Education and Information Technologies (2024) 29:9541–9574
https://doi.org/10.1007/s10639-023-12181-x

Comparing the effects of plugged-in and unplugged activities on computational thinking development in young children

Yan Lin[1,2] · Hongjian Liao[1] · Suxian Weng[3] · Wanqi Dong[1]

Received: 20 June 2023 / Accepted: 24 August 2023 / Published online: 15 September 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

Children's preschool years are crucial for the development of computational thinking (CT) skills. However, debate continues regarding whether CT should be developed through plugged-in or unplugged activities. This study investigated the similarities and differences between plugged-in and unplugged activities with similar learning content and assessment methods for cultivating computational thinking (CT) in young children. Twenty-four young children (aged 5–6 years) from a kindergarten in Foshan, China, were randomly assigned to either the plugged-in or unplugged group to participate in a five-week study. In the plugged-in group, Dodobot was used in the classroom, while in the unplugged group, unplugged materials such as paper, pencil and tangram puzzles were used. Research results indicate that 1) both plugged-in and unplugged activities significantly improved the young children's CT skills after a short-term educational intervention, but there were no significant differences between the two groups; 2) the extent to which the plugged-in and unplugged activities promoted subdimensions of CT was different, with the plugged-in group demonstrating significant improvements in hardware, algorithm, and modularity and the unplugged group demonstrating significant improvements in terms of representation; 3) the children from both the plugged-in and unplugged groups showed high motivation; And 4) the children in both the plugged-in and unplugged groups showed cooperative behaviors. The frequency of cooperative behavior was more related to materials, and cooperation quality was more related to teacher guidance.

Keywords Computational thinking · Plugged-in activities · Unplugged activities · Young children · Comparative study

✉ Hongjian Liao
liaohongjian@gzhu.edu.cn

1 School of Education, Guangzhou University, Guangzhou 510006, China

2 The Second Kindergarten of Beijiao, Foshan 510006, China

3 School of Foreign Studies, Guangzhou University, Guangzhou 510006, China

1 Introduction

Computational thinking (CT) is the thought process involved in formulating a problem and expressing its solution(s) in such a way that a computer or machine can effectively carry out (Wing, 2017). The emergence and development of such thinking requires learners to have abstract and logical thinking abilities. Young children are transitioning from figurative thinking to abstract thinking. If specific reference systems are provided to guide their thinking at this stage (Angeli & Valanides, 2020; Gibson, 2012), they will be able to use abstract thinking. Therefore, the preschool years are supposed to be a critical period for young learners' CT education and coding and programming skills development (Flannery & Bers, 2013; Torres et al., 2018). Researchers have emphasized the significance of CT learning at the early childhood stage (Su & Yang, 2023). Early access to CT and coding interventions in childhood may result in long-term benefits, similar to improvements in academic achievement associated with early acquisition of literacy skills (Relkin et al., 2021). Children can develop a variety of CT skills, such as pattern recognition, sequencing, and algorithm design, through both activities that plugged and unplugged (Saxena et al., 2020).

The importance of developing CT in early childhood is generally acknowledged. However, due to young children's unique cognitive characteristics (figurative thinking predominates, and abstract thinking is developing), insufficient research has been conducted on which teaching practices in early childhood education are most effective in developing children's CT (Bati, 2021; Bers et al., 2014; Botički et al., 2018). Currently, there are two main types of activity for cultivating young children's CT: plugged-in and unplugged. The term "plugged-in activities" refers to learning activities that use computers, electronic devices, or other digital tools to enhance students' computational thinking abilities (Grover & Pea, 2013). By contrast, unplugged activities are those that do not involve computers, electronic devices, or other forms of technology. Instead, they rely on paper and pencil, physical objects, and games, among other non-digital methods, to cultivate students' computational thinking abilities (Looi et al., 2018). Plugged-in activities (e.g., visual programming and programming robots) are regarded as the mainstream approach due to the following advantages (Bers, 2010; Caballero-Gonzalez et al., 2019; Stoeckelmayr et al., 2011). (1) Figurative. Complex and abstract CT can be visualized and made more accessible to young children (Sullivan et al., 2015). (2) Observable. Teachers can observe children's programming behaviors to understand their level of CT and implement interventions (Bers, 2018a). (3) Effectiveness. Empirical studies have demonstrated that plugged-in activities are more effective in teaching CT (Elkin et al., 2016). However, there are certain dilemmas in using plugged-in activities to develop CT: (1) with relatively high cost, the use of programming robots may be restricted from widespread use in large classes (Lye & Koh, 2014; Xinogalos et al., 2017). (2) Few teachers have educational backgrounds in computer-related majors (Bell et al., 2009; Bers et al., 2002). (3) It is difficult to integrate such activity into traditional courses (Bell et al., 2009).

In recent years, unplugged activities have attracted increasing attention, as they have become a new way to cultivate CT in young children and have the following advantages: (1) Screen-free. It does not harm children's eyesight (Rodriguez et al., 2017; World Health, 2019). (2) Low threshold. Young children without programming skills may also benefit from foundational unplugged CT experiences with physical, hands-on play, etc. (Rodriguez et al., 2017). (3) Low cognitive load. Young children do not need to spend time or cognitive resources learning programming language syntax (Bell & Vahrenhold, 2018; Yadav et al., 2018). However, unplugged activities also have some shortcomings: (1) measuring learning outcomes is challenging, as there is no fixed, uniform learning content or assessment tools for computational thinking in unplugged activities (Brackmann et al., 2017; Jun, 2018; Tsarava et al., 2018). (2) course design and development is difficult because few learning resources are available. and (3) it may not be favorable for young children's future coding education. Although unplugged activities can involve children in computational thinking, they do not improve their ability to use programming languages, which is the basis of coding (Bers, 2018b).

In summary, although previous studies have proven that both plugged-in and unplugged activities can develop young children's CT, few studies have examined and compared their effectiveness. Therefore, the present study sets out to investigate the similarities and differences between plugged-in and unplugged activities in developing young children's CT, with the effect measured in terms of overall and subdimensional development. To ensure that the results of this comparative study are as objective and accurate as possible, similar teaching content and educational interventions were implemented for the two groups, and similar assessment tools were used to measure the learning outcomes of both types of activities. Moreover, a variety of test questions, observation scales and field diaries were also applied for cross-validation. We expect the findings of this study to provide empirical support for cultivating young children's computational thinking (e.g., blended plugged-in and unplugged models and how to blend them).

2 Literature review

2.1 Computational thinking framework and evaluation in early childhood

2.1.1 Computational thinking framework

The CT framework comprises the dimensions of computational thinking and its contents. To better develop and evaluate students' CT, a variety of CT frameworks have been proposed by various scholars and organizations. The Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) (2011) have developed a framework for K-12 that includes nine core ideas: data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and processes, automation, parallelization, and simulation. This framework provides an appropriate reference for addressing the question of what CT content to teach and how to teach it in K-12 education (Barr & Stephenson, 2011). Brennan and Resnick developed a new framework based on Scratch teaching practices that divided CT into three key dimensions: CT concepts, CT practices, and CT perspectives. In contrast to the K-12 Computational Thinking Framework, this framework focuses on technology-related changes in learners' thinking that are generated during the learning process (Brennan & Resnick, 2012). Based on literature analysis, Selby and Woollard (2013) proposed a five-element framework that includes abstraction, decomposition, generalization, evaluation, and algorithmic thinking. Due to the specific nature of their cognitive development and learning styles, young children in kindergarten require a more targeted CT evaluation framework. Bers (2018a) proposed seven dimensions of CT suitable for young children: algorithms, modularity, control structures, representation, hardware/software, design process, and debugging. She claims that these seven dimensions are closely related to the concepts and skills that young children need to develop and master in their early years. While the above frameworks are based on the plugged-in environment, CSunplugged.org proposes an unplugged-based framework that classifies CT into six dimensions: abstraction, decomposition, algorithmic thinking, generalization or patterns, evaluation, and logic.

Although CT has received considerable attention over the past few years, there is not yet a full consensus on the subdimensions included in the CT framework (Coulter et al., 2010; Barr & Stephenson, 2011). In terms of its specific content, there is a high degree of consistency. Among them, the seven powerful ideas proposed by Bers constitute the dominant framework for studying CT in young children and is adopted in our study.

2.1.2 Computational thinking evaluation

With the introduction of the CT framework, corresponding assessment instruments for evaluating CT skills have been developed (de Ruiter & Bers, 2022). There are four main types of tools for preschool children's CT evaluation: test questions (including questionnaires, paper/electronic test questions, or hands-on tasks), observation scales, programming tasks, and field diaries or interviews.

(1) Test questions. The paper-and pencil-based test (TechCheck-K) was developed by Relkin et al. (2021) and is currently a well-accepted tool for evaluating children's CT in the early years. It includes six dimensions of Bers' computational thinking framework: algorithms, modularity, control structures, representation, hardware/software, and debugging. Test question-based instruments have the advantage of being time-efficient and not dependent on a specific environment (Bers et al., 2022; Relkin et al., 2021). TechCheck has good psychometric properties (The observed $\alpha = 0.68$; criterion validity at $r = 0.53$) as shown in a validation study (Relkin et al., 2020). It is noteworthy that the percentage of correct responses for each item on TechCheck-K closely paralleled that observed with the original TechCheck in first and second grade students (Relkin & Bers, 2022).

(2) Observation scales. Observation scales are based on the way CT is evaluated through expert observation and rating children's manipulative behaviors. Relkin (2018) developed CT with the KIBO Interactive Play Rating Scales

(IPS-KIBO) based on Bers' seven dimensions. Each dimension is subdivided into five observable operational behaviors at increasing levels on a scale of NS to 4. In unplugged environments, few observation scales are available, possibly due to the lack of uniform learning materials and the difficulty of quantifying learning behavior.

(3) Programming task. Programming tasks are a way to evaluate CT in terms of the extent to which children complete the tasks. For example, the Coding Stages Assessment (CSA) tool developed by Ruiter and Bers is used to assess young children's CT levels. It is an open-ended question evaluation that includes two types of question assessments. The first type of question requires only a verbal response, in which young children need only to answer the researcher's questions in words. The second type of question is task-based, in which young children need to manipulate ScratchJr to complete programming tasks proposed by the researcher. The advantage of task-based evaluation tools is that they can measure CT and assess young children's programming proficiency. However, these evaluations can be time-consuming (de Ruiter & Bers, 2022).

(4) Field Diaries. Field diaries are a way to evaluate CT by observing and recording learners' behaviors or conversations in the field. Using this approach, Brackmann et al. (2017) observed that students displayed motivation while participating in unplugged activities. Caballero-Gonzalez et al. (2019) also used this approach to record students' behavioral performances in learning Bee-Bot, thus supporting the results of an evaluation based on a programming task. Due to its subjectivity, this approach is generally not used alone and is often used as a supplementary research method along with quantitative research tools.

A review of assessment instruments for young children's CT reveals the following: (1) assessment tools are contextual in nature and need to be used in specific scenarios in practice. (2) the TechCheck-K test is an effective assessment tool for evaluating the effects of both plugged-in and unplugged activities on preschoolers. (3) there are more well-developed observation scales for plugged-in activities, but few observation scales are designed specifically for unplugged activities.

2.2 Comparison of plugged and unplugged activities to cultivate computational thinking in kindergarten children

Most of the current literature focuses on the effects of a single plugged-in or unplugged activity on young children's CT development. A thorough review found that there are still very few studies comparing the effects of these two activities (Bati, 2021). Moreover, current comparative research focuses mainly on learning outcomes (i.e., level of CT skills) and learning processes (cooperative behavior, motivation, etc.)

2.2.1 Comparison of learning outcomes

Wohl et al. (2015) used test questions and interviews to measure and compare UK schoolchildren (5–7 years old) who participated in plugged-in (Scratch, Cubelets) and unplugged activities, showing that both activities enabled young children to grasp core concepts in computing subjects but that the unplugged activities appeared to promote a better understanding of 'algorithmic, logical prediction, debugging' concepts than the Scratch and Cubelets tools in the plug-in activities. Cubelets, which have a toy-like shape, were the most popular of the three teaching tools. These findings also indicate that there is not always a positive correlation between learners' preferences for learning tools and academic outcomes. According to the authors, special attention should be given to ensuring students' focus on learning concepts rather than tools. Messer et al. (2018) evaluated three groups of students who participated in iPad-based programming, paper and pen-based programming, and pencil and paper mathematical addition and subtraction tasks using mathematical competency as a criterion and found significant improvement in test scores for all groups but did not find either programming tool to be more effective than the addition and subtraction tasks, nor was the iPad more effective than paper and pencil. The authors suggested that the educational experience is more important than the medium through which it is delivered. Yang et al. (2022) used TechCheck to compare the effects of building block play and programming robots on the development of CT in young children and found that both contributed to significant improvements in CT levels over a short period, but children in the robot programming group showed greater improvements in sequencing skills. The block play in this study were self-directed, whereas programming learning took place in a group setting with teacher intervention, which indicates that the level of teacher intervention in the two activities was not the same. Hence, it may take further research to determine whether differences in teacher intervention lead to differences in CT.

Furthermore, Polat and Yilmaz (2022) compared the effects of unplugged and plugged-in activities on children's basic programming achievement and computational thinking skills. The results show that there is no difference between the two activities in terms of CT skill development, but children in the unplugged group had significantly better basic programming scores than those in the plugged group. The unplugged activities helped students better focus on the subject matter by eliminating the destructive effects of computers or other digital tools. The findings of the study support the idea that unplugged activities have a positive impact on teaching CT and basic programming.

Generally, 1) most comparative studies focus on assessing and comparing the overall levels of CT rather than its subdimensions. 2) In terms of assessment tools, test questions are commonly used to compare the effectiveness of these two activities because they save time and effort while not being restricted by the environment. There are, however, some disadvantages associated with paper-based tests, such as the possibility of children guessing multiple-choice questions correctly, which in some respects do not reflect their true abilities. 3) It is obvious that using only one assessment tool in a study may not comprehensively cover all dimensions of CT, which may result in distorted results. Therefore, there have been attempts to combine a process evaluation tool with a summative evaluation tool in plugged-in activities (Relkin, 2018). However, few have tried to combine multiple assessment tools in comparative studies on plugged-in vs. unplugged activities. There are indeed numerous challenges associated with doing so. First, maintaining consistent subdimensions of CT between these two activities is a problem; second, the development of a process observation scale for unplugged activities is still lacking.

2.2.2 Comparison of learning processes

In addition to learning outcomes, the comparison also involved cooperative behaviors and motivation shown by the children in the learning process (Heljakka et al., 2019; Zhan et al., 2022). In terms of collaborative behaviors, Polat and Yilmaz (2022) argued that the choice of learning tool (e.g., screen-based or paper-and-pencil) influences the frequency of collaborative behaviors, while Sun et al., (2021a) argued that the difficulty of the designed activity is one of the most important influencing factors. Critten et al. (2022) used both plugged-in (robotics) and unplugged (traditional paper and pencil) activities to develop CT in preschool children aged 2–4 years and found that the children's development of CT was accompanied by an increase in their communication and cooperation skills. He attributed this to factors such as materials, choice of activities, and teacher guidance. However, the study did not further discuss whether there were differences between plugged-in and unplugged activities in terms of promoting cooperation skills. Sun et al., (2021a) observed that children in the plugged-in group were more motivated, with teachers indicating in their interviews that the paper and pencil manipulation involved in the unplugged activities made students feel like they were doing exercises and thus were less motivated to learn. Sun also identified the way teachers presented programming concepts in unplugged activities as a significant factor in students' motivation. On the other hand, Zhan et al. (2022) found that unplugged activities contribute positively to young children's engagement and motivation levels due to the interactions they have with each other and with their teachers. For young children, the specific factors in CT education that influence their motivation and cooperative behavior need to be further explored.

Because previous comparative studies have not reached a definite conclusion regarding the respective benefits and drawbacks of plugged-in and unplugged activities for fostering CT, some researchers have started to investigate using a blended model of the two activities (Bers, 2017; Huang & Looi, 2021; Metin, 2022; Thies & Vahrenhold, 2013). For example, Saxena et al. (2020) employed an unplugged method to help young children learn pattern recognition and sequencing, using the Bee-Bot robot to help them learn algorithm design. Their study shows that plugged-in and unplugged activities might reinforce one another rather than being mutually exclusive. However, there is still a lack of sufficient evidence to explain why certain subdimensions are developed through plugged-in activities and others through unplugged activities. Thus, to further explore the respective advantages of the two activities and to serve as a guide for the creation of a blended model incorporating both plugged-in and unplugged activities, it is necessary to compare the effects of plugged-in and unplugged activities on the development of different subdimensions under the same CT development and assessment framework.

2.3 Research gap

A review of the literature reveals that there are still several issues worth further exploration in the comparative study of plugged-in and unplugged activities. (1) Studies have compared the effects of plugged-in and unplugged activities on the overall development of young children's CT, but in-depth comparisons of the two activities at the subdimensional developmental level are still relatively rare. Furthermore, they have focused mainly on primary and secondary school students, but preschool children's cognitive characteristics greatly differ from those of primary and secondary school students. (2) Currently, empirical research on the effectiveness of unplugged activities is insufficient, primarily due to the lack of evaluation tools. In particular, no observational scales for assessing the learning process of young children participating in unplugged activities have been developed, which has also led to a lack of cross-validation studies using multiple assessment tools in combination. (3) In both activities, factors influencing young children's motivation and cooperative behaviors, and their explanations are still very controversial and require further study.

2.4 Research questions

To fill this knowledge gap, this study focuses on the following three research questions.

RQ1: Is there a significant difference in overall CT between the young children in the plugged-in group and the unplugged group under short-term educational intervention?
RQ2: Is there a significant difference in the subdimension of CT between the young children in the plugged-in group and the unplugged group under the short-term education intervention? If so, what are the differences?
RQ3: What are the similarities and differences in learning motivation and cooperative behavior of the young children in the plugged-in group and the unplugged group beyond CT skills?

3 Research design

Based on the above analysis, 1) two types of learning activities (plugged-in and unplugged) were designed for children 5 to 6 years old based on the CT framework proposed by Bers. Instructional design, teacher instruction, and free play time are strictly controlled to minimize the interference of irrelevant variables. 2) Observation scales that are suitable for unplugged activities are developed by referring to those of plugged-in activities. 3) Similar evaluation tools, such as TechCheck-K test questions, were used to assess the overall development and subdimension development of CT learning of children in the plugged-in group and the unplugged group and to compare the similarities and differences in the effectiveness of these two activities.

3.1 Participants

The participants were 24 young children (M = 66.08 months, SD = 3.798) from a public kindergarten in Foshan, Guangdong Province, China. These children were randomly assigned to either the plugged group or the unplugged group, with their parents providing informed consent in terms of enrolling their child in the study. The participants included 10 girls (41.7%) and 14 boys (58.3%), with 12 children in the plugged-in group (50%) and 12 children in the unplugged group (50%). Based on the measurement results from TechCheck-K, there is no significant difference ($p = 0.820 > 0.05$) in the level of CT between the plugged-in group (M = 8.421) and the unplugged group (M = 8.501). These young children have the following characteristics. (1) There was an uneven level of CT at the beginning. The highest score was 14 points, and the lowest score was 4 points. A total of 14 students (58.33%) scored less than 8 points (including 8 points), and the overall level was below the medium level (M = 8.46, SD = 2.206). (2) None of the children had previous experience in systematic thinking training or coding. (3) Children's logical thinking begins to germinate at 5–6 years old, and they can carry out preliminary induction and reasoning. Throughout the study, all children participated in sessions and no data was missing.

3.2 Materials

(1) Dodobot R1 was used as the material for the plugged-in group. Dodobot R1 is a drawing robot designed by iFLYTEK's Alpha Egg to educate children over 3 years old in programming. The main body of Dodobot R1 is a car driven by two main wheels. With color and trajectory recognition modules at the bottom, it acts according to fixed instructions through coding. In addition, Dodobot R1 has DIY expansion holes on both the left and right sides that can be combined with small LEGO bricks to equip R1 with a variety of equipment and shapes. Thus, the fun can be increased, and children's imagination and hands-on ability can be improved. Figure 1 shows Dodobot1 and the marker used to draw the route.
(2) The material for the unplugged group was paper and other ancillary materials such as tangram puzzles and combination locks. The instructional design includes path planning, pattern recognition, password decryption, and treasure hunting. Figure 2 shows some of the material.

3.3 Measurement

3.3.1 Evaluation tools

Three tools were employed in this study to assess young children's CT: the TechCheck-K test questions, classroom observation scale, and field diary. Table 1 shows the application scenarios for each tool.

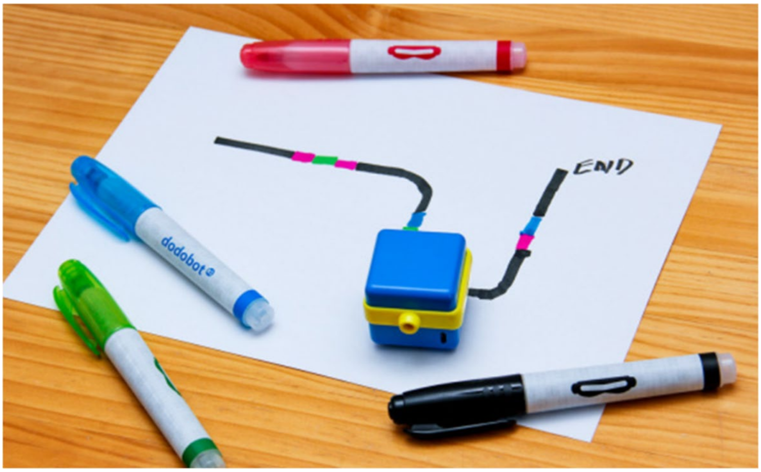


Fig. 1 Dodobot R1 for plugged-in activities

(1) TechCheck-K. TechCheck-K is a CT test question developed by Relkin designed for kindergartners. It consists of 15 questions presented in a forced-selection multiple-choice format with three options. Observation scales were employed as an auxiliary evaluation tool since TechCheck-K questions cover only six dimensions of Bers' seven powerful ideas (apart from the design process). A sample TechCheck-K item is shown in APPENDIX A.
(2) The Observation Scale for CT-Dodobot (OSCT-Dodobot) was adapted from the "Computational Thinking with KIBO Interactive Play Rating Scale". The evaluation consists of six powerful concepts (algorithm, hardware/software, con-

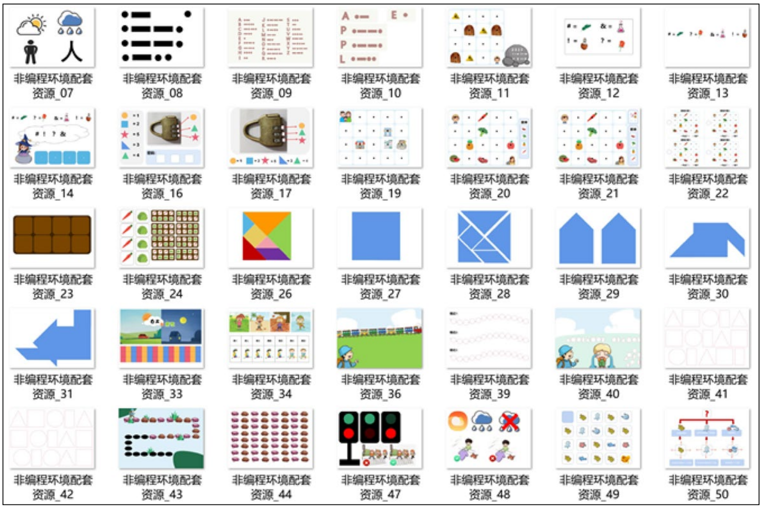


Fig. 2 Materials used in unplugged activities

Table 1 Evaluation tools in this study

Instrument	Source	Tool type (items)	Usage times	Usage environment
TechCheck-K	Relkin et al., (2021)	The question base (15)	Pretest/Posttest	Plugged-in and Unplugged
Observation Scale for CT-Dodobot (OSCT-Dodobot)	Derived form Relkin and Bers (2022)	The scale base (7)	Each Classroom activities During experiment	Plugged in
Observation Scale for CT in Unplugged (OSCTU)	Derived form Relkin and Bers (2022)	The scale base (15)	Each Classroom activities	Unplugged
Field diary	/	/	During experiment	Plugged in and Unplugged

trol structure, debugging, modularity, and design process). Each dimension is subdivided into five levels (NS, 1, 2, 3, 4), which represent the lowest to highest performance levels. It is a teacher-friendly grading method since each dimension is described with a typical behavioral description. A sample OSCT-Dodobot item is shown in APPENDIX B.

(3) The Observation Scale for CT in Unplugged (OSCTU)was developed using the Delphi method based on the "Computational Thinking with KIBO Interactive Play Rating Scale" and the CT framework proposed by CS unplugged.org. A questionnaire was sent to five computational thinking experts by e-mail, along with a description of the background and content of the OSCTU. Experts evaluated each indicator's importance and modified it when necessary. Based on feedback collected in the first round of expert appraisal, the questionnaire was edited and sent to the experts again for evaluation. This process was repeated several times until all five experts reached a consensus on the content and wording of the OSCTU. The coordination coefficient (Kendall's W) and variation coefficient (Vi) were collected from each round of expert appraisal and were calculated. The overall coordination coefficient obtained was 0.51 (<1); in addition, the coefficient of variation in the five dimensions was between 0 and 0.28 (<0.3). Such results showed that the coordination of expert opinions on all of the indicators was good and credible. Based on such findings, the present study put forward a final version of the OSCTU, featuring 5 dimensions and 15 indicators. Kendall's W, Vi and A sample OSCTU items are presented in APPENDIX C, D and E.

(4) Researchers used field diaries to record and analyze teachers' and children's main behavior and language. Diaries are used to analyze and reflect upon the reasons for children's motivational and cooperative behavior in the classroom, as well as how teachers guide and support these behaviors.

3.3.2 Evaluation process

Similar instructional interventions were applied for both the plugged-in and unplugged groups. Before the treatment, both groups of children received a test of CT ability, which showed that both groups had no significant differences. Then, both groups took part in a 5-week experiment that included a learning activity of 45 to 60 min per week. During the experiment, observation scales (OSCT-Dodobot, and OSCTU) and field diaries were used to observe and record the children's performance in both groups. After five weeks of learning activities, the TechCheck-K was used to assess the children in both groups.

4 Procedure

The ADDIE model is used to develop plugged-in and unplugged activities (Branch, 2009). Instructional design, teacher instruction, and free play time are tightly controlled in both activities, ensuring that differences in computational thinking development are caused primarily by differences in learning resources. There is a fixed protagonist in both learning activities, and each activity focuses on the problems and challenges encountered by this protagonist. Plugged-in activities are presented in Table 2, and unplugged activities are presented in Table 3.

Figure 3 shows the learning process for plugged-in activities versus unplugged activities.

5 Results

5.1 Quantitative results

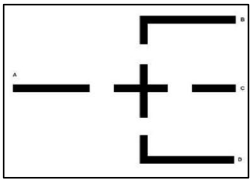
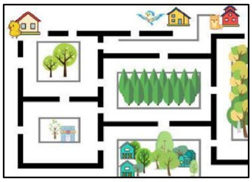
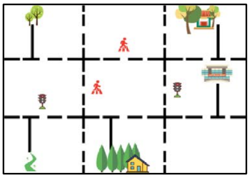
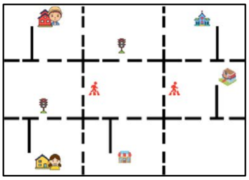
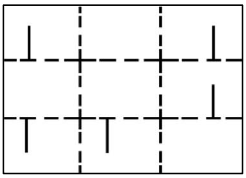
5.1.1 Comparison of overall development of computational thinking

(1) Comparison of overall level based on the TechCheck-K test questions.

To investigate the effect of plugged-in and unplugged activities on children's CT improvement and the differences between the two activities in promoting children's CT development, we conducted independent t tests and univariate analyses. When comparing within-group differences, we assessed the effects of both plugged and unplugged activities on children's CT development using an independent t test. When comparing between-group differences, we used a one-way analysis of covariance test, with baseline scores entered as covariates.

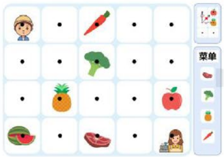
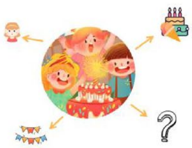
Table 4 displays the within-group and between-group differences between the plugged-in and unplugged groups. From the within-group comparison, there was a significant difference between the pretest and posttest in the plugged group ($p < 0.05$), which indicated that the activity in the plugged group could effectively improve the CT level of the toddlers. There was a significant difference between the pretest and posttest in the unplugged group ($p < 0.05$), which indicated that the unplugged activity could also effectively improve the CT level of the toddlers. In terms of group comparison, there was no statistically significant difference between the plugged-in and unplugged groups.

Table 2 Schedule of plugged-in activities

Activity	Explanation	Main pillars
	**Hello, Dodobot:** Using hardware such as switches and charging locations, kindergartners learn how to code Dodobot by using the colors red, green, and blue, and then move onto rabbits, elephants, and bears.	Hardware Software Representation Algorithms Debugging Design process
	**Cops and Robbers:** A map with many animals and houses is shown to kindergartner, and they need to locate the thief who stole the kitten's necklace and devise a route to catch him.	Hardware Software Representation Algorithms Debugging Design process Modularity
	**Go for a Drive:** Kindergartners need to plan a route for a joyride. However, Dodobot has to slow when it encounters a sidewalk and wait three seconds when it encounters a red light.	Hardware Software Representation Algorithms Debugging Design process Modularity Control Structures
	**Go to school:** The task cards allow kindergartners to check the reasons Dodobot is not able to walk to school successfully individually and then find the right route to help the Dodobot return to school successfully.	Hardware Software Representation Algorithms Debugging Design process Modularity Control Structures
	**Dodobot's weekend:** In cooperation with their peers, kindergartners can set tasks and challenges using various buildings and character cards.	Hardware Software Representation Algorithms Debugging Design process Modularity Control Structures

(2) Cross-validation based on observation scales.

Table 3 Schedule of the unplugged activities

Activity	Explanation	Main pillars
	**Great Adventure:** A variety of tasks related to decryption are shown to kindergartners. They have to discover and use the transformation relationship between symbols to solve problems and complete the treasure hunt (e.g., unlock a combination lock). Other examples: potion ratio, secrets on the stone, etc.	Representation Algorithms Debugging Design process
	**Asen's Weekend:** A map with many food options is shown to kindergartners. They have to find the shortest route to buy the food on the menu and get to the cashier (e.g., shopping). Other examples: go to school, planting, etc.	Representation Algorithms Modularity Debugging Design process
	**Go Hiking:** Identify patterns, copy, populate, expand, and more to complete challenges.	Representation Algorithms Modularity Debugging Design process
	**Treasure Diving:** Kindergartners play the game "Treasure Hunt Under the Sea", moving in accordance with the specific instructions represented by different animals. For example, the turtle indicates that if you encounter a turtle, you can only go forward.	Representation Algorithms Modularity Debugging Design process Control Structures
	**Asen's Birthday Party:** Kindergartners have to work together to breakdown tasks, identify all the steps needed to solve them, and plan a party for Asen.	Representation Algorithms Modularity Debugging Design process Control Structures

Since TechCheck-K did not fully cover all seven dimensions (the design process dimension was not included because it is not easily measured by test questions), we then compared the overall level of CT using the observation scale OSCT-Dodobot with the OSCTU outcome data to validate the results of the TechCheck-K test questions. The results are shown in Table 5.

We conducted a t test analysis on the average scores of five classroom observations for the plugged-in and unplugged children, and the results showed that there was no statistically significant difference in the development of CT between the

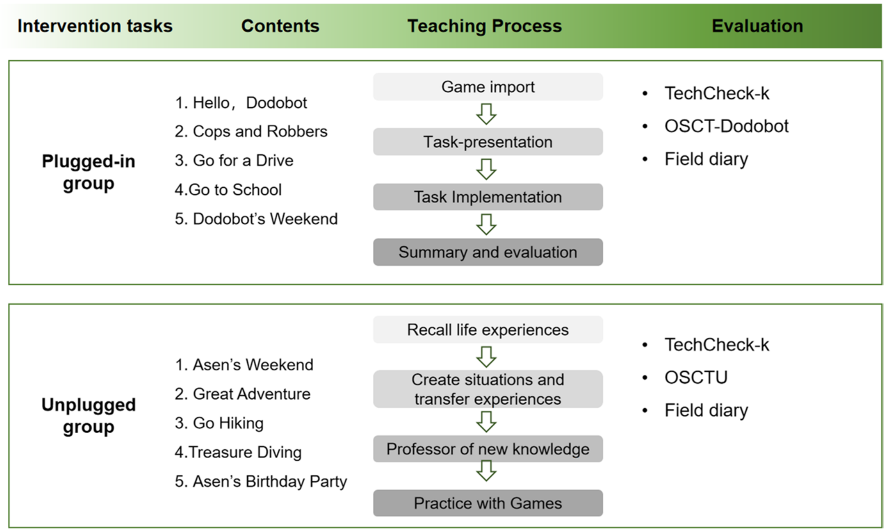


Fig. 3 Teaching process for plugged-in and unplugged activities

plugged-in group and the unplugged group of children ($p > 0.05$), which is consistent with the results of the TechCheck-K test.

5.1.2 Comparison of subdimension development level of computational thinking

(1) Comparison of subdimensions based on the TechCheck-K test.

To further compare and evaluate the effects of plugged-in and unplugged activities on the CT subdimensions, an independent t test was conducted to compare the differences in the CT subdimensions before and after within each group, and one-way analysis of covariance was used to compare the differences in the CT subdimensions between the two groups. Cohen's test was used to measure the magnitude of

Table 4 The overall results of performance in the TechCheck-K

		Mean	Within-group			Between-group		
			t	p	Cohen's d	F	p	Cohen's d
Plugged-in	Pretest	8.421	-3.370	0.003*	1.372	0.327	0.573	0.015
	Post-test	11.752						
Unplugged	Pretest	8.501	-3.272	0.004*	1.334			
	Post-test	11.253						

* $p < 0.05$

Table 5 The overall results of performance in the OSCT-Dodobot and OSCTU

	N	Mean	SD	t	p	Cohen's d
OSCT-Dodobot	12	10.883	1.949	-2.202	0.055	1.258
OSCTU	12	12.946	2.944			

* $p < 0.05$

As the observation scale for unplugged activities does not include the software and hardware dimension; this dimension is not included in the comparison

the effect of improvement in CT for each dimension within the group and between groups for each dimension.

As shown in Table 6, in terms of within-group differences, the plugged-in group had significant differences in hardware, algorithm, and modularization based on TechCheck-K scores ($p < 0.05$), indicating that the plugged-in activity had the best enhancement effect on these subdimensions. The unplugged activity had a significant difference in the representation dimension ($p < 0.05$), indicating that it had the most effective improvement in this dimension. In terms of between-group differences, the difference in the hardware dimension was the most significant between the two groups ($p < 0.05$), while there were no significant differences in other dimensions.

(2) Evaluation and cross-validation of subdimension development based on the observation scale.

To cross-validate the reliability of TechCheck-K in the development of subdimensions, this study combined OSCT-Dodobot and OSCTU to cross-validate the development level of subdimensions under the two types of activities. The observation scale was used to evaluate children's CT levels through observation and rating by the classroom teacher. To visually compare the results of the evaluation of each dimension, a radar chart was created, as shown in Fig. 4. Notably, in the unplugged activities, as no computer devices or robots were used, the children were unable to acquire knowledge of software/hardware through operation. This dimension was not suitable as an evaluation item on the observation scale of unplugged activities. Figure 4 shows the intergroup difference in the five dimensions of "algorithm/modularization, debugging, representation, design process, and control process" between the two activities.

Figure 4 shows that the scores for the four subdimensions of "algorithm/modularization, debugging, representation, and design process" were similar between the plugged-in and unplugged groups, which is generally consistent with the results of the intergroup comparison of the TechCheck-K test. The scores for the "design process" dimension, which is not included in the TechCheck-K test, were also similar for both groups. However, there was a slightly larger difference between the two groups in the "control structure" dimension, which was not consistent with the results of the TechCheck-K test. In the TechCheck-K test, the children in both unplugged and plugged groups scored full marks in the "control structure" dimension (it cannot be

Table 6 Pretest and posttest scores by group and assessment

	Plugged-in					Unplugged							
	Pretest M (SD)	Post-test M (SD)	Within-group Cohen's d	Within-group F	Within-group p	Pretest M (SD)	Post-test M (SD)	Within-group Cohen's d	Within-group F	Within-groupp	Between-group Cohen's d	F	p
Hardware	0.17 (0.389)	1.00 (0)	0.83	13.750	0.000*	0.33 (0.492)	0.67 (0.492)	0.69	88.000	0.111	0.95	5.500	0.039*
Software	0.50 (0.522)	0.67 (0.492)	0.33	1.375	0.430	0.75 (0.452)	0.992 (0.289)	0.64	5.436	0.294	0.80	2.302	0.147
Algorithms	2.75 (1.357)	4.17 (1.193)	1.11	0.155	0.013*	2.58 (1.240)	3.170 (1.267)	0.47	0.600	0.267	0.95	0.571	0.059
Debugging	1.58 (0.793)	1.83 (0.389)	0.40	0.152	0.191	1,67 (0.651)	1.830 (0.389)	0.30	2.708	0.455	0.28	0.478	0.496
modulari-zation	0.78 (0.452)	1.17 (0.718)	0.65	0.957	0.000*	1.33 (0.492)	1.500 (0.522)	0.34	1.375	0.430	0.53	1.692	0.207
Represen-tation	0.92 (0.669)	1.08 (0.900)	0.20	2.859	0.612	0.42 (0.515)	1.170 (0.718)	1.20	0.311	0.008*	0.11	0.062	0.804
Control Struc-tures	1.83 (0.389)	2.00 (0.000)	0.62	13.750	0.166	1.50 (0.798)	2.00 (0.000)	0.87	35.200	0.053	-	-	-

* $p < 0.05$

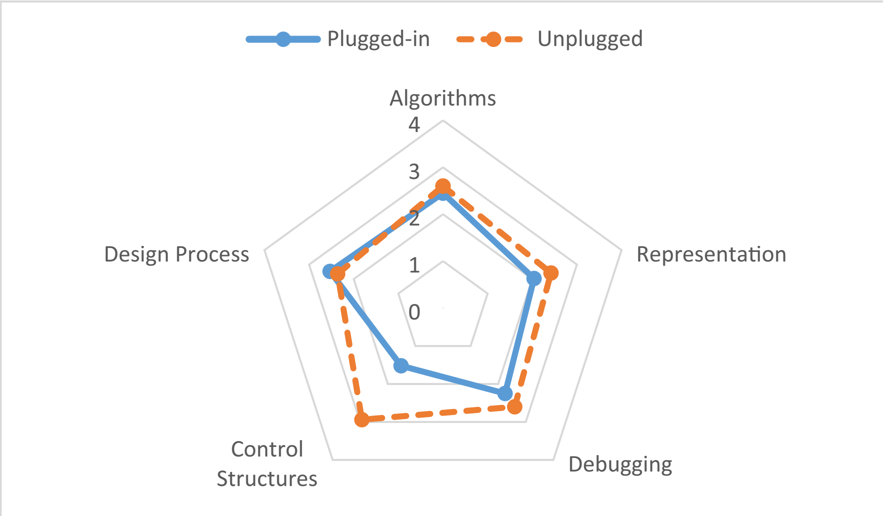


Fig. 4 The average scores of the plugged-in and unplugged groups in each dimension

excluded that the children might have guessed the multiple-choice answers), which may indicate that the test did not fully assess the children's true levels in this dimension. However, through observation, we found that the children in the unplugged group performed better in the control structure dimension.

In conclusion, the evaluation results of the Observation Scale and TechCheck-K test on the subdimensions of CT of the two groups were consistent, indicating that the level of CT demonstrated by the children in practical exercises was consistent with the level of theoretical tests. Alternatively, it can be said that the CT level reflected by the children in theoretical testing was verified in practical activities.

5.2 Qualitative results

5.2.1 Learning motivation

To answer RQ3, the field diaries were coded based on Grounded Theory. Among the coded concepts were learning motivation, cooperative behavior, and so on. The plugged-in and unplugged groups were analyzed and compared according to the coded entries and content, and qualitative results were drawn regarding RQ3.

As expected, the children in the plugged-in group showed high motivation in the learning activities. None of them had previous experience with robots, and programming the robots to move or make sounds greatly sparked their interest. The children in the unplugged group also showed a high level of motivation, which was unexpected, as we assumed that the materials were common to them. However, they provided feedback that the activities were different from their usual activities, and they found them interesting. The following are observations taken verbatim from the researcher's field diary.

- In the activity “Hello, Dodobot”, children O and L from the plugged-in group tried multiple times to make the robot move according to their programmed route, yet they failed. They actively sought help from the teacher, who discovered with them that the robot’s mode had accidentally been changed to “free mode”. The teacher guided O in adjusting the mode, and after successfully doing so, O and L continued their exploration.
- In the activity “Great Adventure”, children L and H from the unplugged group worked together to unlock a code lock. After L entered the wrong password, H and L checked the answer together again and found that A had written a number incorrectly. Because the first group completed the task, the two of them cheered and received the “treasure” inside the password box.
- In the activity “Go Hiking”, after completing the first task, all the children were filled with confidence and urged the teacher to quickly tell them what would happen next and what kind of challenging tasks the second main character Asen would encounter.
- In the Cops and Robbers activity, children were able to observe the thief hidden in the story map. Child Z first tried to trace the route on the map with his finger, and then together with his peers, they drew a color code to make Dodobot move to the thief’s location.
- In the activity “Go for a Drive”, children Z and C from the plugged-in group encountered some difficulties. They disagreed about whether they should turn left or right to reach the designated location. At the same time, C confused the color codes for left and right turns. Although Z was a little angry, he still allowed C to have a try. In the end, C realized that he was wrong, and the two successfully designed the route to help Dodobot reach the correct destination.

5.2.2 Frequency and quality of cooperative behaviors among children

Positive cooperative behaviors were exhibited by children in both plugged-in and unplugged activities, and the teacher’s observational guidance had some impact on the quality of the children’s cooperation. The following are observations taken verbatim from the researcher’s field diary.

- In the activity “Do go school”, P and F shared their joy of success with the teacher when they successfully completed the challenge, proudly stating that they had worked together to achieve it. P shared that he and H had encountered difficulties during the operation when the robot could not recognize the color they used. As a result, the robot did not walk along the predetermined path. However, they eventually completed the task together.
- In the activity “Go for a Drive”, H used the wrong color code, so Dododot did not follow the established route. By checking the color code drawn together, H and L found that the code was wrong, and L told H to wait and suggested that they could walk on one path first and then another to reach the destination. L then used a new color code according to the idea just proposed. However, a new problem arose, as the robot could not recognize the code. H discovered that the order of the color composition was wrong and should be green−blue instead of

blue−green. With both H and L's cooperation, they successfully adjusted the route.

- In the activity "Asen's Weekend", Z kept failing in the process of completing the second jigsaw puzzle challenge. The teacher guided L to determine the difficulties Z encountered and asked L whether he could help Z. After being reminded by the teacher of the difficulties Z encountered, L decided to take the initiative to help him, but instead of helping him directly, he showed him how to do it.
- During the group discussion process in the activity "Asen's Birthday Party," H was slightly idle, and a few children were chatting. Through observation, the teacher found that a few stronger children were always the ones talking and doing things. Thus, the teacher reminded them that they could divide the work, with someone organizing the discussion, and someone else responsible for taking notes, and so on. After the teacher's intervention, the children had clearer roles and responsibilities.

6 Discussion

6.1 RQ1: Is there a significant difference in terms of the overall CT competency between the children in the plugged-in group and the unplugged group under short-term educational intervention?

The present study found that young children's CT can be significantly improved by both plugged-in and unplugged activities within a short period, without significant differences between the two groups. As shown in Table 4, the plugged-in group had a statistically significant improvement in the TechCheck-K score between the pretest and the posttest ($p < 0.01$); meanwhile, the effect size of the improvement in the plugged-in group was "large" ($d = 1.37$). Similarly, a statistically significant prepost improvement in the TechCheck-K score is found in the unplugged group ($p < 0.01$), which also indicates a 'large' effect size ($d = 1.33$). All these findings are aligned with the conclusions of Messer et al. (2018) and Yang et al. (2022). This study used the same evaluation tool to compare the learning effects of the two activities, which is similar to other studies that analyzed plugged-in versus unplugged activities. Furthermore, the present study utilized the OSCT-Dodobot and OSCTU as process observation measures to cross-verify the TechCheck-K results, generating similar findings. Such cross-validation of the multiple evaluation tools increases the validity of the assessment results.

It interests us that unplugged activities can be as effective as plugged-in activities in developing children's CT through short-term interventions. This finding is consistent with Polat and Yilmaz (2022) findings that unplugged activity has a positive impact on the development of young children's CT. Consequently, parents who rush to enroll their children in programming classes may obtain inspiration from this finding and hence reduce their parental anxiety. Based on such preliminary findings, we further analyzed the reasons unplugged activities can be effective and found the following explanations: young children are transitioning from concrete to abstract thinking. During this time, well-designed unplugged activities can provide young children with positive experiences and hence activate their thinking processes. These activities support young children in abstract thinking and facilitate the transition to the next stage of their development.

Three reasons could account for this phenomenon. First, the thinking development of young children in the early stage needs to use concrete things that exist objectively to form an effective understanding (Sun, et al., 2021b). Unplugged activities use figurative materials such as cards, tangrams, and combination locks that are linked to concrete life experiences and provide scaffolding for children's thinking. Second, since children's preexisting experiences can have an important impact on their ability to learn new information (Heikkilä & Mannila, 2018), unplugged activities direct children to make connections between activities and their daily lives (del Olmo-Muñoz et al., 2020). For example, CT could be integrated into the familiar life experience of purchasing vegetables to facilitate children's learning and understanding of "design process" and "algorithms". Finally, it is possible that unplugged activities are more effective, but the benefits may diminish over time. Children benefit quickly from play in unplugged activities without having to learn complex grammar and instructions, whereas children may lag behind in plugged-in activities, as they have to learn the robot's operations and programming instructions in advance. For example, using the CT test, Sun and et al., (2021a) found that students who participated in unplugged activities improved their CT skills more rapidly, but the degree of improvement faded over time, with the benefit of unplugged activities gradually weakening. Therefore, it can be tentatively concluded that the development of computational thinking is more effectively promoted by participating in unplugged activities before plugged-in activities. This result is in accordance with previous findings (Saxena et al., 2020; Sun, et al., 2021a).

6.2 RQ2: Is there a significant difference in the subdimension of CT between the children in the plugged-in group and the unplugged group under the short-term education intervention? If so, what are the differences?

The results show that plugged-in and unplugged activities have different improvement effects on the subdimensions of CT. Table 6 presents the results of the within-group and between-group comparisons for the plugged-in and unplugged activities. The between-group comparison result indicates that the children in the plugged-in group significantly improved in the dimensions of hardware and algorithm (large effect size, $d = 0.83$, $1.11 > 0.8$) and moderate improvements in modularity (medium effect size, $d = 0.65 > 0.5$). In addition, the unplugged activity group showed significant improvement in the dimension of representation (large effect size, $d = 1.2 > 0.8$).

Plugged-in activities have significant advantages in fostering algorithms and modularization (Relkin et al., 2021; Saxena et al., 2020; Yang et al., 2022). This advantage is revealed by the "timely feedback" advantage in the plugged-in activities. Young children can benefit from more trials and errors in plugged-in activities since the robots can offer more immediate and specific feedback. In contrast, unplugged materials cannot provide such “timely feedback”, which makes it difficult for children to determine whether their answers are accurate. There is little chance that they will find problems, make mistakes, and resolve them on their own since young children tend to rely more on teachers for verification.

The present study has demonstrated the advantages of unplugged activities in promoting the development of young children’s “representation” skills. This outcome is contrary to that of Relkin (2018), who found that children in a Coding as Another Language (CAL) group performed significantly better in “representation” than those in a non-CAL group. In fact, “representation” is a measure of a child’s ability to switch between representation systems (Bers et al., 2019). In fact, programming instructions are usually represented differently by different programming robots. For example, BeeBot uses a left-facing arrow (graphic) to indicate a left turn, while Dodobot uses red (color) to indicate a left turn. In this regard, we believe that the way programming instructions are rendered may influence the development of “representation”. It is more challenging for children to comprehend programming directions composed of color permutations rather than those composed of graphics, which may increase the cognitive burden on young children and adversely affect their ability to master the concept. In Bell’s opinion, computational thinking based on programming may increase the cognitive workload of young children (Bell & Vahrenhold, 2018). Thus, unplugged activities that are close to children’s lives may better promote “representational” development than programming learning.

The above findings could have some implications for the practice of computational thinking with a combination of plugged-in activities and unplugged activities. Previous studies have advocated combining plugged-in and unplugged activities in chronological order, and the conclusions of this study can be used to provide more specific guidelines. What dimension of CT is most effectively developed using plugged-in activities, and what dimension is most effectively developed using unplugged activities? To complement each other’s advantages, upon combining them in sequential order, instructors are also suggested to explore the possibility of synchronously combining plugged-in and unplugged activities within the same activity, i.e., shifting from “asynchronously combining” to include “synchronously combining”.

6.3 Apart from CT, are there any other similarities and differences in learning motivation and cooperative behavior between the children in the plugged-in group and those in the unplugged group?

6.3.1 Both the plugged-in and unplugged groups showed a high level of motivation for learning.

In the present studies both the plugged-in and unplugged groups showed a high level of motivation for learning. The following factors could be causes of this result:

(1) Attractive learning materials. The enthusiasm for learning is primarily attributed to the curiosity children in the plugged-in group had for robots and the robot’s “timely feedback” on learning behavior, providing the children with an opportunity to adjust their answers until they arrive at the correct answer, which greatly

motivated them to keep learning. In addition, the motivation for the children to learn was derived from the unplugged materials that were fun, operative, and directly linked to daily lives. In short, physical objects provided the young children with more scaffolds and hence engaged and motivated them to think actively.

(2) Vivid instructional scenarios. Both the plugged-in and unplugged activities used the situational teaching method. Preschoolers are full of imagination, and the vivid and ups and downs of a fairy tale-like activity context were another important reason for attracting the children to participate in the activities.

(3) Sufficient operational time. In the process of learning, the children have adequate time to operate independently, trying to discover and adjust problems dynamically. Such affirmative freedom fully mobilized learning enthusiasm and empowered the children to be in charge of their own learning (Bers, 2008; Curzon et al., 2009).

(4) Hierarchical tasks. Based on Vygotsky's zone of proximal development, appropriate challenging tasks not only meet young children's desire for challenge but also provide them with the opportunity to develop self-confidence, which in turn motivates them to engage in new challenges in the future. The present study set challenging tasks with different levels of difficulty for each plugged-in and unplugged activity, stimulating the children's enthusiastic participation. Consequently, they felt a sense of achievement and actively shared their joy with their teachers.

In summary, children's enthusiasm for learning is influenced by the selection of activity materials, the design, and the organization of activities. Learning motivation in young children cannot be attributed solely to programming tools. Content that is relevant to the lives of young children is very important for their learning. This finding confirms (Heikkilä & Mannila, 2018) the claim that cultivating computational thinking in unplugged environments requires challenging scenarios that cater to children's interests, are appropriate for their ages, and involve them in self-operative activities.

6.3.2 Cooperative behavior was observed in both the plugged-in and unplugged groups, with frequency more closely related to the material and quality more closely related to teachers' guidance.

Previous CT courses have been shown to promote cooperative behaviors (Critten et al., 2022; Monteiro et al., 2021). In this study, the cooperative learning method was applied to both the plugged-in and unplugged activities, ensuring that the two groups had equal cooperation opportunities and frequency (Zhan et al., 2022). The study found that when the children were participating in unplugged activities that involved paper and pencil materials, their cooperation decreased. Some groups had one person manipulating while others watched. In contrast, when the children were exposed to manipulable materials such as jigsaw puzzles and cipher boxes, their verbal and physical contact increased, along with the frequency of cooperative behavior, implying that the frequency of the children's cooperative behavior was related to the manipulability of the materials and whether or not they were visually appealing. This finding was also obtained by Sun and et al., (2021a).

In terms of cooperation quality, direct intervention from teachers as an external force cannot effectively resolve the cooperation problem and improve children's cooperation ability when their cooperation produces contradictions and conflicts. In contrast, allowing young children to resolve nondangerous disputes on their own promotes increased cooperation. As an example, when observing a conflict between two children, L and Z, over the completion of a task, the teacher did not intervene directly. Instead, she observed the children's reactions from a distance. Since the teacher determined that the conflict stemmed from disagreements over route planning, she did not directly judge which idea was correct but rather asked whether both could be tried in practice. Finally, under the teacher's guidance, the two children resolved their dispute through practice and successfully completed the task. This shows that it is important for teachers to observe and analyze children's behaviors and intervene and guide appropriately to improve the quality of cooperation.

7 Conclusion and limitations

This study aims to compare and contrast the similarities and differences in the effectiveness of plugged-in and unplugged activities with similar learning content and assessment methods. The most obvious finding from the analysis is that both plugged-in and unplugged activities significantly improved the young children's computational thinking after a short-term educational intervention, and there were no significant differences between the two groups. Another finding is that there was a difference in the extent to which the plugged-in and unplugged activities promoted the development of subdimensions of computational thinking. Specifically, the plugged-in group demonstrated significant improvements in the dimensions of hardware, algorithm, and modularization, whereas the unplugged group demonstrated significant improvements in the dimension of representation.

Observations of the learning process generated two findings. First, the young children showed high motivation to study in both the plugged and unplugged activities. Second, despite the young children exhibiting cooperative behaviors in both the plugged-in and unplugged activities, the frequency of cooperative behaviors was more related to the learning materials, whereas the quality of cooperation was more related to the teacher's guidance.

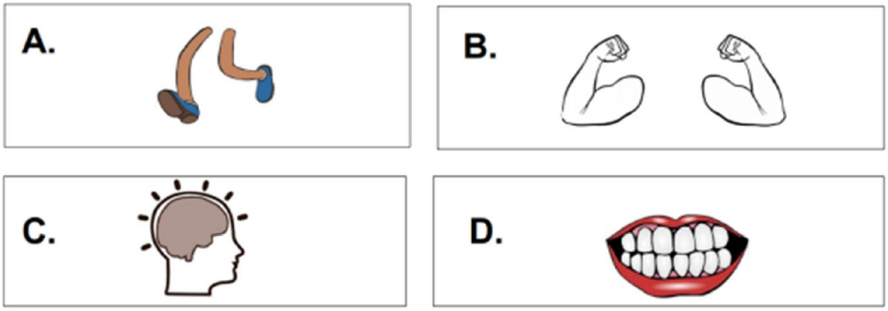
Apart from the above research findings, this study has added to the literature by developing an observation scale for unplugged activities. Data from the experiment indicate that the scale has high validity and can be used to assess children's computational thinking ability developed by participating in unplugged activities.

A limitation of this study is that observation scales and test questions were primarily quantitative evaluations based on adult perspectives, ignoring the idea of young children as "curriculum experiencers", which could have a negative impact on the completeness and objectivity of the results. An additional uncontrolled factor is that there was a limited sample size in this study, which might reduce the generalizability of the results. Consequently, in a follow-up study, we will consider using a mosaic research method to further understand the learning experience of preschoolers and obtain more qualitative materials from larger samples to increase the credibility of the research results.

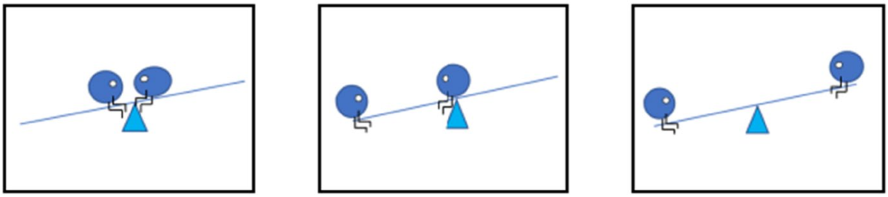
Appendix A

TechCheck-K sample items (Relkin et al., 2020)

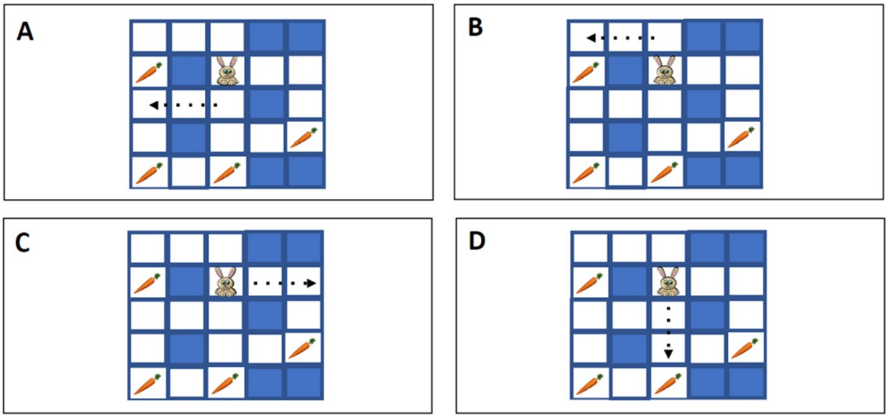
Which works the most like a computer?



This seesaw isn't going up and down. How can it be changed so it works?



The bunny can only hop one white square at a time. What is the fastest way for the bunny to get ONE carrot?



Appendix B

Sample of observation scale for CT in plugged activities (OSCT-Dodobot)

Instructions: Circle the level (1 – 4) in each category that best describes the child's highest level of ability in each category based on observation of their activities during interactive play with KIBO. If there is insufficient evidence to score a particular category, circle "NS" (Not Scorable). Do not give more than one rating per category or write in fractional ratings (e.g.: "Level 2.5").

Dimension	Level	Typical Behaviors
Algorithms/Modularity	NS	Not Scorable (child did not exhibit any programming behaviors related to algorithms/modularity in the play)
	1	Using color codes in random or nonsensical order Doesn't create meaningful sequences in Dodobot's color
	2	Correctly use of 2 types of color codes but does not create or show understanding of longer sequences Does not break programming tasks into smaller parts
	3	Correctly use of 2 types of color codes in syntactically correct sequences Begins to divide more complicated programming tasks into simpler steps
	4	Breaks up programming task into parts that are inter-dependent or recursive Uses clusters of blocks as units in larger programs

Dimension	Level	Typical Behaviors
Control Structure	NS	Not Scorable (child did not exhibit any programming behaviors related to algorithms/modularity in the play)
	1	No understanding or use of repeats or conditionals
	2	No understanding or use of repeats or conditionals and usually make errors
	3	Understanding or use of repeats Understanding conditionals but occasionally make errors
	4	Understanding or use of repeats or conditionals

Since each computational thinking learning activity may not necessarily cover all dimensions of computational thinking, corresponding observation items can be attached based on the taught content

Appendix C

Coefficient of variation of OSCTU

Dimension	Evaluation metrics	Mi	σi	Vi
Algorithm/Modularity	objective	4.40	0.548	0.1245
	steps	4.60	0.548	0.1191
	order	4.60	0.548	0.1191
Control Structures	comprehend	4.40	0.548	0.1245
	identify	5.00	0.000	0.0000
	apply	5.00	0.000	0.0000
	migrate	5.00	0.447	0.0894
Representation	realize	4.80	0.000	0.0000
	comprehend	4.20	0.447	0.1064
	apply	5.00	0.000	0.0000
Debugging	Identify problems	4.80	0.447	0.0931
	Define the problem	5.00	0.000	0.0000
	Correct errors	5.00	0.000	0.0000
Design process	plan	3.80	1.095	0.2888
	Share and assessment	3.60	0.894	0.2483

Appendix D

Coordination coefficient of OSCTU

dimension	Kendall's W	X^2	P
Algorithm / Modularity Control Structures Representation Debugging Design process	0.501	35.083	0.01**

** $p < 0.01$

Appendix E

Sample of Observation Scale for CT in Unplugged Activities (OSCTU)

Instructions: Circle the level (1 – 4) in each category that best describes the child's highest level of ability in each category based on observation of their activities during interactive play with KIBO. If there is insufficient evidence to score a particular category, circle "NS" (Not Scorable). Do not give more than one rating per category or write in fractional ratings (e.g.: "Level 2.5").

dimension	Evaluation metrics	Level	Typical Behaviors
Algorithms / Modularity	Obeject	NS	Problem-solving blindly without any direction or goal, incapacity to understand tasks or problems
		1	Can not fully understand the task or problem, have a basic outline of the goal to be solved but not clear
		2	Basic understanding of tasks or problems, and clear goals
		3	Have a comprehensive understanding of the task or problem, and have clear goals
		4	Fully understanding the task or problem, having clear and specific goals for solving the problem or completing the task
	Steps	NS	Lack of awareness of solving problems in steps
		1	There is a preliminary awareness of solving problems step by step, but do not know how to do it
		2	May try to solve problems step by step, but individual steps are not clear or simple enough
		3	It is possible to reasonably divide the steps, and each individual step is clear and simple
		4	The steps are logically divided, each step has a clear direction, very simple and clear
	0rder	NS	The order of the steps is disorderly, and there is no logical relationship between them
		1	When it comes to simple sorting, it is possible to reasonably design the sequence of steps, but there may be one or two mistakes
		2	When it comes to simple sorting, one can design the steps in a reasonable and correct order, but cannot sort the steps involving complex sorting (selective judging) in order
		3	Able to sort the steps involving complex sorting (selection judgment), but with a high error rate
		4	Being able to correctly sort steps involving complex sorting (selection judgment)
Control Structures	memorize and comprehend	0	Cannot memorize and comprehend any logical relationships
		1	Able to master the basic sequence
		2	Good at mastering order, but have difficulty in understanding complex logical relationships such as loops and repetitions
		3	Basic understanding of complex logical relationships such as repetition and looping
		4	It is easier to understand complex logical relationships such as loops, conditional statements, and so on

Since each computational thinking learning activity may not necessarily cover all dimensions of computational thinking, corresponding observation items can be attached based on the taught content

Supplementary information The online version contains supplementary material available at https://doi.org/10.1007/s10639-023-12181-x.

Funding The Development Planning Project of Philosophy and Social Science in Guangzhou (2023GZYB72); Guangdong Education Science Planning Project (2018JKZ021).

Data availability The datasets generated during and analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest None.

References

Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior, 105*, 105954. https://doi.org/10.1016/j.chb.2019.03.018

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *Acm Inroads, 2*(1), 48–54. https://doi.org/10.1145/1929887.1929905

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

Bell, T., & Vahrenhold, J. (2018). CS unplugged—How is it used, and does it work? In H. J. Böckenhauer, D. Komm, & W. Unger (Eds.), *Adventures Between Lower Bounds and Higher Altitudes. Lecture Notes in Computer Science* (1st ed., pp. 497–521). Springer. https://doi.org/10.1007/978-3-319-98355-4_29

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology, 13*(1), 20–29.

Bers, M. U. (2008). *Blocks to robots: Learning with technology in theearly childhood classroom*. Teachers College Press.

Bers, M. U. (2010). The TangibleK robotics program: Applied computational thinking for young children. *Early Childhood Research & Practice, 12*(2), 1–20. https://www.ecrp.uiuc.eduv12n2/bers.html

Bers, M. U. (2018a). Coding as a playground: Programming and computational thinking in the early childhood classroom. *Routledge*. https://doi.org/10.4324/9781315398945

Bers, M. U. (2018). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. *European Journal of STEM Education, 3*(3), 08. https://doi.org/10.20897/ejsteme/3868

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education, 72*, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

Bers, M. U., González-González, C., & Armas-Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education, 138*, 130–145. https://doi.org/10.1016/j.compedu.2019.04.013

Bers, M. U., Strawhacker, A., & Sullivan, A. (2022). The state of the field of computational thinking in early childhood education. *OECD*. https://doi.org/10.1787/19939019

Bers, M. U., Ponte, I., Juelich, C., Viera, A., & Schenker, J. (2002). Teachers as designers: Integrating robotics in early childhood education [Paper Presentation]. *Association for the Advancement of Computing in Education (AACE).* Retrieved June 19, 2023 from https://www.learntechlib.org/primary/p/8850/

Botički, I., Pivalica, D., & Seow, P. (2018). The use of computational thinking concepts in early primary school [Paper Presentation]. International Conference on Computational Thinking Education, Hong Kong. Retrieved June 19, 2023 from https://www.eduhk.hk/cte2018/

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school [Paper Presentation]. *12th workshop on primary and secondary computing education,* New York, NY, United States. https://doi.org/10.1145/3137065.3137069

Branch, R. M. (2009). *Instructional design: The ADDIE approach* (Vol. 722). New York: Springer. https://doi.org/10.1007/978-0-387-09506-6

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking [Paper Presentation]. *Annual meeting of the American Educational Research Association, Vancouver, Canada*. https://www.researchgate.net/publication/265797241

Caballero-Gonzalez, Y. A., Munoz, L. & Munoz-Repiso, A. G. V. (2019) Pilot experience: Play and program with bee-bot to foster computational thinking learning in young children [Paper Presentation]. 7th International Engineering, Sciences and Technology Conference (IESTEC). https://doi.org/10.1109/iestec46403.2019.00113

Coulter, B., Lee, I., & Martin, F. (2010). Computational thinking for youth. Citeseer

Critten, V., Hagon, H., & Messer, D. (2022). Can pre-school children learn programming and coding through guided play activities? A case study in computational thinking. *Early Childhood Education Journal, 50*(6), 969–981. https://doi.org/10.1007/s10643-021-01236-8

Curzon, P., Peckham, J., Taylor, H., Settle, A., & Roberts, E. (2009). Computational thinking (CT): On weaving it in. *SIGCSE Bulletin, 41*(3), 201–202. https://doi.org/10.1145/1562877.1562941

de Ruiter, L. E., & Bers, M. U. (2022). The Coding Stages Assessment: Development and validation of an instrument for assessing young children's proficiency in the ScratchJr programming language. *Computer Science Education, 32*(4), 388–417. https://doi.org/10.1080/08993408.2021.1956216

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

Elkin, M., Sullivan, A., & Bers, M. U. (2016). Programming with the KIBO robotics kit in preschool classrooms. *Computers in the Schools, 33*(3), 169–186. https://doi.org/10.1080/07380569.2016.1216251

Flannery, L. P., & Bers, M. U. (2013). Let's dance the "robot hokey-pokey!" children's programming approaches and achievement throughout early cognitive development. *Journal of Research on Technology in Education, 46*(1), 81–101. https://doi.org/10.1080/15391523.2013.10782614

Gibson, J. P. (2012). Teaching graph algorithms to children of all ages [Paper Presentation]. *17th ACM Annual Conference on Innovation and Technology in Computer Science Education, Haifa, Israel.* https://doi.org/10.1145/2325296.2325308

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Heikkilä, M., & Mannila, L. (2018). Debugging in programming as a multimodal practice in early childhood education settings. *Multimodal Technologies and Interaction, 2*(3), 42. https://doi.org/10.3390/mti2030042

Heljakka, K., Ihamäki, P., Tuomi, P., & Saarikoski, P. (2019). Gamified coding: toy robots and playful learning in early education [Paper Presentation]. *2019 International Conference on Computational Science and Computational Intelligence,* Las Vegas, NV, USA. https://doi.org/10.1109/CSCI49370.2019.00152

Huang, W., & Looi, C. K. (2021). A critical review of literature on "unplugged" pedagogies in K-12 computer science and computational thinkin. *Computer Science Education, 31*(1), 83–111. https://doi.org/10.1080/08993408.2020.1789411

Jun, W. (2018). A Study on development of evaluation standards for unplugged activity [Paper Presentation]. *2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju,* Korea. https://doi.org/10.1109/ICTC.2018.8539505

Looi, C. K., How, M. L., Longkai, W., Seow, P., & Liu, L. (2018). Analysis of linkages between an unplugged activity and the development of computational thinking. *Computer Science Education, 28*(3), 255–279. https://doi.org/10.1080/08993408.2018.1533297

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51–61. https://doi.org/10.1016/j.chb.2014.09.012

Messer, D., Thomas, L., Holliman, A., & Kucirkova, N. (2018). Evaluating the efectiveness of an educational programming intervention on children's mathematics skills, spatial awareness and working memory. *Education and Information Technologies, 23*(6), 2879–2888. https://doi.org/10.1007/s10639-018-9747-x

Metin, S. (2022). Activity-based unplugged coding during the preschool period. *International Journal of Technology and Design Education, 32*(1), 149–165. https://doi.org/10.1007/s10798-020-09616-8

Monteiro, A. F., Miranda-Pinto, M., & Osório, A. J. (2021). Coding as literacy in preschool: A case study. *Education Sciences, 11*(5), 198. https://doi.org/10.3390/educsci11050198

Polat, E., & Yilmaz, R. M. (2022). Unplugged versus plugged-in: Examining basic programming achievement and computational thinking of 6th-grade students. *Education and Information Technologies, 27*(7), 9145–9179. https://doi.org/10.1007/s10639-022-10992-y

Relkin, E., de Ruiter, L. E., & Bers, M. U. (2020). TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education. *Journal of Science Education and Technology, 29*(4), 482–498. https://doi.org/10.1007/s10956-020-09831-x

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*, 1–15. https://doi.org/10.1016/j.compedu.2021.104222

Relkin, E., Bers, M. U. (2022). TechCheck-K: A measure of computational thinking for kindergarten children [Paper Presentation]. *IEEE Global Engineering Education Conference (EDUCON)*, Vienna, Austria. Retrieved June 19, 2023 from https://ieeexplore.ieee.org/abstract/document/9453926

Relkin, E. (2018). *Assessing Young Children's Computational Thinking Abilities* [Unpublished master dissertation]. University of Tufts. Retrieved June 20, 2023 from https://sites.bc.edu/devtech/wp-content/uploads/sites/113/2015/10/Relkin-thesis.pdf

Rodriguez, B., Kennicutt, S., Rader, C., & Camp, T. (2017). Assessing computational thinking in CS unplugged activities [Paper Presentation]. *2017 ACM SIGCSE Technical Symposium on Computer Science Education*, Seattle Washington, USA https://doi.org/10.1145/3017680.3017779

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2020). Designing unplugged and plugged activities to cultivate computational thinking: An exploratory study in early childhood education. *The Asia-Pacific Education ResearcHer, 29*(1), 55–66. https://doi.org/10.1007/s40299-019-00478-w

Selby, C. C., & Woollard, J. (2013). Computational thinking: The developing definition[Paper Presentation]. *18th annual conference on innovation and Technology in Computer Ccience Education,* Canterbury, UK. Retrieved June 19, 2023 from https://eprints.soton.ac.uk/356481/1/Selby_Woollard_bg_soton_eprints.pdf

Stoeckelmayr, K., Tesar, M., Hofmann, A. (2011). Kindergarten children programming robots: A first attempt [Paper Presentation]. *the 2nd International Conference on Robotics in Education (RIE)*, London, UK.

Su, J., & Yang, W. (2023). A systematic review of integrating computational thinking in early childhood education. *Computers and Education Open*, 4, 100122. https://doi.org/10.1016/j.caeo.2023.100122

Sullivan, A., Elkin, M., & Bers, M. U. (2015). KIBO robot demo: Engaging young children in programming and engineering [Paper Presentation]. *14th International Conference on Interaction Design and Children,* New York, NY, United States. https://doi.org/10.1145/2771839.2771868

Sun, L., Hu, L., & Zhou, D. (2021a). Single or combined? A study on programming to promote junior high school students' computational thinking skills. *Journal of Educational Computing Research, 60*(2), 283–321. https://doi.org/10.1177/07356331211035182

Sun, L., Guo, Z., & Hu, L. (2021b). Educational games promote the development of students' computational thinking: a meta-analytic review. *Interactive Learning Environments,* 1–15. https://doi.org/10.1080/10494820.2021.1931891

Thies, R., & Vahrenhold, J. (2013). On plugging unplugged into CS classes [Paper Presentation]. *44th ACM Technical Symposium on Computer Science Education,* Denver, CO, USA. https://doi.org/10.1145/2445196.2445303

Torres, N. B., González, R. L., & Carvalho, J. L. (2018). Roamer, un robot en el Aula de Educación Infantil para el Desarrollo de Nociones Espaciales Básicas. *RISTI-Revista Ibérica de Sistemas e Tecnologias de Informaçâo, 28*, 14–28. https://doi.org/10.17013/risti.28.14-28

Tsarava, K., Moeller, K., & Ninaus, M. (2018). Training computational thinking through board games: The case of Crabs & Turtles. *International Journal of Serious Games, 5*(2), 25–44. https://doi.org/10.1080/08993408.2021.1877988

Wing, J. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology, 25*(2), 7–14. https://doi.org/10.17471/2499-4324/922

Wohl, B., Porter, B., & Clinch, S. (2015). Teaching computer science to 5–7 year-olds: An initial study with scratch, cubelets and unplugged computing [Paper Presentation]. *Workshop in Primary and Secondary Computing Education, New York, NY, United States.* https://doi.org/10.1145/2818314.2818340

World Health Organization. (2019). Guidelines on physical activity, sedentary behaviour and sleep for children under 5 years of age. Retrieved on June 20, 2023 from https://www.who.int/publications/i/item/9789240015128

Xinogalos, S., Satratzemi, M., & Malliarakis, C. (2017). Microworlds, games, animations, mobile apps, puzzle editors and more: What is important for an introductory programming environment? *Education and Information Technologies, 22*, 145–176.

Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education, 28*(4), 371–400. https://doi.org/10.1080/08993408.2018.1560550

Yang, W., Ng, D. T. K., & Gao, H. (2022). Robot programming versus block play in early childhood education: Effects on computational thinking, sequencing ability, and self-regulation. *British Journal of Educational Technology, 53*(6), 1817–1841. https://doi.org/10.1111/bjet.13215

Zhan, Z., He, W., Yi, X., & Ma, S. (2022). Effect of unplugged programming teaching aids on children's computational thinking and classroom interaction: With respect to Piaget's four stages theory. *Journal of Educational Computing Research, 60*(5), 1277–1300. https://doi.org/10.1177/07356331211057143

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.