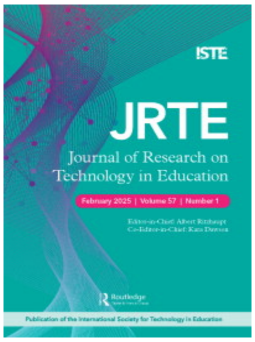


Journal of Research on Technology in Education

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/ujrt20

Pathways to inclusive early childhood computational thinking education: unveiling young students' strategies with multiple representations

Kristina M. Tank, Tamara J. Moore, Anne Ottenbreit-Leftwich, Zarina Wafula, Lin Chu & Sohheon Yang

To cite this article: Kristina M. Tank, Tamara J. Moore, Anne Ottenbreit-Leftwich, Zarina Wafula, Lin Chu & Sohheon Yang (2025) Pathways to inclusive early childhood computational thinking education: unveiling young students' strategies with multiple representations, Journal of Research on Technology in Education, 57:1, 2-25, DOI: 10.1080/15391523.2024.2410194

To link to this article: https://doi.org/10.1080/15391523.2024.2410194

Published online: 09 Oct 2024.

JOURNAL OF RESEARCH ON TECHNOLOGY IN EDUCATION
2025, VOL. 57, NO. 1, 2–25
https://doi.org/10.1080/15391523.2024.2410194

Routledge
Taylor & Francis Group

Pathways to inclusive early childhood computational thinking education: unveiling young students' strategies with multiple representations

Kristina M. Tank[a], Tamara J. Moore[b], Anne Ottenbreit-Leftwich[c], Zarina Wafula[a], Lin Chu[c] and Sohheon Yang[c]

[a]Iowa State University, Ames, Iowa, USA; [b]Purdue University, West Lafayette, Indiana, USA; [c]Indiana University, Bloomington, Indiana, USA

ABSTRACT

This study investigates the integration of computational thinking (CT) into early elementary literacy, focusing on kindergarten to second grade students, using multiple representations to understand their ideas of CT. Through clinical task-based interviews with 12 students, we found that concrete manipulatives, pictorial/graphical representations, and language-based strategies were key to facilitating CT comprehension. The findings indicate no significant gender differences in CT engagement, and instead we need to emphasize the need for inclusive, multi-representational teaching methods in early education. By using multiple representations, we may be able to nurture early STEM interest and confidence in computer science and related fields.

ARTICLE HISTORY

Received 7 April 2024
Revised 13 September 2024
Accepted 20 September 2024

KEYWORDS

Computational thinking; early childhood education; multiple representations; computer science; gender comparisons; early STEM interest

Introduction to the problem

Importance of broadening participation in computing (BPC) in CS education

Computer science (CS) education is essential for preparing students for the future workforce (Tissenbaum & Ottenbreit-Leftwich, 2020). However, CS education lacks inclusivity and excludes certain groups of students, especially historically marginalized populations (Code.org et al., 2023). This gap harms the diversity and quality of the CS workforce, and the representation and inclusion of women's perspectives and needs in technology. This exclusion also reduces the potential for innovation and creativity from diverse perspectives in the field (Shah & Yadav, 2023; Tissenbaum et al., 2021).

Addressing the lack of diversity in CS education is not merely a matter of social justice; it is a necessity for fostering innovation and addressing complex global challenges. Diverse teams bring a range of viewpoints and problem-solving approaches, which are invaluable in tackling multifaceted problems in technology and beyond (National Science Board, 2020). Without proactive measures to broaden participation, we risk perpetuating homogeneity within the tech industry, stifling innovation and limiting the potential for societal advancement (Shah & Yadav, 2023). Therefore, it is critical for educational institutions and policymakers to prioritize inclusive strategies that ensure equitable access to CS education for all students, regardless of their background or demographic characteristics (Vakil, 2018).

Scholars have suggested that the lack of diversity in CS is related to an early loss of interest and confidence that many female students experience in science, technology, engineering, and

CONTACT Kristina M. Tank kmtank@iastate.edu Iowa State University, 0624E Lagomarcino, 901 Stange Rd, Ames, IA 50011-1041, USA.

mathematics (STEM) and CS subjects (Kucuk & Sisman, 2020; Metz, 2007). Research has shown that female students tend to perform well and express interest in STEM and CS in elementary school, but their engagement and self-efficacy decline by late elementary and middle school, especially in mathematics and CS (Cvencek et al., 2011; Lei et al., 2019; Master & Walton, 2013). Therefore, if we want to engage more historically marginalized populations of students in CS, we need to begin at the early elementary grades when there are fewer gender differences in STEM-related motivation, interest, belonging, or ability (Master et al., 2023). However, teaching CS at the elementary level has faced a variety of challenges, with one of the largest being a lack of time due to other subject areas (e.g. Israel et al., 2015). Integrating CS into subject areas that are already emphasized at this level could provide support for teachers who have limited instructional time for other subjects, such as CS, that are not tied to testing (e.g. Rich et al., 2019). In early elementary, literacy receives substantial attention/time and it has been suggested literacy could be an ideal area for CS integration given similarities in language learning and CT (Bers, 2019; Delacruz, 2020). Currently, there is limited information on how to engage elementary students in CS within the context of literacy and if or how this integrated approach supports early understandings of CS.

Research questions

To better understand how to engage early elementary students in CS within the context of literacy, we used the following research questions to investigate CT integrated literacy activities: (1) What aspects of the CT and literacy learning tasks support early understandings of CT competencies for all students and, (2) how do early elementary students, especially girls, experience and participate in literacy integrated CT instruction?

Inclusivity in computer science education

Although computer science (CS) is a field that offers many opportunities for innovation, creativity, and social impact, it is also a field that suffers from a lack of diversity (National Science Board, 2020). The national CS for All initiative has notably boosted interest in computer science education, focusing heavily on K-12 to support the CS pipeline (Code.org et al., 2023). However, most CS education for young learners does not resonate with their everyday experiences (Tissenbaum et al., 2021). This can lead to young learners' disengagement from computer science, especially among those groups that are historically marginalized from computing and STEM fields (Valla & Williams, 2012). This decline in interest is influenced by several factors, such as the lack of exposure and access to CS curriculum and activities (Lin et al., 2012; Su et al., 2023), the negative stereotypes and social norms that discourage girls from pursuing STEM and CS (Kucuk & Sisman, 2020; Metz, 2007), the low sense of belonging and identity that girls feel in these fields as well as the lack of feedback and encouragement from teachers, parents, and peers (Cvencek et al., 2011; Lei et al., 2019; Master et al., 2023). Master et al. (2021) also found that stereotypes of girls being less interested in CS than boys persisted as young as first grade and may contribute to gender gap issues as children get older. Other studies have also shown that girls as young as second grade have expressed lower self-efficacy and attitudes toward CS before any interventions (Kind, 2017) and have observed gender differences showing that boys were more likely to describe themselves as skilled programmers (Kjällander et al., 2021). Therefore, building students' CS knowledge, confidence, and interest at the K-2 levels may be enough to build students' positive STEM/CS identity (Martin-Hansen, 2018; Su et 2023) during this pivotal time.

To establish an inclusive and diverse computer science education ecosystem, we need to examine the content in the CS curriculum, the design of the learning activities, and the focus on our purpose for CS education (Vakil, 2018). In other words, when designing CS education experiences, we need to make efforts to create curricula that are designed for all students. We can accomplish this through three pedagogical practices as suggested by Ryoo (2019): (1) making CS more relevant by connecting concepts to everyday examples and contexts, (2) focusing on issues that are relevant to those students, and (3) providing opportunities for student choice and agency.

Early CS/CT and literacy

Research within early childhood education and CS/CT have found that young children can successfully demonstrate early CT skills (Saxena et al., 2020) and effective instruction has been shown to enhance early CT and coding skills as well as improve mastery of CT and programming concepts (e.g. Bers, 2019; Relkin et al., 2021; Su & Yang, 2023). Developmentally appropriate practices in CS have typically included unplugged activities (without computers or other computational devices), computational toys (such as robots), and/or plugged activities (using a digital app or coding platform like ScratchJr). Studies have found that unplugged activities in conjunction with plugged activities have been shown to help students learn CT better than plugged activities alone (del Olmo-Muñoz et al., 2020), especially when utilizing these approaches through concrete experiences (Bati, 2022).

Although CS and CT are often linked to STEM disciplines, researchers such as Bers (2019) have suggested that "due to the critical foundational role of language and literacy in early years, the teaching of computer science can be augmented by models of literacy learning" (p.499). Incorporating CS into literacy education could provide a significant approach to creating inclusive CS learning for young learners by centering them on an active agent in their learning. Storytelling and imaginative play, which are integral parts of early literacy, serve as ways for children to make sense of and interpret the world around them (Cremin & Flewitt, 2016). CT is seen as a fundamental skill for all, like literacy, essential for navigating daily life (Tsortanidou et al., 2021), offering potential when combined with literacy instruction. In fact, Macrides et al. (2022) found that integrating programming into literacy and storytelling exercises can serve as a suitable method for developmentally appropriate learning for young learners. Furthermore, programming, a key element in CS, is akin to mastering a new language, enabling expression of ideas and acquiring knowledge of programming languages involves understanding sequence and problem-solving skills, thereby teaching young students analytical and logical reasoning (Bers, 2019). This suggests that when looking at the integration of CS/CT into early elementary classrooms, an integration into literacy could not only build on the idea that learning to code involves learning how to use a new language, but it also builds on the practical aspect that literacy and language is a foundational component within early elementary instruction. This further supports recommendations that early CS/CT instruction needs to be integrated into subject areas that are already emphasized at this level as elementary teachers have limited time for additional content and instruction (Israel et al., 2015; Rich et al., 2019).

Early CS/CT and multiple representations

Moreover, when looking at CS/CT with young learners, exploring how multiple representations and translations between them can be a meaningful approach to assess young learner's CS performance because they communicate through more than just spoken words. Valuing multi-modal representations, such as gesture, images, and movement, can promote equitable and inclusive learning environments by focusing on the varied symbols used by young learners (Flewitt, 2016). Moore et al. (2020) found that as early elementary students engaged with CT tasks developed to look at specific representational translations, individual students created and used additional intermediate representations in order to successfully complete these CT tasks. These intermediate representations included such things as using embodiments to keep track of where they were in their coding and reorienting their bodies to make a visual representation more manageable. Through these research studies, it is seen that welcoming different symbolic tools enhances active participation in tasks among young learners.

Along with studies examining the importance of multiple representations for young learners (e.g. Flewitt, 2016; Moore et al., 2020), there have been studies investigating how spatial reasoning, as a way of representing young learners' problem-solving, is intertwined with CS/CT (Berson et al., 2023; Brainin et al., 2022; Margulieux, 2019). Young learners' spatial skills are an important indicator of how they successfully solve problems and make achievements in CS education (Margulieux, 2019). Some studies especially talk about the use of robot programming and spatial reasoning. For example, Berson et al. (2023) discovered that programming with the Sphero robot (as a concrete representation) forms an essential foundation for spatial reasoning and computational thinking; young learners, for instance, used their bodies to indicate spatial awareness by positioning themselves near the robot. Also, according to Brainin et al. (2022), robot programming could enhance young learners' spatial reasoning, such as spatial relations and mental rotation skills, by providing a hands-on cognitive tool that allows them to work on their problem-solving process. Also, enhancing spatial skills helps encode non-verbal data and recognize landmarks (Margulieux, 2019), that can be potentially utilized for CS/CT tasks. Thus, investigating how young learners utilize their spatial skills such as visualization in CS/CT tasks could be meaningful in looking at the relationship of early CS/CT and multiple representations.

Theoretical framework

As we were examining the ways in which the students engaged and interacted with the K-2 CT tasks, we used the Lesh Translation Model (LTM) as the lens through which we examined our data. Building on Bruner's (1966) theory of enactive, iconic, and symbolic representations, the LTM (Lesh & Doerr, 2003) provides a framework for understanding how students develop fluency in using different representational modes. LTM expands on Bruner's work by identifying five key modes: (1) real-world experiences, (2) written symbols, (3) spoken or written language, (4) pictures and graphics, and (5) concrete manipulatives. This model emphasizes that deep conceptual understanding hinges on a student's ability to not only represent concepts in these various forms but also to flexibly translate between and within them (Lesh & Doerr, 2003). The more seamlessly a student can navigate these representational modes, the better they grasp the underlying concept. Importantly, each mode highlights different aspects of the concept, and a complete understanding often requires considering information across multiple representations. Computer science can be taught using multiple representations such as real life examples, everyday language, pseudocode, flowcharts, code tracing/tracking charts and tables, coding languages, etc. (Malik et al., 2019). However, most representations of algorithms and debugging within coding are much too abstract for children as a place to start (Fessakis et al., 2013). Therefore, it is important that students, especially K-2 students, are introduced to computational concepts using multiple representations starting with more concrete ideas and then progressing to more abstract ones; while at the same time, making connections between the representations.

Our tasks are designed such that students participate in multiple types of representations, other than written symbols as they are still learning to read. It is also important to note that many representations can be classified across representational modes. In this research, we chose to classify the representations based on how students were using them. All tasks are set in real-world experiences by tying in images, games, toys, picture books, etc. and so due to space limitations, we have chosen not to present representations specific to real world experiences.

Materials and methods

We employed a clinical, task-based interview design (Clement, 2000; Goldin, 2000) to examine 12 K-2 students' understanding of sequencing as a foundational CT skill through a series of task-based interviews (Goldin, 2000). The task-based interviews target attributes of CT/CS learning including how students develop CT knowledge, how students work through the tasks, and what causes them to successfully complete or get stuck. The multiple task-based interview study design allowed for the in-depth exploration of multiple learners across K-2 and their early understandings of CT.

Data collection

Our research team collected eight task-based interviews from each of the 12 K-2 students across four data collection sessions. Our 12 participants included two girls and two boys from each of the three grade levels and each student participated in all eight of the tasks. All of our students were from three schools in the Midwest region of the United States. Our Kindergarten students were from a rural area, our first grade students were from an urban area, and our second grade students were from a rural area. Table 1 provides relevant information about our participants. It is important to note that we did not ask students what their preferred genders were, but asked our project partner teachers to identify boys and girls based on how it was communicated by the district and parents. Within the states where this research was conducted, it was not a viable option to ask students for their preferred genders. Additionally, our partner teacher at each grade level helped us choose the participants from the students whose parents had consented to the request for research and photo release as defined by our approved IRB protocol and by considering a range of abilities in the students chosen.

Our data were video recordings from the task-based interviews which offered a structured approach to studying student thinking (Goldin, 2000) about the foundational CT concept of sequencing. In these interviews, we presented individual students with carefully designed sequencing tasks (Maher & Sigley, 2020). Students then worked through the tasks talking aloud as they saw fit or were prompted. The task-interview protocols first had the researcher providing directions for the students. For example, from our Task 1 interview protocol, the researcher would say, "Can you help me by using these direction cards to put these pictures into the correct order? You can do whatever you want to do to help you put these pictures in order." Then the protocol instructs the researchers to observe and take notes, only helping the students when they are stuck. The protocols also allow for researcher questions to get at the student thinking if there is something that is unclear in order to reveal their reasoning and thought processes (Clement, 2000). For example, a researcher may interject with a question like: "Isabella, why did you put the cards in this arrangement?" Our task-interview protocols allow for in-depth exploration of how students engaged with the concepts, including how they developed knowledge, navigated challenges, and utilized representations (Maher & Sigley, 2020). The following section describes the tasks and their development.

Table 1. Participant information.

			Session			
Grade	Pseudonym	Biological Sex	1	2	3	4
K	Lily	female	Task 1 → Task 2	Task 3 → Task 4	Task 5 → Task 6	Task 7 → Task 8
	Henry	male	Task 2 → Task 1	Task 4 → Task 3	Task 6 → Task 5	Task 8 → Task 7
	Sophia	female	Task 2 → Task 1	Task 4 → Task 3	Task 6 → Task 5	Task 8 → Task 7
	William	male	Task 1 → Task 2	Task 3 → Task 4	Task 5 → Task 6	Task 7 → Task 8
1	Isabella	female	Task 2 → Task 1	Task 3 → Task 4	Task 6 → Task 5	not present
	Joe	male	Task 2 → Task 1	Task 4 → Task 3	Task 6 → Task 5	Task 7 → Task 8
	Jonathan	male	Task 1 → Task 2	Task 3 → Task 4	Task 5 → Task 6	Task 8 → Task 7
	October	female	Task 1 → Task 2	Task 4 → Task 3	Task 5 → Task 6	Task 7 → Task 8
2	Charlie	male	Task 2 → Task 1	Task 4 → Task 3	Task 5 → Task 6	Task 8 → Task 7
	Connor	male	Task 1 → Task 2	Task 3 → Task 4	Task 6 → Task 5	Task 8 → Task 7
	Mallory	female	Task 2 → Task 1	Task 4 → Task 3	Task 5 → Task 6	Task 7 → Task 8
	Melody	female	Task 1 → Task 2	Task 3 → Task 4	Task 6 → Task 5	Task 8 → Task 7

Description of tasks

The design of the tasks is an important component when implementing a task-based interview approach as the tasks are what drives the eliciting and collection of participants' knowledge and thinking around the desired concept. In the ReCT project, we are breaking down CT into different concepts to examine student learning for early elementary students in CT. For this particular study, the identified CT concept was algorithm development, which can be further broken down into sequencing, repetition structures, and selection structures (Gao & Hew, 2022). Sequencing, or the ability to order steps and understand their relationships, was chosen as the focus as it has been suggested as critical to early CT and positive links have been found between sequencing and later CT performance (Kazakoff et al., 2013; Su & Yang, 2023).

The tasks were developed and piloted by the ReCT team along with three K-2 teacher fellows to elicit the following student understanding of sequencing: simple ordering logic, simple ordering logic with identification of beginning and end of sequences, reverse sequencing, and ordering with multiple logic paths (see Table 2). For each of these concepts, a set of paired tasks were developed: one CT focused and one CT and literacy focused. Four pairs, or eight total tasks, were developed (Tank et al., 2024) and as suggested by the research (Su & Yang, 2023), each of the tasks have multiple versions – unplugged (paper based), embodied (enacted by students), plugged with computational toy (such as push button robots), and plugged with digital devices (such as iPads). The students did these tasks in different orders to examine whether the sequence of versions and/or whether literacy or non-literacy tasks changed the students' experiences (see Table 1).

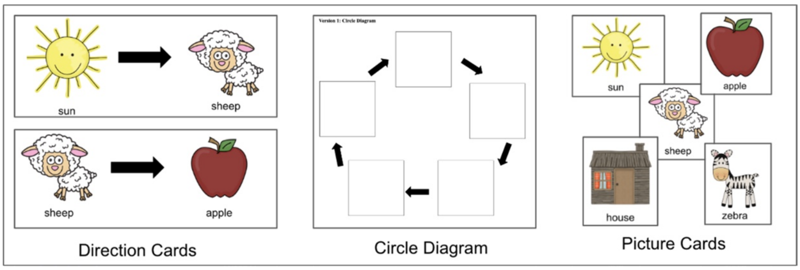
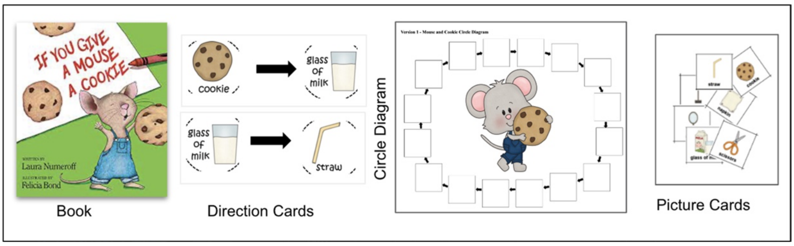
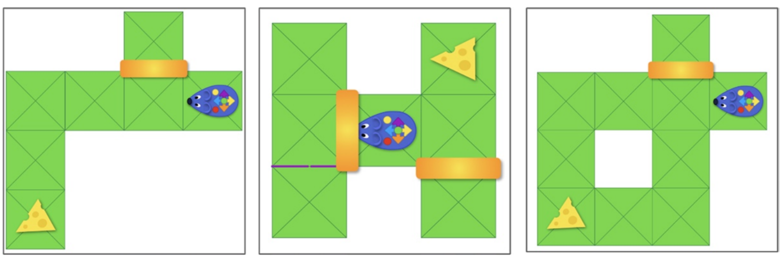
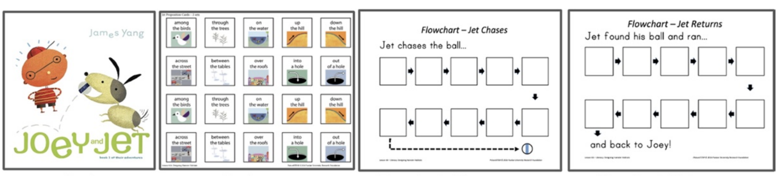
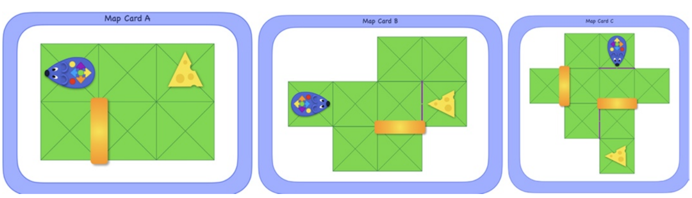
Data analysis

In this study, we utilized a task-based interview approach (Goldin, 2000) to capture detailed observations and actions of students as they engaged with four computational thinking (CT) tasks. We used a constant comparative analysis method (Corbin & Strauss, 2014) to iteratively code, compare, and condense data to identify emerging patterns of how students used and translated between representations. We analyzed over 48 h of video recordings collected across the 12 students, with coding conducted by a team of six researchers over three rounds.

Initially, a priori codes were applied based on the representation framework of the Lesh Translation Model (LTM) and gestures. The codes included concrete (e.g. physical objects and movements), pictorial (e.g. course maps and student drawings), symbolic (e.g. code cards and inscriptions), language (e.g. spoken or written words), and gestures (e.g. body movements for representational purposes) (Lesh & Doerr, 2003). To enhance the validity and reliability of our analysis, we employed the LTM as our analytical framework, focusing on the students' interactions with various representational modes across each task. The LTM, rooted in Bruner's (1966) theory of enactive, iconic, and symbolic representations, provided a structured lens for understanding how students develop fluency in using different representational modes. Our analysis paid close attention to students' ability to navigate and translate between these modes, noting similarities, differences, and patterns in their interactions.

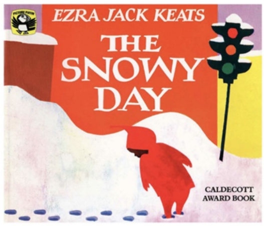
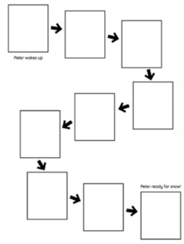
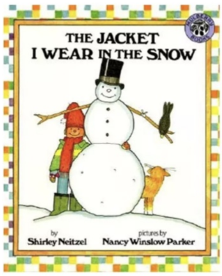
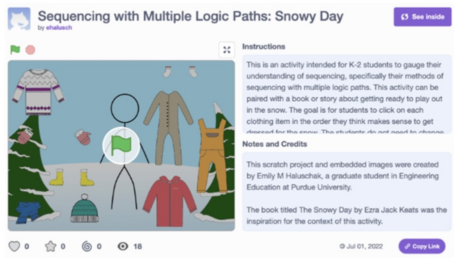
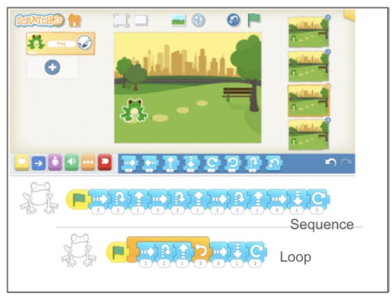
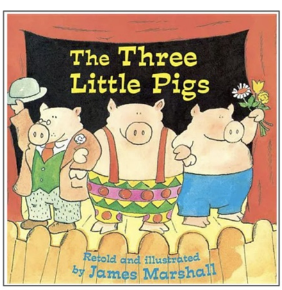
In our coding process, one researcher focused on capturing how students used existing and invented representations during the tasks. The initial researcher selected for analysis looked for one particular representational mode. For example, one researcher reviewed the videos for concrete representations whereby students were moving physical objects and movements to represent their CT understandings. Another researcher focused on the language representations, looking for examples of when students used spoken or written words to represent their CT understandings. Once these examples were identified, these findings were then discussed with two additional researchers, who also reviewed and analyzed the videos. Together, they expanded the coding to include strategies that individual students employed as they translated between and within

Table 2. Overview of tasks.

Task: concept	Short description
1: Simple Ordering Logic	 Students use picture-based direction cards to place picture cards into circular organizer by following A ⟶ B, B ⟶ C, C⟶D logic to determine the order (3 levels of difficulty)
2: Simple Ordering Logic + Literacy	 Students sequence a circular story using either picture-based direction cards and/or the story to place picture cards into a circular organizer representing the A ⟶ B, B ⟶ C, C⟶D logic (There are 3 versions with the first being a practice).
3: Reverse Sequencing	 Students plan and program a path for a robot mouse to get a piece of cheese and then reverse that path (Multiple versions: Unplugged, Embodied, & CT Toy)
4: Reverse Sequencing + Literacy	 Book Cards Flowchart- Chase Flowchart- Return Students retell a story by putting the major events in order for how a dog retrieved a ball (forward) and then returned it to Joey (backward)
5: Sequencing with Multiple Logic Paths	 Students plan a path for the Robot Mouse with multiple possible paths to reach the cheese (Multiple versions: Unplugged, Embodied, & CT Toy)

(*Continued*)

Table 2. Continued.

Task: concept	Short description
6: Sequencing with Multiple Logic Paths + Literacy	 Book  Set of Winter Clothing Cards  Winter Clothing Flowchart  Book  Scratch Activity Students determine the sequence for getting dressed in Winter Gear to go out into the snow (Multiple versions: Unplugged, Embodied, & CT Toy)
7: Sequencing with Programmed Multiple Logic Paths	 Level 1 - walk  Level 2 - hop Students program a path for the frog with multiple possible codes to get to the bench (Two versions: Walk & Hop).
8: Sequencing with Programmed Multiple Logic Paths + Literacy	 Book  Scratch Activity Students retell the story by programming the big bad wolf, with multiple possible codes, to visit the houses of the three little pigs in the correct sequence.

representations. The other three researchers subsequently re-coded the videos to identify elements previously recognized by the research team, with all researchers working to reach consensus. From this coding process, themes were developed, and the results are presented both from each task interview and from the overarching themes identified.

At the conclusion of the coding process, we identified that student understanding within CT was primarily expressed through three representational modes: concrete manipulatives, pictorial/graphical representations, and language. We then identified sub-themes within these categories to provide a more detailed account of how students utilized these representations during the tasks. By comparing the experiences of boys and girls across tasks, we aimed to present a holistic portrayal of how they engaged with sequencing as a foundational CT skill.

The research team comprised six members, all of whom had experience teaching computer science in elementary classrooms. The three lead researchers, who were also professors, conducted the task interviews with the students. Throughout this process, the researchers remained aware of their positionality, recognizing that their backgrounds in computer science education could influence the interpretation of data. Efforts were made to mitigate this bias through collaborative discussions and consensus coding, ensuring that the analysis reflected a balanced and accurate representation of the students' experiences. For example, before beginning the grant, the three lead researchers had extensive conversations to identify different CT components and how we would measure these. Furthermore, we worked closely with three early childhood classroom teachers to design the tasks, ensuring they were developmentally appropriate and focused on critical CT skills.

Limitations

Some limitations of this study include the small sample size, as only 12 students participated, which may not fully represent the diversity of student experiences and abilities. Additionally, the study focused on students from specific geographical areas, which might limit the generalizability of the findings to other regions or populations. The use of task-based interviews, while providing in-depth data, may also introduce observer effects, where the presence of researchers could influence how students perform the tasks. Moreover, the study's reliance on video recordings might overlook subtle nuances in students' thinking processes that could be captured through other data collection methods. Finally, the researchers' backgrounds in computer science education, despite efforts to mitigate bias, could have influenced the interpretation of the data.

Results

To better understand how to engage early elementary students in CS within the context of literacy, we examined what aspects of the CT and literacy learning tasks supported early understandings of CT and how early elementary students engaged in these tasks. Results from this study will be presented using multiple representations as a framework for student understanding within CT and will be presented by primary representational mode: concrete manipulatives, pictorial/graphical representations, and language. Much of the results provide evidence of translation between or within representational mode, so this will be discussed within the results as appropriate.

Concrete manipulative representations

Concrete manipulative representations are described as tangible, physical objects or tools that students interact with to perform tasks. These can include items like coding cards, robots, or other hands-on materials that require students to physically move, arrange, or manipulate them to solve problems or demonstrate understanding of computational thinking (CT) concepts. These representations served as cognitive scaffolds, helping to externalize and organize students' thoughts, and to bridge their understanding of abstract CT concepts by providing a more concrete, tangible experience.

Tracking movement physically

Task 3 and 5 required students to program the mouse to get to the cheese. While observing students during these tasks, it became evident that there were two approaches through which they tracked the path of the mouse on the maze. Eight out of 12 students, that included three boys and five girls, used the physical mouse as a placeholder by picking it up and moving it around the maze to visualize the next steps (sequence) of the path. For example, Sophie used the paper mouse in our unplugged version to help her keep track of the movement (see Figure 1), while Isabella used the actual computational toy robot mouse to help her keep track of the movement (see Figure 2 - left image). This approach allowed them to physically simulate the movement of the mouse—using the orientation of the mouse and the push buttons on the back—to plan its trajectory on the maze. By physically manipulating the mouse, they could better understand the spatial relationships and anticipate the sequence of direction cards required to reach the cheese. The second approach—which many students did, but only two students used for any length of time—was to use their finger or hand to keep their place on the maze. For example, Joe started out using his memory to track the mouse on the maze, but when that presented challenges, he then used his finger to map out the course on the maze (see Figure 2 - right image) then as it got more complex, he used the mouse to track where he was in some instances. Both of these approaches facilitated their problem-solving process enabling them to successfully program the mouse to get to the cheese.

Physically pushing buttons with each step

As an example of a directly related concrete use of the manipulatives, both Lily and Isabella physically pushed buttons on the device to input a command and then manually moved the mouse to follow the programmed path through each step. This action represents a highly concrete form of engagement with computational thinking (CT), where the students not only determined the sequence of steps but also physically enacted the outcome. This tactile and interactive approach allowed them to see the immediate effects of their programming decisions

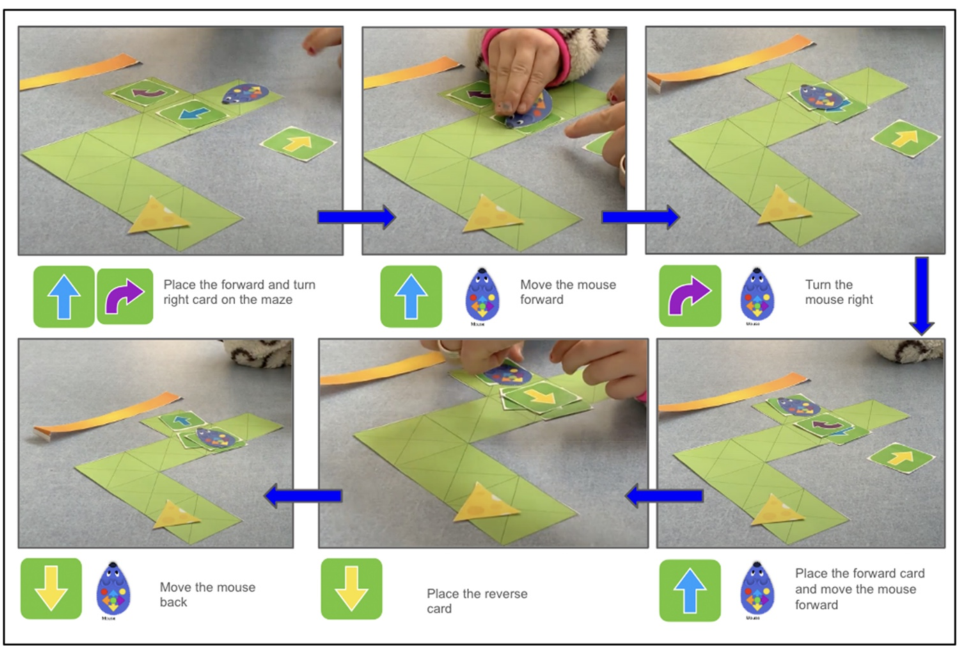


Figure 1. Sophia (K) Picking up and moving around the mouse as a placeholder to determine the sequence of the mouse on the maze.

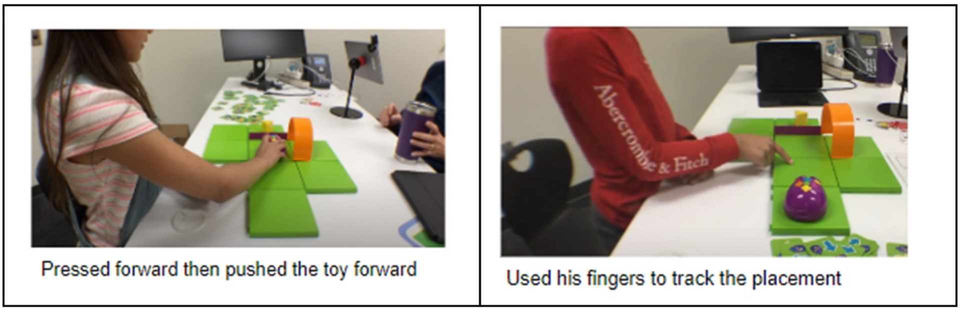


Figure 2. (a) Isabella (1) used the mouse robot to track maze placement and (b) Joe (1) used his fingers to track the placement of the mouse..

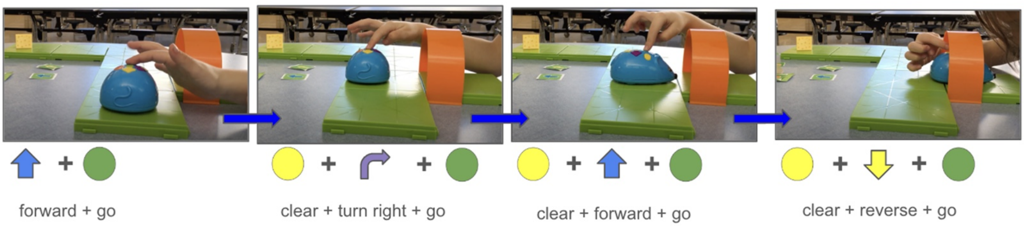


Figure 3. Lily's example of using the mouse to code each step.

and track the code more easily. By physically interacting with the technology, Isabella and Lily (see Figure 3) bridged the gap between abstract CT concepts and real-world application, demonstrating a hands-on learning process that made the computational task more comprehensible and grounded in their experience.

Representational style of direction cards

Tasks 1 and 2 were designed for students to use the direction cards (A→B; B→C) to order the picture cards onto a circle diagram mat. The way that the students managed this task related to how they used the concrete representations of the direction cards. There were four primary ways that the students organized the cards to complete the tasks: hunt-and-peck style, end-to-end, top-to-bottom, and discard pile. Table 3 shows the styles each student used throughout the Tasks 1 and 2.

Hunt-and-peck representation style

The hunt-and-peck style of using the direction cards was employed by nine of our 12 students. This entailed looking for the card needed with little to no organization of the direction cards and only moving direction cards to be able to see them. Here, the students started with a direction card (often the closest one or the one they saw first) and then placed both corresponding picture cards on the mat. Then they would hunt for the next direction card (the one where the picture on the left matched the last card's right picture). Then place the appropriate picture card on the mat. Only three students, Lily (K), Henry (K), and Melody (2) stuck with this representation style throughout both tasks and all versions. Jonathan (1) used this on Task 1 which was his first task, and William (K) (see Figure 4), Connor (2), and Mallory (2) both used this later.

Discard pile representation style

The discard pile strategy was similar to the hunt and peck strategy; students did not try to organize the direction cards before placing the picture cards on the mat. Here though, students

Table 3. The representation style used by each student in Tasks 1 and 2. Some students used multiple representational styles and, therefore, have multiple check marks. The girls' names are underlined to make biological sex easy to identify.

		Representation style			
Grade	Pseudonym	hunt-and-peck	discard pile	end-to-end	top-to-bottom
K	*Lily*	✓			✓
	Sophia				✓
	Henry	✓			
	William	✓			✓
1	*Isabella*			✓	
	October				✓
	Joe	✓	✓		
	Jonathan	✓		✓	
2	*Mallory*	✓	✓	✓	
	Melody	✓			
	Charlie		✓		
	Connor	✓	✓		

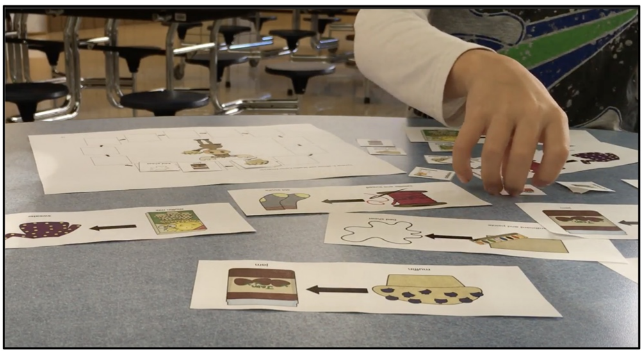


Figure 4. William using the hunt-and-peck method for Task 2 – Moose and Muffin.

would move (or discard) any direction card that they had already used as a way to manage knowing what cards they had left to consider. Four students used this strategy: Joe (1), Mallory (2) (see Figure 5), Charlie (2), and Connor (2).

End-to-end representation style

The end-to-end strategy has students chaining the direction cards so that matching pictures are next to each other. Three of our students used this strategy: Isabella (1), Jonathan (1), and Mallory (2). Two students laid out the direction cards so that all cards were in a single chain, while two students also laid them out in left-to-right chains of a few cards and then made rows down. Isabella did both (see Figure 6). Interestingly, Isabella placed her card in both ways across all of her tasks, sometimes she placed a direction card in her chain and then a picture on her mat (e.g. Figure 6 - left image) and other times she laid out all of her direction cards then all of her picture cards (e.g. Figure 6 - right image).

Top-to-bottom representation style

The strategy of laying cards from top-to-bottom had the students creating a column of cards such that the right picture of one card was the left picture on the card below it. Four students used this strategy: Sophia (K), William (K), Lily (K), and October (1) (see Figure 7). October always organized her direction cards first, then placed all of the picture cards.

The hunt and peck and the discard pile strategies had students alternating between finding a direction card and placing the picture card on the mat. These strategies generally showed the students decomposing the task into single operations—find and place the next card. The

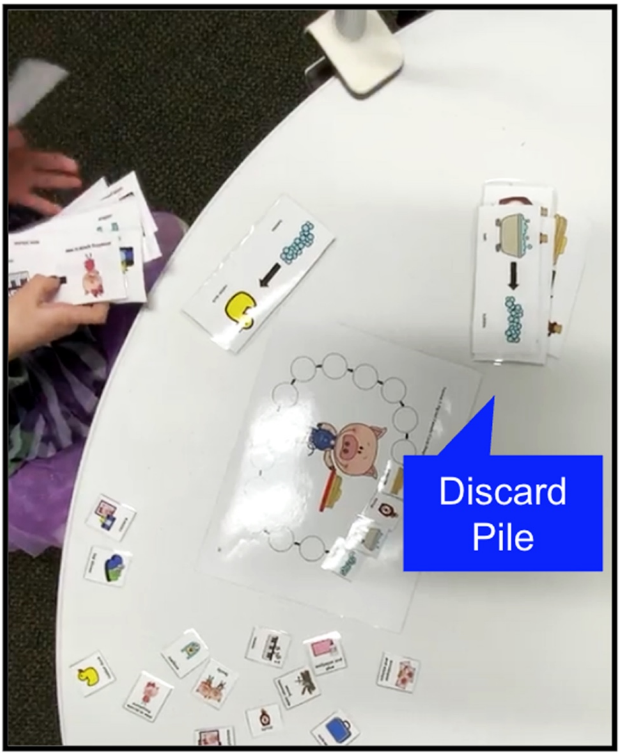


Figure 5. Mallory uses the discard pile strategy as she works through Task 2 – If You Give a Pig a Pancake.

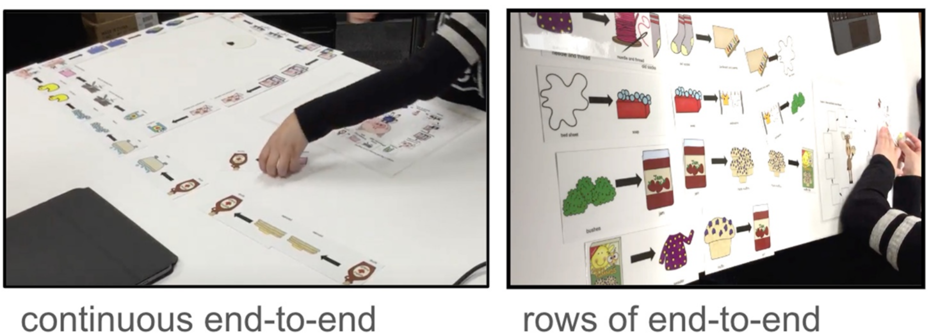


Figure 6. Isabella with her cards end-to-end both one continuous chain and rows of chains.

Figure 7. October placing her direction cards in the top-to-bottom strategy before placing any picture cards on the mat.

end-to-end and top-to-bottom strategies allowed students to either alternate between finding a direction card and placing a picture card on the mat (like the previous strategies) or order all direction cards first and then place all of the picture cards on the mat. This latter strategy showed the students decomposition of the task had a holistic quality—find the pattern, then place the picture cards.

Ineffective card placement on the maze

The use of the concrete representation of the coding direction cards for the robot mouse (Task 3 and 5) was a barrier to them coding the mouse correctly. One way that this was seen was when the students laid the coding cards directly on the maze. Several of our 12 students gravitated toward this concrete action in a way that suggested that they were using the cards as a placeholder to manage where they were in planning out the path of the robot mouse. This caused confusion because: (1) students often did not understand how the cards represented the movement and (2) the placement of the first card on the maze determined where the second card would need to go in terms of spacing.

A common issue with placing the cards on the robot mouse maze was that students had a misunderstanding about how the cards represented movement. One way that this played out was when the maze included a corner, it takes three cards to represent the mouse traveling through a turn (e.g. forward, right, forward) and the students often only placed one (e.g. right). In Figure 8, Melody (Grade 2) placed a single left turn arrow card on the corner. We saw this error across most students who placed the cards directly on the maze. To get the students past this type of error, the researcher always had to suggest laying out the cards off of the maze, which was the case for Melody as well.

Another way that this played out was that students placed the arrow cards rotated to match the path intended. Figure 9 showed Melody as she began Task 3 with the computational toy version. This time, she used just one arrow to represent the path the mouse will take. She does this both for the corner and for the tunnel as marked with red arrows on the image. In this case, the arrow cards chosen are meant to represent the mouse moving forward (blue arrow) or backward (yellow arrow).

A second common issue was spatial consistency throughout placing the cards on the maze. When the student placed the first card, which was usually a forward movement, they placed it either where the mouse started, on the seam between two squares, or where the mouse would land. Once this choice was made it was necessary to keep this consistent spacing throughout the remainder of the placing of cards. However, most students did not recognize the need for consistent spacing. This was common among both the girls and boys in our study at all grade

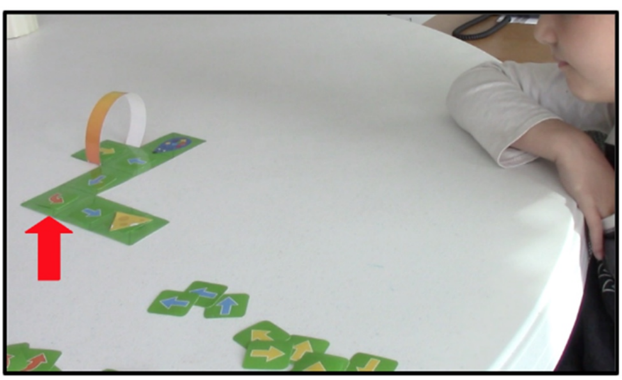


Figure 8. Melody laying cards directly on the unplugged version of Task 3, using a single left arrow to represent a turn. Her code also uses only one card per square which led to coding problems beyond just the corner turn.

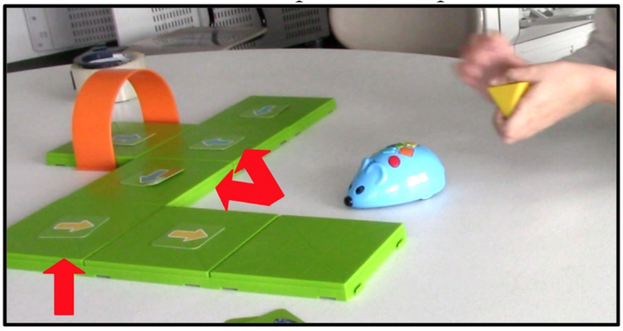


Figure 9. Melody laying cards directly on the computational toy version of Task 3, using rotated forward and backward arrows represent the path the robot mouse will take.

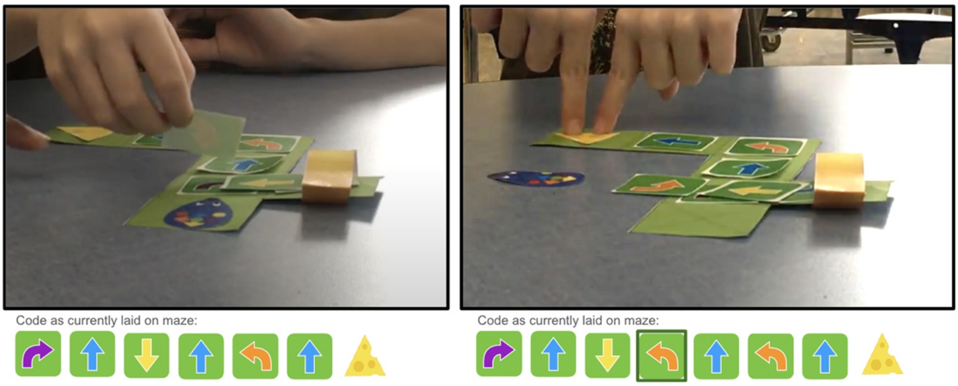


Figure 10. William (K) laying out cards on the maze with spatial issues creating barriers to completion.

levels. Each of the students was unable to solve the task until the researcher prompted them to reorganize the cards off of the maze. In one example, William (K) began by placing cards on the path (Figure 10 - left image). When the mouse did not get to the cheese, William recognized that he was missing a step, and added the turn (orange left arrow card) beside the backwards (yellow arrow) card creating a pile of cards (Figure 10 - right image). When the mouse did not end up at the cheese, he struggled with what to do next as the cards seemed to fill the space. He found it hard to figure out what cards he was missing and how to manage the cards due to his spacing and placement of the cards.

Pictorial/graphical representations: laying out cards in the shape of the board

Pictorial/graphical representations referred to the use of visual elements like images, diagrams, or graphical organizers that students use to understand and solve computational thinking (CT) tasks. These representations can depict sequences, processes, or spatial relationships in a visually interpretable manner, allowing students to conceptualize and reason about CT concepts more effectively. They served as a bridge between concrete manipulatives and abstract thinking, helping students visualize and internalize the steps involved in coding and computational tasks.

In the last concrete representation examples, we saw that the concrete manipulative use of laying the cards on the maze was generally problematic. Four students, three girls and one boy, laid out the direction cards for the robot mouse off of the maze, but in a shape that represented the path of the mouse through the maze. Because the layout of the cards was a graphic representation of how the mouse traveled, we classified this a representational translation from concrete to pictorial/graphical.

In one example, Jonathan (Grade 1) first laid out the cards on the maze in the path that the mouse was traveling on the maze (see Figure 11-left image). The researcher prompted him to move the cards off of the maze so that the mouse could travel on the maze. Jonathan then moved the cards off of the maze in the same shape as the path. He tested the code by coding the mouse and allowing it to travel the maze (see Figure 11- middle image). This code had the mouse coming off of the track past the tunnel. The pictorial representation of the cards allowed him to recognize spatially where the code went wrong by comparing where the mouse went off the maze to that same point in his pictorial layout of the cards. He removed the extra blue arrow and tested again. He continued to debug in this manner (one mistake at time) until he got the mouse to the cheese (Figure 11 - right image).

Task 3 was intended to see if students could reverse the sequence. The researcher asked him to keep the forward corrected code on the table:

> Okay. I want you to leave this [pointing to the direction cards on the table], how it is–because this is his [the mouse's] forward movement. I want you to pretend that our robot mouse can pick up this cheese and take it back to his home. But I want him to do everything he did–backwards. So if [the mouse] went forward, I want it to go backwards.

To accomplish this, he did not consider his original correct code and try to reverse it, rather he went through a very similar process to going forward, which was layout the cards and test. This time, though, he did keep the cards off of the board. Figure 12 shows a few shots of progression through this cycle. The backwards task was more difficult, so while he did use the spatial reasoning to determine where his mistake in the code was compared to where the mouse went wrong on the maze, this cycle he would wipe everything out past the mistake and rebuild, rather than just replace or remove a card like he did in the forward movement of the mouse.

This mode of laying cards in a pictorial representation was present in other students and tasks as well. Figure 13 shows Isabella (Grade 1) laying out her direction cards from Task 3 in a shape that somewhat represented the path of the mouse (left image). However, her organization proved to be difficult for her to know the order of the code, so she changed her organization to a linear representation (right image).

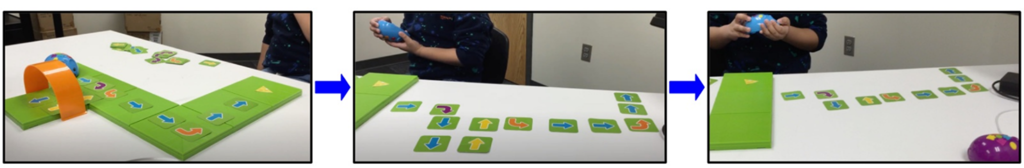


Figure 11. Jonathan (Grade 1) translating his concrete representation of the cards on the maze (left) to a pictorial path-like representation on the table (middle) and then correcting his code (right).

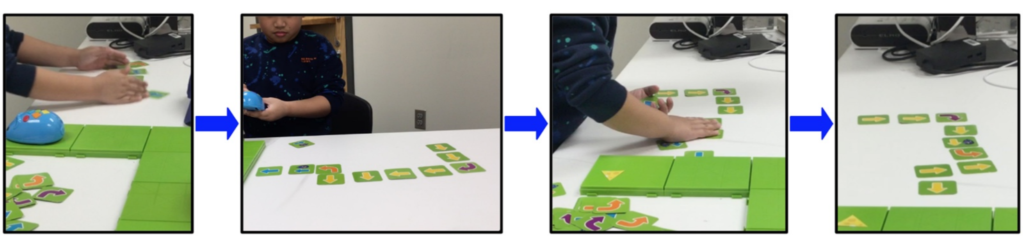


Figure 12. Jonathan working on the reverse code for Task 3. He chose to wipe the cards at the beginning (left image) and with each mistake (third image). This cycle continued until the code was completed.

Figure 13. Isabella representing her code in the shape of the path the mouse should travel (left) then linearly (right).

Language-based representations

This results section explores two key language-based representations that emerged during our analysis of students engaging in computational thinking (CT) tasks: self-talk and stories. We found that all students used self-talk to support their problem-solving processes and verbalize their understanding. Additionally, stories, both created and utilized by the students, served as a common approach to grasping CT concepts and reasoning through tasks. The following sections will delve deeper into these themes with specific examples showcasing how students leveraged self-talk and stories to develop their early understandings of CT.

Self-talk to get through a task

Throughout the tasks, we found that all students used language-based representations of self-talk to support early understandings of CT. Self-talk here is defined as the talk the students did unprompted by the researcher and not directed at the researcher. In the examples of self-talk, the students were frequently seen using self-talk as an intermediary representation to help them either make sense of a representation they were trying to use or to help them as a mediator between representations, such as from the concrete robot mouse to the pictorial representation of the arrow on the coding card. Across the cases of self-talk, the students were seen using this representational form as a way to persist in the task when they got stuck. For instance, Mallory (Grade 2) utilized self-talk effectively during a debugging process, where she verbalized her observations and thoughts, thus externalizing her problem-solving process. She recognized an anomaly in the sequence of her task, labeling it as "weird" and logically deducing that a piece might be missing. As she looked at two direction cards, the muffin → jam and the jam → muffin, she started searching for another direction card with jam on it: “Ok, that’s weird. Is there another jam over here?” Her methodical search for a solution, expressed through her dialogue with herself, highlighted her analytical approach to resolving the sequence challenge. Similarly, Henry (K) and October (1) employed self-talk while interacting with the robot mouse, reading aloud the instructions on the arrow cards or the colors on the arrows. This strategy likely helped them lessen the mental effort needed to convert these abstract representations into physical actions, demonstrating how verbalizing thoughts can play a crucial role in understanding and solving CT tasks.

Use of stories

All 12 of the students incorporated storytelling across the CT tasks, especially those involving the robot mouse in Tasks 3 and 5. During the embodied maze activity (Task 3), this was evident in how Charlie and Conner (Grade 2) physically acted like mice. Charlie pretended to sniff the cheese at the end, while Conner chirped in a playful imitation. Similarly, Sophie (K) added a narrative while controlling the paper mouse, saying, "He can really smell that cheese!" (shaking

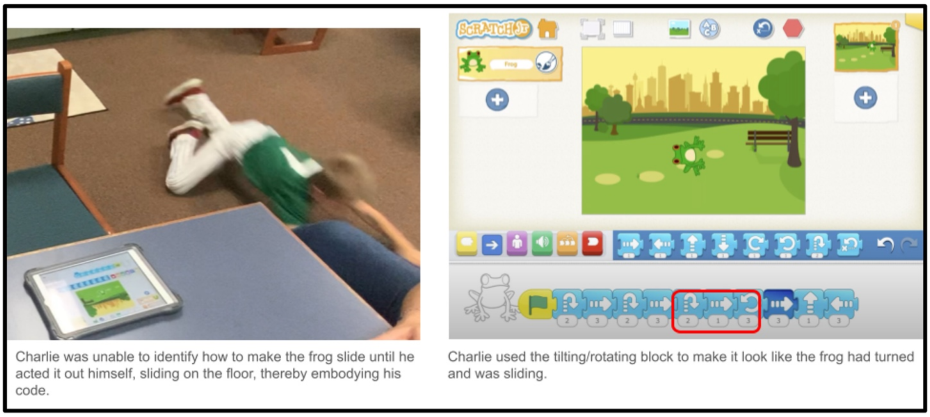


Figure 14. Charlie embodying the code by acting out the slide (left) and the code that matched that action in his ScratchJr code.

the mouse) and then adding sound effects as she moved it forward ("yum, yum, yum"). These examples showcase how storytelling and role-playing bridge the gap between abstract CT concepts and students' experiences, making learning more engaging.

This creative storytelling extended beyond explicit prompts. For instance, when tasked with programming a frog in ScratchJr (Task 7), Charlie (Grade 2) wasn't satisfied with a simple hop. His love for baseball inspired him to create a narrative where the frog would "slide into home-base" (See Figure 14). He even acted out the slide for the researcher, demonstrating his desire to incorporate this movement. This playful integration of narrative highlights how students can use stories to make CT tasks more meaningful and solve problems in unique ways.

In conclusion, these examples demonstrate the power of storytelling in supporting early CT development. Students across all ages and grades consistently used or created narratives to make sense of tasks and develop their CT understanding. Whether prompted by the activity or self-generated, these stories provided a context and structure that aided problem-solving, particularly in more complex sequencing tasks. This suggests that educational activities that encourage storytelling can be a valuable tool for nurturing early CT skills.

Discussion

Based on our study, several conclusions and recommendations can be drawn about integrating computational thinking (CT) in early elementary literacy and how to support competencies for all students. Although we initially sought out to explore differences between boys' and girls' experiences, our analysis across students and tasks indicated no significant gender-based differences in the ways that students engaged with and benefited from these computational thinking tasks. Both boys and girls demonstrated similar strengths and faced comparable challenges during the task-based interviews. This finding suggests that the instructional strategies we develop may not necessarily be gender-specific but should instead focus on addressing the common barriers and leveraging the affordances observed across all students at this level. Thus, our approach to shaping instruction should aim to universally enhance learning experiences and outcomes in computational thinking for every student, irrespective of gender.

However, it may also be that by using multiple representations (concrete manipulatives, pictorial/graphical, and language-based strategies), we were able to enhance inclusivity by accommodating diverse learning styles, which is particularly helpful for marginalized groups (e.g. Ryoo, 2019). Research suggests (Bati, 2022) that girls, along with other historically underrepresented groups in STEM, often benefit from more concrete and relatable instructional methods due to socialized differences in early education experiences. By integrating multiple forms of representation, these learners are better supported due to the following: (1) **concrete manipulatives** (e.g. coding toys) provide hands-on learning, which can reduce intimidation and increase engagement, particularly for students who may feel disconnected from abstract computational concepts (Bati, 2022), (2) **pictorial/graphical representations** help students visualize and plan their actions, thus bridging the gap between abstract concepts and tangible understanding. This is key for young learners, including girls, who may have lower confidence in STEM subjects compared to boys (Moore et al., 2020), and (3) **language-based strategies**, such as self-talk and storytelling, allow students to verbalize and structure their understanding. This cognitive tool can foster deeper engagement and confidence, particularly for students who benefit from narrative-driven learning, a method shown to engage girls in science and math (Cremin & Flewitt, 2016).

The integration of CT into early childhood education aligns with previous findings that highlight the effectiveness of unplugged activities combined with plugged activities in enhancing CT skills among young learners. For example, del Olmo-Muñoz et al. (2020) and Su and Yang (2023) emphasized that this combination was particularly effective, suggesting that your approach to using multiple representations such as concrete manipulatives, pictorial/graphical representations, and language-based strategies is well-supported by existing literature.

Multiple representations enhance understanding

When looking across the findings, there were some overarching trends in how the students engaged with and utilized multiple representations across the tasks that align with previous recommendations for engaging early learners with CT. Concrete manipulatives like coding cards and robots were seen to help our students grasp abstract CT concepts by allowing them to physically manipulate and visualize the problem-solving process, which aligns with recommendations for the importance of concrete experiences for younger learners (Bati, 2022), as well as the findings that suggest that coding with physical robots support spatial reasoning understandings (Berson et al., 2023; Brainin et al., 2022).

Similar to findings from Moore et al. (2020), pictorial/graphical representations that students used with the direction cards helped to bridge the gap between concrete and abstract elements, enabling students to conceptualize and plan their actions. Finally, language, particularly in storytelling and self-talk, was seen to play a crucial role in processing and articulating CT tasks, helping to serve as a cognitive tool that aids understanding and problem-solving (Cremin & Flewitt, 2016). While there were some general trends seen within representations, the overarching analysis across students and tasks revealed that individual students utilizing a diversity of approaches and representational styles when engaging with and solving the tasks. Although this was a small sample of students across K-2 grades, we believe our results showcase the importance of using varied representational modes.

Early childhood CT education has typically focused on developmentally appropriate practices and the use of broader conceptualizations like unplugged and plugged activities (e.g. Rich et al., 2019). However, our research underlines the importance of adopting a multi-modal representations approach to early childhood CT education. This broader perspective acknowledges the various ways children interact with and comprehend CT concepts, suggesting that a combination of concrete manipulatives, pictorial/graphical representations, and language-based approaches can provide a more holistic and developmentally appropriate learning experience. Both Wohl et al. (2015) and del Olmo-Muñoz et al. (2020) emphasized that integrating different modes, such as unplugged activities and computational toys, can help young learners grasp complex concepts by gradually moving from concrete to more abstract forms of representation. This supports our findings that the use of concrete manipulatives, pictorial/graphical representations, and language could be used as stepping stones in CT education. Furthermore, using multiple representations can lead to a more inclusive and effective CT education, catering to the diverse needs and preferences of a wide range of early elementary students (Moore et al., 2020). By doing so, educators can create a more engaging and supportive learning environment that fosters early interest and confidence in CS and STEM, thereby contributing to a broader and more diverse future in these fields.

Language as a cognitive tool and integrating CT with literacy

Language plays a pivotal role in learning computational thinking (CT) and in helping students process and articulate their understanding (Bers, 2019). The use of language has been shown to facilitate a deeper connection between students and the abstract concepts inherent in CT tasks (Macrides et al., 2022) and this was seen in our tasks through the use of stories and self-talk. We found that language and storytelling were significant in helping students articulate and process their computational thinking tasks. For instance, during a task with a robot mouse, a student named Sophia created a story where she navigated the mouse through a maze, using language to plot each step, such as moving forward, turning, and reversing. This storytelling approach allowed her to not only visualize and sequence the actions needed, but also helps in breaking down complex CT tasks into relatable and comprehendible parts, allowing students to make sense of and navigate through these tasks more effectively.

To leverage language effectively in teaching CT, educators can integrate literacy activities into the regular curriculum, using stories and dialogues as tools to introduce and explain CT concepts. This approach not only enhances the relatability and comprehensibility of CT but also promotes a more holistic educational experience where language and computation support and enrich each other (Bers, 2019). By doing so, teachers can create a learning environment where language is used strategically to reduce the complexity of CT concepts, helping students to organize their thoughts, solve problems, and express their understanding in a coherent and structured manner. By embedding CT concepts within familiar stories and personal narratives, students could better understand and articulate the steps involved in problem-solving, thereby enhancing their engagement and learning in computer science education.

Implications

Early introduction of CS concepts

Others have found that introducing CS concepts at the elementary level, especially through literacy integration, can lay the groundwork for future learning and interest in STEM fields (e.g. Martin-Hansen, 2018). Based on the results of the study, we found that integrating CT into literacy practices using multiple representations, especially language, could potentially bridge gaps in CS education and create a more inclusive and engaging learning environment for young students, particularly in the early grades.

Integrating CT with literacy and promoting language as a cognitive tool

Teachers should deliberately foster an environment where students are encouraged to use storytelling and self-talk as strategies to articulate and process their understanding of computational thinking (CT) concepts. By blending CT with literacy activities and intentionally guiding students to create narratives, educators can make abstract CT principles more tangible and relatable (Bers & Horn, 2010). This approach not only aids in externalizing students' thought processes but also makes problem-solving more manageable. Utilizing digital storytelling tools like ScratchJr allows students to integrate programming elements into their narratives, enriching their learning experience and helping them to develop a deeper understanding of CT through language and narrative construction.

Multiple representations as an inclusive approach

Teachers should use a variety of representational modes, including concrete manipulatives (like coding cards and robots), pictorial/graphical representations, and language-based instructions. This approach caters to different learning styles and helps in making CT concepts more accessible for all students.

However, despite the importance of CS education, disparities in participation persist. Computer science education often excludes marginalized groups, including female students and those from underrepresented racial and socioeconomic backgrounds (Cvencek et al., 2011; Kucuk & Sisman, 2020). Without interventions that address these disparities, many students lose interest and confidence in CS early on, which perpetuates a lack of diversity in the field (Shah & Yadav, 2023).

The findings of this study highlight how using multiple representations, such as concrete manipulatives and pictorial/graphical tools, can provide an inclusive learning experience that fosters early engagement in CS for marginalized groups. By offering varied entry points into CS learning, these approaches can help address existing disparities by ensuring that all students, regardless of their background, have access to developmentally appropriate and engaging CT instruction (Cremin & Flewitt, 2016; Moore et al., 2020).

Introduce CT at early grades

Introducing computational thinking (CT) concepts at early grade levels is crucial for building foundational skills and fostering interest in STEM fields. Making CT relevant and engaging by connecting it to students' everyday experiences and interests can demystify computer science and make it more accessible to all students. Our observations show that there are no significant differences in engagement with CT between boys and girls at these ages, yet we know from other research that as students progress in their education, these differences appear. This highlights the importance of starting CT education early and ensuring an inclusive learning environment. By doing so, teachers can provide equal access and opportunities for every student to benefit from CT education, thus supporting the development of CT skills in young learners and enhancing their interest in pursuing STEM education and careers in the future.

Acknowledgements

For blinded purposes, there will be an acknowledgment of the project funder and for some of the images in a published version of this paper.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

Google's Computer Science Educational Research Grant 2021 and the National Science Foundation, Awards: #2122785, #2122736, #2122742.

Notes on contributors

Kristina M. Tank is an Associate Professor of Science Education at Iowa State University, specializing in STEM education and engineering integration in K-12 classrooms. Her research focuses on supporting teachers in implementing project-based learning and integrating engineering practices into science instruction to enhance student learning and engagement.

Tamara J. Moore is a Professor of Engineering Education at Purdue University, with expertise in integrating engineering concepts into K-12 STEM education. Her research focuses on developing and implementing engineering-based curricular materials, particularly through project-based learning, to support teachers in enhancing students' problem-solving and critical thinking skills. She is also recognized for her work in creating interdisciplinary approaches to STEM education.

Anne Ottenbreit-Leftwich is a Professor of Instructional Systems Technology at Indiana University, specializing in the integration of technology into K-12 education. Her research focuses on supporting preservice and inservice teachers in using technology to enhance teaching and learning, with an emphasis on computational thinking and AI education.

Zarina Wafula is PhD student at Iowa State University, specializing in educational technology and the integration of computer science into early elementary education. Her work focuses on developing and testing curriculum materials that combine computational thinking and literacy, with a particular emphasis on activities for K-2 students that foster foundational skills like sequencing and algorithm development.

Lin Chu is a doctoral student in the Instructional Systems Technology program at Indiana University. Her research focuses on Computer Science Education, Computational Thinking, and Data Science. With a background in children's English education from Taiwan, she is committed to advancing technology education and instructional design, particularly in K-12 environments.

Sohheon Yang is a PhD student in Instructional Systems Technology at Indiana University. Her research interests include technology integration in early childhood education, STEAM education, and equitable learning, with a focus on using technology to improve educational outcomes for young children. She has a background in early childhood education from South Korea and actively works on projects like integrating computational thinking into K-2 literacy programs.

ORCID

Kristina M. Tank http://orcid.org/0000-0001-9011-2273
Tamara J. Moore http://orcid.org/0000-0002-7956-4479
Anne Ottenbreit-Leftwich http://orcid.org/0000-0002-4055-314X
Zarina Wafula http://orcid.org/0000-0002-2771-0016
Lin Chu http://orcid.org/0000-0002-4148-8554
Sohheon Yang http://orcid.org/0009-0003-8339-696X

References

Bati, K. (2022). A systematic literature review regarding computational thinking and programming in early childhood education. *Education and Information Technologies*, *27*(2), 2059–2082. https://doi.org/10.1007/s10639-021-10700-2

Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, *6*(4), 499–528. https://doi.org/10.1007/s40692-019-00147-3

Bers, M. U., & Horn, M. S. (2010). Tangible programming in early childhood: Revisiting developmental assumptions through new technologies. In I. R. Berson & M. J. Berson (Eds.), *High-tech tots: Childhood in a digital world* (pp. 49–70). Information Age Publishing. https://tidal.northwestern.edu/media/files/pubs/Bers-Horn_May1809.pdf

Berson, I. R., Berson, M. J., McKinnon, C., Aradhya, D., Alyaeesh, M., Luo, W., & Shapiro, B. R. (2023). An exploration of robot programming as a foundation for spatial reasoning and computational thinking in preschoolers' guided play. *Early Childhood Research Quarterly*, *65*, 57–67. https://doi.org/10.1016/j.ecresq.2023.05.015

Brainin, E., Shamir, A., & Eden, S. (2022). Robot programming intervention for promoting spatial relations, mental rotation and visual memory of kindergarten children. *Journal of Research on Technology in Education*, *54*(3), 345–358. https://doi.org/10.1080/15391523.2020.1858464

Bruner, J. S. (1966). *Toward a theory of instruction*. Harvard University Press.

Clement, J. (2000). Analysis of clinical interviews: Foundations and model viability. In A. E. Kelly, & R. A. Lesh (Eds.), *Handbook of research design in mathematics and science education* (pp. 341–385). Routledge.

Code.org, CSTA, & ECEP Alliance. (2023). *2023 State of Computer Science Education*. https://advocacy.code.org/stateofcs

Corbin, J., & Strauss, A. (2014). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage publications.

Cremin, T., & Flewitt, R. (2016). Laying the foundations: Narrative and early learning. In T. Cremin, R. Flewitt, B. Mardell, & J. Swann (Eds.), *Storytelling in early childhood: Enriching language, literacy and classroom culture* (pp. 13–28). Routledge. https://doi.org/10.4324/9781315679426

Cvencek, D., Meltzoff, A. N., & Greenwald, A. G. (2011). Math–gender stereotypes in elementary school children. *Child Development*, *82*(3), 766–779. https://doi.org/10.1111/j.1467-8624.2010.01529.x del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of primary education. *Computers & Education*, *150*, 103832. https://doi.org/10.1016/j.compedu.2020.103832

Delacruz, S. (2020). Starting from Scratch (Jr.): Integrating code literacy in primary grades. *The Reading Teacher*, *73*(6), 805–812. https://doi.org/10.1002/trtr.1909

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, *63*, 87–97. https://doi.org/10.1016/j.compedu.2012.11.016

Flewitt, R. (2016). Equity and diversity through story: A multimodal perspective. In T. Cremin, R. Flewitt, B. Mardell, & J. Swann (Eds.), *Storytelling in early childhood: Enriching language, literacy and classroom culture* (pp. 150–168). Routledge. https://doi.org/10.4324/9781315679426

Gao, X., & Hew, K. F. (2022). Toward a 5E-based flipped classroom model for teaching computational thinking in elementary school: Effects on student computational thinking and problem-solving performance. *Journal of Educational Computing Research*, *60*(2), 512–543. https://doi.org/10.1177/07356331211037757

Goldin, G. A. (2000). A scientific perspective on structured, task-based interviews in mathematics education research. In A. E. Kelly, & R. A. Lesh (Eds.), *Handbook of research design in mathematics and science education* (pp. 517–545). Routledge.

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, *82*, 263–279. https://doi.org/10.1016/j.compedu.2014.11.022

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, *41*(4), 245–255. https://doi.org/10.1007/s10643-012-0554-5

Kind, A. (2017). Computing attitudes: Will teaching 2nd grade students computer science improve their self-efficacy and attitude and eliminate gender gaps. *Computer Science and Self-Efficacy*, *8*, 1–34. https://www.smcm.edu/mat/wp-content/uploads/sites/73/2015/06/Amanda-Kind-2015.pdf

Kjällander, S., Mannila, L., Åkerfeldt, A., & Heintz, F. (2021). Elementary students' first approach to computational thinking and programming. *Education Sciences*, *11*(2), 80. https://doi.org/10.3390/educsci11020080

Kucuk, S., & Sisman, B. (2020). Students' attitudes towards robotics and STEM: Differences based on gender and robotics experience. *International Journal of Child-Computer Interaction*, *23–24*, 100167. https://doi.org/10.1016/j.ijcci.2020.100167

Lei, R. F., Green, E. R., Leslie, S., & Rhodes, M. (2019). Children lose confidence in their potential to "be scientists," but not in their capacity to "do science. *Developmental Science*, *22*(6), 1–8. https://doi.org/10.1111/desc.12837

Lesh, R., & Doerr, H. M. (2003). Foundations of a models and modeling perspective on mathematics teaching, learning, and problem solving. In R. Lesh & H. M. Doerr (Eds.), *Beyond constructivism: Models and modeling perspectives on mathematics problem solving, learning, and teaching* (pp. 3–34). Lawrence Erlbaum.

Lin, C., Liu, E., & Huang, Y. (2012). Exploring parents' perceptions towards educational robots: Gender and socio-economic differences. *British Journal of Educational Technology*, *43*(1), E31–E34. https://doi.org/10.1111/j.1467-8535.2011.01258.x

Macrides, E., Miliou, O., & Angeli, C. (2022). Programming in early childhood education: A systematic review. *International Journal of Child-Computer Interaction*, *32*, 100396. https://doi.org/10.1016/j.ijcci.2021.100396

Maher, C. A., & Sigley, R. (2020). Task-based interviews in mathematics education. In S. Lerman, (Ed,), *Encyclopedia of Mathematics Education* (pp. 821–824). Springer. https://doi.org/10.1007/978-3-030-15789-0_147

Malik, S. I., Shakir, M., Eldow, A., & Ashfaque, M. W. (2019). Promoting algorithmic thinking in an introductory programming course. *International Journal of Emerging Technologies in Learning (iJET)*, *14*(01), 84–94. https://doi.org/10.3991/ijet.v14i01.9061

Margulieux, L. E. (2019). Spatial encoding strategy theory: The relationship between spatial skill and STEM achievement. In R. McCartney, A. Petersen, A. Robins, & A. Moskal, (Eds.), *Proceedings of the Fifteenth Annual Conference on International Computing Education Research* (pp. 81–90). ACM. https://doi.org/10.1145/3291279.3339414

Martin-Hansen, L. (2018). Examining ways to meaningfully support students in STEM. *International Journal of STEM Education*, *5*(1), 53. https://doi.org/10.1186/s40594-018-0150-3

Master, A., Meltzoff, A. N., & Cheryan, S. (2021). Gender stereotypes about interests start early and cause gender disparities in computer science and engineering. *Proceedings of the National Academy of Sciences*, *118*(48), 1–7. https://doi.org/10.1073/pnas.2100030118

Master, A., Tang, D., Forsythe, D., Alexander, T. M., Cheryan, S., & Meltzoff, A. N. (2023). Gender equity and motivational readiness for computational thinking in early childhood. *Early Childhood Research Quarterly*, *64*(3), 242–254. https://doi.org/10.1016/j.ecresq.2023.03.004

Master, A., & Walton, G. M. (2013). Minimal groups increase young children's motivation and learning on group-relevant tasks. *Child Development*, *84*(2), 737–751. https://doi.org/10.1111/j.1467-8624.2012.01867.x

Metz, S. S. (2007). Attracting the engineering of 2020 today. In R. J. Burke, M. C. Mattis, & E. Elgar (Eds.), *Women and minorities in science, technology, engineering and mathematics: Upping the numbers* (pp. 184–209). Edward Elgar Publishing.

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

National Science Board. (2020). *Vision 2030*. National Science Foundation. https://www.nsf.gov/nsb/publications/2020/nsb202015.pdf

Relkin, E., de Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers & Education*, *169*, 104222. https://doi.org/10.1016/j.compedu.2021.104222

Rich, K. M., Yadav, A., & Schwarz, C. V. (2019). Computational thinking, mathematics, and science: Elementary teachers' perspectives on integration. *Journal of Technology and Teacher Education*, *27*(2), 165–205. https://www.learntechlib.org/p/207487/

Ryoo, J. J. (2019). Pedagogy that supports computer science for all. *ACM Transactions on Computing Education*, *19*(4), 1–23. https://doi.org/10.1145/3322210

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

Shah, N., & Yadav, A. (2023). Racial justice amidst the dangers of computing creep: A dialogue. *TechTrends*, *67*(3), 467–474. https://doi.org/10.1007/s11528-023-00835-z

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

Su, J., Yang, W., & Zhong, Y. (2023). Influences of gender and socioeconomic status on children's use of robotics in early childhood education: A systematic review. *Early Education and Development*, *34*(4), 910–926. https://doi.org/10.1080/10409289.2022.2078617

Tank, K. M., Ottenbreit-Leftwich, A., Moore, T. J., Yang, S., Wafula, Z., Kim, J., Fagundes, B., & Chu, L. (2024). Investigating sequencing as a means to computational thinking in young children. *International Journal of Computer Science Education in Schools*, *6*(3), 67–77.

Tissenbaum, M., & Ottenbreit-Leftwich, A. (2020). A vision of K-12 computer science education for 2030. *Communications of the ACM*, *63*(5), 42–44.

Tissenbaum, M., Weintrop, D., Holbert, N., & Clegg, T. (2021). The case for alternative endpoints in computing education. *British Journal of Educational Technology*, *52*(3), 1164–1177. https://doi.org/10.1111/bjet.13072

Tsortanidou, X., Daradoumis, T., & Barberá, E. (2021). A K-6 computational thinking curricular framework: Pedagogical implications for teaching practice. *Interactive Learning Environments*, *31*(8), 4903–4923. https://doi.org/10.1080/10494820.2021.1986725

Vakil, S. (2018). Ethics, identity, and political vision: Toward a justice-centered approach to equity in computer science education. *Harvard Educational Review*, *88*(1), 26–52. https://doi.org/10.17763/1943-5045-88.1.26

Valla, J. M., & Williams, W. (2012). Increasing achievement and higher-education representation of under-represented groups in science, technology, engineering, and mathematics fields: A review of current K-12 intervention programs. *Journal of Women and Minorities in Science and Engineering*, *18*(1), 21–53. https://doi.org/10.1615/JWomenMinorScienEng.20

Wohl, B., Porter, B., & Clinch, S. (2015). Teaching computer science to 5-7 year-olds: An initial study with Scratch, Cubelets and unplugged computing. In J. Gal-Ezer, S. Sentance, & J. Vahrenhold (Eds.), *Proceedings of the workshop in primary and secondary computing education* (pp. 55–60). Association for Computing Machinery. https://doi.org/10.1145/2818314.2818340