Journal of Science Education and Technology (2020) 29:19–34
https://doi.org/10.1007/s10956-020-09812-0

# Multiple Representations in Computational Thinking Tasks: A Clinical Study of Second-Grade Students

Tamara J. Moore[1] · Sean P. Brophy[1] · Kristina M. Tank[2] · Ruben D. Lopez[1] · Amanda C. Johnston[1] · Morgan M. Hynes[1] · Elizabeth Gajdzik[1]

Published online: 14 February 2020


## Abstract
Computational thinking requires high cognitive load as students work to manage multiple tasks in their problem-solving environment. Through research in K-2 classrooms on computational thinking, we noticed that students lack the representational fluency needed to move from one form to another—such as moving from physical to more abstract representations. Therefore, the following research question was studied: How do second-grade students use and translate among representations to solve computational thinking tasks using the robot mouse game? To address this, we employed a task-based interview approach with three second-grade students who were engaged in four computational thinking tasks using the Code and Go Robot Mouse Coding Activity Set developed by Learning Resources. Through four clinical tasks involving the robot mouse, students solved puzzles set up to force them to make particular representational translations. Each translation involved a level of cognitive complexity the students needed to manage to successfully complete the task. We found that students translated between many different representations using concrete representations to ease translations, language as a scaffold between translations, and embodied movements as representations or to assist with translation. Furthermore, the levels of representational maturity showed by the students varied with the difficulty of the task, and the spatial orientation was particularly difficult for them. These results provide important insights into how learners may develop their ability to engage with abstract representations that will be part of future practices associated with activities in science, mathematics, engineering, and computational thinking.

**Keywords** Computational thinking · representations · representational fluency · early elementary

Computational thinking (CT) is highly valued in STEM education because it involves problem-solving skills associated with using technology as part of the process. Engineers and scientists continuously transform what they observe in the physical world and represent it in abstract models they can use to help solve problems as they design and build theories about how something works. Often the practice involves using abstract models to develop algorithms that a computer can perform. Without this assistance from technology to automate complex computational tasks, engineers and scientists, as well as many others, could not accomplish their work. Further, as technology becomes increasingly more ubiquitous in our workplace, a need for partnering with technology to support thinking is becoming even more important.

Young children adapt quickly to digital and mechanical technology. Computers, mobile devices, and building kits engage children in puzzles and games and allow them to invent and build their own worlds. For decades, educational researchers have speculated that engagement with technology can provide a natural way for students to develop thinking skills they can use in everyday problem-solving and their future life's work (e.g., Papert 1980; Resnick 2006, 2007). The reason is the cognitive demand associated with learning to teach a machine (a computational device) how to autonomously perform a task involves much of the same cognition necessary to solve everyday problems and problems associated with STEM professions (Aurigemma et al. 2013; Paas et al. 2003; Sweller et al. 1998). However, it is not totally understood how we learn to think computationally, particularly young learners. Through research in K-2 classrooms, we have been studying CT demonstrated by young students engaged in

✉ Tamara J. Moore
tamara@purdue.edu

[1] Department of Engineering Education, Purdue University, Wang Hall, Suite 3500; 516 Northwestern Avenue, West Lafayette, IN 47906, USA

[2] School of Education, Iowa State University, 0624c Lagomarcino; 901 Stange Road, Ames, IA 50011, USA

engineering design-based STEM integration challenges. As part of this work, we noticed that students lack the representational fluency needed to move from one form to another – such as moving from physical to more abstract representations. In this paper, we explore how young children demonstrate their ability to interpret and interact with various representational forms as they design algorithms to instruct a robotic mouse on how to execute a solution to a variety of maze puzzles. With each maze puzzle is an increasing level of complexity needed to comprehend and solve the puzzle problems, setting up an environment to examine how students engage with and move among multiple representations as they manage the cognitive demand of various computational tasks. The following research question guided this work: *How do second-grade students use and translate among representations to solve computational thinking tasks using the robot-mouse game?*

## Literature Review and Theoretical Framework

Here we present definitions of computational thinking and a framework for thinking about representational fluency. These items provide the rationale for a clinical study to investigate how second-grade students approach the task of programming a robot mouse to solve a maze puzzle.

### Computational Thinking and Problem-Solving

Computational thinking can easily be thought of as only computer science activities, but it has been expanded into education as a systematic way of thinking that encompasses analytic and problem-solving skills as well as a language for expression and communication (Bers 2018). Since the early 1980s, Papert argued that computer programming could further the development of students' complex cognitive processes at schools. Programming computers allowed students to represent their ideas in a virtual environment, such as artists making up crafts, while they developed problem-solving skills at the same time (Papert 1980; Papert and Harel 1991). More recently, Wing (2006) likened the term "computational thinking" to the process of "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (p. 33). Thus, a growing interest in the area of CT has led to a number of frameworks for understanding and assessing computational thinking (e.g., Barr and Stephenson 2011; Brennan and Resnick 2012; Grover and Pea 2013; Kalelioglu et al. 2016). For this research, we follow the definition of Shute et al. (2017), who systematically reviewed 70 documents to propose a theoretical framework for CT based on problem-solving. Shute et al. defined CT as "the conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts" (p. 151), and they highlighted the following six competencies or facets of CT from the literature: *decomposition*, *abstraction*, *algorithms*, *debugging*, *iteration*, and *generalization*. *Decomposition* refers to breaking down complex problems into more manageable parts. *Abstraction* means to extract the essence of a system through three sub-processes: data collection and analysis, pattern recognition, and modeling. In *algorithms*, one creates a set of instructions for humans or machines to follow to solve a problem. This facet has four subcategories: algorithm design, parallelism, efficiency, and automation. *Debugging* refers to identifying and addressing problems that inhibit progress toward task completion. *Iteration* refers to repeating the problem-solving process to refine the solution toward an optimal solution (task completion), which involves debugging. Finally, *generalization* is defined as the ability to transfer the CT skills to other problem-solving contexts. This view of CT as a foundation for problem-solving recognizes that CT encompasses a set of thinking skills that are rooted in computer science but share similarities with thinking skills in other disciplines such as mathematics, engineering, design, and science (Bers 2010; Shute et al. 2017). Thus, this perspective does not limit CT only to concepts related to coding or programming but instead suggests a broader focus on problem-solving and the ability to approach problems in a systematic way.

### Computational Thinking and Multiple Representations

CT is a systematic way of thinking that offers multiple paths for problem-solving, communication, and expression relating to well-structured, ill-structured, and real-world problems (Wing 2010). As students are developing CT skills, the types of tasks associated with these problems often require translation among multiple representations, including concrete, symbolic, pictorial, motor, and language representations in order to solve problems and communicate solutions (Bers 2018). For example, students need to be able to comprehend, interpret, and use information from representational media to address CT tasks. Thus, children have a double challenge as they are asked to translate across information encoded in different representations while they are also developing the competencies of CT (Bers 2018). Since CT is broader than coding and is not limited to using computers, the tools used for teaching CT provide opportunities to integrate physical manipulation, movement, and motor skills (Bers 2018; Byers and Walker 1995). Furthermore, it has been argued that CT is more than just a problem-solving process as it requires students to solve problems algorithmically and develop a level of technological fluency and language (Papert 1980; Bers 2010) as they learn to communicate and express their ideas within the language of code. Thus, they are "learning about" and "using" language and symbols as they use a formal constructed language to communicate instructions or codes in their own project (Bers 2018). Overall, the development of CT requires students to move within and among multiple representations. This intrinsic relationship between CT and representations needs to be further studied.

## Representations

Designing relevant computational thinking tasks and expectations for students requires knowing the level of competence that is achievable and reasonable for young students. Therefore, the aim of this study is to better understand the computational thinking processes that students use when solving CT tasks. However, it is difficult to assess and measure this understanding, especially in young students, because we cannot directly assess their internal understandings and their partial completion of tasks. Because we can observe the actions they undertake and how they use the resources available to them, we can assess their use of the tools used in CT. Student use of representation in problem-solving is one of these tools, and how students use different representations can give us important insights into their cognition and development. As a result, we use representational fluency as a proxy for understanding the cognitive actions involved in the tasks in this study. Although this cannot give us a complete picture of their thinking, it can yield valuable insights about the computational thinking processes of young students.

Students' ability to solve CT tasks depends on their representational fluency. Representational fluency is the ability to translate within and among representations of a concept and is an important cognitive skill that students need for deep understanding (Greenes and Findell 1999; Johri and Olds 2011; Lesh and Zawojewski 2007; Suh and Moyer 2007). Representations and translations between them support computational offloading, provide more information through re-representation, and express abstract information (Ainsworth 2006). However, in order to utilize a representation, learners must understand how the information is encoded in the representation, how it relates to its context or domain, how to choose the best representation according to the problem, and how to construct new representations when necessary (Ainsworth 2006). We operationalize the representations and translations for this study using the Lesh translation model and research on gestures.

## The Lesh Translation Model

The Lesh translation model (LTM), shown in Fig. 1, is a framework for understanding representation use and representational fluency (Lesh and Doerr 2003). Lesh built upon the work of Bruner (1966), who introduced three different types of representations that are used to understand the world: through doing it (enactive representations), through depicting it (iconic representations), and through symbolic means, such as language (symbolic representations). The LTM (Lesh and Doerr 2003) goes further to include five types of representations: (1) representation through realistic, real-world, experience-based metaphors, (2) written symbolic representation, (3) language-based representation (spoken or written), (4) pictorial or graphic representation, and (5) concrete representation (concrete, manipulatable models). Students often use different representational media when struggling with a task (Dienes 1960; Thomas et al. 2010). For example, the mental rotation of an object with multiple details can be more easily completed by manipulation of a concrete model rather than just the mental processing of a picture (Stieff et al. 2016; Stull et al. 2012). Within the LTM, students may embody certain concepts within certain representations, but students' ability to move among representations demonstrates more understanding than static representations alone. However, this model of representational fluency does not explicitly include gestures, which is a needed representational media when considering students' CT processes.

## Gestures as Representations

Gestures are representations people of all ages use to describe the world around them and to support their thinking during problem-solving (Kita et al. 2017). Thus, the use of gestures is often seen while speaking. Gestures used in conjunction with language can help with self-regulation to support high cognitive loads in computational thinking tasks, as well as other forms of problem-solving (Goldin-Meadow et al. 2001; Thurnham and Pine 2006). However, these behaviors are often different in children and adults. As children develop problem-solving skills, their use of language develops, "allowing language to become a tool for planning, controlling, and evaluating actions" (Alarcón-Rubio et al. 2014, p. 96). Additionally, children's abilities to solve cognitively difficult

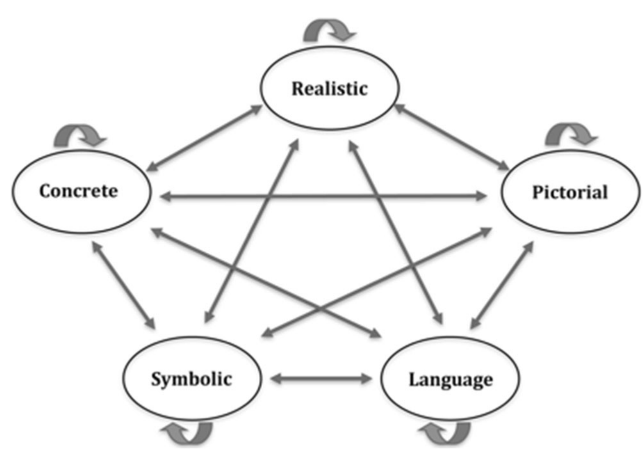


**Fig. 1** The Lesh translation model

tasks are correlated to their use of internalized private speech (Alarcón-Rubio et al. 2014). In addition to language and gestures, pointing, depicting objects with the shape of hands, and indicating spatial locations also act as a representation of thought to convey ideas (Hostetter and Alibali 2008). However, gestures not only fill the role of conveying ideas to others but may also, in fact, be "integral to the child's cognitive processes" (Thurnham and Pine 2006, p. 47). While there is still a debate about the role of embodiment, gestures, and movement in cognition, the concept of embodied cognition has accumulated significant evidence relating learning and cognitive processing to an individual's use of gestures and body movements (Wellsby and Pexman 2014). Research in mathematics learning has suggested that gestures can help reduce cognitive load of learners as they balance multiple ideas or numbers in their mind (Goldin-Meadow et al. 2001). Given the low number of items that can be stored in working memory (Miller 1956), the use of gestures and body movements has also been linked to an individual's learning and recall, development of language and concrete concepts, and development of abstract concepts (Alibali and Nathan 2012; Macedonia and von Kriegstein 2012; Roth and Lawless 2002). Indeed, "gesture activates, manipulates, packages and explores spatio-motoric information for the purposes of speaking and thinking" (Kita et al. 2017, p. 262). Therefore, for our framing of representations, we have added gestures as an additional representation beyond the five representations in the LTM.

### Representations and Spatial Reasoning

The variety of different types of representations students use and the translations between them can require varying levels of proficiency in spatial reasoning. Spatial reasoning is an important skill linked with mathematical and computational thinking (Bruce and Hawes 2015; Gunderson et al. 2012; Román-González et al. 2017; Wai et al. 2009) because it encompasses "building and manipulating two- and-three-dimensional objects; perceiving an object from different perspectives; and using diagrams, drawings, graphs, models, and other concrete means to explore, investigate, and understand abstract concepts such as algebraic formulas or models of the physical world" (Kinach 2015, p. 535). Spatial reasoning is a developed ability, and the spatial reasoning skills of children vary greatly from those of adults (Thommen et al. 2010). Young children often rely on more concrete representations to aid spatial thinking because a certain level of cognitive flexibility is required for internal spatial manipulation and re-representation of objects in other forms (Ebersbach and Hagedorn 2011; Harwood and Usher 1999). These enacted representations are key first steps in perceiving and making sense of new experience (Bruner 1966). Although students' spatial reasoning abilities vary, they can be improved with practice and instruction (Feng et al. 2007; Uttal et al. 2013).

This is also true within a computational thinking environment. Fessakis et al. (2013) found that kindergarten students struggled with spatial orientation and directions within a computer programming interface. The students faced difficulty in decoding angle and turn symbols within the coding environment to orient the ladybug (the actor within the environment) in an appropriate direction. They also had difficulty with position directions when the ladybug faced a direction that the students could not orient themselves to be in alignment. These difficulties caused students to interact with the teacher or each other and with the software. The study found that these issues tended to persist throughout the study but faded some over time. Together, the ideas regarding representational fluency, including gestures and spatial reasoning, will be used to help analyze the data in this study.

## Methods

In this study, we are interested in how young students utilize representations when engaged in CT tasks. Therefore, we address the research question: *How do second-grade students use and translate among representations to solve CT tasks using the robot-mouse game?* To do this, we employed a task-based interview approach (Goldin 2000), which is a form of a clinical interview (Clement 2000; Ginsburg 1981; Hunting 1997). The interview is designed to have the participants interact with the interviewer while performing a carefully constructed problem-solving task. The task is designed to elicit participants' knowledge, thinking, or representations of particular ideas and ways of reasoning as they attempt to solve the problem (Maher and Sigley 2014). The task-based interview approach provided a structured environment that can be somewhat controlled and therefore allows for systematic and in-depth exploration of a specific topic (Goldin 2000), which, in this research, targets students' use and translations among representations during CT tasks.

### Context and Participants

This study is part of a larger externally funded project that is focused on integrating STEM and CT for K-2 students in both in-school and out-of-school environments. As part of this larger project, the project team wanted to understand the computational abilities of young students as they relate to coding. The current study was conducted at a K-8 public charter school located in a small city in the Midwest that included 57% students on free or reduced lunch and 22% students of color. This study was conducted as part of a 4-week, in-school enrichment time called "Puzzles and Robots." This enrichment program was part of the school day and included four

1-hour sessions associated with integrating CT with STEM through various learning experiences. These sessions spanned 4 weeks and contained several options in which students could self-select to participate. This study was conducted during the first 2 weeks of the enrichment program. All students in K-2 had to choose among various enrichment activities, and nine of the K-2 learners chose to attend "Puzzles and Robots," which included our lessons with the Code and Go™ Robot Mouse Activity Set as the first two sessions. The activities for these two sessions were led by members of the project research team. The students in the study had no prior formal experience with CT.

Informed consent was obtained from all individual participants in our enrichment activity. Seven of the nine students in the program and their parents assented/consented to participate in our data collection, and the three second-grade students who consented were chosen as the focus of this study. We chose the second-grade students because the additional tasks we designed were most accessible to the older students in the program (Goldin 2000). These three students were interviewed as they worked through a series of CT tasks using the Code and Go™ Robot Mouse Activity Set (described below). The students were in groups of two as they traveled to stations around the room and participated in four intentionally designed tasks across the two sessions. Two of the focus students, Lily and Jason, participated as a pair throughout all the activities, and the third student, Beth, worked with another student (a first grader with consent) on three of the activities and alone for one of the activities. Names of participants are pseudonyms.

## The Task Environment

The computational thinking tasks within this study utilized the Code and Go™ Robot Mouse Activity Set developed by Learning Resources. The activity set is set up as a game-like environment (Fig. 2) in which the player generates a sequence of steps that instruct the robot mouse on how to move through a course or maze to find the cheese. The activity set includes a battery-operated, programmable robot mouse named Colby, a cheese wedge that makes Colby's nose light up when touched, 16 square tiles that can be interlocked into various

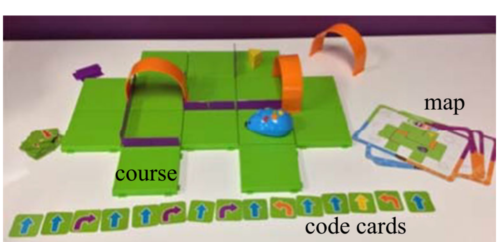


**Fig. 2** The Code and Go™ Robot Mouse activity. Set up as the game was intended to be played

configurations to make a larger course and are the size of one forward movement of the mouse, 30 code cards printed with a direction arrow or action symbol (the action symbol is a lightning bolt), and 10 double-sided activity cards depicting different course maps of maze puzzles to be solved (Learning Resources n.d.). The robot mouse can be programmed to move forward or backward, to pivot left or right 90°, or to perform one of three random actions that include making a mouse squeak noise, chirping while lighting up the eyes, or moving forward then backward. Each command is controlled by a different colored button on the back of the mouse that is either an arrow pointing in the corresponding direction (for the movement of the mouse) or a circle (for start, clear, and action). Every button represents a discrete operation. For example, the mouse turn (caused by pushing the left or right arrow) includes no forward motion; rather, the mouse pivots in place. Additional course pieces to the tiles, including walls and tunnels, can be added as obstacles and constraints to solving the puzzle.

The Code and Go™ Robot Mouse Activity Set was chosen as the basis for our CT task-based interviews because the activity resembles the basic features of programming associated with higher-level languages and was developed for use by this age child. Computer programming typically involves planning an algorithm using flowcharts and pseudocode, which are independent of a programming language. Therefore, the algorithm designer can use natural (everyday) language to think about the process without having to worry about the details specific to generating the program for a device. In the mouse activity, as the algorithm designer talks or thinks through the plan (pseudocode), the code cards can be used to plan an algorithm; therefore, the code cards function as flowcharts. The programmer will translate the plan into a set of instructions (i.e., code) the machine can perform (e.g., move forward, back, pivot left, pivot right). The programming environment provided is the set of buttons on the back of the mouse. With this interface, the programmer can input a sequence of instructions for motion into the robot mouse's memory, clear all instructions if a mistake is made, and request to compile and execute the code. The motion buttons contain multiple indicators to indicate action, an arrow pointing in the direction of motion to be performed, and a unique color. The buttons multiple indicators are important because they have two symbolic (linguistic) labels that are equivalent in meaning (e.g., forward = blue, backward = yellow, left = purple, right = orange). The code is stored in memory until the programmer is ready to execute the code. When a command to execute is performed (with the green "go" button), then the code is translated into operations that control the mechanical hardware (e.g., move the motors for a specific amount of time in a specific manner). The output of the code is the movement of the mouse on the physical course, which the programmers will use to test and debug their code. They will test their code by observing if the mouse successfully moves through the course maze as planned, and they could debug their code if they notice mistakes. If the programmers use the code cards (which are analogous to a flowchart), then they will have an added resource (or representation) to help them manage their cognitive load while programming and debugging.

The first day of the enrichment program had students playing the *game as intended* – meaning that designers from Learning Resources designed this task. This served as an introduction to coding and to the robot-mouse coding environment. The students were given a very brief overview of the basics on how to play the game and use the mouse as described earlier. The goal was to have the pair of children use the tiles to build the physical course from a map on the selected activity card. They were provided some scaffolding from the researchers to lay out a coding plan using the code cards, program the mouse to traverse the course using the buttons on the mouse's back, test their code by having the mouse execute the code, and debug their code by evaluating the success of the mouse to achieve the goal of finding the cheese and following all criteria (e.g., path includes passing through all the tunnels before reaching the cheese). Children could reprogram their mouse when needed and try again. Our goal was for each child to play the roles of course builder, code planner, and programmer. Throughout approximately 43 min, each student needed to play each role at least two times as they progressed through increasingly more difficult activity cards. More difficult activity cards involved longer routes, more path options, and more sub-goals to perform according to the rules of the game. The students used the activity cards from the set in this order: 2, 4, 5, 8, and then any card bigger than 10 (except 12), which increased in complexity.

The second day of the enrichment program had students engage in translation activities that extended beyond the basic intent of the game. Pairs of students rotated through four stations, but only the three second-grade students were part of the study. (The fourth station was just a chance to let them free play with the game and did not provide the same intentionally structured task environment.) We were interested in how the students interacted with a variety of representations as they performed the CT tasks. Therefore, using the same Code and Go™ Robot Mouse Activity Set, the research team designed additional activities for the second session that required the students to use and translate between representations in different ways. The three structured task interview stations were (1) design a course (with game pieces) and translate it to drawing a map (activity card), (2) follow code cards to place cheese at the final location, and (3) use a map to code the mouse by pushing the buttons. Images of each task are included with the results. The *design-course-and-translate-to-drawing-a-map* station was set up with the context of having the students making new courses for a friend who had finished the game and therefore needed new activity cards. The students designed and built a physical course together using the equipment provided in the game and then individually drew a map of their course on an activity card template. The *follow-the-cards-to-place-the-cheese-on-the-course* station had students interpret an algorithm defined by a sequences of code cards laid out next to an already constructed physical course. The students were asked to determine where the cheese should be located if the mouse follows the path indicated by the cards. The students followed the directions laid out by the code cards to track the path of the mouse, placing the cheese where the mouse would end its path. They performed 2 rounds with 10 cards on the first round and 14 on the second round. At the *coding-directly-from-the-map* station (number 3 above), the students were given an activity card and the mouse but could not view the physical course for reference. Students were asked to program the mouse to reach the cheese. After programming the mouse, they tested it on the physical course (which was hidden from view during their programming). Debugging occurred back at the activity card and away from the physical course. Students had an opportunity to do this for two activity cards, with the second being more difficult. Each activity was presented at a different station in the same classroom, monitored by a different project researcher, and designed to make specific representational translation moves, which will be explained in the results section. At each station, students had approximately 15 min to complete the tasks before they moved on to the next station.

## Data Collection and Analysis

The task-based interview approach used in this study needed to capture observations and actions that occurred during the interview (Goldin 2000). Therefore, data were captured at each station with both audio and video recording devices across the two sessions. One video camera on a tripod and one audio recorder on the table were at each station for the entirety of each session. The audio was used as a supplement to the video data in case the video recorder did not pick up the spoken words of the students. These qualitative data focused on the three second-grade students as they worked through the four CT tasks described above. For each task, the students sat down with one of the three researchers, who set up and explained the task at hand and asked them questions to probe their thinking around the task as they were working.

Analysis of the data followed a method of constant comparative analysis (Corbin and Strauss 2014) that included an iterative process of coding, comparing, and condensing the data to allow for the emergence of patterns related to how students used and translated between representations within and across the four tasks. Analysis started with the 177 min of video recording data that were collected across the four tasks over the two sessions. The recordings were discussed and coded by four researchers across three rounds of coding using a priori codes that were based on the representation framework of the LTM plus gestures described above. The representation codes included concrete (which includes the physical course, the mouse movement, the mouse buttons, and any additional objects the students use), pictorial (which includes the course map and any drawings the students made), symbolic (which includes the code cards and any inscriptions students made of the symbols on the code cards), language (spoken or written words), and gestures (body movements used for representational purposes). The first round of coding was completed by one researcher and intended to capture evidence of how the students used existing and invented representations while they performed the interview tasks. The initial findings were discussed with two other researchers who also watched and analyzed the videos. The three researchers again coded not only how the students used existing and invented representations but also the strategies that individual students followed as they translated within and among representations. Finally, a fourth researcher coded the videos, looking for the elements previously identified by the research team. All researchers coded to consensus. The audio recordings were used to triangulate the video data, especially related to student talk when it was not clear from the video data what was being said by the students. From the coding, themes were developed. The results and discussion are presented first from each task interview then from the themes.

### Limitations

In this study, our analysis included three second-grade students. Although this allowed us to deeply analyze these students, we cannot make claims that extend to all students. Additionally, the students self-selected into the program. We collected our data over a limited time period of two 1-hr interactions with the students that were 1-week apart. During their task interviews, the students worked closely with an adult, which might have affected their actions. Often, in spite of prompting from the adults, the students often did not explain their actions or thoughts, which limited the claims we can make about their actions. Finally, although the robot mouse activity set closely mirrors many aspects of common programming languages, it does have some limitations, such as not having looping capabilities, which limited the tasks that we could lay out for the students.

## Results and Discussion by Activity

In this section, we describe the representations that students used and translated among during each of the four activities and how that related to the CT tasks. Figures display the representations the students used during each task. In each figure, the first diagram illustrates the intended path of translation between representations associated with performing an activity. The second diagram illustrates the various paths students demonstrated during the task.

### Introductory Activity: Game as Intended

In the introductory activity, the researcher explained to the students how to play the game as it was originally designed (see Fig. 2) and how to use each representation (map [activity card], physical course, code cards) within the game. The students were provided scaffolding to help them directly translate between the representations as shown in Fig. 3, first using the map to build the course, then using the course to define the solution path with the code cards, and finally using the code cards to program the mouse by pressing the buttons. Some translations were fluent; however, others required transitioning through alternative representations to navigate the translation. This section describes the specific representations used and translations expressed by the students while playing the game.

Two translations were fluent for the students – (1) from the map (activity card) to the course and (2) from the code cards to the mouse buttons. In the first translation, the students collected the information from the picture on the activity card (pictorial) to build the physical course (concrete). Although there were a few instances where they misinterpreted the map and thus misplaced some course pieces (fences, tunnels, etc.), the students were able to move fairly smoothly from the map picture to the physical course. Often students were able to build the course fairly independently by looking at the picture and then putting together the pieces. In a few instances, as seen in the below excerpt, students used spoken language between each other and with the facilitator and used gestures as they moved from the map picture to building the physical course.

> Jason: [starts to build]. So, like that... [pauses and looks from picture to physical course]
> Lily: Remember it is... [points to and counts on the picture] one, two...two straight [gestures to course]
> Jason: When you tell me to do it, then I don't do it. [keeps building]
> Lily: Oh, you're doing the outside first?
> Jason: Yes. [laying down the purple borders] One, two. [fills one square and picks up two more pieces] Ok, I messed up... [picks up the piece, pauses, looks at the picture]
> Lily: No, you didn't.
> Jason: Ok. [keeps building the last pieces, finishes the course and places the mouse in the correct place].

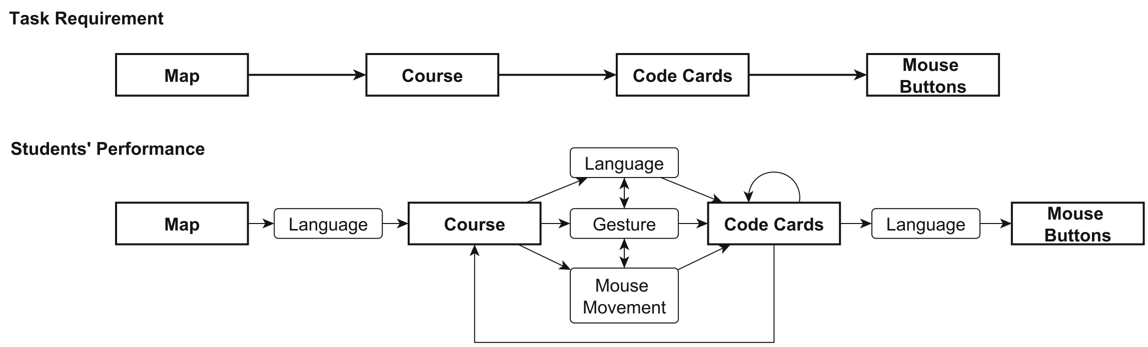


**Fig. 3** Translations between the representations used by the students in the game as intended activity. Figure shows the contrast between the representational translation of the task and the student translations

The students' translation from code cards to the mouse buttons was also fluent, and they used language and gestures to explain their actions as they worked to use the code cards to program the mouse. During this part of the task, students often looked at the code card, spoke its color or direction aloud, and then moved over to pushing the button on the mouse. Here you can see Beth using language and gesture as she enters the program on the mouse from the code cards:

> Facilitator: So, two forwards and a purple, what next?
> Beth: [Pointing to the forward mouse button.] You press one (pause) one, there [pressing the right turn].

During the playing of the game, the most cognitively challenging translation for the students was from the course (concrete representation) to the code cards (symbolic representation). This part of the task asks students to determine a sequence of code cards that represent the movement of the mouse from its starting position to the cheese. In order to generate this algorithm, students needed information from the course, such as the direction and number of steps as well as the orientation of the mouse during each move. Students frequently used concrete representations (mouse and gestures) as intermediate steps while completing this task. For example, Lily identified the code cards for her algorithm by moving the robot mouse on the physical course and then said the needed code card aloud as she physically moved the mouse.

> Lily: So, it needs to do this [puts right-turn code card in front of her, pauses, and looks at the course].
> Facilitator: Turn right.
> Lily: [Nods. Picks up mouse and moves it while talking] Then go straight [puts the mouse down on the course and picks up a forward code card that she places next to the right turn card. Picks up the mouse again]. Then go straight again [moves the mouse one space, then puts it down. Places another forward card in her sequence]... [pauses]...[looks at the course] Wait, what? [Confused, she pulls the correct forward card away from her sequence]. Does it go straight again? [Picks up the mouse and physically moves the mouse in the next direction, but her sequence is now off because she removed a code card but did not move the mouse back to the correct spot]. Then turn purple. [Puts mouse down, gets a right turn card and continues moving and coding the mouse].

As they moved through this challenging translation, students were seen using different types of representations to get them from where they needed the mouse to go on the physical course to laying out the algorithm with the code cards. Both Lily and Jason placed the mouse on the course or used their hands as mock-mouse movement and, based on that movement, chose the code card. Beth preferred to lay the code cards directly on the course as a concrete representation of the mouse's path. While the three students were determining which code cards to place, they all used language to represent their thinking, either spontaneously or promoted by the facilitator. For instance, Jason said, "I know what to do. I need two blue ones [referring to the forward arrow's color], one that goes this way [gesturing to right]...a purple... then two more blues."

The robot mouse activity contains several representations children must interpret to accomplish the task of representing a solution as a sequence of well-defined steps, i.e., code. These major representations include a physical course (concrete), a map of the course (pictorial), and code cards (symbolic), representing the discrete steps required by the mouse to complete the task. Coding the mouse by pushing the buttons is the final translation between the code cards (symbolic) and button pushes (concrete). The translation between each of these representations involves intrinsic cognitive load (Aurigemma et al. 2013; Goldin-Meadow et al. 2001; Thomas et al. 2010) that a learner must manage to accomplish a task and a level of element interactivity between representations. This intrinsic cognitive load is different between each of these major representations and may be simple for some students but difficult for others, which could be associated with the development of their representational fluency. The smooth translations from the map (activity card) to the physical course could result from the one-to-one alignment between elements in each representation. Each element (tile, tunnel, cheese) in the map has a corresponding element in the physical course. Therefore, once students obtain a frame of reference that aligns the two representations, there is very little additional information that needs to be interpreted between the two representations. The main challenge we saw in the students was assembling the physical course, which could be a motor skill and/or an issue of evaluating the interlocking system of the tiles. The researcher assisted with assembly of the course and not with interpretation of the map. Similarly, the code cards and the mouse buttons have a one-to-one pictorial alignment, which puts little load on working memory to process. Also, the element interaction at each step is extremely low and only requires the matching of a card to a button.

A major observation from this task for our three participants is that they invented various intermediate representations, or cognitive supports, to help them manage their cognitive loads. The process of generating an algorithm to navigate the space required managing the current location of the mouse, locating the end goal, choosing the next step, and remembering each of the steps along the way. This combination can increase the load on a student's working memory (Paas et al. 2003). Our findings illustrate that the source of the cognitive load is the element interactivity within the task and how students invented strategies to manage this load relative to the task (Paas et al. 2003). Also, as students develop more experience with the activity and practice with the representations, we may begin to see a reduction in the need for these intermediary representations, showing a maturity for the specific task and potentially representational fluency (Paas et al. 2003; Sweller et al. 1998). This idea is discussed further in the section on embodiment.

## Course Translated to Drawing Map Activity

Drawing the map from a physical course designed by the students is a spatial reasoning task where students need to translate from the physical course to a map drawing that they create as shown in Fig. 4. This complex task required the students to decompose the drawing task into discrete actions, and students used different decomposition strategies to manage this task. For example, Jason drew one square grid at a time, and as he progressed to each square grid, he added elements within each grid, such as walls, action lightning bolt, cheese, etc. Beth and Lily had a slightly different approach as they drew the entire course board (matrix of tile pieces) before adding in the other elements of the physical course. Beth started drawing at the position of the mouse, but after this initial step, she randomly drew the elements of the course on her activity map with no apparent pattern to which course element she added next. Lily drew all the same elements at the same time, starting by identifying all the lightning bolts, then the maze walls, and finally the tunnels.

Drawing the details of the course on the map is a process of abstraction where students needed to interpret and translate the information from the concrete representation (physical course). They collected information by counting the number of grid pieces and elements in the physical course, identifying how to orient and position each object, and translating that information in order to draw the map. To do this, the students relied on the position of the mouse as their frame of reference for the other elements. Initially, Jason and Lily were standing on the same side of the course, which was on the opposite side from the mouse. Lily changed her position by walking around to the other side of the course so that her perspective was the same as the mouse's orientation. Figure 5 shows the initial and final positions of both students and how the maps that they drew differed. Lily and Jason were at the same station and were therefore drawing their maps from the same physical course. However, while Lily's map correctly represented the physical course, Jason misrepresented the position of the objects, drawing a "mirror" image of the physical course. He was not facing the same direction as the mouse in his drawing and, therefore, had to rotate the physical course in his mind in order to draw the map. Jason drew the tunnels 90° differently than they should have been, which would have been correct from his own perspective

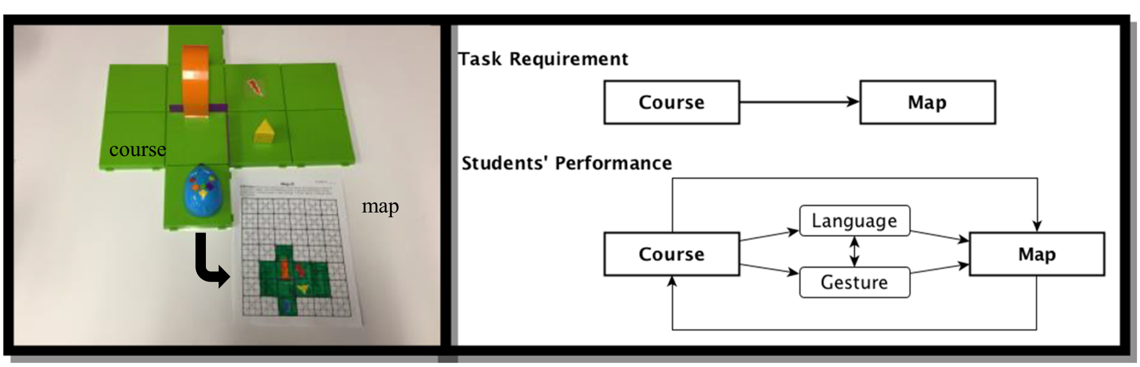


**Fig. 4** Translations between the representations used by the students in the course translated to drawing map activity showing the contrast between the representational translation of the task and the student translations

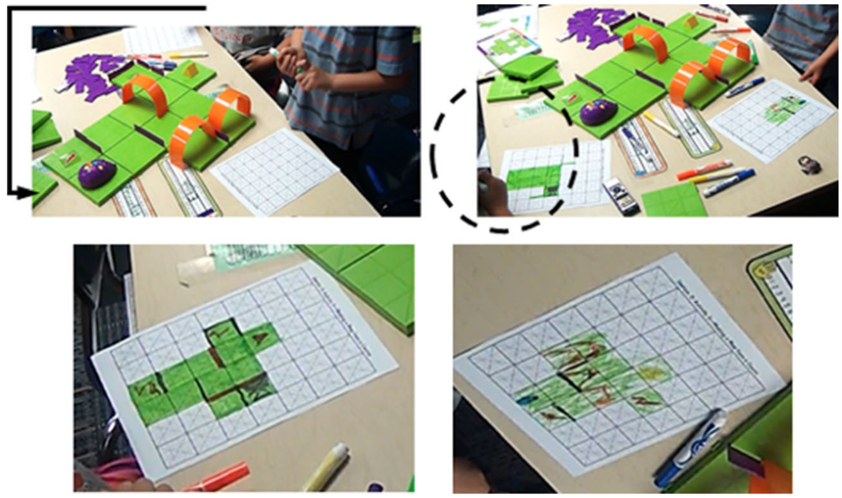

**Fig. 5** Lily and Jason's initial positions for the map drawing activity (top left). Lily (outlined in dotted circle) and Jason's final positions (top right). Final maps drawn by Lily (bottom left) and Jason (bottom right) representing the same invented course

but not from the mouse's orientation. Furthermore, he represented them on his paper as curved tunnels rather than the straight lines that the game maps and the other students used. However, Jason did debug his map drawing when he was finished constructing it. As the facilitator was collecting his map, Jason stood up and looked at the course again, saying, "Wait, I messed up," and pointed to one of the tunnels that was incorrectly placed in his drawing and tried unsuccessfully to correct it.

Generating a pictorial representation of a physical course can also be quite cognitively challenging for the young students. Accomplishing this task involves negotiating multiple possible moves as demonstrated by the children in this study. Jason appears to use a strategy of decomposing the drawing task into drawing each individual object in the set (tiles, walls, tunnels, and cheese), starting at a single point of reference (the start). From there he switched his attention between each adjacent object as he moved from the starting position. Alternatively, Beth and Lily appeared to segment the task by objects, starting with the largest orienting feature first, the matrix of tiles that are the base of the course. From there the girls filled in the smaller elements at the appropriate relevant location. This difference among these three cases suggests multiple methods for organizing the objects into sets and then systematically operating on the sets. Beth's method of organizing objects by types could be viewed as a very systematic process that lowers the element interactivity. That is, her working memory only needs to focus on the next element in the set and its location in a set. This process of structuring elements by type and processing each is a common strategy a programmer might use to respectively step through data. Lily's method is similar but part way through the process her approach appears more random. Since working memory is of limited size (Miller 1956), it is possible that Lily lost track of objects already processed; therefore, the algorithm is less efficient with respect to time. Jason appears to be organizing his thinking at a very local level to a matrix tile and processing all information local to that tile before moving to the next tile. This organizational style also gives some precursors to computational thinking that will transfer well to designing automated methods for systematically processing a large amount of information. Deciding if one method is better than another is difficult to determine from three case studies and a single intervention. However, this activity and observation raises questions like, "Does one method of organizing the task lead to a decomposition process that will be more productive in other tasks?" or "Does one approach lead to a higher representational fluency?". Future research can help to explore these questions in more detail. However, the current study does show that these young learners can invent a process to manage the generation of an abstract representation (map drawing) of a physical model. Further, their inventions are analogous to basic algorithms associated with autonomously processing data with a machine. The data presented here support and add to the study by Fessakis et al. (2013) in which they found young children sometimes struggled with orientation-related issues in coding environments. Here we see some of the same issues related to orientation but also more issues when trying to relate a concrete physical environment to the creation of a drawing of the environment.

## Follow Cards to Place the Cheese on the Course Activity

In the *Follow the Code Cards Activity*, the students needed to transfer between a preset series of cards and the course (Fig. 6). The only way in which students directly transferred between these representations correctly was when they used their hands to point to both the code card and

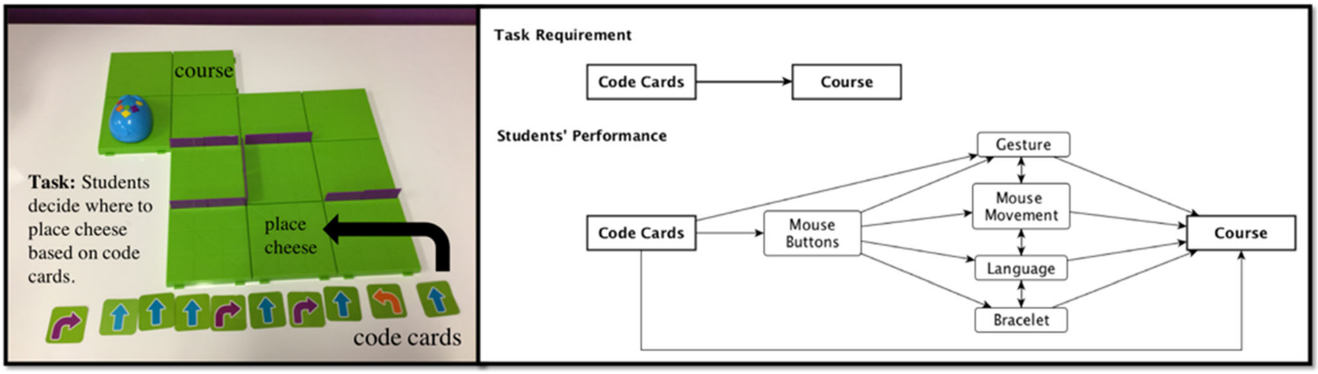


**Fig. 6** Translations between the representations used by the students in the Follow the Cards to Place the Cheese on the Course Activity showing the contrast between the representational translation of the task and the student translations

the location on the course in unison to follow the path. More often, the students added in representations to aid their transfer. The students primarily used the mouse and their hands as concrete representations as they traced the algorithm defined by the code cards. They used these representations to track the mouse's movement on the course and followed the code cards one step at a time, tracing both the direction and position of the mouse. When they were not able to use the concrete representations directly, the students developed substitutes. For example, when Jason was told that he could not use the mouse directly on the course, he grabbed a bracelet toy that he had been fidgeting with and said:

> Jason: Wait, I want…I think I can do it. [Places the mouse in one hand directly in front of himself on the table and his bracelet in the other on the course. He aligns them in the same direction]. So, boom [moves both mouse and bracelet forward one space. Looks at next code card]. And then boom [turns both mouse and bracelet, looks at next code card]. Boom [moves them both forward one space, looks at next code card]. Boom [turns both, looks at next code card].

However, the use of this representation was frustrating for him, and at one point, he insisted, "It's too hard. I have to use the mouse." In addition, the students used their language to support their transfer. For example, when Beth was prompted by the facilitator to explain where the cheese would be, the following occurred:

> Beth: [Pointing first to the code card on the table with one hand.] It goes forward. [Places the pointer of her other hand on the course and moves it forward one square before looking back at the code cards.] Then she turns [moves her finger along the course] then goes forward…
> Beth continued to quietly talk through each step, mostly to herself, as she pointed to the code card and then looked to the course and pointed the location where the mouse would be on the physical course.

This task required students to simultaneously track multiple pieces of information with each step. A student needs to keep track of which discrete step they are executing in the code cards, encode what the next step requires, execute the action on the physical track while remembering the location and direction of the mouse, and then transition back to the card to get the next instruction. Maintaining the state of each of these elements can overload the child's working memory, resulting in their need for physical pointers as reminders of where they are in the process. This process represents what a computational machine must do to process the information represented by the coding cards. However, humans can use other strategies to execute the code cards by finding patterns they can chunk into a single action. For example, seeing a pattern of repetitive steps (e.g., five forward steps) in the coding cards could be transferred to a simple move of counting tiles on the physical map. This action reduces the number of times the student needs to refer back to the original representations. From a CT perspective, the student is learning to find patterns in the sequence, which decomposes the algorithm into fewer discrete actions. Therefore, the student could be developing representational fluency by learning more about how to think with the representation (e.g., finding patterns in the sequence). These skills of noticing patterns and generalizing them into simpler repetition patterns are fundamental to computational thinking, and they are common skills observed of experts who chunk information to support their thinking on future tasks (e.g., Miller 1956). These skills are also needed for understanding code and the ability to manage debugging. The work we see from the three students in our study supports the idea that CT requires students to solve problems algorithmically and develop a level of technological fluency and language (Papert 1980; Bers 2010). However, in this study, the three students did not demonstrate finding a pattern and chunking it into a single rule, which would have reduced their cognitive load.

### Coding Directly From the Map Activity

In the *Coding Directly from the Map Activity*, the students were tasked to transfer between a map drawing and pressing the buttons to make the mouse move, which is a complex pictorial-to-concrete translation. In reality, the students did not directly translate correctly between the intended representations as illustrated in Fig. 7.

The students used other representations as intermediaries between the map and buttons. For example, they transferred between the map drawing and the physical mouse, as a way to track position, and then to the buttons, which is a pictorial (map) to concrete (mouse as a manipulative) to concrete (pressing a button) translation. They used the mouse as a concrete manipulative in a variety of ways, including using the mouse to trace a path on the map or on the table. They were also able to make translations through the representation of language by either talking to the facilitator or to themselves as they moved from the pictorial (map) to the concrete (pushing the buttons on the mouse). Additionally, the students used gestures in conjunction with the mouse and language representations. One way that Jason did this was by explaining the mouse's movements as he moved the mouse on the table:

> Jason: Ok, I can do this. So, turn [rotates mouse 90 degrees]. Go [moves mouse forward]. And then this way [rotates mouse 90 degrees]. And go [moves mouse forward]. Turn that way [rotates mouse 90 degrees]. Then go [moves mouse forward]. Turn that way [rotates mouse 90 degrees]. Go [moves mouse forward]. And that way [rotates mouse 90 degrees]. Go [moves mouse forward].

In another instance, Lily held the map in one hand and the mouse in the other, and as she moved through the path, she held the mouse's nose on top of the map, explaining the mouse's movements out loud. These examples demonstrate how the students used the representation of language to themselves, their partner, or the facilitator.

## Discussion by Theme

We identified five major themes across the CT activities that categorized our findings. The first three include how students aided their cognition with (1) concrete representations, (2) language, and (3) embodied movements. The other two categories relate to their developmental level observed in (4) various levels of representational maturity and (5) difficulty with spatial orientation related to performing a task. The following section describes each of these in more detail.

### Students Used Concrete Representations to Ease Translations

The students most often used concrete representations to provide perceptual cues as they moved from one representation to another. When students struggled to complete the task or if one of their attempts failed, they increased their use of concrete representations. These concrete representations were used as tracking mechanisms within the physical space (location) and orientation during these tasks, and students often created their own concrete representations that were beyond the requirements of the translation they were attempting to make.

The ways we saw the concrete representations being used were often intertwined. For example, in the *Follow the Code Cards to Place the Cheese Activity*, at first, the students tried to follow the path by simply looking at the cards and the course. When this did not work, they often used a concrete representation, such as the mouse, their hand, or other item (e.g., bracelet) to help themselves follow the movements of the mouse. Another instance occurred in the *Coding Directly from the Map Activity*, the students used the mouse to track the

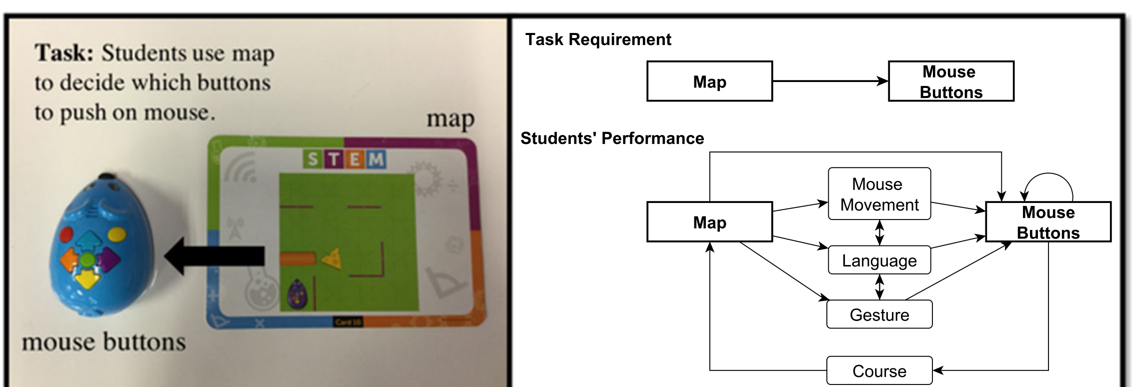


**Fig. 7** Translations between the representations used by the students in the coding directly from the map activity showing the contrast between the representational translation of the task and the student translations

motions based on the map, either by moving the mouse on the table next to the map or pointing the mouse's nose right on the map. These representations became more abstract as the students become more comfortable managing the task. For example, after some initial success, the students tracked the mouse's motion less closely, such as only turning the mouse but not moving it forward or by increasing the distance between the mouse and the other representations, such as moving it higher off the course or further from the map.

### Students Used Language as a Scaffold Between Translations

Another theme we identified was around the students' use of language while they worked through the tasks. Often, when the students were translating between representations, they used language, both to themselves and to others, as a way to scaffold their translations. For example, when they translated from the representation of the code cards to the buttons on the mouse, they often said the color aloud or when they were tracking the movements of the mouse on the course:

> Facilitator: And tell me which buttons you're pushing and in what order.
> Jason: Blue, orange.
> Lily: I basically pressed orange, then blue, and then blue again. And then you press squeak *[i.e., the action button]*, and then you press purple, and go like that. And then you go, blue, blue and then you go purple.
> Jason: [talking over Lily] Ok, I did it.
> Lily: And it goes like that. And you go blue, blue.

These uses of language were often invented by the students, not given to them by an adult or the game itself. Although sometimes the students were prompted to use language to explain what they were doing, they also used language in unprompted instances. These examples suggest that students were using their language as a tool to aid their problem-solving and computational thinking (Alarcón-Rubio et al. 2014; Fessakis et al. 2013). Their language is another instance of their use of representations to balance the cognitive load of the computational thinking tasks they were working through (Goldin-Meadow et al. 2001; Thurnham and Pine 2006). Similarly to their use of concrete representations, their language use became more abstract as they became more comfortable with the tasks, demonstrated by their language becoming quieter, more internalized, and less specific.

### Students Used Embodied Movements as Representations or to Assist With Translation

Students used gestures, such as pointing, to support their tracking of or translating between representations. For example, in *Follow the Code Cards Activity*, to help herself keep track of where the mouse would be on the course based on the code cards, Beth followed the cards with one hand and the location on the course with the other hand. Pointing helped her to keep track of where the mouse would be and track which code card she was using. However, this strategy broke down when her arms crossed over each other and she could no longer follow both paths without pausing her pointing, and she was unable to continue. In other situations, students used an object to track and simulate actions. For example, in the *Follow the Code Cards Activity*, Jason was told not to use the mouse to track the movements and reverted to using a bracelet to act as a placeholder for each movement as described above. This was likely another strategy Jason enacted to reduce the cognitive load of the task. However, he quickly became frustrated as the bracelet did not work as well as the mouse since it was difficult to track the orientation of the bracelet as compared to the mouse. Therefore, the students used their hands, the mouse, and the bracelet as both a marker for their location in the problem space and as a marker for the direction they were heading as they completed the task. These memory aids chunk information into one visual object, which helps the child keep track of the information as they translate between two representations and helps to reduce the cognitive load of the translation between representations (Aurigemma et al. 2013; Goldin-Meadow et al. 2001; Thomas et al. 2010).

The development of computational thinking pushes students to develop abstract concepts, which may be supported with various strategies to manage cognitive limits. From the perspective of embodied cognition, physical placeholders or manipulatives may support students in developing representational fluency in computational thinking tasks. The literature mentioned previously highlighted important connections between body movements and learning of abstract concepts (Wellsby and Pexman 2014). Hostetter and Alibali (2008) saw in their research that individuals who were simulating physical actions in their mind often used gestures to track or imitate the action they were simulating. They also found that individuals with stronger spatial skills were more likely to use gestures. While we may view these sorts of strategies as reducing cognitive load in the translation between computational thinking representations, an alternative view may be that the student is developing stronger spatial skills in enacting a third representation through gesture. Just as the use of manipulatives in mathematics and science education have shown to aid learning (Dienes 1960; Suh and Moyer 2007), such manipulatives may aid students in developing representational fluency in computational thinking. Bers (2018) also notes the importance of allowing young children the opportunity to experience the concepts or big ideas of computational thinking through hands-on play and manipulatives and suggests the use of programming platforms similar to that of the robot mouse that allow for the use of physical manipulatives.

### Levels of Representational Maturity Varied With the Difficulty of the Task

We saw varying levels of representational maturity throughout the tasks, showing evidence that the students were better able to utilize representations in some activities over others. By saying "representational maturity," we do not intend to claim that students developed a mature level of representational fluency by performing the proposed CT tasks. Instead, we refer to how students used representations and translations on a maturity continuum in each task. When the tasks were straight forward for the students, we saw that they could do the translations directly. However, when they began to struggle either because the task had multiple steps or when they could not get the translation directly, we saw that students often reverted back to concrete representations as intermediary tools. As an illustration, recall that Jason had gone through several representations during the *Follow Cards to Place the Cheese on the Course Activity*. He used his bracelet (concrete) and the mouse (concrete and realistic) while at the same time talking through the steps (language) in order to figure out what the cards (symbolic) were indicating. We saw these differing levels of representational maturity manifest during times of ease versus times of difficulty. This is not surprising as learners must be able to utilize many skills to thoroughly use representations (Ainsworth 2006).

The use of additional representations to aid in transfer was particularly evident when students were having trouble transferring between two different representations. In the beginning, the students often added in other representations to support their transfer. The *Coding Directly from the Map Activity* is not a natural translation. In order to manage the task, students used self-talk as they pressed the buttons while at the same time also using the mouse as a concrete manipulative as a way to manage the location of the mouse on the map. Once they seemed more comfortable, they began to take away the additional representations. For example, they often stopped the self-talk that explained their steps as they were pushing buttons. However, on some of the more difficult translations, students continued to use the intermediary concrete representations to manage their task. In these examples, we observed different levels of representational fluency (Greenes and Findell 1999; Johri and Olds 2011; Lesh and Zawojewski 2007; Suh and Moyer 2007), which corresponded to the complexity of the task.

### Spatial Orientation Was Difficult for the Students

The importance of perspective can be seen very clearly in the *Course Translated to Drawing Map Activity* when the students are drawing their own map. In this activity, Lily and Jason built their course together and individually drew their maps. When they were first handed their paper to draw on, they were standing on the same side of the table, which happened to be the opposite side from where they had placed the mouse's starting location. Before starting her drawing, Lily walked around to the other side of the table, aligning her perspective with the initial perspective of the mouse, saying aloud that it would be easier that way, while Jason remained opposite the mouse and had less success with this task. However, in a different activity, when Jason was unsure of where the mouse would go, he moved around to the other side of the table and tilted his head to be in the mouse's perspective. In these scenarios, the students were unable to complete the task when they were in a different spatial orientation from the mouse, suggesting that their mental rotation ability was not sufficiently developed, and thus they needed to physically move their body instead. Similarly, the students were also seen using the mouse or their hand as a marker or spatial orientation tool as they engaged in some of the more complex tasks that required them to be able to visualize the mouse's position and movement in their head. This supports other work that states that young children often rely on more concrete representations to support their spatial reasoning (Ebersbach and Hagedorn 2011; Harwood and Usher 1999). For example, in the *Follow Cards to Place the Cheese Activity*, Jason used his hand to represent the mouse when he was restricted from using it to assist with the spatial orientation aspects of the task. Lily was seen using pointing as a strategy to help with spatial orientation as she was able to follow the symbols with her finger when they were in a straight line, but when the mouse turned, she used the mouse to follow along in order to keep track of the mouse's perspective. Our results support the literature that students' spatial reasoning abilities differ from adults (Thommen et al. 2010) but that they have the skills to improve their spatial reasoning with practice and support (Feng et al. 2007; Uttal et al. 2013).

## Conclusions and Implications

Students in this study demonstrated that the translation between each of the various representations highlighted in these four computational tasks involves an intrinsic cognitive demand that a learner must manage to accomplish a task. For the more difficult translations, we saw students invent various intermediate representations, or cognitive supports, to help them manage this load. The major observations for our case studies of three second grade students illustrate the translations between representations that accompany different computational tasks and how students invent strategies to manage them. Understanding these strategies may help educators or curriculum developers embed scaffolds to support representational transitions that provide students opportunities to offload some of the cognitive load associated with these translations. Also, we expect that, as students develop more experience with computational thinking representations and translations, we would see a reduction in the need for intermediary representations, which may indicate that the students' representational fluency is maturing.

Future research could include investigating the themes and actions presented here with a larger population to confirm if the strategies are generalizable beyond the three students we observed. Similarly, we could conduct similar task interviews with different coding platforms to understand how the physical or virtual environments cue students' representational and translational strategies. We believe that such results could be helpful in the development of curricula and pedagogical strategies for supporting students' computational thinking development.

**Funding Information** This material is based upon work supported by the National Science Foundation under Grant No. 1543175. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## Compliance with Ethical Standards

**Conflict of Interest** The authors declare that they have no conflict of interest.

**Ethical Approval** All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. This article does not contain any studies with animals performed by any of the authors.

**Informed Consent** Informed consent was obtained from all individual participants included in the study.

## References

Ainsworth, S. (2006). DeFT: a conceptual framework for considering learning with multiple representations. *Learning and Instruction*. https://doi.org/10.1016/j.learninstruc.2006.03.001.

Alarcón-Rubio, D., Sánchez-Medina, J. A., & Prieto-García, J. R. (2014). Executive function and verbal self-regulation in childhood: developmental linkages between partially internalized private speech and cognitive flexibility. *Early Childhood Research Quarterly*. https://doi.org/10.1016/j.ecresq.2013.11.002.

Alibali, M. W., & Nathan, M. J. (2012). Embodiment in mathematics teaching and learning: evidence from learners' and teachers' gestures. *The Journal of the Learning Sciences*. https://doi.org/10.1080/10508406.2011.611446.

Aurigemma, J., Chandrasekharan, S., Nersessian, N. J., & Newstetter, W. (2013). Turning experiments into objects: the cognitive process involved in the design of a lab-on-a-chip device. *Journal of Engineering Education*. https://doi.org/10.1002/jee.20003.

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*. https://doi.org/10.1145/1929887.1929905.

Bers, M. U. (2010). The tangible K robotics program: Applied computational thinking for young children. *Early Childhood Research and Practice*. http://ecrp.uiuc.edu/v12n2/bers.html.

Bers, M. U. (2018). *Coding as a playground: Programming and computational thinking in the early classroom*. New York: Routledge.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *2012 annual meeting of the American Educational Research Association* Retrieved from https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Bruce, C. D., & Hawes, Z. (2015). The role of 2D and 3D mental rotation in mathematics for young children: what is it? Why does it matter? And what can we do about it? *ZDM*. https://doi.org/10.1007/s11858-014-0637-4.

Bruner, J. S. (1966). On cognitive growth. In J. S. Bruner, R. R. Olver, & P. M. Greenfield (Eds.), *Studies in cognitive growth: A collaboration at the center for cognitive studies* (pp. 1–29). New York: Wiley.

Byers, J. A., & Walker, C. (1995). Refining the motor training hypothesis for the evolution of play. *The American Naturalist*. https://doi.org/10.1086/285785.

Clement, J. (2000). Analysis of clinical interviews: Foundation and model viability. In R. Lesh & A. E. Kelly (Eds.), *Research design in mathematics and science education* (pp. 547–589). Hillsdale: Erlbaum.

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

Dienes, Z. P. (1960). *Building up mathematics*. London: Hutchinson Educational.

Ebersbach, M., & Hagedorn, H. (2011). The role of cognitive flexibility in the spatial representation of children's drawings. *Journal of Cognition and Development*. https://doi.org/10.1080/15248372.2011.539526.

Feng, J., Spence, I., & Pratt, J. (2007). Playing an action video game reduces gender differences in spatial cognition. *Psychological Science: A Journal of the American Psychological Society / APS*. https://doi.org/10.1111/j.1467-9280.2007.01990.x.

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 in Education*. https://doi.org/10.1016/j.compedu.2012.11.016.

Ginsburg, H. (1981). The clinical interview in psychological research on mathematical thinking: aims, rationales, techniques. *For the Learning of Mathematics, 1*(3), 4–11.

Goldin, G. A. (2000). A scientific perspective on structured, task-based interviews in mathematics education research. In R. Lesh & A. E. Kelly (Eds.), *Research design in mathematics and science education* (pp. 309–325). Hillsdale: Erlbaum.

Goldin-Meadow, S., Nusbaum, H., Kelly, S. D., & Wagner, S. (2001). Explaining math: gesturing lightens the load. *Psychological Science*. https://doi.org/10.1111/1467-9280.00395.

Greenes, C., & Findell, C. (1999). Developing students' algebraic reasoning abilities. In L. V. Stiff & F. R. Curico (Eds.), *Developing mathematical reasoning in grades K-12, 1999 yearbook* (pp. 127–137). Reston: National Council of Teachers of Mathematics.

Grover, S., & Pea, R. (2013). Computational thinking in K-12: a review of the state of the field. *Educational Research*. https://doi.org/10.3102/0013189X12463051.

Gunderson, E. A., Ramirez, G., Beilock, S. L., & Levine, S. C. (2012). The relation between spatial skill and early number knowledge: the role of the linear number line. *Developmental Psychology*. https://doi.org/10.1037/a0027433.

Harwood, D., & Usher, M. (1999). Assessing progression in primary children's map drawing skills. *International Research in Geographical and Environmental Education*. https://doi.org/10.1080/10382049908667613.

Hostetter, A. B., & Alibali, M. W. (2008). Visible embodiment: gestures as simulated action. *Psychonomic Bulletin & Review*. https://doi.org/10.3758/PBR.15.3.495.

Hunting, R. P. (1997). Clinical interview methods in mathematics education research and practice. *The Journal of Mathematical Behavior*. https://doi.org/10.1016/S0732-3123(97)90023-7.

Johri, A., & Olds, B. M. (2011). Situated engineering learning: bridging engineering education research and the learning sciences. *Journal of Engineering Education*. https://doi.org/10.1002/j.2168-9830.2011.tb00007.x.

Kalelioglu, F., Gulbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing, 4*(3), 583–596.

Kinach, B. M. (2015). Fostering spatial vs. metric understanding in geometry. *The Mathematics Teacher, 105*(7), 534–540.

Kita, S., Alibali, M. W., & Chu, M. (2017). How do gestures influence thinking and speaking? The gesture-for-conceptualization hypothesis. *Psychological Review*. https://doi.org/10.1037/rev0000059.

Learning Resources. (n.d.). Code & Go™ robot mouse activity set. https://www.learningresources.com/code-gor-robot-mouse-activity-set Accessed 9 July 2019.

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

Lesh, R., & Zawojewski, J. (2007). Problem solving and modeling. In F. K. Lester Jr. (Ed.), *Second handbook of research on mathematics teaching and learning* (pp. 763–804). Charlotte: Information Age Publishing.

Macedonia, M., & von Kriegstein, K. (2012). Gestures enhance foreign language learning. *Biolinguistics, 6*(3–4), 393–416.

Maher, C. A., & Sigley, R. (2014). Task-based interviews in mathematics education. In S. Lerman (Ed.), *Encyclopedia of mathematics education*. Dordrecht: Springer.

Miller, G. A. (1956). The magical number seven, plus or minus one: some limits on our capacity for processing musical information. *Psychological Review*. https://doi.org/10.1037/h0043158.

Paas, F., Tuovinen, J., Tabbers, H., & Van Gerven, P. W. M. (2003). Cognitive load measurement as a means to advance cognitive load theory. *Educational Psychologist*. https://doi.org/10.1207/S15326985EP3801.

Papert, S. (1980). *Mindstorms. Children, computers and powerful ideas*. New York: Basic Books.

Papert, S., & Harel, I. (1991). Situating constructionism. In S. Papert & I. Harel (Eds.), *Constructionism* (pp. 1–12). Cambridge: MIT Press.

Resnick, M. (2006). Computer as paint brush: Technology, play, and the creative society. In D. G. Singer, R. M. Golinkoff, & K. Hirsh-Pasek (Eds.), *Play = learning: How play motivates and enhances children's cognitive and social-emotional growth* (pp. 192–208). Oxford: Oxford University Press.

Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. In proceedings of the *6th ACM SIGCHI Conference on Creativity & Cognition* (pp. 1-6). ACM.

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*. https://doi.org/10.1016/j.chb.2016.08.047.

Roth, W.-M., & Lawless, D. (2002). Scientific investigations, metaphorical gestures, and the emergence of abstract scientific concepts. *Learning and Instruction*. https://doi.org/10.1016/S0959-4752(01)00023-8.

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

Stieff, M., Scopelitis, S., Lira, M. E., & Desutter, D. (2016). Improving representational competence with concrete models. *Science Education*. https://doi.org/10.1002/sce.21203.

Stull, A. T., Hegarty, M., Dixon, B., & Stieff, M. (2012). Representational translation with concrete models in organic chemistry. *Cognition and Instruction*. https://doi.org/10.1080/07370008.2012.719956.

Suh, J., & Moyer, P. S. (2007). Developing students' representational fluency using virtual and physical algebra balances. *Journal of Computers in Mathematics and Science Teaching, 26*, 155 Retrieved from http://www.editlib.org/index.cfm?fuseaction=Reader.ViewFullText&paper_id=22799.

Sweller, J., van Merrienboer, J. J., & Paas, F. G. W. C. (1998). Cognitive architecture and instructional design. *Educational Psychology Review, 10*(3), 251–296.

Thomas, M. O. J., Wilson, A. J., Corballis, M. C., Lim, V. K., & Yoon, C. (2010). Evidence from cognitive neuroscience for the role of graphical and algebraic representations in understanding function. *ZDM, 42*(6), 607–619. https://doi.org/10.1007/s11858-010-0272-7.

Thommen, E., Avelar, S., Sapin, V. Z., Perrenoud, S., & Malatesta, D. (2010). Mapping the journey from home to school: A study on children's representation of space. *International Research in Geographical and Environmental Education*. https://doi.org/10.1080/10382046.2010.496975.

Thurnham, A. J., & Pine, K. J. (2006). The effects of single and dual representations on children's gesture production. *Cognitive Development*. https://doi.org/10.1016/j.cogdev.2005.09.005.

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*. https://doi.org/10.1177/0963721413484756.

Wai, J., Lubinski, D., & Benbow, C. P. (2009). Spatial ability for STEM domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of Education & Psychology*. https://doi.org/10.1037/a0016127.

Wellsby, M., & Pexman, P. M. (2014). The influence of bodily experience on children's language processing. *Topics in Cognitive Science*. https://doi.org/10.1111/tops.12092.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*. https://doi.org/10.1145/1118178.1118215.

Wing, J. M. (2010). *Computational thinking: What and why?* Unpublished manuscript. https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf.