Using Computational Thinking as a Metacognitive Tool in the Context of Plugged Vs. Unplugged Computational Activities

Ceren Ocak, Aman Yadav and Victoria Macann

September 12, 2023

International Society of the Learning Sciences

Using Computational Thinking as a Metacognitive Tool in The Context of Plugged vs. Unplugged Computational Activities

Ceren Ocak, Georgia Southern University, cocak@georgiasouthern.edu
Aman Yadav, Michigan State University, ayadav@msu.edu
Victoria Macann, Michigan State University, macannvi@msu.edu

Abstract: This paper explored how K-5 teachers incorporated computational thinking (CT) to support young children's development of metacognitive knowledge and abilities. Two 4th-grade mathematics teachers' lesson videos were analyzed to understand how K-5 teachers used CT as a metacognitive tool. One teacher incorporated CT ideas and practices into her teaching without using any computational device (i.e., unplugged), whereas the other used Dash & Dot robots to engage his students in CT (i.e., plugged). Within those activities, teachers used CT to engage students in a variety of metacognitive strategies, such as attending to critical features of a problem, creating a mental model of a problem, and monitoring solution paths. Our findings provided insight into how K-5 teachers can leverage CT to enhance their students' metacognitive knowledge and abilities.

Introduction

In recent years, computational thinking and computer science (CS) education researchers have shown a growing interest in metacognition (Prather et al., 2022). Metacognition is one's awareness of own cognitive processes, and it is critical to learning per se for enabling students to control and monitor their cognitive processes (Flavell, 1976). Similarly, CT—a systematic approach to solving and formulating problems—involves processes where one must constantly reflect on one's own thinking and actions, such as finding and fixing own errors (e.g., debugging) (Yadav et al., 2016). Given these overlaps, scholars have called for investigating the connection between CT and metacognition for their students' academic benefit (Yadav et al., 2022).

Some scholars have proposed that students' metacognitive awareness can be developed by engaging them in a systematic problem-solving process during programming (Loksa et al., 2022). Others have claimed that CT through unplugged activities (i.e., without any computational device) could also help develop students' metacognitive abilities. For example, Yadav et al. (2022) suggested that CT potentially enables and overlaps with several metacognitive processes (e.g., identification of steps to solve a problem, execution of the steps serially or in parallel, and solution-monitoring); therefore, CT could be one way to teach metacognition in classrooms explicitly. In this context, our purpose was to explore how elementary teachers can leverage CT to support young children's development of metacognitive strategies. To explore that premise, we asked:

1. In what ways do elementary teachers use CT to support students' metacognition?
    - How does teachers' use of CT as an approach to engaging their students in metacognitive strategies differ between a plugged and an unplugged CT approach?

To address this desideratum, we video-recorded two 4th-grade teachers' CT-integrated mathematics lessons. One teacher used Dash and Dot robots to teach the concepts of area and perimeter, while the other teacher implemented CT ideas into the arrays lesson without using any computational device. Treating each case separately, we elicited the metacognitive strategies that teachers taught during computational problem-solving. Our purpose was to expand the list of metacognitive strategies that teachers use to bring metacognitive experiences to elementary mathematics classrooms through unplugged and plugged CT activities. Our question is relevant and timely and contributes meaningfully to ongoing and current questions about using CT as a metacognitive tool to enhance learning and instruction.

Computational thinking and metacognition

Computational thinking

In 2006, Jeanette Wing defined CT as "solving problems, designing systems, and understanding human behavior by drawing on the concepts fundamental to computer science" (p.33) and proposed that CT is not only for computer scientists but for every child. Since then, the efforts to bring CT into elementary education have only grown stronger. Built on the momentum generated by Wing's (2006) article, several educational initiatives and reforms have focused on developing children's CT knowledge and skills by integrating computing into K-12 classrooms (e.g., International Society for Technology in Education [ISTE], American Computer Science

Teachers Association [CSTA], and Next Generation Science Standards [NGSS]). Those initiatives are critical in that they reflect the growing presence of computing in K-12 education. At the same time, education scholars have also called for meaningful CT integration practices as CT has made its way through standards and K-12 curricula. To date, CT has been incorporated across disciples for a wide range of purposes, including but not limited to its use to support problem-solving skills, modeling, analyzing and representing data, and, more recently, helping students understand how computing impacts society. The most common perspective around CT integration conceives it as a problem-solving approach (Kafai et al., 2020) that includes several practices, such as decomposing problems into manageable parts (i.e., problem decomposition), using a set of steps to solve a problem (i.e., algorithmic thinking), seeing whether the solution could be transferred to similar problems (i.e., abstraction). Yadav et al. (2022) have also argued that these sub-practices overlap with several metacognitive strategies that have been shown to support students' academic outcomes.

Metacognition

Metacognition has been extensively studied for over fifty years. In 1976, Flavell pioneered the notion of metacognition and framed it as a phenomenon associated with one's awareness of own cognitive processes, such as memory and problem-solving. Flavell (1976) proposed that metacognition had two major components: metacognitive knowledge (MK) and metacognitive experiences (ME). MK refers to knowledge of one's own strengths and weaknesses when dealing with a task, while experiences emerge when MK is called on during problem-solving (Efklides, 2002). The scope of metacognition, however, is over and beyond what one declares about their own cognitive states; it extends to how one takes action in controlling one's own cognitive processes, which overall entail "thinking about the learning process, planning for learning, monitoring of comprehension or production while it is taking place, and self-evaluation of learning" (O'Malley et al., 1985, p. 560).

Computational thinking and metacognition

Similar to metacognition, CT involves processes where one must constantly reflect on own thinking and actions. A growing literature suggests that CT naturally enables and offers mechanisms to engage learners in self-reflective practices through debugging, iterations, and abstraction (Allsop, 2019). Those processes naturally engage learners in retrospective and prospective decision-making to help reach an equilibrium between their current actions and future goals. In this way, students '*debug [their] own thinking*' retrospectively while engaging in processes help them move closer to achieving their final goal (Kafai & Burke, 2016, p. 321). Considering these connections between CT and metacognition, young children, who are taught CT, have a better chance of mastering their own learning and cognitive processes, which could also benefit their long-term academic success. However, still little is known about the shape and degree of how CT overlaps with metacognition. In this paper, we aim to fill this gap by examining metacognitive strategies that in-service teachers employ during CT tasks in plugged and unplugged contexts—which can provide insight into how CT could be used as a pedagogical tool to teach metacognitive strategies explicitly.

Method

The study employed Bezemer's (2015) analytical framework for video-based multimodal analysis for social interaction to analyze 45-minute length videos collected from two classrooms. Multiple channels of communication (e.g., spoken dialogue, body-based gestures) are translated into a multimodal transcript through a set of steps—such as choosing a methodological framework, designing a transcript, and defining transcription conventions—where images and text are combined for fine-grained analysis. This approach helped us elicit metacognitive strategies (e.g., gestures) that are multimodal, going beyond teachers' discourse or dialogue only.

Context

The videos were collected as part of a research project, CT4EDU, that focused on supporting elementary teachers in integrating CT into their science and mathematics teaching. Elementary teachers incorporated CT practices in their mathematics lessons, and we video-recorded their lessons and the interactions with the students using Swivl—a robotic mount for a camera that helped record the actions of a moving teacher. We used these videos to form a preliminary understanding of how teachers use CT practices as a metacognitive tool within their instruction. We randomly selected two 4th-grade mathematics lessons from two teachers who volunteered to participate in our study. We chose mathematics as a subject because it naturally enables CT integration and the development of metacognitive skills for being at the heart of problem-solving. All data were kept confidential, and the participants were given pseudonyms, as displayed in Table 1.

Participants

Participants included two elementary teachers from two schools in the Midwestern United States, as displayed in Table 1. The teachers had participated in a professional learning experience that focused on preparing elementary teachers to integrate CT into their mathematics and science instruction. The focus of this study was CT-integrated mathematics lessons that the two teachers facilitated. The first teacher, Michael, used dash and dot robots to teach the concepts of area and perimeter, while the second teacher, Jill, taught simple arrays without using any digital computational tools.

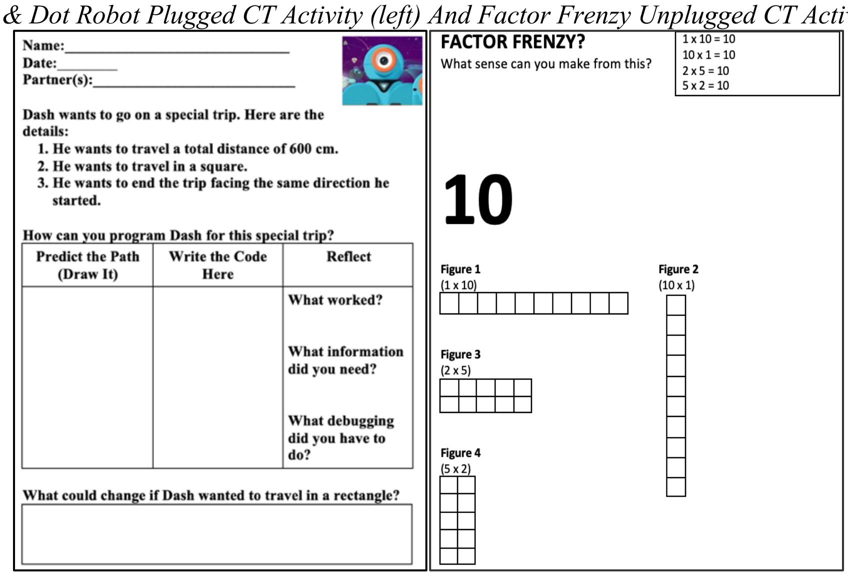
Table 1
Teacher Demographics and Recorded Lessons

Pseudonym	Gender	Grade	Class Size	Activity
Michael	Male	4	23	Dash & Dot Robot Plugged CT
Jill	Female	4	23	Factor Frenzy Unplugged CT

Research context

Dash and Dot plugged CT activity: In this activity, CT ideas and practices were used to teach basic mathematical concepts (e.g., perimeter and directions) when programming Dash robots. The major idea was to program the robot Dash to travel in a square. Therefore, our first teacher, Michael, instructed students to program the robot Dash for a special trip following the preset rules, such as traveling a total distance of 600 centimeters and ending the trip facing the same direction he started (see Figure 1, left column).

Figure 1
Dash & Dot Robot Plugged CT Activity (left) And Factor Frenzy Unplugged CT Activity (right)



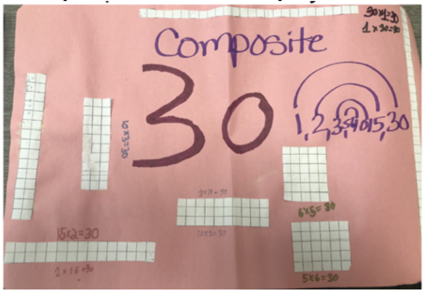
Factor frenzy unplugged CT activity: The second teacher, Jill, used factor frenzy to teach factors in mathematics. At the same time, the activity largely drew on the use of debugging, abstraction, and decomposition to teach arrays, factors, and products. The students first used base ten blocks to write different multiplication equations for 10 in the form of arrays (see Figure 1, right column).

This activity was also scaffolded by giving a lower or higher number other than 10; an example for the number 54 might be as follows:

1. Students were reminded to begin with one row for the number 54 (i.e., 1 x 54).
2. Students then found different ways to create the factors of 54 (e.g., 2 x 27, 3 x 18, 6 x 9).
3. Students were asked to find the array's vertical and horizontal orientation (i.e., all the factor pairs).
4. Students then recorded their arrays on a construction paper, using the grid paper and correctly labeling their arrays (see Figure 2 for a sample construction paper)

Figure 2

A Sample Construction Paper



Data analysis

The video data was first transcribed verbatim and analyzed by three raters in segments according to where we identified both CT practices and metacognitive strategies. First, we identified CT practices such as *decomposing* large tasks into smaller sub-goals, using *patterns*, and thinking *algorithmically* to develop efficient solutions. Then, we explored the emergent metacognitive strategies associated with each CT practice. This approach allowed us to figure out overlapping elements of CT and metacognition. We then collated the information into overarching themes. The themes were identified through consensus building in recursive meetings with the raters (Baxter & Jack, 2008).

Results

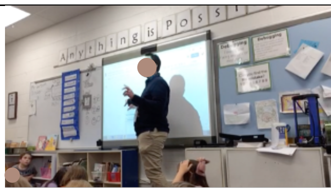
Our video-based multimodal analysis in the context of plugged and unplugged computational activities revealed a set of strategies on how CT could help enhance young learners' metacognition in mathematics classrooms. The metacognitive processes and associated strategies used by the teachers are detailed in the following, and their connection to CT ideas, sub-skills, and practices are explained in a narrative form within the cases. Our focus was on the ways in which teachers' used CT as a metacognitive strategy rather than how the students reacted to the teachers' strategies.

Case 1

The analysis of the data from plugged computational activity (i.e., Dash & Dot) revealed three major themes of how CT could be used as metacognitive strategies: (1) developing an understanding of the critical features of a problem, (2) feeling for the constraints of the problem, and (3) monitoring solution paths.

(1) Developing an understanding of the critical features of a problem: The first metacognitive strategy that the students engaged in during CT was the critical analysis of a task. Our first teacher, Michael, first began by prompting the students to develop an understanding of the critical features of a problem (see Table 2). For example, after introducing the Dash & Dot activity, he asked his students: "What are some things that you need to think about with these preset rules, and what is important to know about Dash's trip". These questions helped students assess the problem's existing condition and better understand its critical features to attend to in the later stages of problem-solving. This is also a critical process of computational problem-solving, where students engage in abstraction as a CT practice (i.e., focusing on the most relevant and essential details of the problem).

Table 2
A Moment of Abstraction Through Developing An Understanding of The Critical Features of a Problem

Dialogue	Interaction	Transduction
Michael: So, with the people sitting next to you, I want you to turn and talk about what are some things that you need to think about with these directions [e.g., the total distance and shape of the path that robot must follow] **Michael:** … do some early thinking now. Ready, set, turn …	 [prompts the students to think about the critical features of a problem]	→ [metacognitive strategy: critical analysis of a task] Michael engages the students in **abstraction**, which involves focusing on the most relevant and essential details of the problem.

(2) Feeling for the constraints of the problem: The second metacognitive strategy that the students were taught during CT was feeling for the problem's constraints. Michael combined verbal prompts with

perspective-taking to teach students how to set goals that are sensitive to problem constraints, which is a significant metacognitive activity (Paulson & Bauer, 2011). Table 3 displays how Michael first mentioned the limitations of the robot's movement. He said: "Dash can only walk forward, backward, turn right, and turn left." And then, following the verbal prompt, he visually executed those movements algorithmically as if he was the Dash robot. This interaction is significant for showcasing how algorithmic thinking and perspective-taking emerged as strategies to support students' goal-setting process—that is, a critical element of one's metacognition.

Table 3
A Moment of Algorithmic Thinking Through Verbal Prompts and Perspective-taking

Dialogue	Interaction	Transduction
Michael: Some of the groups are finding out that they made a plan that involves their robot traveling in certain directions. And then what they found out is that there are some limits that …Dash can only walk forward, backward …	 [prompts questions to feel for the constraints of a problem]	→ [metacognitive strategy: planning ahead] Michael prompts students to become cognizant about the problem constraints, executing a sequence of the robot's movements (i.e., CT practice of **algorithmic thinking**).
Michael: … turn left …	[Michael turns right, imitating the robot] 	
Michael: … or turn right … **Michael:** …So, there are really some things that you need to think about … that are a little bit different … So, if your planning involved just moving and sliding around, you might have to add in different things and try that out.	[He prompts the students to think about the critical features of a problem] 	→ [metacognitive strategy: planning ahead]

(3) Monitoring solution paths: The third metacognitive strategy that the students engaged in during CT was metacognitive monitoring. Metacognitive monitoring is "evaluating the process of learning or current state of knowledge" (Rivers et al., 2017, p. 549), and controlling that process is the ability to make changes to the original plan when it does not work as planned. Throughout the process, one asks oneself introspective questions, such as "am I following my plan? Is this strategy working?" (Martinez, 2006, p.698).

The following exchange between Michael and the students suggested that as students engaged in **debugging** (i.e., finding and fixing problems), they were given an opportunity to enhance their strategies associated with metacognitive monitoring and control. The following excerpt from Michael's instruction is strong evidence of how those initial strategies were later used by the students as part of their metacognitive monitoring at the end of the lesson, as appears in a set of brackets:

> What was nice is that they [i.e., small groups] kind of made a plan, and when they weren't sure [when the plan worked out well], they went back [monitored the solution], and they found some of the important information that was in these three things here [revisited the preset rules]. So, they said, well, I don't know, maybe I can do this, this, and this … [thought algorithmically & envisioned possible solutions]. And then when they were like, well, I'm not sure if I forgot an important part [checking task information to validate

comprehension]. They went back, and they were looking at the three important rules here because the big idea is that he wants to go on a trip.

Case 2

The analysis of the data from unplugged computational activity (i.e., Factor Frenzy) revealed three major themes of how CT could be used as metacognitive strategies: (1) discovering patterns, (2) activating students' relevant background knowledge, (3) creating a mental map of a problem.

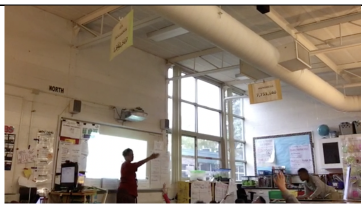
(1) Discovering a pattern: Our second teacher, Jill, first displayed the pair of numbers on a whiteboard, which gives a 10 when multiplied (aka factor pairs of 10), such as 1 x 10 = 10; 2 x 5 = 10. Like Michael's case, Jill began the lesson by encouraging the students to analyze a problem critically. She asked: "What sense can you make from this slide?" This exchange suggests that assessing the problem's initial condition is critical and common to CT and metacognition.

Jill: Take a look at the board. It says factor frenzy. *What sense can you make from this slide?*
Student: If that's factors frenzy, then the factors are like the multiplication numbers.
Jill: So, she remembers from last year. Good! That a factor is our multiplication numbers.
Jill: Tell me more. What does that mean?

(2) Activating students' relevant background knowledge: Next, Jill activated students' relevant background knowledge, which is an important metacognitive strategy (Lai, 2011). She asked, "Do you remember the name of these 10s? What were they called from last year?" (see Table 4).

In this case, activating students' relevant background knowledge went hand in hand with the CT practice of pattern recognition. Jill enabled her students to explore patterns based on the types of problems they had solved in the past. She used a variety of questions that engaged students in pattern recognition. She then used those patterns to help her students to understand the characteristics of a problem and figure out the kind of operations needed to solve a problem. As displayed in Table 4, she prompted her students to remember the nature of the problems that sum can be used: "Because remember … If we look over there [points at the CT posters in the classroom], sum goes with what kind of problems …"

Table 4

Dialogue	Interaction	Transduction
Jill: Anything else you notice? Somebody else? **Student:** … in the equation … All of them have 10s in. **Jill:** Do you remember the name of these 10s? What were they called from last year? Student: … Sum? **Jill:** Not the sum! Because remember … If we look over there [points at a poster in the classroom], sum goes with what kind of problems?	 [prompts the students to think about the givens of a problem]	→ [metacognitive strategy: activating relevant background knowledge] Jill prompts them to recall previous knowledge, using patterns to detect the inherent characteristics of a problem.

(3) Creating a mental map of a problem: The last metacognitive strategy Jill facilitated was when she mapped out "givens, a goal, and obstacles" (Davidson et al., 1994) of a problem for the students by posing several inquiry questions and suggesting strategies as appear in a set of brackets. For example, she said:

> I'm modeling (i.e., the problem) right now because you're going to be doing this. Today, you're going to be finding as many factors as you can for a number [setting a goal]. This is our target number [pointing at the givens of a problem]. And we were trying to find all the factors we could. We use the arrays to help us find the factors [cueing about reaching the goal]. And we use the arrays to represent the equations. That's what computer scientists do. That's called abstraction [cueing about when to use what CT skill].

Being able to establish a relationship between *givens, a goal, and obstacles* of a problem is helpful in decomposing a problem and abstracting the essentials of that problem, and those are also critical skills of CT. The exchange above is, therefore, strong evidence of the connection between CT and metacognition.

Discussion

This study presented how two elementary teachers used CT to support students' metacognition in the context of plugged and unplugged CT activities as a part of their mathematics instruction. The findings suggested that CT could be used as a pedagogical tool to explicitly teach metacognitive strategies to young children in elementary mathematics classrooms. In both CT activities, with or without a computational device, our teachers brought in diverse metacognitive strategies that helped strengthen their students' metacognition during the different stages of CT; those strategies and processes are listed in Table 5.

Table 5
Metacognitive Strategies Emerged During Plugged & Unplugged Computational Activities

Name of the CT Practice (Yadav et al., 2022)	**Associated Metacognitive Strategy**	**How Our Teachers Used CT As a Metacognitive Strategy**
Abstraction: "Focusing on the most relevant and essential details of the problem that needs to be solved." (p. 409)	• Critical analysis of a task • Paying attention to important ideas • Creating a mental model of a problem	Asking students questions to develop intuition about critical features of a problem: 1. "What's going to be important?" *2.* "What's important to know about this trip?"
Pattern recognition: "Finding the similarities and differences between problems." (p. 409)	• Activating students' relevant background knowledge through recognizing patterns • Understanding the inherent characteristics of a problem through those patterns	Asking students questions to encourage them to discover a pattern in the problem: 1. "What sense can you make from this?" 2. "What kind of problems does this pattern go with?"
Algorithmic thinking: "Designing a step-by-step solution to a problem." (p. 409)	• Perspective-taking to design a step-by-step solution	Asking students to think of the robot as an embodied agent, thinking of Dash as a human: 1. "Dash wants to go on a special trip. How can you program dash for his special trip?"
Decomposition: "Simplifying complex tasks by breaking them down into smaller parts." (p. 409)	• Creating a mental model of possible solutions	Designing lessons in a way that they have a built-in prediction component (e.g., "you have to predict the solution and draw the path")
Debugging: "Finding and fixing errors." (p. 409)	• Monitoring solutions	Asking questions to encourage students to think about their solution paths (e.g., "what makes you say that").

Our teachers often strengthened the metacognitive elements of the CT-integrated mathematics units by asking questions that made the students constantly think about their own decisions during the entire problem-solving process. We also observed that both teachers designed CT-integrated mathematics lessons in a way that both had a built-in prediction component, which is also crucial for metacognition. This helped learners to predict several possible solutions without executing them and select the appropriate strategies based on the expected outcome. As shown in Table 5, strategies used by our teachers are naturally a part of CT (first column) and essential to metacognitive skill development (second column).

Conclusion

This study is an initial attempt at exploring how CT can be used as an approach to support teachers to teach metacognitive strategies in elementary classrooms explicitly. Our findings hold several implications for the future of CT in elementary education. While much of the focus on CT in K-12 classrooms has been as a pathway to introduce computer science, our findings suggest that elementary teachers can use CT to teach metacognitive strategies to support disciplinary learning. Introducing those strategies might help students assess the initial conditions of a problem, devise solution paths responsive to the problem's constraints, and predict multiple solutions that could be applied as the problem conditions change (Liu & Liu, 2020). It should also be noted that one teacher used CT practices as a metacognitive tool in the context of the plugged activity, while another used CT as a metacognitive tool in the context of an unplugged activity. Future work should expand on this line of research to examine how CT can support students' learning in the core disciplines while improving their problem-solving skills.

References

Allsop, Y. (2019). Assessing computational thinking process using a multiple evaluation approach. *International Journal of Child-computer Interaction*, *19*, 30-55. https://doi.org/10.1016/j.ijcci.2018.10.004

Baxter, P., & Jack, S. (2008). Qualitative case study methodology: Study design and implementation for novice researchers. The qualitative report, 13(4), 544-559. https://www.nova.edu/ssss/QR/QR13-4/ baxter.pdf

Bezemer J (2015) How to transcribe multimodal interaction? A case study. In S. Norris & C. D. Maier (Eds.). *Texts, images and interaction: A reader in multimodality* (pp. 155–159). Netherlands: Mouton de Gruyter

Davidson, J., Deuser, R., & Sternberg, R. J. (1994). *The role of metacognition in problem-solving.* In J. Metcalfe & A. P. Shimamura (Eds.), Metacognition: Knowing about knowing. MIT Press.

Efklides, A. (2002). The systemic nature of metacognitive experiences. In *Metacognition* (pp. 19-34). Springer, Boston, MA.

Flavell, J. H. (1976). Metacognitive aspects of problem solving. *The nature of intelligence*.

Kafai, Y., Proctor, C., & Lui, D. (2020). From theory bias to theory dialogue: embracing cognitive, situated, and critical framings of computational thinking in K-12 CS education. *ACM Inroads*, *11*(1), 44-53. https://doi.org/10.1145/3381887

Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational psychologist*, *50*(4), 313-334. https://doi.org/10.1080/00461520.2015.1124022

Lai, E. R. (2011). Metacognition: A literature review research report. *Research Reports*, *41*.

Loksa, D., Margulieux, L., Becker, B. A., Craig, M., Denny, P., Pettit, R., & Prather, J. (2022). Metacognition and self-regulation in programming education: Theories and exemplars of use. *ACM Transactions on Computing Education (TOCE), 22*(4), 1-28. https://doi.org/10.1145/3487050

Liu, S., & Liu, M. (2020). The impact of learner metacognition and goal orientation on problem-solving in a serious game environment. *Computers in Human Behavior, 102*, 151-165. https://doi.org/https://doi.org/10.1016/j.chb.2019.08.021

Martinez, M. E. (2006). What is metacognition? Phi Delta Kappan, 87(9), 696–699

O'Malley, J. M., Chamot, A. U., Stewner-Manzanares, G., Kupper, L., & Russo, R. P. (1985). Learning strategies used by beginning and intermediate ESL students. *Language learning*, *35*(1), 21-46. https://doi.org/10.1111/j.1467-1770.1985.tb01013.x

Paulson, E. J., & Bauer, L. (2011). Goal Setting as an Explicit Element of Metacognitive Reading and Study Strategies for College Reading. *NADE Digest*, *5*(3), 41-49.

Prather, J., Margulieux, L., Whalley, J., Denny, P., Reeves, B. N., Becker, B. A., ... & Bosch, N. (2022). Getting by with help from my friends: Group study in introductory programming understood as socially shared regulation. In Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1 (pp. 164-176). https://doi.org/10.1145/3501385.3543970

Rivers, M. L., Dunlosky, J., & Persky, A. M. (2020). Measuring metacognitive knowledge, monitoring, and control in the pharmacy classroom and experiential settings. *American Journal of Pharmaceutical Education*, *84*(5).

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

Yadav, A., Ocak, C., & Oliver, A. (2022). Computational thinking and metacognition. *TechTrends*, *66*(3), 405-411. https://doi.org/10.1007/s11528-022-00695-z O

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*(6), 565-568. https://doi.org/10.1007/s11528-016-0087-7