Lin, X.-F., Wang, J., Chen, Y., Zhou, Y., Luo, G., Wang, Z., Liang, Z.-M., Hu, X., & Li, W. (2023). Effect of a Reflection-Guided Visualized Mindtool Strategy for Improving Students' Learning Performance and Behaviors in Computational Thinking Development. *Educational Technology & Society*, *26*(2), 165-180. https://doi.org/10.30191/ETS.202304_26(2).0012

Effect of a Reflection-Guided Visualized Mindtool Strategy for Improving Students' Learning Performance and Behaviors in Computational Thinking Development

Xiao-Fan Lin[1,2,3*], Jing Wang[3,4], Yingshan Chen[3], Yue Zhou[3], Guoyu Luo[3], Zhaoyang Wang[3], Zhong-Mei Liang[5], Xiaoyong Hu[3,6] and Wenyi Li[2]

[1]Guangdong Provincial Philosophy and Social Sciences Key Laboratory of Artificial Intelligence and Smart Education, Guangdong Engineering Technology Research Center of Smart Learning, South China Normal University, Guangzhou, P.R. China // [2]Guangdong Provincial Institute of Elementary Education and Information Technology, Guangzhou, P.R. China // [3]School of Education Information Technology, South China Normal University, Guangzhou, P.R. China // [4]Teacher Education College of Guangdong-Hong Kong-Macao Greater Bay Area, South China Normal University, Guangzhou, P.R. China // [5]Zhixin South Road Primary School, Guangzhou, P.R. China [6].Institute of Artificial Intelligence in Education, South China Normal University, Guangzhou, P.R. China// linxiaofan@m.scnu.edu.cn // jjw15683078248@163.com // 2021020874@m.scnu.edu.cn // 15107152939@163.com // 2475317832@qq.com // m17754831067@163.com // 351632147@qq.com // 472275060@qq.com // liwenyi@pku.edu.cn

[*]Corresponding author

ABSTRACT: Computational thinking (CT) is an imperative competency in the 21st century. Mindtools can assist students in understanding concepts and decomposing tasks during CT development through programming. However, students may encounter challenges in complex CT problem-solving tasks due to being confused when using mindtools without proper guidance. Research evidence shows the potential of reflection in complex CT problem-solving by regulating cognitive activities. Accordingly, this study designed a reflection-guided visualized mindtool strategy to address CT development challenges. A quasi-experiment and lag sequential analysis were conducted by recruiting 97 junior high school students to examine the effects of the proposed strategy on CT development and to explore students' behavior patterns. Results revealed that the proposed approach improved students' CT performance, CT disposition, meta-cognitive awareness, and learning motivation. Students learning with the proposed strategy exhibited more key behaviors of facilitating CT problem-solving (e.g., generalizing the knowledge, re-designing the algorithm scheme, and evaluating the feasibility of their proposed schemes) than students in the control group, revealing the essential process of CT development and enlightening teachers on guiding students to produce such learning processes when cultivating CT.

Keywords: Reflection, Mindtool, Computational thinking, Behavior, Junior high school students

1. Introduction

Computational thinking (CT) is considered an imperative competency for everyone in the 21st century (Denning, 2017; Hsu et al., 2018). Existing studies show that learning CT benefits students' high-level thinking abilities, such as problem-solving, critical thinking, creativity, and collaboration (Denner et al., 2019; Scherer et al., 2019). CT has been incorporated into the national curricula of many countries, especially in programming courses, which are regarded as one of the most effective approaches to developing CT (Ezeamuzie & Leung, 2021; Shute et al., 2017; Wing, 2006). Nevertheless, there are challenges for novice CT learners to break down problems into small subproblems in the problem-solving process, generalize what they have learned in programming courses, and transfer it to solve problems in authentic and complex contexts (Zhao et al., 2022). Furthermore, it was found that students might continually encounter certain CT challenges related to programming concepts due to the abstraction and complexity of concepts such as event handling, conditionals, and manipulation of variables (Grover et al., 2016; Mouza et al., 2020). Thus, it is suggested that additional guidance or tools be provided for students to better grasp complex CT concepts (Lye & Koh, 2014).

Mindtools have been widely recognized in CT development through computer programming courses (Zhang et al., 2021). Concept maps are one of the mindtools that are believed to help students understand complex knowledge and to promote higher-order thinking, such as problem-solving (Jonassen & Carr, 2020). With the support of concept maps, students can easily divide an entire task into smaller subtasks in programming (Zhao et al., 2022). Although existing evidence has shown the role of concept maps in CT development (Krieglstein et al., 2022), students face several challenges when dealing with complex CT problem-solving tasks with concept maps (Wang et al., 2017). Specifically, self-generated concept maps may be incomplete and incorrect (e.g., unable to create meaningful connections) because of the lack of teachers' guidance (Eshuis et al., 2021; Wong et al., 2021).

Besides, students may also be confused about identifying gaps and misconceptions in their knowledge even when they have constructed a concept map (Eshuis et al., 2021). CT is not only a cognitive but also a meta-cognitive thinking process that regulates one's cognitive activities (Chen et al., 2021). Therefore, it is necessary to introduce appropriate guidance for students to help them review knowledge and regulate the meta-cognitive thinking process for CT development during problem-solving.

Reflection, which is a meta-cognitive strategy (Medina et al., 2017), has the potential to improve CT because students can be aware of their cognitive process and use this awareness to regulate their problem-solving (Colbert et al., 2015). According to Schön (1987), the mastery of a subject depends on a person's ability to reflect on the spot. Previous studies have indicated that reflection increased students' self-efficacy, learning motivation, meta-cognitive awareness, and CT performance by comparing new and old knowledge and integrating different opinions to understand errors and misconceptions (Chen et al., 2021; He et al., 2021; Lin et al., 2022a). Moreover, several scholars have also reported the effectiveness of reflection in figuring out connections between theory and practice (Radović et al., 2021), identifying misconceptions and deficiencies in knowledge (Cavilla, 2017), and developing a deeper understanding (Ghanizadeh, 2017). Therefore, the integration of mindtools and reflection may offer potential advantages for students to address complex learning problems (Chang & Hwang, 2022).

1.1. Research gaps

Although previous studies (e.g., Chang & Hwang, 2022) have noted the role of mindtools and reflection in complex problem-solving contexts, they did not regard the incorporation of mindtools and reflection as a teaching approach and measure its effect on CT education. In addition, there are conflicting findings regarding whether the use of mindtools and reflection enhances students' learning outcomes. For example, Chang and Hwang (2022) argued that reflection facilitated students' structural knowledge when they reflected on feedback generated by a mindtool, while Eshuis et al. (2021) indicated that integrating reflection prompts into a mindtool did not work as expected and could not help students improve their learning. Although there is a lack of relevant studies on adopting the combination approach of mindtool and reflection in CT development, this controversial situation may also be transferred to students' CT development. A general challenge is that CT is usually developed through programming in diverse operationalized ways, which may ignore cultivating the complex problem-solving ability of CT, potentially accounting for some debatable issues and conflicting findings (Ezeamuzie & Leung, 2021). Despite a plethora of CT studies in the general area of programming (e.g., Wu et al., 2019), robust empirical research investigating the effectiveness of a mental process to cultivate CT is more limited and often focused on the role of cognitive skills for CT development (Ezeamuzie & Leung, 2021). Rather mechanical understandings of CT development formerly prevailed, leading to students' frustration and interest reduction when they were faced with programming challenges (Sun et al., 2021). More recently, effective studies on CT development have conceptualized it as processes in which students have to actively participate in the cognitive and meta-cognitive process and be aware of regulating their problem-solving (Chen et al., 2021). In order to clarify what differentiates more effective from less effective reflective practices, there is a need to incorporate reflective approaches with mindtool-based CT development in relation to the thinking process (Lin et al., 2022a). To sum up, reflection could be considered an effective approach to promote students' active participation in the meta-cognitive process, and visualized mindtools could be regarded as a useful support strategy for facilitating cognitive development in CT learning. Therefore, it seems necessary to move beyond operationalized CT processes in various contexts of research and take into consideration a novel instructional strategy for problem-solving with a more simplified procedure (e.g., discover, extract, create, and assemble) (Ezeamuzie, 2022). In addition, few studies have used Lag sequential analysis (LSA) to focus on how students participate in the process of CT development and to what extent learning is facilitated from the perspective of behaviors (He et al., 2021).

1.2. Research questions

To address these gaps, the present study proposed a reflection-guided visualized mindtool strategy and assessed its effect on CT development. We experimented with the strategy and investigated its impact on students' performance and CT disposition. Measuring CT disposition is important for assessing whether students will work consistently in the CT development process despite being frustrated and failing due to challenges posed by complex problems (Jong et al., 2020). Besides, we were also interested in students' motivation as the result of the strategy, considering that reflection often plays a role in enhancing learners' CT learning motivation (Fang et al., 2022). It is also worth considering that students' CT can be developed by improving their metacognition (Chen et al., 2021). Measuring meta-cognitive awareness provides a good indicator that reflects changes in students' metacognition after incorporating reflection into the mindtool-based CT development process (Lin et al., 2022b). To understand how students' CT works through programming in a comprehensive and specific way, it is necessary for us to observe the details of students' behavior in the understanding process of the CT concept and problem-solving from a process-oriented perspective and at a micro level (Sun et al., 2021). LSA has been noted as an effective method to explore students' behavior patterns and learning performances with a contextualized reflective mechanism (Lin et al., 2022b). Accordingly, we conducted a quasi-experiment in a Fire Extinguishing AI Robot task of a programming course and examined students' CT development behaviors with LSA to explore their learning processes. We aimed to visualize the patterns and detect the sequential relationships between each behavior. This is one of the few studies that has used behavior analysis to explore the essential process of students' CT development behaviors in an experimental study with the reflection-guided visualized mindtool strategy. Specifically, the current study addressed the following research questions:

- RQ1. Are there any significant differences in students' CT performance (RQ1.1) and CT disposition (RQ1.2) in the Fire Extinguishing AI Robot task of a programming course which adopted the reflection-guided visualized mindtool strategy (RVMS) and the visualized mindtool strategy (VMS)?
- RQ2. Are there any significant differences in students' learning motivation (RQ2.1) and meta-cognitive awareness (RQ2.2) in the Fire Extinguishing AI Robot task of a programming course which adopted RVMS and VMS?
- RQ3. Are there any differences between the behavior patterns of those learning with RVMS and VMS?

2. Literature review

2.1. Computational thinking and programming

Computational thinking was first defined by Wing (2006) as a way of thinking to solve problems, design systems, and understand human behaviors with the use of fundamental computer science concepts. Although CT has been defined from different perspectives since Wing (2006), common points include that CT is the ability to use computer science concepts to solve problems, including computational concepts and computational practices in programming (Ezeamuzie & Leung, 2021). Buitrago Flórez et al. (2017) claim that students could better develop solutions for complicated problems in the real world with CT. Besides, several previous studies have shown that CT could benefit students' academic achievement (Lei et al., 2020), cognitive benefit (Scherer et al., 2019), problem-solving, and computer science attitudes (Denner et al., 2019).

Researchers have indicated that CT could be developed and facilitated through proper approaches (e.g., programming, robotics, and simulations) (Shute et al., 2017). It is generally believed that programming is a prominent and effective way to cultivate CT (Ezeamuzie & Leung, 2021; Lye & Koh, 2014), as it involves breaking down a problem into smaller problems and expressing a solution in the form of computational steps and algorithms (Merino-Armero et al., 2022). However, there is a lack of in-depth analysis of the way of thinking used in CT, which leads to students' difficulty in developing CT when engaged in programming learning (e.g., understanding of logics and semantics) (Shute et al., 2017). Additionally, previous research has pointed out the challenges in CT development practices, including difficulties understanding and applying complex CT concepts (Mouza et al., 2020), failure to decompose problems, and being unable to solve problems effectively by using CT and implementing solutions practically due to a lack of detailed mental models (e.g., mental maps, Venn diagrams) (Buitrago Flórez et al., 2017). To address the above difficulties of CT development, some studies have proved that scaffolding from appropriate visualized tools is essential for developing students' programming knowledge and thinking skills (Buitrago Flórez et al., 2017; Lye & Koh, 2014; Omer et al., 2020). Researchers have noted that mindtools play an important role in representing and organizing knowledge in computer programming courses (Zhang et al., 2021), which could help students understand complex knowledge and decompose tasks (Jonassen & Carr, 2020; Zhao et al., 2022).

2.2. Mindtool

Mindtools have been suggested as an effective method to engage students in organizing and presenting their knowledge through computer application programs (Jonassen et al., 1998; McAleese, 1998). Concept maps, as mindtools, have been widely applied in educational settings (Chang et al., 2022). With concept maps, individuals can generate meaningful learning by representing concepts with nodes, and the relationships between concepts with links (Yue et al., 2017). Moreover, students engage in a deep cognitive process while combining and representing conceptual knowledge structures (Novak & Gowin, 1984).

Chang et al.'s (2022) study shed light on the potential of concept maps to improve computer science conceptual understanding. To help students understand complex knowledge, researchers have applied concept maps in complex problem-solving processes, such as breaking down the whole programming task, presenting logical thinking, and applying knowledge to practice, which might be difficult for students (Wang et al., 2017; Zhao et al., 2022). There is significant interest in two instructional methods of concept maps: self-generated concept maps and constructing on a scaffold (also known as the fill-in-the-blank construct) (Chang et al., 2001). Although self-generated concept maps have been applied less than fill-in-the-blank concept maps in studies, they offer a high degree of flexibility that can benefit students with different learning styles (Oliver, 2008). While fill-in-the-blank concept maps are helpful for short-term learning, their restriction of freedom of content and structure may become a new constraint during long-term learning or in the case of complex knowledge (Wong et al., 2021). In a similar context for conceptual understanding and complex problem-solving, it is reasonable to apply the above-mentioned pedagogies (e.g., self-generated and fill-in-the-blank concept maps) to students' CT development. Although studies have confirmed that self-generated concept maps are more conducive to students' permanent development, constructing concept maps is challenging for students with low prior knowledge without extra support (Wong et al., 2021; Chuang et al., 2018). To improve students' programmable logic controllers knowledge performance, previous literature has noted the importance of promoting students' active participation in both cognitive and meta-cognitive learning processes for their CT development (Chen et al., 2021). Otherwise, students have difficulties eliminating the misunderstandings of the concepts by improving concept maps, which still existed even after completing concept maps (Eshuis et al., 2021).

2.3. Reflection

Reflection is regarded as an imperative activity in education research, including work by Dewey (1933). He defined reflection as continuously evaluating one's performances or behaviors to gain a deeper understanding of one's experiences (Dewey, 1933). Rodgers (2002) explained Dewey's concept of reflection as a four-stage process: presence to experience, description of experience, analysis of experience, and intelligent action, and noted that the action stage was often overlooked. Schön (1987) further studied reflective practice and divided the reflective practice into reflection-before-action, reflection-in-action, and reflection-on-action. Among these practices, Schön (1987) regarded reflection-in-action (occurring while performing the task) as the centre of art through which practitioners can cope with troublesome practices.

The effectiveness of reflection has been well recognized in the literature. Reflection is regarded as an essential meta-cognitive strategy for obtaining meaningful learning from specific experiences (Medina et al., 2017). Research has shown that reflection could facilitate complex problem-solving by critiquing the initial understanding of phenomena and constructing new descriptions (Schön, 2017). Besides, reflection is also helpful when cognitive knowledge is lacking. With the help of reflection, learners can observe and evaluate themselves, examine the gaps in their understanding, and their thinking and behavior can evolve (He et al., 2021; Lin et al., 2022a). There are many approaches to support and direct students' reflection, including prompting and guiding questions, think-aloud protocols, and peer assessment with assessment criteria (Radović et al., 2021). Studies have identified the benefits of these methods in students' learning performance (Fang et al., 2022; Radović et al., 2021). Meta-cognitive prompts were found to direct students' attention to important aspects of CT during their problem-solving processes and trigger their self-reflection (Chen et al., 2021). The adoption of think-aloud protocols has been suggested as students' cognitive processes can be verbalized and they can better understand CT practices (Lye & Koh, 2014). Fang et al. (2022) indicated that students could be encouraged to improve their work and tendency to solve complicated problems more systematically when receiving constructive comments from their peers.

Regarding CT, scholars have indicated that students are mainly situated in learning contexts that focus on the passive use of syntax and algorithms while lacking opportunities to engage in in-depth thinking about systematic approaches to problem-solving, which could lead to difficulty in facing real problems (Buitrago Flórez et al., 2017). Reflection can inspire students to have a stronger sense of motivation to engage in CT problem-solving and can develop their confidence (Chen et al., 2021). Besides, reflection has the potential to trigger students' metacognition in the CT development process (Buitrago Flórez et al., 2017). Previous studies suggested the likely effect of metacognition on directing students' attention to the critical aspects of CT and on helping students evaluate and question their solutions from different perspectives and discovering the limitations of their thinking during their problem-solving process (Chen et al., 2021; He et al., 2021), which could support the generalization of CT concepts and practices in more complex problem-solving contexts. Accordingly, there is a necessity to fill the research gap due to the lack of relevant studies on adopting the combination approach of mindtools for cognition and reflection for meta-cognitive thinking in CT development. In addition, there are conflicting findings concerning the effectiveness of integrating reflection prompts into mindtools (Chang & Hwang, 2022; Eshuis et al., 2021). In order to clarify what more effective CT practices are, there is a need to incorporate a reflective approach with mindtools for CT development (Lin et al., 2022b). Therefore, we proposed a reflection-guided visual mindtool strategy to engage students in learning CT from a cognitive to meta-cognitive perspective.

3. Programming teaching design within a reflection-guided visualized mindtool strategy

3.1. System structure

In this study, we developed a reflection-guided visualized mindtool strategy system. The system structure includes a reflection-guided strategy, a mindtool promoting CT algorithm design mechanism, and several databases. The reflection-guided strategy enabled students to complete the reflection activity with the guidance of reflective prompts, reflective evaluation rubrics, and stimulated recall reflections. With the guidance of reflective prompts, students are more engaged in adaptation, transfer, synthesis of information, or asking for directions to create a better CT outcome. With the guidance of reflective evaluation rubrics, every student is responsible for the judgment results of peers' projects. With the guidance of stimulated recall reflections, students can explain or reflect on certain behavior patterns. Mindtool promoting CT algorithm module may provide students with an overall view of the cognitive knowledge structure and their cognitive process by providing self-generated and fill-in-the-blank concept maps. With the guidance of fill-in-the-blank concept maps, students can easily construct CT knowledge maps and better discover the connection between knowledges. With the guidance of self-generated concept maps, students can visualize their thinking process, helping them break down the task and clarify the step of the algorithm design. The mindtool promoting CT algorithm module also provides materials to connect the knowledge and thinking process for CT development with the open source software module and the open source hardware build module for completing CT tasks.

3.2. The context of the fire extinguishing AI robot task

In this self-developed online inquiry CT development environment, students are required to solve problems. They need to decompose the fire extinguishing AI robot task of a programming course (Figure 1) into several big steps, with one-by-one detailed solutions, in this process to develop their CT. The whole learning process is completed with the guidance of a reflection prompt. Reflective prompts may set up initial boundaries, trigger students to think more deeply, and make the most appropriate connection of concepts during the CT online inquiry activity. It can be inferred that when students are given reflective prompts such as specific content-based concept mapping prompts corresponding to CT development materials for the fire extinguishing AI robot, they can develop a more elaborated CT disposition than those without prompts. Also, reflective prompts provide an anchoring structure to which new information can be attached to existing schemas.

Figure 1. Operating scenario of the fire extinguishing AI robot

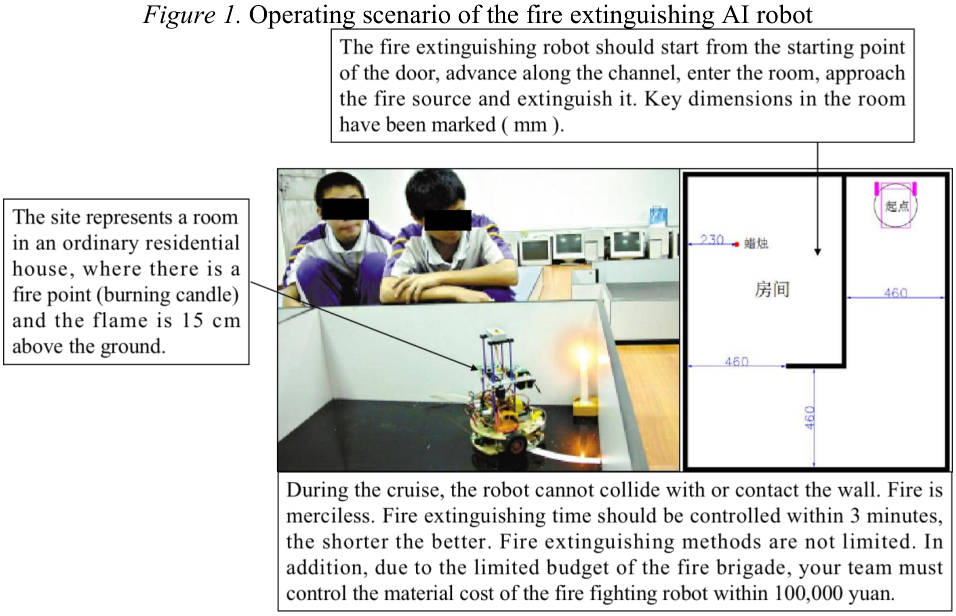
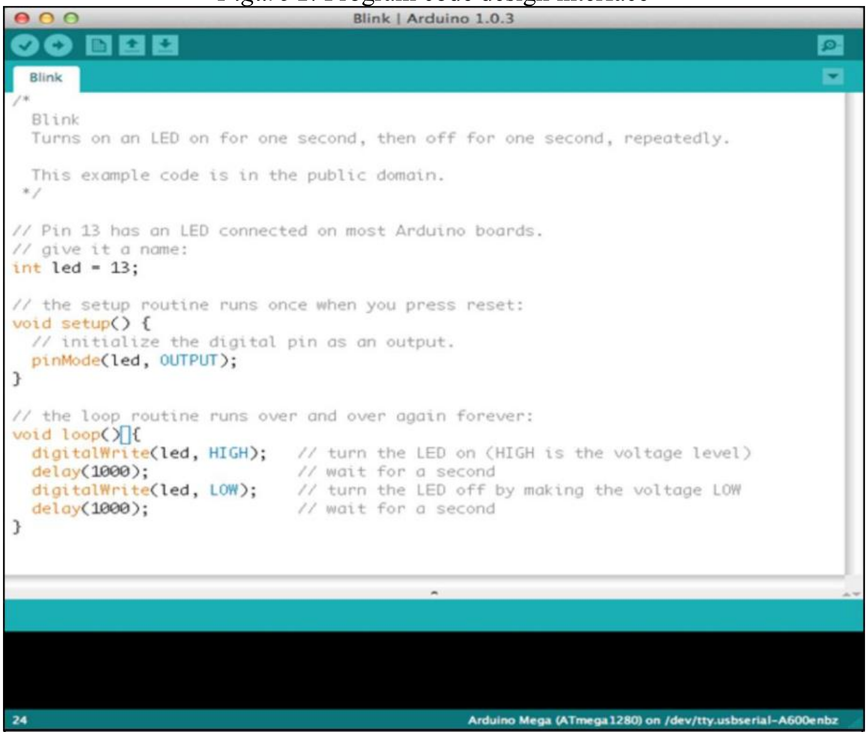


Figure 2 shows the algorithm design. In this process, students used a mindtool to plan and design a structured and complex algorithm script to implement the critical actions that robots needed to include in the fire extinguishing process. Besides, it might also help students identify and correct individual mistakes they have made in the individual design, construct a more logical structure, and rearrange their thinking logically. Students used Arduino 1.0.3 to write code to realize the algorithm design program. Figure 3 shows the open source software. Students used Corona SDK to develop mobile phone software and generalize to solve problems in similar situations. Figure 4 shows the hardware. Students used this hardware to build robots and complete the key actions designed in the program.

Figure 2. Program code design interface

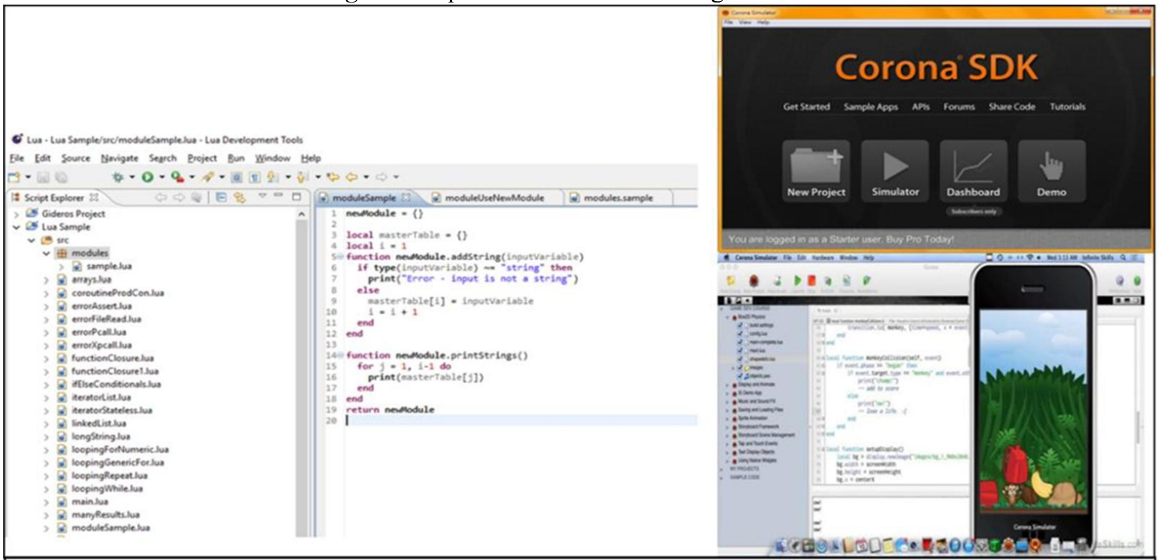


Using Arduino 1.0.3 to write code to implement the key program of the robot designed by the algorithm

The students completed the preliminary model building, found the model's shortcomings through the evaluation: Can the prototype of the fire extinguishing robot built at present meet the initial needs of requirements?; What is the reason?; What are the imperfections of the prototype?; What is the key gap?; and What is the improvement direction? With the guidance of reflective evaluation rubrics, every student was responsible for the judgment results of their peers' projects. After assigning scores for each dimension, students checked the rubrics to make sure that they had given an appropriate score to each dimension. After checking the rubrics, they reviewed the scores again to confirm whether their grading tasks were well done. Checking the rubrics was very important for students in the reflective reviewing task for CT tasks. Students spent a while checking the evaluating rules. They needed guidelines for measuring their peers' performance. With the rubrics, they could know whether their peers' projects met the expected criteria. Before choosing the scores for each dimension, they went through the evaluation rubrics and awarded the appropriate scores to their projects.

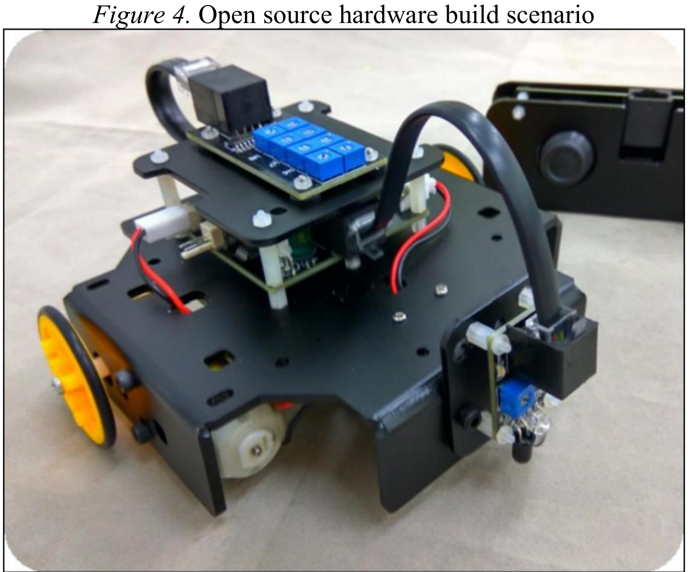
To promote CT development, the teacher videotaped a 1-hour lesson. During this period, the students were asked to review their programming peers' initial projects. During another 1-hour lesson, students were asked to revise their own projects based on reviewers' feedback. Then, the instructor used the first-round stimulated recall reflections to identify students' thought processes (i.e., behaviors) during the reviewing and revising phases (1 hour). With the replay of the video recordings, these students were allowed to see and explain their actions during the CT development within online inquiry tasks. Therefore, the categories, as well as behavior patterns were found. Finally, with the unfolding of some behavior patterns during CT development, the second-round stimulated recall reflections were administered with the think-aloud protocols for the collected data. The think-aloud protocols used reflections to involve students to describe their actions as they were watching video recordings of their online tasks. Thus, they were more likely to interpret or reflect on some specific behavior patterns in the consecutive stimulated recall reflections.

Figure 3. Open source software design interface



Using Corona SDK open source mobile phone software

Figure 4. Open source hardware build scenario



Building robots with open source hardware to implement key actions

4. Method

4.1. Participants

We employed a quasi-experimental research design on the Fire Extinguishing AI Robot task of a programming course. The study participants ($N = 97$) were 42 female (43.3%) and 55 male (56.7%) junior high school students in southern China. The students' average age was 15.25 years, ranging between 13 and 16 years. These students were selected because they had all previously participated in a mindtool teaching project. We randomly divided the 97 students into experimental and control groups according to the experimental design, with 48 students assigned to the experimental group adopting RVMS, and 49 assigned to the control group adopting the VMS. All the students were taught by the same teacher who had taught information technology courses for nearly 10 years with enriched experience teaching programming.

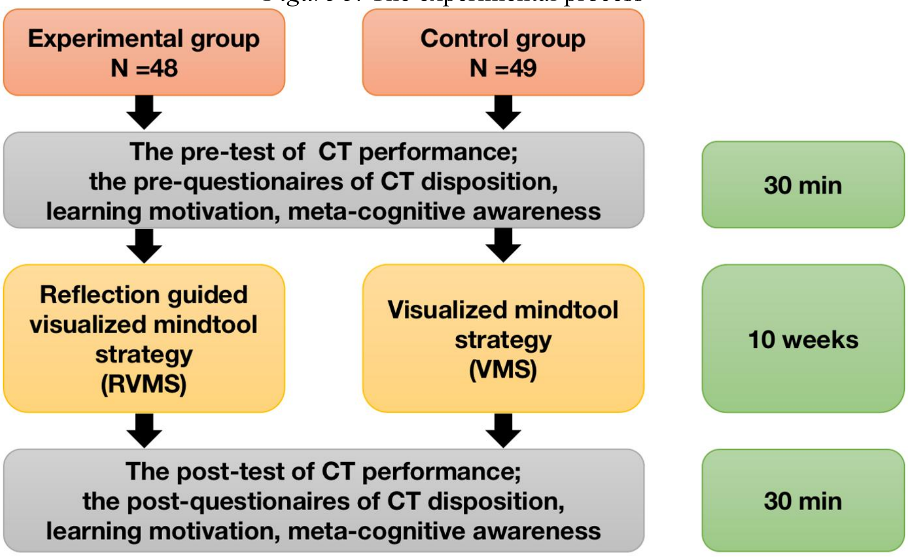
4.2. Experiment process

Figure 5 shows the experimental procedure of the study, which was conducted for 10 weeks. Before the learning activities, all students were trained to familiarize themselves with the basic structure of the mindtool (e.g., concept mapping tools) needed in the later learning. Students then completed the 30-minute pre-tests and pre-questionnaires, which aimed to exclude the effects of students' achievement and perception.

The learning materials were the same during the learning activities, including Arduino open-source hardware, the handbook of fire extinguishing AI robots, programming tools, and the system. Both groups of students used the Arduino device to accomplish the same task of fire extinguishing AI robots, and applied the programming knowledge they learned with the help of concept maps.

The difference between the two groups was that the students in the experimental group learned with the reflection-guided visualized mindtool strategy, while those in the control group adopted the visualized mindtool strategy. For instance, to identify cognitive gaps, students in the experimental group were asked to complete a concept map with the aid of several reflective activities. Through the reflective prompt activities, the concept maps were optimized several times to deepen understanding of programming knowledge to solve complex problems. Unlike the experimental group, control group students understood the CT concepts and decomposed problems without the assistance of reflection. That is, the teacher would directly tell them the correct answer when they encountered problems or had any doubts. Afterwards, the post-tests and post-questionnaires were administered for 30 minutes to examine whether their CT performance and disposition, learning motivation, and meta-cognitive awareness had improved.

Figure 5. The experimental process



4.3. Instrument

To collect data regarding CT cultivation, the instruments used for this study are as follows:

CT performance consisted of a pre- and post-test, to examine the effects of the proposed strategy. The tests consisted of two multiple-choice items, two open-ended questions, and one programming question (100 points). An example multiple-choice item is, "What are the intelligent functions of AI that help us complete our work?" An example open-ended question is, "Suppose you are going to design a smart fire alarm for your school. Please write down your ideas and the design proposals." The programming question is, "Please make an intelligent fire extinguishing robot and draw a flowchart of the program."

The pre-and-post questionnaires based on Tsai et al. (2021) were used to determine the level of CT disposition. This questionnaire could measure participants' perspectives on how they tried to think about and use skills related to CT. The original version of the CT disposition questionnaire was in English. Some words were modified to ensure that all items were clearly expressed in Chinese according to the students' opinions. Then, the adopted questionnaire was given to each participant of the two groups before and after teaching interventions. The questionnaire consisted of 25 items with a 5-point Likert scale for five sub-dimensions: abstraction, algorithm, evaluation, decomposition, and generalization, scored from strongly disagree to strongly agree. The Cronbach's α of the pre-test and post-test were 0.87 and 0.89, respectively. Indicative items for each sub-dimension are as following:

- Abstraction: I think I will try to think about how the program problems and the results are presented.
- Algorithm: I think I will try to develop detailed steps to solve the programming problem.
- Evaluation: I think I will try to find the right solution to the program problem.
- Decomposition: I think I will try to think about the possibility of a programming problem being decomposed.
- Generalization: I think I will try to determine if there are similarities between different programs.

The meta-cognitive awareness questionnaire was modified from the measuring tool proposed by Lin et al. (2022b), consisting of five items with a 5-point Likert-type rating scheme ranging from 1 (strongly disagree) to 5 (strongly agree). The Cronbach's alpha value of the questionnaire was 0.92. An example item is: I can discover the relationship between the critical issues in the Fire Extinguishing AI Robot task of a programming course that cause the program to fail.

Adapted from the instrument developed by Tapingkae et al. (2020), the learning motivation questionnaire consisted of eight items (e.g., When participating in online inquiry-based CT development in the Fire Extinguishing AI Robot task of a programming course, I always find that the learning is very interesting; Cronbach's alpha = 0.88). Students rated these items on a 5-point Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree).

4.4. Data collection and data analysis for the behavior analysis

4.4.1. The use of video recording and stimulated recall reflections

To observe students' learning behaviors during the online inquiry-based CT development activities, we recorded each student's behaviors throughout the process via video. We collected the video data in normal AI class situations. The teacher videotaped a 1-hour lesson. Students were asked to review their programming peers' initial projects and a 1-hour lesson during which they revised their projects based on reviewers' feedback. In addition, we used the video to identify students' thought processes (i.e., behaviors) during both the reviewing and revising phases (1 hour). With the replay of the video recordings, these students were allowed to see and explain their actions during the CT development within online inquiry tasks. Therefore, the categories as well as patterns of their behaviors were found for both groups.

4.4.2. Coding process

The video data were captured using FASTCAPTURE software installed on the students' computers. The students' behaviors were then video recorded through the CT development activity. Then, we replayed the video files for coding the reviewing and revising phases (180 minutes). LSA was conducted to explore the students' learning behavior patterns in the CT development activities by using the GSEQ software. To analyze the CT development behaviors, we developed an initial coding scheme by synthesizing the conceptual framework of CT from Tsai et al. (2021) and Ehsan et al. (2021). Then we carried out a pilot test by reviewing the recorded video to capture the primary behavior in the CT-oriented programming learning process and constructed the final coding table, as shown in Table 1.

Table 1. The coding table of learning behaviors

Code	Phase/content	Description
A	Abstracting the gap	Thinking about a problem from a whole point of view to find the gaps rather than looking at the details
B	Examining the rubrics	Reading the online inquiry CT development rubrics.
C	Searching the Internet	Browsing the Web and reading the information on the Internet.
D	Decomposing	Breaking down a difficult CT problem into more manageable sub-problems
E	Algorithming	Writing codes with algorithms for a problem
F	Generalizing	Recognizing the specific type of practicing solutions and applying them to similar problems.
G	Evaluating	Finding the status of the best solution and resources
H	Re-algorithming	Rewriting codes with better algorithms for a problem

Two researchers coded the video data based on the coding scheme. Both researchers have undergone comprehensive training on the operational definitions of the behavior codes. Considering that each student could averagely spend at least 20 seconds on the same behavior, the researchers underwent a real-time 20 second-to-20 second data coding. The lead coder was the first researcher (the principal investigator of this study) was the lead coder. During the coding process, the second researcher met regularly with the first researcher in this study to discuss coding disagreements and assess the inter-rater kappa criterion of 0.86.

5. Result

5.1. CT performance

After conducting the learning activity, we performed an analysis of covariance (ANCOVA) on the CT performance results to test the relationships between the two groups' post-test results. Before the ANCOVA, the Levene's test of determining homogeneity of variance was not violated (F = 0.39, p = .09 > .05), and the homogeneity of regression slopes was confirmed (F = 1.35, p = .15 > .05). Therefore, ANCOVA was conducted. Results in Table 2 showed that the CT performance of the experimental group students was significantly better than that of the control group, thus responding to RQ1.1. Moreover, the η^2 value was 0.09, indicating that the finding had a medium effect size (Cohen, 1988).

Table 2. The ANCOVA result of CT performance

Group	N	Mean	SD	Adjusted Mean	Std. error	F	η^2
Experimental group	48	90.27	9.10	91.37	1.97	4.87*	0.09
Control group	49	85.76	12.95	84.38	1.91		

Note. $^*p$ < .05.

5.2. CT disposition

Before conducting ANCOVA to analyze students' CT disposition, the Levene's test of homogeneity of variances was applied to examine whether variances across samples were equal. The result of this test was not significant (p = .12 > .05), suggesting that the difference between the variances for all groups was not significant. Also, the result (F = 2.36, p > .05) indicated that the homogeneity of regression coefficients was not violated. Therefore, ANCOVA was performed.

According to the results (F = 3.51, p < .05), the difference between the two groups was statistically significant (see Table 3). The CT disposition score of the experimental group was higher than that of the control group, which responded to RQ1.2. Furthermore, the η2 of the proposed method is 0.09, indicating a medium effect size.

Table 3. The ANCOVA result of CT disposition

Group	N	Mean	SD	Adjusted Mean	Adjusted SD	F	η^2
Experimental group	48	4.35	0.78	4.35	1.82	3.51*	0.09
Control group	49	3.82	0.89	3.82	1.82		

Note. $^*p$ < .05.

5.3. Meta-cognitive awareness

Before the ANCOVA, the homogeneity of variance assumptions and homogeneity of regression coefficients were tested to examine the effect of the proposed strategy on students' meta-cognitive awareness, controlling for the pre-questionnaire scores. Levene's test for equality of variances was not significant (F = 5.87, p > .05). Hence, the homogeneity of variance assumption was not violated. Also, the result (F = 3.79, p > .05) indicated that the assumption of homogeneity of regression coefficients was not violated. Therefore, the ANCOVA was conducted.

The adjusted means and standard error were 4.21 and 0.89 for the experimental group, and 3.68 and 0.94 for the control group (see Table 4). The ANCOVA results indicated that the meta-cognitive awareness scores of the two groups showed a significant difference (F = 8.71, p < .05). As a response to RQ2.1, the meta-cognitive awareness score of the experimental group was statistically higher than that of the control group. Furthermore, the η^2 of the proposed approach was 0.68, indicating a large effect size.

Table 4. The ANCOVA result of meta-cognitive awareness

Group	*N*	Mean	*SD*	Adjusted Mean	Adjusted *SD*	*F*	η^2
Experimental group	48	4.21	0.67	4.21	0.89	8.71^*	0.68
Control group	49	3.68	0.89	3.68	0.94		

Note. $^*p < .05$.

5.4. Learning motivation

In terms of the learning motivation scores, Levene's test for equality of variances was $F = 3.56$ ($p = .12 > .05$), indicating no significant difference between the two groups' learning motivation. The homogeneity of regression coefficients was examined to understand whether there was an interaction between the covariate and independent variables ($F = 6.21$, $p > .05$). It was observed that there was no interaction between the pre- and post-tests, indicating that the regression coefficient within the group did not reach a significant level. As the homogeneity assumption was satisfied, the ANCOVA could be performed.

The ANCOVA result is shown in Table 5. The results of the questions on learning motivation showed that the experimental group's learning motivation test score was significantly higher than that of the control group ($F = 8.04$, $p < .05$) with a large effect size ($\eta^2 = 0.76$). The adjusted mean scores of the experimental and control group were $M = 4.32$ and $M = 3.90$, respectively. Based on the results, it is concluded that the learners who used RVMS had better learning motivation compared to those using VMS, responding to RQ2.2.

Table 5. The ANCOVA result of learning motivation

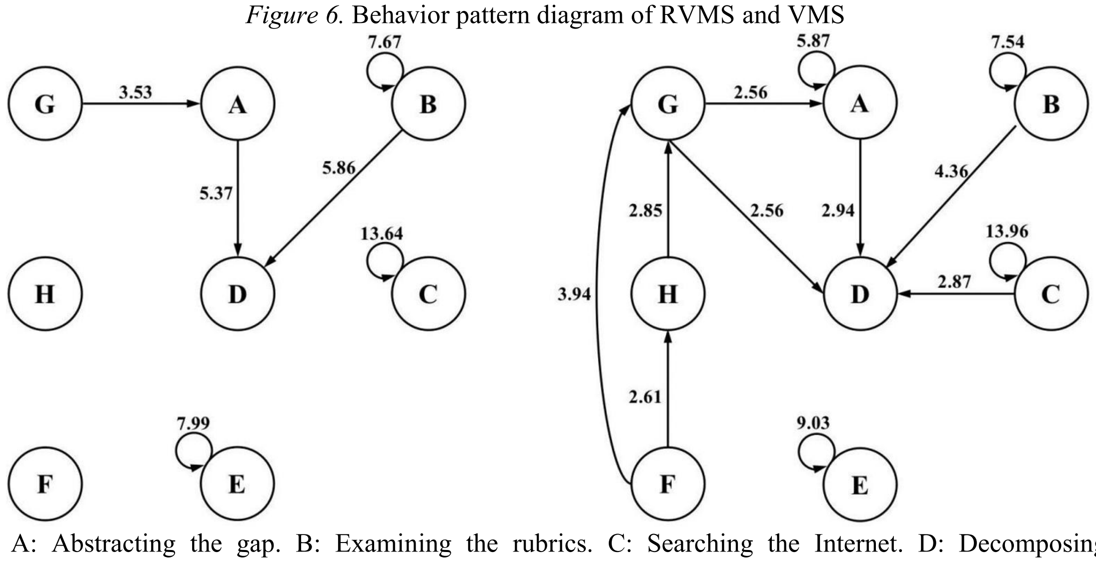
Groups	*N*	Mean	*SD*	Adjusted Mean	Adjusted *SD*	*F*	η^2
Experimental group	48	4.32	0.81	4.32	0.98	8.04^*	0.76
Control group	49	3.90	0.88	3.90	0.64		

Note. $^*p < .05$.

5.5. Comparisons of behaviors of the two groups of students

In response to RQ3, this study examined the students' behaviors patterns of the experimental and control groups with LSA in their CT learning. In Figure 6, it can be seen that the two groups were similar in six sets of significant sequences, including B→B, C→C, E→E, A→D, B→D, and G→A. In other words, all students, no matter which group they belonged to, demonstrated continuity in abstracting the gap (B→B), searching for the information on the Internet (C→C), and algorithming (E→E). Besides, they all showed three unidirectional sequences from Abstracting the gap to Decomposing (A→D), Examining the rubrics to Decomposing (B→D), and Evaluating to Abstracting the gap (G→A).

Figure 6. Behavior pattern diagram of RVMS and VMS



Note. A: Abstracting the gap. B: Examining the rubrics. C: Searching the Internet. D: Decomposing. E: Algorithming. F: Generalizing. G: Evaluating. H: Re-algorithming.

However, in the experimental group, more sets of significant sequences were involved in students' behavior patterns, revealing that the experimental group demonstrated special sequences, including C→D, F→G, F→H, H→G, G→D, and A→A. During the CT development activities, experimental group students attempted to decompose after searching the Internet (C→D), while those in the control group did not. Besides, the behavior of abstracting the gap (A) of the experimental group students was repeated continuously, which meant that they tended to think deeply and fully abstract their gaps. Furthermore, after generalizing, students turned to re-algorithm then went to evaluation (F→H, H→G) or turned to evaluate directly (F→G). They then tried to decompose (G→D); that is, the special sets of sequences present in the experimental group are the key to the difference in performance of the two groups.

6. Discussion

This study proposed a reflection-guided visualized mindtool strategy to improve students' CT performance, CT disposition, learning motivation, and meta-cognitive awareness. The study further explored the effect of reflection integrated into mindtools on students' behaviors during CT development.

Concerning CT performance and CT disposition, the experimental group had significantly better learning achievements. The improvement in CT performance and disposition implied that the proposed strategy could promote the development of students' CT by effectively incorporating reflection into the use of mindtools during the process of programming for CT. With mindtools, students could regulate CT concepts and make connections between each concept easily and visually. Additionally, the findings suggested the likely effects of reflection on continually directing students to review the process of CT problem-solving and improving their solutions to successfully solve the complex problem of CT. This finding, however, differed from the conclusion of past research (Eshuis et al., 2021), which reported that the combination of reflective prompts and concept maps could not promote students' learning achievement. It might be that students could not actively reflect on the provided information, and their reflection was too superficial only with reflective prompts. The present study adopted various reflective approaches rather than just reflective prompts, including reflective evaluation rubrics and stimulated recall reflections. With the guidance of reflective prompts, students could think more deeply and make the appropriate connection of concepts during the CT online inquiry activity. With the evaluation rubrics, students could measure their peers' performance and know whether their peers' and their own projects met the expected criteria by scoring the CT tasks. In the consecutive stimulated recall reflections, students were allowed to interpret or reflect on some specific behavior patterns during the CT development within online inquiry tasks to improve their performance in the next stage.

In terms of behavioral transition diagrams, both the experimental and control group learning with the mindtool strategy generated the behavior of decomposing after abstracting the gap (A→D) or examining the rubrics (B→D). This means that CT development with the mindtool strategy allowed students to plan and examine a precise and sophisticated structure of concepts and further decompose all possible main or specific scenes node-by-node to construct frames to complete the online inquiry-based CT tasks. These findings are consistent with the results of Zhao et al. (2022), which also showed the positive effects of mindtools on the cultivation of CT. However, students using RVMS generated more specific behavior sequences than those learning with VMS. In terms of the repeated sequence (A→A), it was found that, with the guidance of reflective evaluation rubrics, students in the experimental group could clearly understand the goal of the task and then accurately discover the limitations of their ideas. Moreover, students were more likely to generalize the knowledge they had learned, re-design their algorithms for the tasks, evaluate the feasibility of their proposed schemes, and then better decompose the task (F→H→G→D, F→G→D). These behaviors implied that students could transfer the knowledge to find better solutions after reflecting on their own CT problem-solving process.

The findings based on the different behavior patterns of the two groups in this study indicated that the proposed strategy, which incorporates reflection into the whole process of CT problem-solving by carrying out various reflective activities (i.e., reflective prompts, reflective evaluation rubrics, and stimulated recall reflections), could engage students in deeper reflection and allow them to be more deeply involved in the CT problem-solving process. This may be why students learning with RVMS showed better CT learning outcomes, meta-cognitive awareness, and motivation than those learning with VMS. The study further indicated the value of reflective behaviors in CT cognitive and meta-cognitive processes, including problem and action gap identification, information searching and re-algorithming, and generalizing new ideas and alternative generation. The findings revealed that successful CT development usually depended on individuals' reflection to engage proactively in CT development challenges and persist in attempts to meet the challenges they encountered. Hence, the valuable determinants regarding the reflection-guided mindtool for conducting online inquiry-based CT development in a programming or AI course are advocated. For better CT outcomes, the cultivation of CT should pay more attention to engaging students in certain behaviors, including problem identification, information collection, idea generation, hypotheses making, generalizing, re-algorithming, and evaluation. This study provides practical recommendations including an effective reflection-guided visualized mindtool strategy for improving students' CT learning performance and behaviors from cognition to meta-cognition. This study extends the existing pedagogy in CT learning by highlighting the importance of visualized mindtools to promote students' conceptual knowledge, self-generated thinking, and complex problem-solving for cognitive development in CT learning, and the role of the reflection-guided strategy to actively engage students in the meta-cognitive process with the guidance of reflective prompts, reflective evaluations, and stimulated recall reflections.

7. Conclusion

The present study has provided a thorough look at and understanding of a reflection-guided visualized mindtool strategy aiming to advance students' CT outcomes (i.e., CT performance, CT disposition, learning motivation, and meta-cognitive awareness). Meanwhile, the proposed strategy has great potential to activate students' in-depth reflection (i.e., monitoring and regulating their cognitive activities and practices) during online CT development by incorporating reflection with mindtools into the whole process of CT problem-solving with various reflective methods. The results can guide teachers in optimizing their CT pedagogy. For example, teachers can use visualized mindtools, such as concept maps, to guide students' thinking and task decomposition. Besides, teachers can provide proper approaches or tools to arouse students' reflection before, during, and after CT problem-solving to identify gaps in their thinking and behaviors and improve their CT-related problem-solving abilities. In addition, it is also suggested that teachers pay more attention to cultivating students' certain behaviors before implementing CT development activities based on the behavior analysis results. It was observed that the experimental group generated more specific behaviors, including abstracting the gaps, decomposing tasks, generalizing, re-algorithming, and evaluating, which are critical to learning CT (Grover et al., 2016). Thus, instead of solely teaching syntax and how to code during programming courses, teachers should focus more on guiding students to find ways to solve CT problems. The findings of this study have the potential to promote these valuable behaviors.

Although the experimental results showed that RVMS is effective for CT cultivation, some limitations to this study should be noted. First, this study just recorded students' behaviors during the online inquiry-based CT development activity to observe behavioral changes, which may have ignored some potential behavior critical to CT in face-to-face learning or after class. Second, the present study only identified the effectiveness of the proposed strategy in terms of CT concepts and CT practices; other aspects of CT were not considered. Further studies are needed to investigate the effect of the proposed strategy on other aspects of CT, like CT skills. Lastly, the coding process of behavior analysis was manual, which takes time and may inevitably result in human error. The development of an automated recording and coding system may be considered.

Acknowledgement

This work was supported by the National Natural Science Foundation of China [grant number 62007010]; the Key Project of National Natural Science Foundation of China under grant number 62237001; the National Key R&D Program of China [grant number 2022YFC3303605]; the Science and Technology Projects in Guangzhou [grant number 202102021217]; the Special Funds of Climbing Program regarding the Cultivation of Guangdong College Students' Scientific and Technological Innovation [grant number pdjh2023a0139]; 2023 annual Guangzhou Youth and the Communist Youth League project of "Guangzhou Youths' Participation in Rural Revitalization Research: The I-SEED 'Internet Plus' Cloud Public Welfare to Empower Rural Education Revitalization" [grant number 2023TSW13], Teaching Quality Project of South China Normal University: Professional Development of Artificial Intelligence Teachers under "New Normal" Background [grant number 192]; College Student Innovation and Entrepreneurship Training Program [grant number 202328010]; and South China Normal University "Challenge Cup" Golden Seed Cultivation Project [grant number 22JXKA09].

References

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research*, *87*(4), 834–860.

Cavilla, D. (2017). The Effects of student reflection on academic performance and motivation. *SAGE Open, 7*(3). https://doi.org/10.1177/2158244017733790

Chang, C.-C., & Hwang, G.-J. (2022). A Structured reflection-based graphic organizer approach for professional training: A Technology-supported AQSR approach. *Computers & Education*, *183*, 104502. https://doi.org/10.1016/j.compedu.2022.104502

Chang, C.-C., Hwang, G.-J., & Tu, Y.-F. (2022). Concept mapping in technology-supported K-12 education: A Systematic review of selected SSCI publications from 2001 to 2020. *Journal of Educational Computing Research*, *60*(7), 1637-1662.

Chang, K. E., Sung, Y. T., & Chen, S. F. (2001). Learning through computer-based concept mapping with scaffolding aid: Learning through computer-based concept mapping. *Journal of Computer Assisted Learning*, *17*(1), 21–33.

Chen, C.-H., Liu, T.-K., & Huang, K. (2021). Scaffolding vocational high school students' computational thinking with cognitive and meta-cognitive prompts in learning about programmable logic controllers. *Journal of Research on Technology in Education*, 1–18. https://doi.org/10.1080/15391523.2021.1983894

Chuang, C.-W., Hwang, G.-J., & Tsai, W.-J. (2018). A Peer tutoring-based concept mapping approach to improving students' learning achievements and attitudes for a social studies course. *International Journal of Online Pedagogy and Course Design*, *8*(1), 1–12. https://doi.org/10.4018/IJOPCD.2018010101

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Lawrence Erlbaum Associates.

Colbert, C. Y., Graham, L., West, C., White, B. A., Arroliga, A. C., Myers, J. D., Ogden, P. E., Archer, J., Mohammad, Z. T. A., & Clark, J. (2015). Teaching meta-cognitive skills: Helping your physician trainees in the quest to 'Know What They Don't Know'. *The American Journal of Medicine*, *128*(3), 318–324.

Denner, J., Campe, S., & Werner, L. (2019). Does computer game design and programming benefit children? A Meta-synthesis of research. *ACM Transactions on Computing Education*, *19*(3), 1–35. https://doi.org/10.1145/3277565

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, *60*(6), 33–39.

Dewey, J. (1933). *How we think: A Restatement of the relation of reflective thinking to the educative process*. D.C. Heath and Company.

Ehsan, H., Rehmat, A. P., & Cardella, M. E. (2021). Computational thinking embedded in engineering design: Capturing computational thinking of children in an informal engineering design activity. *International Journal of Technology and Design Education*, *31*(3), 441–464.

Eshuis, E. H., Vrugte, J., Anjewierden, A., & Jong, T. (2021). Expert examples and prompted reflection in learning with concept maps. *Journal of Computer Assisted Learning*, *38*(2), 350–365.

Ezeamuzie, N. O. (2022). Abstractive-based programming approach to computational thinking: Discover, extract, create, and assemble. *Journal of Educational Computing Research, https://doi.org/10.1177/07356331221134423*

Ezeamuzie, N. O., & Leung, J. S. C. (2021). Computational thinking through an empirical lens: A Systematic review of literature. *Journal of Educational Computing Research*, *60*(2), 481-511.

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

Ghanizadeh, A. (2017). The Interplay between reflective thinking, critical thinking, self-monitoring, and academic achievement in higher education. *Higher Education*, *74*, 101–114. https://doi.org/10.1007/s10734-016-0031-y

Grover, S., Pea, R., & Cooper, S. (2016). Factors influencing computer science learning in middle school. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 552-557). Association for Computing Machinery. https://doi.org/10.1145/2839509.2844564

He, Z., Wu, X., Wang, Q., & Huang, C. (2021). Developing eighth-grade students' computational thinking with critical reflection. *Sustainability*, *13*(20), 11192. https://doi.org/10.3390/su132011192

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

Jonassen, D. H., & Carr, C. S. (2020). Mindtools: Affording multiple knowledge representations for learning. In *Computers as cognitive tools, volume two: No more walls* (pp. 165-196). Routledge. https://doi.org/10.1201/9781315045337-8

Jonassen, D. H., Carr, C., & Yueh, H.-P. (1998). Computers as mindtools for engaging learners in critical thinking. *TechTrends*, *43*(2), 24–32.

Jong, M. S. Y., Geng, J., Chai, C. S., & Lin, P. Y. (2020). Development and predictive validity of the computational thinking disposition questionnaire. *Sustainability, 12*(11), 4459. https://doi.org/10.3390/su12114459

Krieglstein, F., Schneider, S., Beege, M., & Rey, G. D. (2022). How the design and complexity of concept maps influence cognitive learning processes. *Educational Technology Research and Development*, *70*(1), 99–118.

Lei, H., Chiu, M. M., Li, F., Wang, X., & Geng, Y.-J. (2020). Computational thinking and academic achievement: A Meta-analysis among students. *Children and Youth Services Review*, *118*, 105439. https://doi.org/10.1016/j.childyouth.2020.105439

Lin, Y.-T., Yeh, M. K.-C., & Tan, S.-R. (2022a). Teaching programming by revealing thinking process: Watching experts' live coding videos with reflection annotations. *IEEE Transactions on Education*, *65*(4), 617–627.

Lin, X.-F., Hwang, G.-J., Wang, J., Zhou, Y., Li, W., Liu, J., & Liang, Z.-M. (2022b). Effects of a contextualised reflective mechanism-based augmented reality learning model on students' scientific inquiry learning performances, behavioural patterns, and higher order thinking. *Interactive Learning Environments*, 1-21. https://doi.org/10.1080/10494820.2022.2057546

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

McAleese, R. (1998). The Knowledge arena as an extension to the concept map: Reflection in action. *Interactive Learning Environments*, *6*(3), 251–272.

Medina, M. S., Castleberry, A. N., & Persky, A. M. (2017). Strategies for improving learner metacognition in health professional education. *American Journal of Pharmaceutical Education*, *81*(4), 78. https://doi.org/10.5688/ajpe81478

Merino-Armero, J. M., González-Calero, J. A., & Cózar-Gutiérrez, R. (2022). Computational thinking in K-12 education. An Insight through meta-analysis. *Journal of Research on Technology in Education*, *54*(3), 410-437.

Mouza, C., Pan, Y.-C., Yang, H., & Pollock, L. (2020). A Multiyear investigation of student computational thinking concepts, practices, and perspectives in an after-school computing program. *Journal of Educational Computing Research*, *58*(5), 1029–1056.

Novak, J. D., & Gowin, D. B. (1984). *Learning how to learn*. Cambridge University Press.

Oliver, K. (2008). A Comparison of Web-based concept mapping tasks for alternative assessment in distance teacher education. *Journal of Computing in Teacher Education, 24*(3), 95-103.

Omer, U., Farooq, M. S., & Abid, A. (2020). Cognitive learning analytics using assessment data and concept map: A Framework-based approach for sustainability of programming courses. *Sustainability*, *12*(17), 6990. https://doi.org/10.3390/su12176990

Radović, S., Firssova, O., Hummel, H. G., & Vermeulen, M. (2021). Improving academic performance: Strengthening the relation between theory and practice through prompted reflection. *Active Learning in Higher Education*. https://doi.org/10.1177/14697874211014411

Rodgers, C. (2002). Defining reflection: Another look at John Dewey and reflective thinking. *Teachers College Record*, *104*(4), 842–866.

Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2019). The Cognitive benefits of learning computer programming: A Meta-analysis of transfer effects. *Journal of Educational Psychology*, *111*(5), 764–792.

Schön, D. A. (1987). *Educating the reflective practitioner: Toward a new design for teaching and learning in the professions*. Wiley.

Schön, D. A. (2017). *The Reflective practitioner*. Routledge.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Sun, D., Ouyang, F., Li, Y., & Chen, H. (2021). Three contrasting pairs' collaborative programming processes in China's secondary education. *Journal of Educational Computing Research, 59*(4), 740–762.

Tapingkae, P., Panjaburee, P., Hwang, G.-J., & Srisawasdi, N. (2020). Effects of a formative assessment-based contextual gaming approach on students' digital citizenship behaviours, learning motivations, and perceptions. *Computers & Education*, *159*, 103998. https://doi.org/10.1016/j.compedu.2020.103998

Tsai, M.-J., Liang, J.-C., & Hsu, C.-Y. (2021). The Computational thinking scale for computer literacy education. *Journal of Educational Computing Research*, *59*(4), 579–602.

Wang, M., Cheng, B., Chen, J., Mercer, N., & Kirschner, P. A. (2017). The Use of web-based collaborative concept mapping to support group learning and interaction in an online environment. *The Internet and Higher Education*, *34*, 28–40. https://doi.org/10.1016/j.iheduc.2017.04.003

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.

Wong, R. M., Sundararajan, N., Adesope, O. O., & Nishida, K. R. A. (2021). Static and interactive concept maps for chemistry learning. *Educational Psychology*, *41*(2), 206–223.

Wu, B., Hu, Y., Ruis, A. R., & Wang, M. (2019). Analysing computational thinking in collaborative programming: A Quantitative ethnography approach. *Journal of Computer Assisted Learning*, *35*(3), 421–434.

Yue, M., Zhang, M., Zhang, C., & Jin, C. (2017). The Effectiveness of concept mapping on development of critical thinking in nursing education: A Systematic review and meta-analysis. *Nurse Education Today*, *52*, 87–94.

Zhang, J.-H., Meng, B., Zou, L.-C., Zhu, Y., & Hwang, G.-J. (2021). Progressive flowchart development scaffolding to improve university students' computational thinking and programming self-efficacy. *Interactive Learning Environments*, 1–18. https://doi.org/10.1080/10494820.2021.1943687

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