerinin ortaokul öğrencilerinin bilgi-işlemsel düşünme becerilerine etkisi. *Ege Eğitim Teknolojileri Dergisi, 2*(1), 19-31. https://dergipark.org.tr/tr/pub/eetd/issue/38495/410699

Aydoğdu, Ş. (2020). Blok tabanlı programlama etkinliklerinin öğretmen adaylarının programlamaya ilişkin öz yeterlilik algılarına ve hesaplamalı düşünme becerilerine etkisi. *Eğitim Teknolojisi Kuram ve Uygulama, 10*(1), 303-320. https://doi.org/10.17943/etku.649585

Aytekin, A., Sönmez Çakır, F., Yücel, Y. B., & Kulaözü, İ. (2018). Geleceğe yön veren kodlama bilimi ve kodlama öğrenmede kullanılabilecek bazı yöntemler. *Avrasya Sosyal ve Ekonomi Araştırmaları Dergisi, 5*(5), 24-41. https://dergipark.org.tr/en/pub/asead/issue/40925/494055

Bal, N. (2019). *Temel robotik eğitiminin ortaokul öğrencilerin 21. yy becerilerine ve bilgi işlemsel düşünme becerilerine etkisi* (Yayınlanmamış Yüksek Lisans Tezi). Hatay Mustafa Kemal Üniversitesi, Hatay.

Brennan, K., & Resnick, M. (2012, June). New frameworks for studying and assessing the development of computational thinking. Paper presented at the 2012 Annual Meeting of the American Educational Research.

Can, A. (2013). *SPSS ile bilimsel araştırma sürecinde nicel veri analizi* (11th ed.). Pegem.

Çatlak, Ş., Tekdal, M., & Baz, F. (2015). Scratch yazılımı ile programlama öğretiminin durumu: Bir doküman inceleme çalışması. *Journal of Instructional Technologies and Teacher Education, 4* (3). 13-25. https://dergipark.org.tr/en/pub/jitte/issue/25088/264774

Çetin, İ., Otu, T., & Oktaç, A. (2020). Adaption of the computational thinking test into Turkish. *Turkish Journal of Computer and Mathematics Education (TURCOMAT), 11*(2), 343-360. https://doi.org/10.16949/turkbilmat.643709

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

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education, 150,* 103832. https://doi.org/10.1016/j.compedu.2020.103832

Delal, H., & Oner, D. (2020). Developing middle school students' computational thinking skills using unplugged computing activities. *Informatics in Education, 19*(1), 1-13. https://doi.org/10.15388/infedu.2020.01

Dikkartın Övez, F.T., & Acar, İ. G. (2022). The effect of block-based game development activities on the geometry achievement, computational thinking skills and opinions of seventh-grade students. *Journal of Educational Technology & Online Learning, 5*(4), 1106-1121. http://doi.org/10.31681/jetol.1151170

Erümit, A. K., Şahin, G., & Karal, H. (2020). Yap programlama öğretim modelinin öğrencilerin bilgi-işlemsel düşünme becerilerine etkisi. *Kastamonu Eğitim Dergisi, 28*(3), 1529-1540. https://doi.org/10.24106/kefdergi.3915

Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education, 29*(1), 12-28.

Geist, E. (2016). Robots, programming and coding, oh my! *Childhood Education, 92*(4), 298-304. https://doi.org/10.1080/00094056.2016.1208008

Hair, J. F., Hult, G. T. M., Ringle, C., & Sarstedt, M. (2013). *A primer on partial least squares structural equation modeling* (2nd ed.). Sage.

ISTE (2011). Operational definition of computational thinking for K–12 education. Access: https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf?_ga=2.143521276.87737870.1700361224-768374804.1700133162

ISTE (2023). ISTE standards: For students. Access: https://iste.org/standards/students#1-5-computational-thinker

Kafai, Y. B., & Burke, Q. (2013). Computer programming goes back to school. *Phi Delta Kappan, 95*(1), 61-65. https://doi.org/10.1177/003172171309500111

Kalelioğlu, F. (2017). Unplugged bilgisayar bilimi (B3) öğretimi. Y. Gülbahar (Ed.). *Bilgi işlemsel düşünmeden programlamaya* içinde (ss. 155-178). Pegem.

Kaya, M., Korkmaz, Ö., & Çakır, R. (2020). Oyunlaştırılmış robot etkinliklerinin ortaokul öğrencilerinin problem çözme ve bilgi işlemsel düşünme becerilerine etkisi. *Ege Eğitim Dergisi, 21*(1), 54-70. https://doi.org/10.12984/egeefd.588512

Kırçalı, A. Ç. (2019). *K12 düzeyinde algoritma öğretiminde kullanılan plugged ve unplugged araçların çeşitli değişkenler açısından değerlendirilmesi* (Yayınlanmamış yüksek lisans tezi). Marmara Üniversitesi, İstanbul.

Kirçali, A. Ç., & Özdener, N. (2023). A comparison of plugged and unplugged tools in teaching algorithms at the K-12 level for computational thinking skills. *Technology, Knowledge and Learning, 28*(4), 1485-1513. https://doi.org/10.1007/s10758-021-09585-4

Korkmaz, Ö., Çakır, R., & Özden, M. Y. (2015). Bilgisayarca düşünme beceri düzeyleri ölçeğinin (BDBD) ortaokul düzeyine uyarlanması. *Gazi Eğitim Bilimleri Dergisi, 1*(2), 143-162.

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior, 72*, 558-569.

Oluk, A., Korkmaz, Ö., & Oluk, H. A. (2018). Scratch'ın 5. sınıf öğrencilerinin algoritma geliştirme ve bilgi-işlemsel düşünme becerilerine etkisi. *Turkish Journal of Computer and Mathematics Education (TURCOMAT), 9*(1), 54-71. https://doi.org/10.16949/turkbilmat.399588

Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (2020). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children?. *Computers in Human Behavior, 105*, 105849. https://doi.org/10.1016/j.chb.2018.12.027

Piedade, J., & Dorotea, N. (2023). Effects of Scratch-based activities on 4th-grade students' computational thinking skills. *Informatics in Education, 22*(3), 499-523. https://doi.org/10.15388/infedu.2023.19

Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education, 128* (2019), 365-376. https://doi.org/10.1016/j.compedu.2018.10.005

Ramazanoğlu, M. (2021). Robotik kodlama uygulamalarının ortaokul öğrencilerinin bilgisayara yönelik tutumlarına ve bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algılarına etkisi. *Türkiye Sosyal Araştırmalar Dergisi, 25*(1), 163-174. https://dergipark.org.tr/en/pub/tsadergisi/issue/61177/736602

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

Román-González, M. (2015, July). Computational thinking test: Design guidelines and content validation. Paper presented at EDULEARN15 conference, Barcelona, Spain.

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior, 72,* 678-691. https://doi.org/10.1016/j.chb.2016.08.047

Scratch (2023). About Scratch. Access: https://scratch.mit.edu/about

Sigayret, K., Tricot, A., & Blanc, N. (2022). Unplugged or plugged-in programming learning: A comparative experimental study. *Computers & Education, 184*, 104505. https://doi.org/10.1016/j.compedu.2022.104505

Sun, L., Hu, L., & Zhou, D. (2021). Improving 7th-graders' computational thinking skills through unplugged programming activities: A study on the influence of multiple factors. *Thinking Skills and Creativity, 42,* 100926. https://doi.org/10.1016/j.tsc.2021.100926

Sun, L., & Liu, J. (2023). Different programming approaches on primary students' computational thinking: a multifactorial chain mediation effect. *Education Technology Research and Development.* Advance online publication. https://doi.org/10.1007/s11423-023-10312-2

Threekunprapa, A., & Yasri, P. (2020). Unplugged coding using flowblocks for promoting computational thinking and programming among secondary school students. *International Journal of Instruction, 13*(3), 207-222. https://doi.org/10.29333/iji.2020.13314a

Tonbuloğlu, B., & Tonbuloğlu, İ. (2019). The effect of unplugged coding activities on computational thinking skills of middle school students. *Informatics in Education, 18*(2), 403-426. https://doi.org/10.15388/infedu.2019.19

Tospaa (2023). Kodlama eğitimine genel bakış. Access: https://tospaa.org/unplugged-unplugged-kodlama-nedir/

Totan, H. N. (2021). *Blok tabanlı kodlama eğitiminin ortaokul öğrencilerinin bilgi işlemsel düşünme becerileri ve kodlama öğrenimine yönelik tutumlarına etkisi: Blocky örneği* (Yayınlanmamış yüksek lisans tezi). Necmettin Erbakan Üniversitesi, Konya.

Uçak, S., & Erdem, H. H. (2020). Eğitimde yeni bir yön arayışı bağlamında "21. yüzyıl becerileri ve eğitim felsefesi". *Uşak Üniversitesi Eğitim Araştırmaları Dergisi, 6*(1), 76-93. https://doi.org/10.29065/usakead.690205

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

Wohl, B., Porter, B., & Clinch, S. (2015). Teaching Computer Science to 5-7 year-olds: An initial study with Scratch, Cubelets and unplugged computing. In Proceedings of the workshop in primary and secondary computing education (pp. 55-60).

Yükseltürk, E., & Altıok, S. (2015). Bilişim teknolojileri öğretmen adaylarının bilgisayar programlama öğretimine yönelik görüşleri. *Amasya Üniversitesi Eğitim Fakültesi Dergisi, 4*(1), 50-65. https://dergipark.org.tr/en/pub/amauefd/issue/1732/21264

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education, 141* (2019), 1-25. https://doi.org/10.1016/j.compedu.2019.103607

Zhao, W., & Shute, V. J. (2019). Can playing a video game foster computational thinking skills?. Computers & Education, 141, 103633. https://doi.org/10.1016/j.compedu.2019.103633