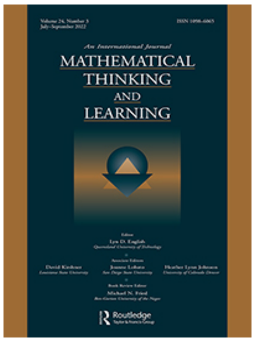

Mathematical Thinking and Learning

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/hmtl20

Kindergarten students’ mathematics knowledge at work: the mathematics for programming robot toys

Jessica F. Shumway, Lise E. Welch, Joseph S. Kozlowski, Jody Clarke-Midura & Victor R. Lee

To cite this article: Jessica F. Shumway, Lise E. Welch, Joseph S. Kozlowski, Jody Clarke-Midura & Victor R. Lee (2021): Kindergarten students’ mathematics knowledge at work: the mathematics for programming robot toys, Mathematical Thinking and Learning, DOI: 10.1080/10986065.2021.1982666

To link to this article: https://doi.org/10.1080/10986065.2021.1982666

Published online: 29 Sep 2021.

MATHEMATICAL THINKING AND LEARNING
https://doi.org/10.1080/10986065.2021.1982666

Routledge
Taylor & Francis Group

Kindergarten students' mathematics knowledge at work: the mathematics for programming robot toys

Jessica F. Shumway [a], Lise E. Welch [a], Joseph S. Kozlowski [a], Jody Clarke-Midura [b], and Victor R. Lee [c]

[a]School of Teacher Education and Leadership, Utah State University, Logan, Utah, USA; [b]Department of Instructional Technology and Learning Sciences, Utah State University, Logan, Utah, USA; [c]Graduate School of Education, Stanford University, Stanford, California, USA

ABSRACT

The purpose of this study was to explore how kindergarten students (aged 5–6 years) engaged with mathematics as they learned programming with robot coding toys. We video-recorded 16 teaching sessions of kindergarten students' (N = 36) mathematical and programming activities. Students worked in small groups (4–5 students) with robot coding toys on the floor in their classrooms, solving tasks that involved programming these toys to move to various locations on a grid. Drawing on a semiotic mediation perspective, we analyzed video data to identify the mathematics concepts and skills students demonstrated and the overlapping mathematics-programming knowledge exhibited by the students during these programming tasks. We found that kindergarten children used spatial, measurement, and number knowledge, and the design of the tasks, affordances of the robots, and types of programming knowledge influenced how the students engaged with mathematics. The paper concludes with a discussion about the intersections of mathematics and programming knowledge in early childhood, and how programming robot toys elicited opportunities for students to engage with mathematics in dynamic and interconnected ways, thus creating an entry point to reassert mathematics beyond the traditional school content and curriculum.

ARTICLE HISTORY
Received 11 January 2021
Revised 16 September 2021
Accepted 16 September 2021

KEYWORDS
Elementary mathematics; technology; elementary programming; spatial reasoning; measurement; number; semiotic mediation

Introduction and aim

Technology changes the way we live and invariably affords new ways of using mathematics. Kaput et al. (2007) argued the importance of studying underlying principles about the role of technology in mathematics education in order to leverage their connections and provide students with new kinds of mathematics experiences. For example, one key force pertinent to technology use is the shift from static to dynamic forms of mathematical interactions (Roschelle et al., 2017). These kinds of dynamic mathematical interactions often rely on movement in space and the counting or measurement of continuous quantities. A context for investigating the role of technology in fostering these kinds of dynamic mathematical interactions is computer programming (Clements & Battista, 1989; Francis et al., 2016; Noss & Hoyles, 1996; Papert, 1972). The practices and thinking processes across mathematics and programming seem to be "natural companions" (English, 2018, p. 2; Weintrop et al., 2016). Perhaps even more fundamentally, programming infuses new forms of symbolic representation into mathematical learning activities. Some would go so far as to argue that the nature of this infusion is not a call for additional symbolic mastery. Rather, it changes the very foundations of how mathematics is manifested and representationally embodied (Wilensky & Papert, 2010). By introducing programming

CONTACT Jessica F. Shumway jessica.shumway@usu.edu School of Teacher Education and Leadership, Utah State University, Logan, Utah, USA

as a symbolic system, how one engages with disciplinary knowledge changes (Sherin, 2001). The technology emerging from computer science and programming education has important implications for how educators and researchers might rethink mathematics teaching and learning for students, such as Google's Blockly for learning group theory (Gadanidis et al., 2018) and LEGO Mindstorms EV3 robots for fostering spatial reasoning (Francis et al., 2016). Hence, programming and its technologies are an important context for pursuing the study of key forces for technology's role in mathematics learning, and in particular, their potential for fostering dynamic mathematical interactions and role in mediating students' learning of mathematics concepts and skills.

While there is evidence that programming in mathematics education is an important context for studying technology's potential to enhance traditional mathematics learning goals, it is also an important context for understanding some of the challenges when integrating technology in mathematics teaching. Hickmott et al. (2018) conducted a scoping review on studies that link computational thinking – a thinking process associated with programming – to the learning of mathematics. They found that the potential of integrating programming to improve mathematics experiences and learning is unclear. Their results indicated that *non-empirical* studies explicitly linking the learning of mathematics concepts to computational thinking were common (e.g., potential mappings between mathematics and computer science domains), but *empirical* studies that explicitly link mathematics and computational thinking were rare. They suggested a need to move beyond the incidental linkages between mathematics and computational thinking to more effectively deliver integrated curricula in elementary schools. A first step in doing this is for mathematics educators on interdisciplinary research teams to empirically identify the mathematics that occurs as students engage in computational thinking activities, thus explicitly linking the mathematics and computational thinking.

Some recent empirical studies on programming technologies and mathematics learning in early childhood are doing just that. Specifically, early childhood researchers have investigated the use of programmable robot toys and young children's development of spatial visualization (Bartolini Bussi & Baccaglini-Frank, 2015; Palmér, 2017), effective settings to promote problem solving and computational thinking (Sung et al., 2017), and the use of number skills and spatial orientation when programming (Fessakis et al., 2013).

Our study is positioned in this emerging body of research examining the tangible and dynamic features of robot coding toys for young children (e.g., Hamilton et al., 2020, see Figure 1) and these toys' potential for mediating mathematics knowledge with or within programming tasks and computational thinking contexts. The overall goal of our study was to identify the ways kindergarten students (ages 5–6 years) engaged with mathematics while learning to program with robot coding toys in their classrooms. Considering the premise that new symbolic systems yield new ways of recognizing mathematics (Wilensky & Papert, 2010), a perspective that recognizes how symbols mediate learning is necessary. The present study contributes to the field by investigating the role of programming technology in mediating students' learning of mathematics concepts and skills (Kaput et al., 2007) and by providing empirical evidence from kindergarten classrooms that illustrate the connections or lack of connections

Botley	Cubetto	Bee Bot	Code-a-pillar
			

Figure 1. A sample of commercially available, tangible, and screen-free robot coding toys.

between mathematics and programming concepts and skills (Hickmott et al., 2018). Although codable robot toys for young children may be primarily used to make aspects of computational thinking more accessible, there are ways in which we expect mathematical ideas to appear in distinguishable ways.

In the following sections, we first situate our research in the theoretical framework of semiotic mediation. Next, we describe the empirical literature we drew upon to delineate the mathematical concepts and programming indicators we claim as evidence of the children's engagement with mathematics. Then, we present the results of our analysis and conclusions.

Theoretical framework

Grounded in a Vygotskian approach of semiotic mediation, Bartolini Bussi and Mariotti (2008) theory of semiotic mediation for mathematics classrooms provides a lens for examining students' processes of using an artifact to accomplish a task and the semiotic potential of the artifact to accomplish a didactic mathematical objective (provided by the teacher). In the process of using an artifact, students and teachers generate *signs* – e.g., language, gestures, body movement, symbols – and *semiotic bundles* – related multimodal signs or systems of signs (Arzarello et al., 2009). Bartolini Bussi and Mariotti argued that "what needs to be emphasized and better explained is the link between signs and the content to be mediated and the way in which these links can be exploited in an educational perspective" (p. 752).

In the present study, we focused on kindergarten students' processes of doing and using mathematics while solving programming tasks. These activities were centered on using an artifact (i.e., robot coding toys) to understand the links between the artifact and the content (i.e., mathematics and programming) being mediated. Bartolini Bussi and Mariotti described the *semiotic potential* of an artifact as the system of relationships among the artifact, task, and content, and is that which must be "constructed and exploited" within students' signs. In the context of our study, students' signs included their language about the movement of the robots, use of the program symbols (i.e., arrows or tiles to symbolize movements), hand gestures, body movements, and gaze among the central artifact (robot coding toy) and its related components (e.g., adventure grid, programming board related to the task).

Figure 2 shows an adapted version of Bartolini Bussi and Mariotti's diagram that illustrates the main processes involved in semiotic mediation, as they relate to our study. The artifacts are the coding robot toys (Cubetto and Botley; center in the figure) that students were directed to use to solve a task (programming task; upper left in the figure). The content to be mediated in our study are the mathematical concepts and skills and the ways they overlap or are linked to programming concepts and skills (lower left in the figure). The teacher (cultural mediator between mathematical concepts/skills and signs) designed an artifact-centered programming task (i.e., a programming task with a coding robot toy), and hence, knows the mathematics concepts and skills that could be mediated by the artifact due to the mathematical nature of the coding toy (i.e., moves in precise, measured, countable movements across a Cartesian grid space). While engaging with the tasks, the kindergarten students construct *artifact signs* or personal meaning through use of the artifact, which are rooted in the robot toy context (upper right in the figure). The teacher then guides the evolution of these signs from *artifact signs* to *mathematical and programming signs*, due to their position of knowing the mathematics/programming knowledge that can be mediated by the artifact (lower right in the figure). *Pivot signs* occur between the artifact signs and mathematics signs, fostering meaning between the context of artifact to the context of mathematics and programming. We were interested in students' semiotic activity and the evolution of signs representing their use of mathematics knowledge and connections between programming and mathematics knowledge.

What is involved in programming?: defining computational thinking and its constructs

Elementary computer science standards include programming skills such as using symbols to represent data, using software that controls computational devices (e.g., robots), and sequencing codes (Computers Science Teachers Association [CSTA], 2017). Central to these standards is *computational thinking* (CT), defined as "the conceptual foundation required to solve problems effectively and

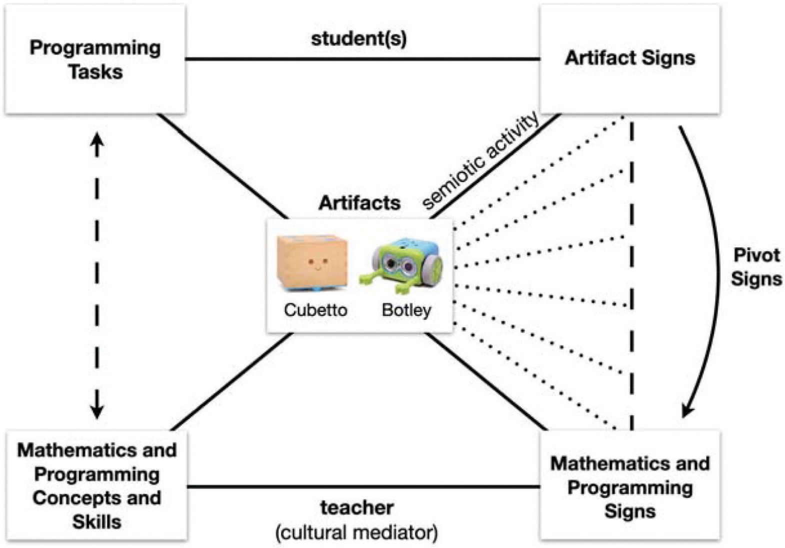

Figure 2. A diagram of the links among the elements in the theory of artifacts as a tool of semiotic mediation (adapted from Bartolini Bussi & Mariotti, 2008; Bartolini Bussi & Baccaglini-Frank, 2015).

efficiently . . . with solutions that are reusable in different contexts" (Shute et al., 2017, p. 151). While the field has yet to reach consensus for the definition and operationalization of CT, there are a variety of frameworks that attempt to identify and categorize the components included in CT (e.g., Brennan & Resnick, 2012; Grover & Pea, 2013; Shute et al., 2017; Weintrop et al., 2016; Wing, 2006). Shute et al. (2017) identified decomposition, abstraction, algorithms, and debugging as the components of CT that occurred most frequently in the research literature.

Algorithmic thinking, debugging, and decomposition were of particular interest for our study. In the context of our work in kindergarten classrooms, we define *algorithmic thinking* as developing and using logical and ordered sequences of instructions for directing an agent to take certain actions (Clarke-Midura et al., 2021). In our context with coding toys, algorithmic thinking included learning how to use codes to program a robot, planning a path and program, building a program, reading a program, and running a program with tiles or arrows (codes). Debugging and decomposition are processes related to algorithmic thinking that emerge as young children engage in tasks with these coding toys (e.g., Lavigne et al., 2020; Rehmat et al., 2020; Wang et al., 2020). *Debugging* is the "use of knowledge of syntax and written programs to recognize an incorrect program, locate the specific error, and fix the program (or bug)" (Clarke-Midura et al., 2021). *Decomposition* in computer science is similar to decomposing numbers, shapes, or problems in mathematics. It involves breaking a problem into smaller, more manageable components and considering the part-whole relationships.

While there are claims that CT is broad and spans ways of thinking within and outside of computer science, we view programming as an important context for developing CT (Bers et al., 2019). CT, computer science, and programming instruction are becoming more common in K-12 classrooms. Programming and CT in early childhood classrooms look different than in secondary settings, and researchers are studying ways to teach these topics to young children with tangible block-based or visual coding toys (e.g., Angeli & Valanides, 2020; Muñoz-Repiso & Caballero-González, 2019).

Connections between mathematics and programming

Seminal research about the connections between mathematics and programming established precedents about what mathematics is possible with coding technology (Clements & Battista, 1989; Clements et al., 1996; Cuneo, 1986; Noss & Hoyles, 1996; Papert, 1972; Sherin, 2001; Wilensky &

Papert, 2010). Since then, coding tools have changed substantially, and there is renewed interest in the related concept of CT (Wing, 2006). Research has highlighted the overlapping thinking processes in CT and mathematical thinking (e.g., diSessa, 2018; Weintrop et al., 2016), the dispositional synergies and habits of mind between these two ways of thinking (Pei et al., 2018; Pérez, 2018), how to build on connections between mathematics learning and CT (English, 2017, 2018), the integration of mathematics and CT curriculum standards (e.g., Israel & Lash, 2020; Rich et al., 2019), and whether or not interventions can improve both programming and mathematical skills (e.g., Sung & Black, 2020).

Mathematics and programming in early childhood classrooms

In the early childhood research literature on mathematics and programming, studies show promising outcomes for young children's mathematics learning in CT and programming contexts (Fessakis et al., 2013; Miller, 2019; Palmér, 2017; Sung et al., 2017; Sung & Black, 2020). Some research studies leveraged young children's existing mathematics knowledge to explore their CT learning (Lavigne et al., 2020), while other studies investigated ways programming activities might support mathematical knowledge (Miller, 2019). For example, Lavigne et al. (2020) developed non-coding tasks that were based on preschoolers' existing mathematics knowledge to investigate how teachers and preschool students interacted with the CT activities sequencing, modularity, and debugging. Conversely, Miller (2019) studied ways that Year 2 students (ages 7 and 8) demonstrated mathematical thinking (patterning and structures) as they engaged in programming tasks. She assigned students to a control group (typical mathematics instruction) or an intervention group, which received three 45-minute sessions with Scratch (a block-based programming application) and three sessions with a coding robot. Results indicated that students in the intervention group made significant mathematical learning gains when compared to the control group, indicating the potential of coding as a context for students to see mathematical patterns and structures in codes.

The design of the activities in these research studies determined the mathematics learning that occurred. Sung et al. (2017) engaged kindergarten and first-grade students in number-line oriented activities within four conditions using two settings: full-embodied CT activities and Scratch Jr. (a block-based coding program on iPads) activities. Their results indicated that the activities benefited both mathematics and programming learning, as students engaged in computational perspectives while enhancing their numeracy, magnitude comparison, and number line estimation. Palmér's (2017) study engaged preschool children in guided interaction with the Bee-Bot coding robot over three to four weeks. The spatial nature of the activity engaged children in decomposition and sequencing while also exploring spatial thinking, counting, and symbols (arrows). Fessakis et al. (2013) aimed to evaluate the learning value of a computer programming activities with kindergarten students using path and maze activities with a Logo-like ladybug applet on the computer. They found that the programming activities offered learning in counting, orientation skills, and angle turn concepts along with programming skills such as command execution and debugging. Both Palmér and Fessakis et al.'s studies reference spatial thinking. The spatial aspects of coding-toy and Logo-like activities appear to be a promising point of connection between mathematics and programming (Clements & Battista, 1989; Sarama & Clements, 2009). Spatial reasoning is an important thinking process in mathematics and tends to be overlooked in U.S. kindergarten curriculum (Uttal et al., 2013). Kindergarten students are expected to use relational language, such as above and next to, to describe the position of objects in space (Common Core State Standards Initiative [CCSSI], 2010), but spatial reasoning involves a coordination of relational language, spatial orientation, spatial relations, and spatial visualization (manipulating shapes mentally; Davis & The Spatial Reasoning Study Group, 2015; Sarama & Clements, 2009).

Overall, three broad mathematics domains emerge in the literature on early childhood mathematics and programming: spatial, measurement, and number concepts and skills. While these findings provide evidence of mathematics learning in specific programming environments, there is a need for further research on the ways young children use their school mathematics learning with technologies, such as tangible coding toys, in classroom settings. This is especially important considering many commercial toys made for young children involve moving a robot on an implicit grid, which could present apt opportunities for spatial reasoning (Francis et al., 2016; Palmér, 2017) and early number line concepts (Sung et al., 2017). Kindergarten in the U.S. marks the beginning of children's formal mathematics instruction, which makes it a particularly interesting age group to study the intersections between school mathematics and technology-mediated mathematics.

The focus of the present study is on the mathematics concepts and skills that kindergarten students use during programming tasks with robot coding toys, and in what ways their mathematics and programming knowledge overlap. We examined students' semiotic activity and its relation to the tasks, artifacts, and content of programming and mathematics. The research questions guiding our inquiry were:

(1) What mathematics concepts and skills do kindergarten students engage with during robot coding toy tasks and how are these concepts and skills demonstrated?
(2) How do mathematical and programming concepts and skills overlap as students solve robot coding toy tasks?

Methods

Participants and setting

The participants were 36 kindergarten students (ages 5–6 years; 15 female and 21 male) from two classrooms within the same public elementary school in the Intermountain West. The demographics of the school included 70% White, 20% Hispanic/Latinx, 7% Asian, 2% Mixed Race/Other, and 1% Black. Almost half (46%) of students in the school were eligible for free or reduced lunch, indicating low-socioeconomic households.

The two kindergarten teachers were experienced early childhood educators and had committed to participating in the larger, ongoing design-based research project with our research team. The teachers conducted Science, Technology, Engineering, Art, and Mathematics (STEAM) rotations every Friday that lasted 30 minutes. Five stations were set up around each room and small groups of 4–5 students (8 groups total) rotated to a different station every Friday. Some stations integrated STEAM subjects while other stations focused on only one STEAM discipline. Each classroom contained a Technology Station, which was facilitated by members of the research team. For the purpose of the present study, two researchers, both former elementary teachers, taught at the Technology Station (one in Classroom A and one in Classroom B). Each researcher-teacher taught groups of students two times over 10 weeks. They taught the same two tasks (one with a Cubetto robot and one with a Botley robot, which are described below) across the different groups each week so that every student had an opportunity to engage in two robot coding toy activities with the researcher-teacher. Prior to engaging in the researchers' two lessons with robot coding toys, the students had previously engaged in unplugged activities (e.g., sequencing Going on a Bear Hunt, "programming" each other to move in their classrooms), board games (e.g., Robot Turtles by ThinkFun), and an activity with Terrapin's BeeBot robot.

Materials: artifacts and tasks

We designed tasks around two screen-free commercial coding toys, Botley and Cubetto (see Figure 1), which have the ability to move forward and backward as well as rotate 90 degrees right and left. The tasks centered on building and debugging programs to move the robots from a beginning location to a stopping location on a grid. The main activities structuring the didactical cycle included activities with the artifacts, individual and collective production of signs (e.g., gestures, symbols in the form of the arrows on the programming board, language), and discussion. What follows is a summary of each robot toy and its accompanying task.

Cubetto robot toy

Primo Toys manufactures Cubetto for children aged three and older. This robot is a rectangular prism with motorized wheels at its base. The toy has a programing board in which accompanying tiles, each representing a unique command, can be placed to build a program. A program is enacted/run by placing the desired tiles on the programming board, then pressing the "go" button. Cubetto operates on a fabric grid of square rows and columns with colorful and engaging pictures in various squares, such as a castle, desert sand, and grass. We used these images to dictate start and end points for Cubetto's "adventures." The length of each square corresponds to the distance Cubetto travels in one linear movement (15 centimeters).

Cubetto task. The researcher-created Cubetto task was titled *Following Directions and Building a Program*. The objective was to program Cubetto to get from one place to another. For example, one of the tasks shown in Figure 3 was to program Cubetto to move from the tree to the desert (from A to B). Students fit coding tiles (C) into the programming board (D) to complete the challenge. Upon completion of each challenge, the students were presented a slightly more complex path and its associated adventure story.

Botley robot toy

Botley the Coding Robot is produced by Learning Resources and designed for children ages five and older. This robot is a rectangular prism with two motorized wheels. Botley is controlled by entering commands into an external remote. Once commands have been entered, a green button is used to execute/run the program. Included with the Botley toy are puzzle pieces which may be assembled to form a grid or path. The length of each puzzle piece is equivalent to one of Botley's linear movements (20 centimeters).

Figure 3. Following directions and building a program activity for Cubetto.

Botley task. The researcher-created Botley task was titled *Crack the Code* (see Figure 4). The objective was to program one robot (Botley 1) to match the path of another preprogrammed robot (Botley 2). For example, the teacher-researcher programmed Botley 2 with the instructions of the hidden program (on a cardboard strip, A) without the students seeing the program. The students watched Botley 2 enact the program on the grid. Next, the students used the programming cards (B), program organizer (C), and remote (D) to program Botley 1 to enact the same program as Botley 2. When the students believed they had replicated Botley 2's program correctly, the cardboard strip was flipped and Botley 2's program was compared with students' program for Botley 1 to see if the programs matched.

Data source and analysis

The main data source was video of 16 sessions (8 sessions per toy, 480 minutes total) of student groups solving the robot coding toy tasks. Special care was made to capture the students' physical interactions with the robot and materials (i.e., remote or programming board, grid, program organizer) and discussions with peers to best observe the students' semiotic activity.

We employed provisional and thematic coding methods (Saldaña, 2015) to analyze the video data. We began with broad, a priori mathematics categories based on the existing literature and pilot work (i.e., spatial, measurement, number, and other). We used group review and interpretation of selected clips of the video data to determine observable evidence to determine nuances within the broad categories of spatial, measurement, and number concepts and skills. We then used kindergarten and first-grade U.S. mathematics standards (Common Core State Standards Initiative [CCSSI], 2010) and research-based mathematics learning trajectories (Sarama & Clements, 2009) to develop provisional codes for identifying mathematics concepts and skills within the broad a priori mathematics categories from the initial analysis. We used programming concepts and skills observed as part of a larger research project on computational thinking to develop our provisional codes for identifying programming concepts and skills (i.e., algorithmic thinking, debugging, and decomposition; Clarke-Midura et al., 2021; Lavigne et al., 2020). Pairs of coders and group review of video and codes led to more refined and/or context-specific codes. This process led to our tables of codes (see Tables 1, Tables 2, and Tables 3 in the Results section), which included context-specific descriptions of major themes (e.g., spatial knowledge in codes) and descriptions of indicators (i.e., students' artifact and mathematics/programming signs demonstrating their concepts and skills). For example, we distinguished

Figure 4. Crack the code activity for Botley.

Table 1. Spatial concepts and skills students used or developed.

Concepts and Skills	Description	Sample of Indicators (Students' Demonstration of Concepts and Skills)
Spatial orientation	Understands and operates on relationships between different positions in space (e.g., the robot's movement in relation to a location on the grid or the robot's position in relation to themselves)	• Manually rotates the robot in relation to the end-point to help plan one's program • Watches one robot move across the grid in a linear movement and determines how to move another robot in the same manner • Traces paths on the grid with hands from a start point to an end point • Considers robot's orientation in relation to the direction it will move
Spatial visualization	Understands and performs imagined transformations of objects (e.g., mental images of movement in the space such as a 90-degree rotation)	• Mentally rotates the robot to know which rotation code to add to one's program and demonstrates with head, hand, or body movements • Adjusts head or body when thinking of path • Describes imagined path or ending destination in a way to indicate they have visualized the robot's movements prior to the movements occurring
Spatial language	Describes movement of robot using spatial language correctly (e.g., forward, backward, rotate left) or intuitively (e.g., straight, down, turn) or describes position of the robot relative to locations or objects (e.g., next to)	• Describes robot's linear movements with terms such as forward, backward, up, down, straight, go • Describes robot's rotational movements with terms such as rotate, turn, left, right, that way, over • Describes position of robot relative to locations on the grid such as next to, almost there, in front of
Spatial knowledge in codes	Connects spatial orientation, spatial movement, and spatial language to a representational system (e.g., codes for the program represented as arrows or tiles)	• Uses a code for a program to symbolize the movement they described • Interprets a code for a program as an instruction for movement on the grid • Describes a movement and names the code that is needed for that movement (e.g., it needs to go forward so we need a green tile; it needs to turn so use the arrow that looks like this)

aspects of spatial knowledge occurring in programming – such as spatial orientation versus spatial visualization – or aspects of number knowledge – such as counting movements from a given point versus counting codes for a program. We relied on prior mathematics education research to interpret students' various signs and how they were related to the mathematics concepts and skills in our three areas of focus: spatial, measurement, and number concepts and skills (see the Discussion section). Finally, we prepared memos detailing the instances of multiple codes and progressions in students' discussions or programming accuracy and developed a narrative summary for each lesson for each group. The memos and narrative summaries guided the selection of transcripts for illustrative examples of the major themes, which we report next.

Results

The results are organized around the two main research questions. First, we present the results of kindergarten students' engagement with mathematics during programming tasks with robot coding toys. We guide readers through the data chronologically for just one group of students with one robot, then we provide additional examples from the data that fall into each of the three mathematics concepts and skills categories (i.e., spatial, measurement, and number). Next, we present the ways mathematical and programming concepts and skills overlapped during students' interactions with the coding toys as they engaged with programming tasks.

Table 2. Measurement concepts and skills students used or developed.

Concepts and Skills	Description	Sample of Indicators (Students' Demonstration of Concepts and Skills)
Units of measure	Understands and operates with a unit of measure, usually one linear forward movement	• Gestures individual linear movements on the grid, pausing between each iterated unit of distance • Describes the number of units needed for a program or makes sounds indicating a unit of measure (e.g., deet, deet, deet) • Describes or simulates a rotation as a unit of distance from one orientation to another (90 degrees for these robots)
Distance measurement	Understands that distance can be measured by units of linear movement either by counting the units of measure or describing or showing a distance from one point to another	• Describes the measurable attributes of distance or the robot's movements • Gestures a distance, for instance, between two hands representing the distance between a start point and end point • Uses words to describe the distance between a start and end point or the robot's position and a desired position on the grid, such as far or long or too short • Compares distances by describing a distance as equal, longer, or shorter using words such as more, less, farther, shorter • Gestures or describes how much more distance the robot needs to cover compared to current position

Table 3. Number concepts and skills students used or developed.

Concepts and Skills	Description	Sample of Indicators (Students' Demonstration of Concepts and Skills)
Counting	Counts movements (a distance quantity) or codes (objects)	• Uses number words to count either movements or objects • Names the correct number of movements or codes when asked how many • Gestures on the board or grid or with fingers a specific number of squares or movements while using number words
Counting on	Counts on from a given space on the grid (i.e., robot's starting point) which involves understanding that they are counting the movements of the robot, not the squares on the grid	• Touches grid as a sliding motion or jumping motion from the starting point to show counting on from a given point in space (e.g., like jumps on a number line) • Explains verbally that counting starts after the initial square because that is the starting point
Coordinating counts	Coordinates the totals of two quantities and/or matches 1-to-1 counting with movements or codes	• Associates number of movements along a path with the number of commands or codes a program needs to complete the path • Connects the number of movements and number of codes (blocks on programming board, arrows on remote, or arrows on program organizer) through a verbal explanation or gestures • Uses number words (e.g., three forwards) instead of spatial language (e.g., forward, forward, forward) for counting movements and coordinated with another quantity (tiles or arrows)
Operations	Uses addition or subtraction to operate on quantities	• Physically adds or subtracts from a quantity of codes, most often ± 1 block or arrow from the programming board or organizer • Adds or subtracts from a quantity of movements most often ± 1, such as "we need one more forward"

Students' engagement with mathematics during coding toy tasks

Focusing on one group: Cubetto, students' semiotic activity, and the mathematics content

We first guide readers through one group's activities with the Cubetto robot coding toy. Figure 5 shows this group – Cruz, Leah, Ramona, Hugo, and the teacher (all names are pseudonyms) – and provides an overview of the four tasks. We provide segments of transcripts from this group's discussions and interactions during the tasks with the Cubetto toy to orient readers to the didactic cycle of each group.

For ease of reading and understanding the spatial orientation students used while programming, in the examples that follow we refer to program codes by the first letter of their directional name. For instance, a program that would move the robot forward then rotate left is represented as FL. On the students' programming board for Cubetto, this program was represented by one green tile and one yellow tile. On the remote control for Botley, this program was represented by pressing the forward arrow and the left rotation arrow.

The Tree to the Desert (FFF) and Desert to the Tree (BBB) tasks: spatial orientation and counting. To illustrate the students' engagement with spatial orientation concepts, we provide examples from the beginning of the lesson as students programmed Cubetto to move from the Tree to the Desert (FFF) and later to move from the Desert to the Tree moving backwards (BBB; see Figure 5's bottom left panel, a and b paths). The students placed the correct tiles on the programming board (FFF). However, a student picked up Cubetto to turn it on, but re-placed Cubetto at the starting point in a backwards orientation in relation to the desert space on the grid (the end location). Cruz noticed that Cubetto was facing the opposite direction and re-oriented the robot correctly, rotating it 180 degrees. The teacher prompted Cruz to explain why he re-oriented the robot, a critical point that later contributes to a pivot sign and ultimate connection to spatial orientation. His response highlighted his spatial orientation reasoning and his use of spatial visualization to predict what would happen if they ran the program with Cubetto oriented the wrong way on the starting square:

Cruz I thought we have to have him face forward.

Teacher That's exactly right, [Cruz]. Because he was facing this way? *((rotating Cubetto 180 degrees))*

Cruz Then he would drive off . . . *((pointing the direction Cubetto would drive, opposite the intended direction))*

Hugo and Ramona later applied Cruz's idea about the starting orientation of the robot as they programmed Cubetto to move three backwards movements to return to the tree. Hugo told the teacher that they do not need to change Cubetto's orientation and the teacher prompted him to share this idea with the group. Hugo used gestures, and Ramona added to his idea using spatial language and physically moving the robot to demonstrate the orientation she referred to (see Figure 6).

Hugo If we keep it this way [facing backward in relation to the tree] it's still gonna work because. . . *((gestures the backward movement that will occur))*

Teacher What do you all think?

Ramona But I know . . . If we turn it this way . . . *((rotates Cubetto 90 degrees))* it would go to the castle then go off the mat *((points to the castle then off the mat))*

Teacher You all are noticing that the direction Cubetto is facing matters.

Both Hugo and Ramona considered the direction Cubetto was facing and expressed their understanding that the robot's orientation is related to the direction of the movements they instruct the robot to move. Hugo noted that they did not have to reorient Cubetto because of the robot's spatial relationship to the tree and gestured the intended backward movement. Ramona built on Hugo's idea,

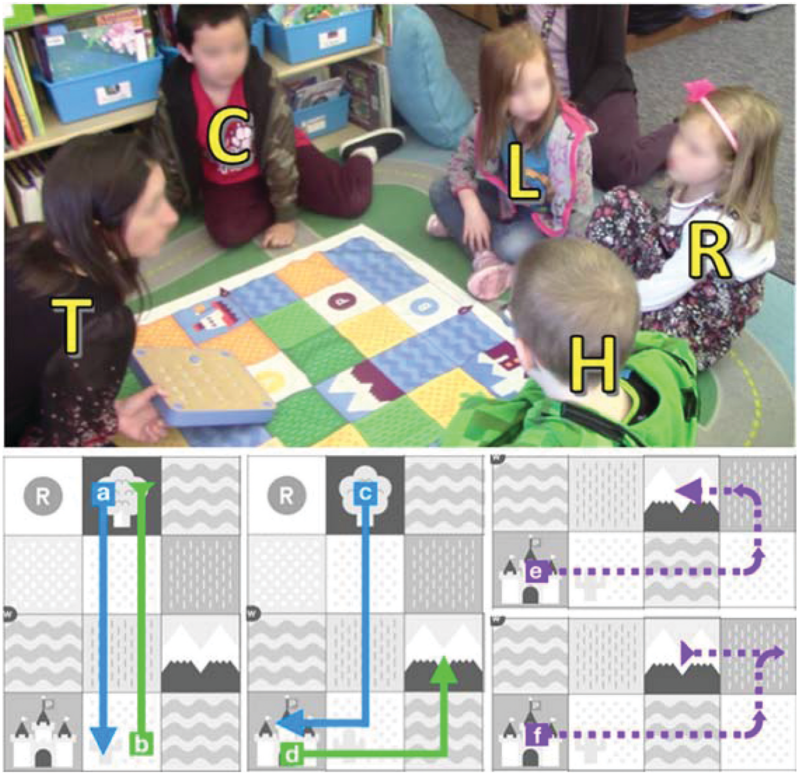

Figure 5. Setting for cubetto tasks and the types of tasks. *Note*. Top panel: Cruz (C), Leah (L), Ramona (R), Hugo (H), and the teacher (T) discussing possible paths for the Cubetto robot toy on the Adventure Grid. The students learned to use Cubetto's programming language: green tiles represent forward movements (F), purple tiles represent backward movements (B), yellow tiles represent left rotations (L), and red tiles represent right rotations (R). Bottom left panel: Task a requires three forward movements (i.e., three green tiles) to move Cubetto from the Tree to the Desert (FFF), and Task b is the Desert to the Tree (backwards) path which requires three backward movements (i.e., three purple tiles; BBB). Bottom middle panel: Task c is the Tree to the Castle path which involved forward movements and a rotation (FFFRF), and Task d is the Castle to the Mountains path which involved linear forward movements and a rotation (FFLF). Bottom right panel: Task d was intended to be FFLF, but this group chose to program the robot to move FFFLFLF (e) and FFFLFRB (f).

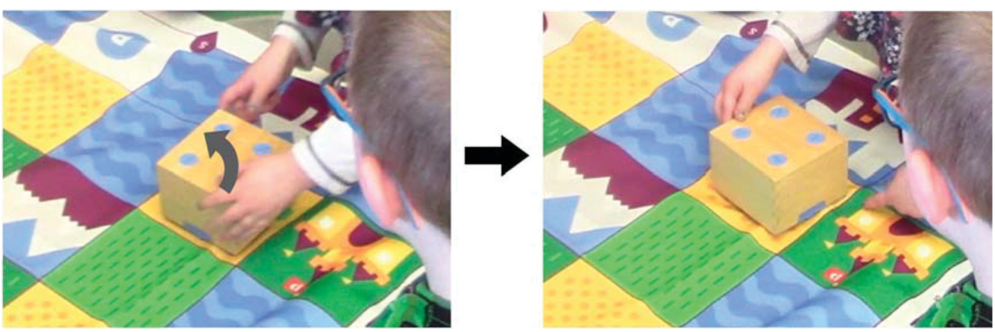

Figure 6. Ramona demonstrated that spatial orientation of the robot matters by rotating Cubetto.

rotating the robot and showing what a backward movement would look like from that orientation. Ramona's semiotic activity acted as a pivot sign that the teacher then mediated to relate Ramona's sign to the mathematics knowledge of spatial orientation.

Spatial orientation was a key concept to explore early on in the series of tasks because of the spatial nature of the toy and the navigation-based task. Similarly, students employed counting as they were learning what each code instructs the robot to do. With little experience early on in the lesson, the students were still trying to figure out what they needed to count: grid squares or the robot's movements. When students focused on the grid spaces, their programs were often incorrect because they included the grid space the robot was starting from rather than counting on from that starting point. In other words, they were counting the squares (4 squares), not the robot's movements through space (3 forward movements). As students learned to count the robot's movements, their counting resembled counting jumps on a number line (which the teacher understood, but not yet the students) in that their gestures started to mirror the pause between the robot's movements (artifact signs). Unlike counting on number lines, the students counted Cubetto's movements from the middle of one square to the middle of the next square rather than from one marked line to the next as they would on a number line, adding complexity to the way they represented these measured movements in space. The example below demonstrates this budding understanding of counting on, in particular, counting on units of linear movement from the robot's starting position. Students worked to determine how many forward tiles were needed to program Cubetto to move from the tree to the desert (see Figure 7):

Hugo One, two, three *((taps each square in the middle in a jumping motion as he counts))*

Ramona One, two, three . . . *((touches each square as she counts))* two more *((slides her hand across the two squares between Cubetto and the desert)).*

Leah So it can get here *((touches the desert square))*

Ramona Two . . . three . . . *((looks at teacher))*

Hugo Three

Cruz No, we need two *((grabs two green tiles and places one on the programming board and holds one))*

Teacher Why two?

Cruz Because if you put three, drive, drive, drive *((while sliding his hand from Cubetto to the yellow square, then to the green square, and finally to the desert square, pausing between each movement))* . . . oh, I thought it would go to the house *((grabs another green tile and places a second and third tile on the programming board)).*

Figure 7. Cruz iterating linear units of measure to determine three green codes. *Note*. Cruz moved his finger from Cubetto to the yellow square, to the green square, and finally to the desert square. This unit iteration, punctuated by his verbal expression "drive," highlighted his budding understanding of counting on and learning to count the robot's movements.

Hugo and Ramona's semiotic bundles (comprised of gesture and language) provided insight into their thinking: tapping each middle part of the square in a jumping motion while counting and sliding a hand across squares imitating the robot's motion while naming "two more." Both students counted on from Cubetto's starting location and identified each unit along the intended path with a sequential count and tap/slide. When Ramona said "two more" she showed an application of coordinating counts in that she was counting movements and understood how this connected to the number of codes needed for the programming board. Ramona and Hugo used tapping to specify the destination of each code whereas Cruz modeled each unit as a "drive," indicated by a unitized movement and iterated forward units. By justifying his reasoning for two forwards, Cruz realized an error in his initial answer and applied operations to add another forward (i.e., 2 green codes plus 1 more green code) which we observed when he grabbed another green tile and placed all three tiles on the programming board. He subtly used subtraction when he said, "oh, I thought it would go to the house," realizing that the house is one movement too far. Overall, this example illustrates the polysemy of the sign "forward" initiated by the teacher, as students' gestures differed from each other. In contrast to Hugo and Ramona, Cruz's thinking was initially observable through the mathematics/programming sign – the green codes. When pressed to explain his response of "two," his semiotic bundle appeared situated as an artifact sign (sliding hand and "drive") but his response then passed from the context of the artifact to the context of the mathematics and programming (i.e., grabbing another green code). This interaction of counting, counting on, coordinating counts, and operations brought the group to agreement to successfully code the robot to move FFF.

Castle to the Mountains task: spatial knowledge in codes and measured movements. In the Castle to the Mountains task (FFLF, FFLFLF, or FFLFRB; see Figure 5d, e, and f paths), we provide examples of students learning to symbolize (with mathematics and programming signs; i.e., color tiles) both their own gestures and language and the robot's movements with the programming codes (i.e., color tiles). This activity of symbolizing their spatial orientation knowledge with codes and connecting symbols to spatial language was an example of a space where both artifact signs and mathematical signs emerged. This challenging abstraction of their spatial knowledge is illustrated in the vignette below. In the Castle to the Mountains task, the teacher intended the program to be FFLF (see Figure 5, bottom middle panel for path d), but the students built the program FFFLFLF (see Figure 5, bottom right panel for path e). This task challenged students to understand the left rotation from two different orientations, and also challenged them to symbolize their spatial knowledge using Cubetto's coding tiles. In this segment, students worked together to build part of the program (FFFLF), and the teacher suggested that they run the program. However, Hugo stopped them because the program was incomplete:

Hugo It's going to go forwards then it needs a turn *((rotates hand gesturing left rotation on board and moves it forward)).*

Teacher Oh, so [Hugo] you think there's even more codes that we need?

Cruz *((grabs tile for a right rotation))* This one.

Leah We need to make it turn.

Teacher Ok. Is that the right one, [Cruz]?

Cruz I think so . . . I forgot . . . *((holds tile for a right rotation with one hand, retraces path with his other hand to simulate the program, emphasizing the second rotation with a sharp left .gesture))*

Hugo Wait, but it's going to have to turn this way *((gestures with his hand on the grid and rotates hand left))* and then turn this way *((rotates hand left then forward to the mountains)).*

Cruz Yeah it's the right one *((moves right rotation tile to the programming board)).*

Teacher What do you think [Hugo]?

Ramona *((reaches over and takes off a forward tile))* Better turn take this one off, because then . . .

Hugo No, we still need it.

Leah We still need it, [Ramona].

Ramona No, no, look at *((points to each code in the program))* deet, deet, deet, turn!

As indicated in the transcript, Hugo and Cruz retraced the path with hand gestures on the grid to understand the left rotation from two different relative positions. Simultaneously, they worked on symbolizing their spatial orientation knowledge and language in codes. This example illustrates how Cruz linked his semiotic bundles of symbol-gesture with Hugo's semiotic bundles of gesture-language, which was an interesting interaction of artifact signs (language and gestures tied to the robot) with the mathematical/programming sign (tile that symbolizes a rotational movement). Leah used spatial language to describe the robot's anticipated movements. Ramona was able to simulate parts of the program using her spatial orientation knowledge, but was also learning to coordinate artifact signs (i.e., spatial orientation gestures) with the mathematical/programming signs (i.e., symbolize the spatial knowledge with the programming tiles). Overall, this transcript segment showed each student in this small group using their spatial orientation knowledge correctly while still learning how to attach each movement to a program code, relying on their own and each other's semiotic activity.

This group's continued discussions and interactions about their program to move Cubetto from the castle to the mountains (see Figure 4 for path e) illustrates one example of students' observable engagement with units of measure and distance measurement. Students moved their hands along the grid, simulating the robot's individual forward movements as they paused between each movement to explain which codes were needed. These nuances in intentional gestures could be considered pivot signs because they are based in the movement of the artifact, but subtle differences in the gesture – a pause for a unit of measure versus a sweeping forward movement across a path – indicate a connection to a mathematical idea. The pausing gesture, for example, showed their ability to recognize the separate units of the planned path and their potential for understanding that one green block represented one measured or precise forward movement. As they learned the sign system, the students considered each movement as a unit and anticipated the distance those movements would make the robot travel.

Spatial concepts and skills

In this section, we continue to answer research question 1, but now provide additional examples from the data that fall into each of the three mathematics concepts and skills categories (i.e., spatial, measurement, and number). Spatial knowledge was a central component of kindergarten students' behaviors and discussions as they made observations of the robots' movements and positions, planned and built programs, and debugged programs. They engaged in spatial orientation, spatial visualization, spatial language, and representing spatial knowledge in codes, which are described in Table 1. The third column in Table 1 provides a sampling of indicators for students' demonstration of these concepts and skills, observed within students' semiotic activity.

In particular, the concept *spatial knowledge in codes* involved complex semiotic activity due to the coordination of multiple signs (i.e., the semiotic bundles) and the evolution of signs (pivot signs) toward more generality. The coding tasks challenged students to make sense of the movements of the robot in order to complete the tasks (using artifact signs), then match their knowledge of spatial movements and language with the robot's coding scheme (e.g., "it needs to move forward then turn to get there so we need to put a green and the yellow turn [on the programming board]"). We called this link between signs and mathematics/programming knowledge *spatial knowledge in codes* (see Table 1). As illustrated in the first vignette with Cruz, Leah, Hugo, and Ramona solving the Tree to the Desert task, the students attended to the spatial orientation of the robot, which was an important first step early on in the series of tasks. Later, spatial orientation became central in understanding rotations from various perspectives. In both of these spatial orientation situations, spatial language (describing movements) and gestures (physically moving the robot to gesturing rotations) emerged as signs mediating the spatial concepts and skills. Specifically, kindergartners demonstrated spatial knowledge through semiotic bundles, such as using spatial language (e.g., turn, move forward, this way, right, left, then here) while using gestures with their hands (e.g., pointing forward, gesturing a rotation, tracing a path) or by moving the robot with their hands (e.g., simulating linear or rotational movements by moving the robot). In other words, students coordinated their spatial orientation knowledge (e.g., robot's movement/orientation in relation to points on the grid) with spatial language (e.g., forward).

This type of semiotic bundle, or combination of sign systems, mediated emergent mathematics concepts and skills. The teacher became a part of a semiotic chain in students' conversations as they negotiated the meaning of these signs. An example from a small group working with Botley further illustrates this type of semiotic mediation. In the example, the students were learning how an individual code directs the Botley robot's individual movements. The teacher introduced the coding cards to help students plan their programs and keep track of what codes to input in the remote (see Figure 8). The green arrows represent forward linear movement and the red arrows represent backward linear movement. As the teacher pulled out each coding card, she asked the students what they thought the code instructed Botley to do:

Teacher *((places green forward arrow card on the floor))* Can anyone take a guess what they think this code is for?

Trevor Straight.

Teacher *((places red backward arrow card on the floor))*

Kelsey Down.

Teacher What does down mean?

Trevor *((based on Botley's orientation, gestures a backward movement with his arm))* Backwards.

Teacher Yep, this one moves Botley backwards *((points to code and physically moves Botley backwards one space))* and this one moves Botley forwards *((points to code and physically moves Botley forwards one space)).*

Fernando *((gestures directions with his hand))*

Figure 8. Botley's green arrow directs it to move forward one space, and its red arrow directs it to move backward one space.

The students relied on their spatial knowledge while learning the codes. Trevor and Kelsey used language ("straight," "down," "backwards") to describe linear movements. Fernando imitated the directional gestures, suggesting that he was considering the robot's movement in space in relation to his own body's movement in space.

In other instances, the teacher ran a program with rotations to illustrate the difference between the left and right programming codes. As students were learning the rotation codes for Cubetto, students learned to represent their spatial knowledge with the robot's codes:

Teacher I have some new pieces. Let me show you what they look like on the programming board *((places right rotation tile and left rotation tile on the programming board)).*

Mateo I want to see what happens.

Sonya They're turns.

Teacher Yeah, they're turns.

Mateo *((presses button to run the program of right rotation, left rotation))*

Teacher Do you see what's happening?

Mateo It turns both ways! *((points to red right rotation tile))* So this way is this turn *((picks up Cubetto and rotates to the right))* and that way is that way *((points to yellow left rotation tile and simulates left rotation with Cubetto)).*

Up until this point, these students had programmed Cubetto using only the forward pieces (the green tiles for Cubetto). Sonya guessed the red and yellow pieces were "turns," possibly by comparing their shapes to the green forward piece. Mateo confirmed this hypothesis by running the program and demonstrated his understanding of Cubetto's rotational coding syntax by enacting each code with Cubetto, stating that it turns "both ways." The students learned that the codes instruct the robot to execute linear or rotational movements. Students then learned to distinguish linear codes from rotational codes and how a robot will enact the movements within the space.

Each example in this section highlighted how students learned programming codes by abstracting robots' movements to symbols, thus pivoting between artifact signs (e.g., hand gestures mirroring the robot's movements) and mathematics and programming signs (i.e., arrow representing the robot's forward movement, coding tile that represents a rotation). A commonality between the Botley and Cubetto coding syntaxes is the arrow design. The arrows suggest movement and direction, which are dynamic and spatial orientation considerations necessary to successfully code the toys.

Measurement concepts and skills

The kindergarten students in the study engaged in measurement concepts and skills, specifically units of measure and distance measurement, which are described in Table 2. While students did not operate with the robots' precise measures of movement (e.g., 15 centimeters forward, 90-degree rotations), the grid-based problem space permitted the students to conceptualize each unit's approximate measure as did features of the robot (e.g., pauses between movements or blinking lights to indicate individual units of movement).

As illustrated in the vignette with Cruz, Leah, Hugo, and Ramona, measurement knowledge was observed during students' discussions about programs that required iterations of a unit of movement (e.g., three forward movements). In particular, when students had opportunities to observe what each code made the robot do, students engaged in discussions around how many iterations of a unit (i.e., the distance of one forward movement) were needed in their programs. Students showed a developing understanding of units of measure when they distinguished units by pauses in their hand movements across the grid, approximating the increments of the robot's linear and rotational movements.

In other groups, students used gestures to indicate distance. For example, the Desert to the Tree task (BBB) elicited conversations about how many backward codes to use based on a comparison of distance between two points. For example, one student explained this to her group saying, "Because we, we went three ... *((swipes a finger in the air, indicating from the tree to the desert))* ... and then to go back three ... " In this example, the student compared the first distance traveled (three forward movements) to the ensuing distance traveled (three backward movements).

Number concepts and skills

As illustrated in the vignette where Ramona, Hugo, and Cruz worked to determine how many forward tiles were needed to program Cubetto to move from the tree to the desert, students in the study engaged in number concepts and skills, which included counting, counting on, coordinating counts, and operations (see Table 3). Counting-related behaviors occurred during students' discussions about building a program or debugging a program and were interconnected with measurement concepts such as units of measure. Operations concepts were observed during debugging situations when students needed to add or subtract codes from a program. For example, students recognized that one more code was needed if a robot was one movement short of reaching the goal (like Cruz's self-correction from two codes to three codes with his "drive, drive, drive" expression).

Coordinating counts also entailed semiotic bundles. Students learned to count the robots' movements, the grid spaces, the number of codes needed, and in Botley's case, the number of pushes on a remote. Coordinating the counting among these different "counts" involved a system of signs. For example, a group of students – Andrea, Feng, Alina, and Dre – worked on the Botley Crack-the-Code task. The students needed Botley to travel two grid spaces forward. Originally, Dre placed two forward arrows on the program organizer. Alina added one more to make three arrows. This action prompted a debugging situation in that Dre and Feng recognized that they had just tried three forwards and the robot went one space too far:

Dre Three straights again?

Feng Again?

Teacher [Alina], did you hear what they just said?

Alina Wait, this out *((pointing to the third arrow she had added))* because then it will go like this, this *((touching the grid square Botley is sitting on then touch the grid square one ahead of it))*. Or probably because she starts here she probably goes like there *((points to the grid square Botley is starting on and makes a long continuous sliding motion with her finger that goes off the board))*

Upon Dre and Feng's remarks, Alina's attention was drawn to the bug in her program and her coordination of signs (gestures and language) directed at the codes (arrows), grid squares, and movement of the robot. Once this group agreed that the program needed two forward codes, Andrea entered the codes on the remote as Dre counted "one, two" then Alina counted the robot's movements of "one, two." In this example, Alina had to understand different kinds of quantities (static discrete arrow symbols and dynamic units of the robot's forward movements), which were expressed in pointing, touching sequential spaces on the grid, gliding motions, and language such as "this, this" and "starts here ... goes like there." Alina's peers used numerical signs like, number words and counting, which together with Alina's artifact signs provided mediation of the mathematics concepts of counting and counting on.

Overlapping mathematical and programming concepts and skills during coding toy tasks

Our second research question was about understanding the ways mathematical and programming concepts and skills overlapped during students' interactions with the coding toys as they solved programming tasks. Overall, our results revealed that students demonstrated spatial knowledge in every programming domain we analyzed (i.e., five algorithmic thinking domains: learning codes, path planning, program planning, building programs, and running programs; debugging; and decomposition). A second result was that during program building tasks, students also demonstrated mathematics signs spanning every mathematics domain we analyzed in the study and in interconnected ways. A third result showed that units of measure and counting emerged as particularly important concepts during students' path decomposition of a program while operations supported students' debugging efforts. In what follows, we provide transcript segments from our video data that illustrate these results.

Pervasiveness of spatial knowledge across programming concepts and skills

Spatial knowledge was a key mathematical facet of students' engagement with robot toy programming activities. This theme was characterized by children's spatial semiotic activity across all of the programming concepts and skills listed in our analytic themes and the way it emerged across robots, groups of students, and tasks. Students' spatial knowledge mediated pivot signs. This occurred when they connected artifact signs with mathematics and programming signs as they were learning to use the codes (spatial knowledge in codes), planning paths and programs for the robot (spatial orientation and spatial language), and running programs (spatial visualization). We provide an example of each of these instances.

Learning to use spatial knowledge in programming codes. One challenging aspect of learning to program robot toys was to learn the syntax of the programming language, which in the case of these robot toys was either color-coded tile pieces (Cubetto) or directional arrows on the remote (Botley). Matching these codes with the movements of the robot was difficult for kindergarten students, because matching codes with movements involved coordinating three key pieces of knowledge: 1) their own spatial orientation in relation to the robot's orientation and movements, 2) the robot's orientation and movements in relation to locations on the grid, and 3) the desired movements in relation to the representation of a movement via the coding tile or arrow. This process required students to coordinate multiple sign systems within mathematics (e.g., spatial orientation) and programming (e.g., spatial knowledge in codes) in relation to the artifact (e.g., the robot's movements on a grid). Learning to represent movement with programming codes and learning to use the codes to elicit a robot's desired movement provided was mediated by the teacher to leverage pivot signs and connect the activity to mathematics.

Path and program planning overlapping with spatial orientation and language. Students generally used two types of planning prior to building their programs for the robot: Path Planning and Program Planning. Both of these algorithmic thinking activities were supported by the students' spatial knowledge; in particular, spatial orientation and spatial movement in language. We observed path planning when students indicated a possible route on the grid by applying language-gesture bundles to describe the path. Path planning consistently evoked rich conversation about the artifact's location on the starting point in relation to the end point. Students used language (e.g., *here, there*, and *that way*) and gestured with varying levels of precision to indicate their intended path. Program planning generally followed path planning. We observed program planning when students discussed the anticipated path using spatial language-signs, such as *forward/straight* and *turn* (sometimes, *left* or *right*) relative to the specific codes they considered using in their program (e.g., "we'll need straights and a turn"). The task context required students to orient themselves and the robot relative to grid locations, and this provided pivot sign opportunities as the students coordinated artifact, mathematical, and programming signs to plan and simulate/run paths and programs.

Running programs overlapping with spatial visualization. We observed students' spatial visualization through the timing of their hand and body gestures in relation to the robot's movements. An anticipated rotation was indicated by a finger point, a flat hand turn at the wrist, or a motion made as if they were to grab the robot to rotate it themselves. We observed students' head movements as well as shifting and rotating the trunk of their bodies to indicate movement direction or rotation. Running the program appeared to cue students' spatial visualization, as if they were thinking one step ahead of the robot in anticipation of evaluating the accuracy of their sequence of codes in the program. An example of this occurred when Cruz, Leah, Ramona, and Hugo ran their program to move Cubetto from the castle to the mountains (see the longer transcript in the section on their group's vignettes). We revisit the transcript where students ran the program to check its accuracy, but we isolate the instance when we observed Leah's spatial visualization through her gestures and Cruz's spatial visualization through his language. Leah ran the program that read FFFLRF (recall the correct program for this path was FFFLFLF). They observed the robot execute FFF in the grid space, and Leah anticipated its left rotation:

Leah It's going to turn, it needs to turn *((rotating hands above grid in a 90-degree motion to the left))*

Teacher *((points to code 4, a left turn))*

Hugo *((holds up 4 fingers))*

Cruz *((Cubetto rotates left))* Oh, we need a straight

Leah anticipated that the robot would turn before Cubetto enacted the rotation and indicated this with a language-gesture bundle, providing a glimpse into Leah's spatial visualization of a 90-degree rotation. Cruz's language sign about how to fix the bug – "Oh, we need a straight" – also evidenced his spatial visualization of what movement was needed (a forward) after the left rotation in order to debug that part of the program. These gesture-artifact bundles informed the teacher of how the students were making connections between their spatial understanding and their progress in learning the coding semantics. Sometimes the students would verbalize their mental image of the transformation saying, "it's going to turn" or "it's going to go the wrong way," connecting artifact signs and mathematical signs to give the teacher insight into the students' spatial visualization in relation to their knowledge in codes.

Program building and mathematics concepts and skills

Just as spatial knowledge was a key mathematical characteristic of students' engagement with robot toy programming activities, the algorithmic thinking activity of program building contained multiple elements of mathematical thinking. *Program building* is when students sequence codes to build a program for a robot toy. In our research context, program building entailed students using coding cards on the program organizer (to know which codes to input to Botley's remote control) or students placing coding tiles on Cubetto's programming board. In this way, program building activities generate pivot signs as students navigate the complex process of coordinating mathematics skills (e.g., spatial knowledge, measurement, number) with programming signs (e.g., robot codes) via an artifact (e.g., the robot coding toy).

Program building and interconnected spatial knowledge. Program building is distinguished from program planning by the purposeful selection and placement of codes into a programming sequence to meet a programming objective. Spatial orientation, spatial language, and spatial knowledge in codes typically co-occurred during the program building activities. This type of interconnected mathematics knowledge occurred as a small group worked together to "crack the code" for a hidden Botley program. After planning their program together, the students began to identify each code needed to move Botley 2 along the same path as Botley 1. Cruz physically moved the robot unit by unit on the grid to determine each code needed for the program, which we call *simulating the robot's movements*, a common program building strategy. Cruz had moved Botley to the beginning position. He placed a right rotation arrow on the program organizer and simulated this code by rotating Botley right. Leah stated, "Yes. I agree so far." Cruz added a forward arrow to the program organizer, and Leah stated, "yes, a straight." Cruz moved Botley back to the starting position, again simulating the right rotation and forward movement again with Leah making statements such as "And then you turn again here" and "No, the same way, the same way." Cruz moved the robot forward, then rotated it to the left. He momentarily placed the robot next to the forward arrow on the program organizer then put the robot back at the beginning grid space, seemingly orienting his perspective to the robot's perspective to determine the next rotation needed. He grabbed a left rotation arrow and placed it on the organizer, and said "Here it turns, it turns, this way."

In this segment, the students worked to coordinate spatial movement (understanding the robot's linear and rotational movements along the path), spatial language (straight, turn, this way), and spatial knowledge in codes (selecting correct codes). When these concepts converge to form semiotic bundles, we noted these moments as interconnected concepts and skills. This type of interconnected spatial knowledge occurred during program building. Sometimes an individual student's interactions or behavior evidenced this interconnected knowledge, other instances involved a convergence of these ideas across the students in the group, such as in the transcript showing students' collective use of language, gesture, movement, and representational signs. This interaction among spatial (movement), verbal (language), and symbolic (programming code) during program building is an example of the monumental role that spatial signs and systems of signs played in this complex algorithmic thinking activity.

Program building and interconnected measurement and number knowledge. Measurement and number knowledge also occurred in an interconnected way as students built programs. Students often demonstrated a consideration for linear units in program building using signs such as tracing a path in the grid space with pauses at each unit or tapping the grid spaces along a path. These signs occurred when justifying code choices while building a program or immediately after the program was complete. Students used language signs to describe the distances the robots traveled, such as if a robot went too far or not far enough. Students also compared distances to justify programming choices. Counting movements and codes was often tied to these actions. The example below illustrates this kind of interconnected mathematics knowledge. Here, students coordinated quantities of distance, movement, and counting. A student recognized that if Cubetto traveled linearly from the tree to the desert, the same number of codes were needed to move from the desert back to the tree, even if the movements were backward rather than forward. The students built a program to move Cubetto from the desert back to the tree (BBB).

Esme We need to do three . . . I think.

Joaquin *((puts a backward tile on programming board))*

Esme Because we did three when we got to there *((pointing to the ending point, the desert, where Cubetto was sitting))*

Teacher What do you think about [Esme]'s idea?

Eva Three *((gives thumbs up))*

Aurora *((reaches over to grab two backward tiles and places them on the programming board for a total of three backward tiles))*

Teacher So [Esme], why do you think three?

Esme Because we, we went three . . . *((swipes a finger in the air, indicating from the tree to the desert))* . . . and then to go back three *((gestures with hands out, palms up))*

Eva *((touches the end point on the grid))*

Joaquin *((pats three squares in incremental units from the green to the yellow to the tree))*

Oscar Three *((points to tree))*, three *((points to Cubetto/desert))*, three! *((points to programming board))*

Eva *((pats three squares in incremental units from the green to the yellow to the tree))*

Esme demonstrated an understanding of length measurement conservation by relating the number of forward commands Cubetto traveled from the tree to the desert with the number of backward commands Cubetto needed to travel the same distance. Esme used number to quantify the distance, which Aurora coordinated with program codes to build the program BBB. Joaquin and Eva affirmed their understanding and agreement by patting each square along the path back to the tree, coordinating units and total number of codes needed. Oscar also indicated his understanding of the similarities between the forward path and backward path with the number of programming codes needed by indicating the start position of each path (first the tree, then desert), then the programming board. Esme's language-gesture bundle communicated her understanding that the total distance in question – indicated by a swipe of her finger – equated to three. Oscar similarly indicated this connection using language and gesture, however, Joaquin and Eva relied solely on gesture signs to affirm their agreement. It is possible that Joaquin and Eva's taps along the path were to confirm their understanding of Esme's suggestion, or to communicate where 3 was represented within the problem space.

The mathematics in debugging and decomposition

The computational thinking processes of debugging and decomposition are intertwined with algorithmic thinking as students plan and build programs. Spatial reasoning, units of measurement, and operations knowledge emerged as helping or hindering forces when students debugged or decomposed programs.

Debugging and operations and spatial knowledge. A mathematics error was often at the core of buggy programs. Sometimes a bug in the program was a result of a misstep in spatial orientation or an error in counting. Many kindergarten students were in the beginning stages of learning how numbers are sequenced and nested within each other, so they did not yet understand how to count on from a given starting point. Mathematics skills that most frequently overlapped with debugging included operations and spatial skills, as illustrated next. In this example, the students built the program FFFLFFF to get Cubetto from the castle to a desert square at the top of the grid. The target program required one more forward at the end: FFFLFFF**F**. The students ran their buggy program. As they observed Cubetto enact the program, Brandon and Leo recognized that Cubetto was not going to reach the goal. Below is their conversation that took place as Cubetto rotated left, moved three spaces forward, then stopped one square short of the goal.

Brandon I think . . . nope.

Leo *((holds thumbs down as Cubetto continues to move, indicating that he anticipates the program not working))*

Joshua Hope we get them. *((holds thumb up))*

Leo *((continues shaking thumb upside down))*

Joshua *((Cubetto stops))* I knew it.

Brandon *((gasps))* We need one more! *((reaches over to place one more forward tile on the program board))*

Leo anticipated a bug mid-program execution. He likely visualized how far Cubetto would travel after the left rotation with the existing program and compared this to the intended destination. Brandon used operations ("one more") to debug, and communicated this with a language-movement bundle, placing a forward tile to the program. We observed the use of operations when students used language and/or gesture signs to indicate the addition or subtraction of a code. Adding a code occurred when a forward movement was missing (like the example with Brandon) and subtracting occurred when the robot traveled too far. Adding and subtracting codes acted as pivot points for students to engage with basic operations in an authentic problem-solving experience while connecting interactions with the artifact to mathematics. Within this use of adding/subtracting was students' use of spatial knowledge to identify a bug and isolate the error and spatial language-gesture bundles to justify debugging strategies.

Decomposition and units. Decomposition was observed when students decomposed paths and programs into individual units or at a change in movement type (e.g., from forwards to a rotation). Decomposition requires students to understand the one-to-one relationship of a programming code with the robot's individual movements. Hence, students' decomposition activity overlapped with measurement units and number concepts. For example, a student demonstrated decomposition with language-gesture signs by simulating the path the robot should take and separating each unit of movement with a sequencing word (e.g., *this, then this*). In this example, the students were solving a Crack the Code task (RFLF) for Botley. The students built the program RF and knew a rotation was needed next. Hyrum thought it should be a left rotation should go next, but Marjorie was not convinced and countered Hyrum's program adjustments by using language-movement bundles to demonstrate the target path unit by unit:

Hyrum Wait. We need ... *((RFL on the program organizer, takes off the left code from the program organizer, takes off the forward and looks at the other cards)).*

Marjorie But if it went this way *((moves Botley right))*, if it went like this *((puts Botley back in its starting position and orientation and then rotates it right))* and then this *((moves Botley forward one space))* and then this *((rotates Botley to the right and then moves it one space forward))*, what?

Marjorie decomposed the path by moving the robot unit by unit, separating each unit with a pause and sequencing language (e.g., *this, then*). Hyrum returned the left and forward commands to the program organizer (now RFL), and Marjorie again decomposed the program into individual units and used language-movement bundles to ensure that each code was accurate:

Marjorie If it went like this *((brings Botley back to the starting position))*. If it went this *((rotates Botley to the right))*, this *((moves Botley one space forward))*, and then this *((rotates Botley to the right and moves it one space forward))*, then it would be *((motions to incorrect location))*. If it, since it did like this *((rotates Botley to the right))*, this *((moves Botley one space forward)), this ((rotates Botley to the left and moves it one space forward))*, it's *yellow ((indicating a left command)).*

Decomposition required students to understand the iterative measure of the robot's dynamic linear and rotational units. The grid squares scaffolded students' use of the linear unit. Each forward and backward code moved the robot from one square to the next, providing students a benchmark for each linear unit. Similarly, the grid's construction permitted students to observe the rotational unit (90 degrees) of the left and right rotation codes.

Discussion

In this study, we examined kindergarten students' engagement with mathematics as they solved programming tasks with robot coding toys. Using semiotic mediation as a lens, we showed how students demonstrated spatial, measurement, and number concepts and skills during the robot coding toy tasks. The results revealed the ways in which programming robot toys elicited opportunities for students to engage with mathematics in dynamic and interconnected ways, thus creating an entry point to reassert mathematics beyond the traditional school content and curriculum (Wilensky & Papert, 2010). Students' connections between artifact signs and mathematical/programming signs allowed us to view not just the ways codable robot toys make computational thinking more accessible, but also the ways these robot toys' symbols mediated mathematics knowledge in distinguishable ways, such as representation of continuous quantities, coordination of quantities, iteration of quantities, and decomposition of continuous quantities.

Spatial and programming knowledge with robot coding toys

Spatial knowledge occurred across algorithmic thinking, decomposition, and debugging concepts and skills and supported students' engagement in the programming activities. In other words, spatial knowledge was at the center of learning to program robot toys and pervaded various dimensions of the CT process (Francis et al., 2016; Palmér, 2017). This finding aligns with Palmér's (2017) findings that showed that students were able to visualize actions and coordinate the relationships among actions, gridded maps, and symbols. Our results suggest that these coding toy activities may offer unique experiences for students to engage with spatial concepts and skills by relating artifact signs generated from interacting with the artifact (i.e., robot) with mathematical/programming signs (i.e., arrows that represent movement in space), which leads to a mediation of dynamic mathematics content (i.e., spatial orientation, quantifying movements).

Program building and interconnected mathematical knowledge

Another key finding was that building programs for the robot toys elicited students' use of interconnected mathematics knowledge. As students learned to sequence codes to build a program, we observed them coordinating various aspects of mathematical and programming knowledge. The complexity of the program building tasks often required interconnected spatial knowledge in which students used codes to express (symbolize) their spatial orientation, spatial movement, and spatial language knowledge, demonstrated in semiotic bundles. Francis et al. (2016) similarly found that this kind of interplay among varied skills elicits opportunities for complex spatial reasoning. In other instances, they used interconnected measurement and number knowledge to coordinate their knowledge of quantities of distance, quantities of movements, and quantities of codes.

Dynamic notions of spatial, measurement, and number knowledge

Students used their emerging notions of space, measurement, quantities, and counting in ways that extend beyond school mathematics, in other words, the kindergarten content standards as delineated by the Common Core State Standards for Mathematics (CCSSM) for most states in the U.S. (Common Core State Standards Initiative [CCSSI], 2010). In kindergarten, students learn to count up to 20 organized or scattered objects (K.CC.B.5; Common Core State Standards Initiative [CCSSI], 2010). In our study, we observed students grapple with counting in a new and challenging context. The context of the robot toys provided them opportunities to count *movements* (as opposed to *discrete objects*), coordinate their counting of related objects and symbolize that relationship (e.g., three linear movements needs three green codes), and grapple with the concept of counting on from a starting point in space (i.e., counting on from point A).

Similarly, the dynamic and precise nature of the spatial knowledge these kindergarten students engaged with during programming activities with the robot toys extended and enriched the spatial knowledge addressed in the mathematics standards for kindergarten. Kindergarten students are expected to " … describe the relative positions of these objects using terms such as *above, below, beside, in front of, behind*, and *next to*" (Common Core State Standards Initiative [CCSSI], 2010). These static notions of space reflect the common requirement for children to sit still in classrooms, which "might limit the development of their spatial reasoning" (Francis et al., 2016, p. 18). Rather than static notions of relative position in space, these programming tasks allowed student to engage in navigation skills and perspective taking as they described the robot's position in terms of its movements or relative positioning using terms such as *forward, backward*, and *rotate*, and relational language such as *this way, from the start, almost to the castle*, and *not far enough*. These forms of spatial experiences provide a viable way to spatialize mathematics curriculum (Davis & The Spatial Reasoning Study Group, 2015), promoting children's balanced development of spatial skills and knowledge through authentic, meaningful tasks (Francis et al., 2016).

The kindergarten measurement standards suggest that students should learn to describe measurable attributes of objects (K.MD.A.1) and directly compare objects to determine and describe the difference of their measurable attribute, such as more or less length or weight (K.MD.A.2; Common Core State Standards Initiative [CCSSI], 2010). In the context of the coding toy tasks, students engaged in distance measurement comparisons and in linear and rotational units of measure. These are challenging concepts when you consider that this age group to falls in Sarama and Clements (2009) learning trajectory levels of *Serial Orderer* and *End-to-End Length Measurer*. Relatedly, in order to build a program for the robot toy, students needed to identify a linear unit of measure, determine the number of iterations, then iterate the unit. In this way, students extended their knowledge of counting discrete objects (K.CC.B.4; Common Core State Standards Initiative [CCSSI], 2010) to counting movements and learning to count on from a given point in space (Siegler & Ramani, 2008). This means that students engaged in measurement in dynamic ways, being able to attend to units of continuous distance, not just discrete amounts. A vast amount of literature has demonstrated that elementary students perform poorly with spatial measurement tasks, and that students think that units of discrete measurement – such as lines on the ruler – are what they are measuring rather than the space between two lines (Blume et al., 2007; Drake, 2014; Smith et al., 2013). Our results are important because they suggest that this coding toy context holds unique opportunities for students to learn measurement concepts, such as a unit of linear length and iterations of that unit in space, that have been historically challenging to address with standard measurement curriculum in the U.S. (Szilágyi et al., 2013).

Finally, the debugging activities that students engaged in while coding provided an authentic problem for using number operations, specifically the concept of one more and one less. The decomposition work students did in program building and debugging provided an authentic context for breaking up a problem to make it easier to solve and attending to part-whole concepts that are critical to successful and efficient mathematical problem solving (Cheng, 2012). Steffe (1992) described two schemes of action and operation regarding composite units: coordinating and segmenting. These relate to the decomposition young children used with units of linear measure, both in terms of sequencing linear units to compose a path or program (coordinating) and looking at a path and decomposing it into smaller individual parts (unit segmenting). This is important because the embedded coordinating and segmenting affordances of the toys and tasks may support the development of composing and segmenting units.

Future research needs

Future research is needed to explore other intersections and connections between programming and mathematics, such as mathematical practices and computational thinking, thereby broadening the relevance of mathematics (Pérez, 2018) or even transforming or revolutionizing the way we characterize mathematics (diSessa, 2018; Wilensky & Papert, 2010). Research is also needed to better understand the complexity of the technologies' affordances and the tasks used in these settings. There are still lingering questions about the role of the robot's design features and how these specific features afford or maybe even hinder mathematics knowledge and learning. For example, does the remote for Botley versus the programming board for Cubetto elicit different kinds of mathematical knowledge? In addition, the tasks in our study were designed to introduce young children to programming. Future research is needed on tasks that use the programming technology, but foreground the mathematics (similar to Lavigne et al., 2020), use the robot toys to teach specific mathematics topics, and explicitly link mathematics and CT (Hickmott et al., 2018).

Conclusion

Understanding the connections between mathematics and programming knowledge is important for learning ways we can "re-vision, a making-visible of the mathematical structures and relationships which are hidden beneath the surface of our realities" (Noss & Hoyles, 1996, p. 255). Within our study, students engaged in dynamic and interconnected mathematics, a deviation from traditional school mathematics which tends to focus on static notions of mathematics and individual skills that are later connected. With a push to integrate programming in elementary mathematics education (Barr & Stephenson, 2011; Hickmott et al., 2018; Israel & Lash, 2020; Kotsopoulos et al., 2017), identifying the kind of mathematics students use in coding has implications for classroom instruction. This knowledge can help kindergarten teachers draw attention to the mathematics which is often hidden beneath the surface and mediate the content using an engaging artifact that encourages young students to produce personally meaningful signs and connect to cultural mathematical signs. The more teachers are aware of the embedded mathematics within these types of programming contexts, the easier it is for them to better leverage connections between school mathematics and mathematics within STEM and computer science. Integrating mathematics and programming learning is an area to further explore in order to help school mathematics evolve with the rapid pace of technological developments in the digital age.

Acknowledgments

The authors would like to thank the members of the Coding in Kindergarten Research Group (Hannah Evans, Rebecca Peterson, Rebecca Cox, Mitchell Atkinson, and Deborah Silvis) for contributions to discussions, coding, and literature reviews. They also recognize the teachers and students at their partner schools and their willingness to undertake STEM +C and Math+C approaches to learning.

Disclosure statement

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

Notes on contributors

Jessica F. Shumway is Assistant Professor of Mathematics Education at Utah State University where she studies instructional practices and technologies that foster mathematics learning in elementary school.

Lise E. Welch is a doctoral student at Utah State University where she studies early spatial and measurement concepts situated in computational thinking activities.

Joseph S. Kozlowski is a doctoral student at Utah State University where he studies mathematical creativity and the intersection of mathematical and computational thinking.

Jody Clarke-Midura is Associate Professor of Instructional Technology and Learning Sciences at Utah State University where she studies digital media, games, and toys for learning and assessment.

Victor R. Lee is Associate Professor in the Graduate School of Education at Stanford University where he studies STEM learning with the support of emerging digital technologies.

Funding

This work was supported by a grant (Award #1842116) from the National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation, Utah State University, or Stanford University.

ORCID

Jessica F. Shumway http://orcid.org/0000-0001-7655-565X
Joseph S. Kozlowski http://orcid.org/0000-0003-4163-2955
Jody Clarke-Midura http://orcid.org/0000-0001-5434-0324
Victor R. Lee http://orcid.org/0000-0001-6434-7589

References

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

Arzarello, F., Paola, D., Robutti, O., & Sabena, C. (2009). Gestures as semiotic resources in the mathematics classroom. *Educational Studies in Mathematics*, *70*(2), 97–109. https://doi.org/10.1007/s10649-008-9163-z

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

Bartolini Bussi, M. G., & Baccaglini-Frank, A. (2015). Geometry in early years: Sowing the seeds towards a mathematical definition of squares and rectangles. *ZDM Mathematics Education*, *47*(3), 391–405. https://doi.org/10.1007/s11858-014-0636-5

Bartolini Bussi, M. G., & Mariotti, M. A. (2008). Semiotic mediation in the mathematics classroom: Artifacts and signs after a Vygotskian perspective. In L. English & D. Kirshner (Eds.), *Handbook of international research in mathematics education*, (2nd ed., pp. 746–783). Routledge.

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

Blume, G. W., Galindo, E., & Walcott, C. (2007). Performance in measurement and geometry from the perspective of the principles and standards for school mathematics. In P. Kloosterman & F. K. Lester (Eds.), *Results and interpretations of the 2003 mathematics assessment of the national assessment of educational progress* (pp. 95–138). National Council of Teachers of Mathematics.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking* [Paper presentation]. American Education Researcher Association, Vancouver, Canada.

Cheng, Z.-J. (2012). Teaching young children decomposition strategies to solve addition problems: An experimental study. *The Journal of Mathematical Behavior*, *31*(1), 29–47. https://doi.org/10.1016/j.jmathb.2011.09.002

Clarke-Midura, J., Silvis, D., Shumway, J. F., Lee, V. R., & Kozlowski, J. (2021). Developing a kindergarten computational thinking assessment using evidence centered design: The case of algorithmic thinking. *Computer Science Education*, *31*(2), 117–140. https://doi.org/10.1080/08993408.2021.1877988

Clements, D. H., & Battista, M. T. (1989). Learning of geometric concepts in a Logo environment. *Journal for Research in Mathematics Education*, *20*(5), 450–467. https://doi.org/10.2307/749420

Clements, D. H., Battista, M. T., Sarama, J., & Swaminathan, S. (1996). Development of turn and turn measurement concepts in a computer-based instructional unit. *Educational Studies in Mathematics*, *30*(4), 313–337. https://doi.org/10.1007/BF00570828

Common Core State Standards Initiative [CCSSI]. (2010). *Common core state standards for mathematics*. http://www.corestandards.org/Math/

Computers Science Teachers Association [CSTA]. (2017). *CSTA K-12 computer science standards, revised 2017*. http://www.csteachers.org/standards

Cuneo, D. (1986). Young children and turtle graphics programming: Generating and debugging simple turtle programs. *Paper presented at the meeting of the American Educational Research Association*. American Educational Research Association.

Davis, B., The Spatial Reasoning Study Group. (2015). *Spatial reasoning in the early years: Principles, assertions, and speculations*. Routledge.
diSessa, A. A. (2018). Computational literacy and "The big picture" concerning computers in mathematics education. *Mathematical Thinking and Learning*, *20*(1), 3–31.https://doi.org/10.1080/10986065.2018.1403544
Drake, M. (2014). The problem with the school ruler. *Australian Primary Mathematics Classroom*, *19*(3), 27–32. https://files.eric.ed.gov/fulltext/EJ1093323.pdf
English, L. D. (2017). Advancing elementary and middle school STEM education. *International Journal of Science and Mathematics Education*, *15*(S1), 5–24. https://doi.org/10.1007/s10763-017-9802-x
English, L. D. (2018). On MTL's second milestone: Explore computational thinking and mathematics learning. *Mathematical Thinking and Learning*, *20*(1), 1–2. https://doi.org/10.1080/10986065.2018.1405615
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
Francis, K., Khan, S., & Davis, B. (2016). Enactivism, spatial reasoning and coding. *Digital Experiences in Mathematics Education*, *2*(1), 1–20. https://doi.org/10.1007/s40751-015-0010-4
Gadanidis, G., Clements, E., & Yiu, C. (2018). Group theory, computational thinking, and young mathematicians. *Mathematical Thinking and Learning*, *20*(1), 32–53. https://doi.org/10.1080/10986065.2018.1403542
Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051
Hamilton, M., Clarke-Midura, J., Shumway, J. F., & Lee, V. R. (2020). An emerging technology report on coding toys and computational thinking in early childhood. *Technology, Knowledge and Learning*, *25*(1), 213–224. https://doi.org/10.1007/s10758-019-09423-8
Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2018). A scoping review of studies on computational thinking in K-12 mathematics classrooms. *Digital Experiences in Mathematics Education*, *4*(1), 48–69. https://doi.org/10.1007/s40751-017-0038-8
Israel, M., & Lash, T. (2020). From classroom lessons to exploratory learning progressions: Mathematics + computational thinking. *Interactive Learning Environments*, *28*(3), 362–382. https://doi.org/10.1080/10494820.2019.1674879
Kaput, J., Hegedus, S., & Lesh, R. (2007). Technology becoming infrastructural in mathematics education. In R. A. Lesh, E. Hamilton, & J. Kaput (Eds.), *Foundations for the future in mathematics education*(pp. 173-192). Routledge.
Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A pedagogical framework for computational thinking. *Digital Experiences in Mathematics Education*, *3*(2), 154–171. https://doi.org/10.1007/s40751-017-0031-2
Lavigne, H. J., Lewis-Presser, A., & Rosenfeld, D. (2020). An exploratory approach for investigating the integration of computational thinking and mathematics for preschool children. *Journal of Digital Learning in Teacher Education*, *36*(1), 63–77. https://doi.org/10.1080/21532974.2019.1693940
Miller, J. (2019). STEM education in the primary years to support mathematical thinking: Using coding to identify mathematical structures and patterns. *ZDM*, *51*(6), 915–927. https://doi.org/10.1007/s11858-019-01096-y
Muñoz-Repiso, A. G. V., & Caballero-González, Y. A. (2019). Robotics to develop computational thinking in early childhood education. *Comunicar*, *27*(59), 63–72. https://doi.org/10.3916/C59-2019-06
Noss, R., & Hoyles, C. (1996). *Windows on mathematical meanings: Learning, cultures and computers*. Kluwer Academic Publishers.
Palmér, H. (2017). Programming in preschool—with a focus on learning mathematics. *International Research in Early Childhood Education*, *8*(1), 75–87. https://files.eric.ed.gov/fulltext/EJ1173690.pdf
Papert, S. (1972). Teaching children to be mathematicians versus teaching about mathematics. *International Journal of Mathematical Education in Science and Technology*, *3*(3), 249–262. https://doi.org/10.1080/0020739700030306
Pei, C., Weintrop, D., & Wilensky, U. (2018). Cultivating computational thinking practices and mathematical habits of mind in Lattice Land. *Mathematical Thinking and Learning*, *20*(1), 75–89. https://doi.org/10.1080/10986065.2018.1403543
Pérez, A. (2018). A framework for computational thinking dispositions in mathematics education. *Journal for Research in Mathematics Education*, *49*(4), 424–461. https://doi.org/10.5951/jresematheduc.49.4.0424
Rehmat, A. P., Ehsan, H., & Cardella, M. E. (2020). Instructional strategies to promote computational thinking for young learners. *Journal of Digital Learning in Teacher Education*, *36*(1), 46–62. https://doi.org/10.1080/21532974.2019.1693942
Rich, K. M., Spaepen, E., Strickland, C., & Moran, C. (2019). Synergies and differences in mathematical and computational thinking: Implications for integrated instruction. *Interactive Learning Environments*, *28*(3), 272–283. https://doi.org/10.1080/10494820.2019.1612445
Roschelle, J., Noss, R., Blikstein, P., & Jackiw, N. (2017). Technology for learning mathematics. In J. Cai (Ed.), *Compendium for research in mathematics education* (pp. 853–876). National Council of Teachers of Mathematics.
Saldaña, J. (2015). *The coding manual for qualitative researchers* (3rd ed.). SAGE Publications.
Sarama, J., & Clements, D. H. (2009). *Early childhood mathematics education research: Learning trajectories for young children*. Routledge.

Sherin, B. (2001). A comparison of programming and algebraic notation as expressive languages for physics. *International Journal of Computers for Mathematical Learning*, *6*(1), 1–61. https://doi.org/10.1023/A:1011434026437
Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003
Siegler, R. S., & Ramani, G. B. (2008). Playing linear numerical board games promotes low-income children's numerical development. *Developmental Science*, *11*(5), 655–661. https://doi.org/10.1111/j.1467-7687.2008.00714.x
Smith, J. P., III, Males, L. M., Dietiker, L. C., Lee, K., & Mosier, A. (2013). Curricular treatments of length measurement in the United States: Do they address known learning challenges? *Cognition and Instruction*, *31*(4), 388–433. https://doi.org/10.1080/07370008.2013.828728
Steffe, L. P. (1992). Schemes of action and operation involving composite units. *Learning and Individual Differences*, *4*(3), 259–309. https://doi.org/10.1016/1041-6080(92)90005-Y
Sung, W., Ahn, A., & Black, J. B. (2017). Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. *Technology, Knowledge, and Learning*, *22*(3), 443–463. https://doi.org/10.1007/s10758-017-9328-x
Sung, W., & Black, J. B. (2020). Factors to consider when designing effective learning: Infusing computational thinking in mathematics to support thinking-doing. *Journal of Research on Technology in Education*, *53*(4), 1–23. https://doi.org/10.1080/15391523.2020 .1784066
Szilágyi, J., Clements, D. H., & Sarama, J. (2013). Young children's understandings of length measurement: Evaluating a learning trajectory. *Journal for Research in Mathematics Education*, *44*(3), 581–620. https://doi.org/10.5951/jresematheduc.44.3.0581
Uttal, D. H., Miller, D. I., & Newcombe, N. S. (2013). Exploring and enhancing spatial thinking: Links to achievement in science, technology, engineering, and mathematics? *Current Directions in Psychological Science*, *22*(5), 367–373. https://doi.org/10.1177/0963721413484756
Wang, X. C., Choi, Y., Benson, K., Eggleston, C., & Weber, D. (2021). Teacher's role in fostering preschoolers' computational thinking: An exploratory case study. *Early Education and Development*, *32*(1), 26-48. https://doi.org/10.1080/15391523.2020.1784066
Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5
Wilensky, U., & Papert, S. (2010). Restructurations: Reformulations of knowledge disciplines through new representational forms. In J. Clayson, & I Kallas (Eds.), Proceedings of the*constructionism 2010 conference.*Library and Publishing Centre. https://ccl.northwestern.edu/2010/wilensky_restructurations_Constructionism%202010-latest.pdf
Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215