International Journal of Child-Computer Interaction 28 (2021) 100250

Contents lists available at ScienceDirect

International Journal of Child-Computer Interaction

journal homepage: www.elsevier.com/locate/ijcci

Research paper

How young children engage in and shift between reference frames when playing with coding toys

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

[a] Instructional Technology and Learning Science, Utah State University, Logan, USA
[b] College of Teacher Education and Leadership, Utah State University, Logan, USA
[c] Graduate School of Teacher Education, Stanford University, Stanford, USA

article info

Article history:
Received 18 March 2020
Received in revised form 29 October 2020
Accepted 3 January 2021
Available online 13 January 2021

Keywords:
Coding toys
Reference frames
Egocentric
Allocentric
Computer science
Early childhood
Syntonicity

abstract

This research explores reference frames and reference frame shifts among young children (ages 5 and 6) as they are learning to program with a commercial coding toy. To date, little is known about reference frames used by young children in toy-based coding. Video recordings of 16 children engaging in two programming tasks were collected. Results from an analysis of 240 min of video data indicate that young children engage in shifts of reference frames when learning to program, however, young children do not have a precise coordination system. The present study proposes two additional reference frames, ProtoEgocentric and ProtoAllocentric, that allow for the consideration of developmental imprecision in children. The most common reference frame shift observed was ProtoAllocentric to ProtoEgocentric, which accounted for 47.4% of all shifts across sites and contexts. Three cases are presented to illustrate reference frame shifts and how children's developmental imprecision affects their experience.

1. Introduction

As of late, a number of toys have appeared commercially that claim to help young children learn computer programming. A substantial number of them take the form of whimsical robots that a child programs to move forward or backward or to turn right or left while making sounds and lighting up (Shumway, Clarke-Madura, Lee, Hamilton, & Baczuk, 2020). For example, the Bee-bot is a robotic bee with buttons on its back that control which direction it turns and how many units forward or backward it will roll (Angeli & Valanides, 2019). Toys like Bee-bot are highly reminiscent of the Logo programming language (Papert, 1980), which involved a virtual turtle that could be programmed to move similarly through commands such as *fd 1, bk 1, rt 90*, and *lt 90*.

When Papert designed the Logo programming language, he introduced the idea of syntonic learning and that children could use their own body knowledge to think through the movements of the Logo turtle (Papert, 1980). The Logo turtle was ''an object to think with'' and had a natural mapping to children's intuitions because it was ''body syntonic''. In other words, the child programming with Logo should be able to make simple mappings between their own bodily movements and the movements of the Logo turtle.

These coding toys, and Logo, leverage embodied understandings of the child. When stuck on what movement to program next, a child should be able to think how they would move if they were in the same position as the turtle and become unstuck. Since Logo, many researchers have used embodiment and body syntonicity as designed means to help children gain a better understanding of computer programming concepts (Bers, Flannery, Kazakoff, & Sullivan, 2014; Città et al., 2019; Fadjo, Lu, & Black, 2009; Martinez, Gomez, & Benotti, 2015; Palmér, 2017; Raabe et al., 2015; Sung, Ahn, & Black, 2017).

These ideas about body syntonicity and its mobilization in learning activities echoes theories of embodied cognition that link action and perception and posit that abstract symbols such as words and numbers, or in the case of these toys, directional arrows, need to be grounded in bodily experience (Gibbs, 2005). Recent studies on tangible toys and robots integrate physical movement and programming by having children physically enact the code. For example, Bers et al. (2014) used this syntonic, physical approach in a study with 53 kindergarten children from three classrooms to better understand how the children developed computational thinking (CT), computer programming, and robotics concepts. Children played group games, such as Simon Says, to reinforce programming skills using physical movement. Specifically, the purpose of these activities was to prepare the children's understanding of action-correspondence. In this group game, the teacher called out 'codes' and the children used their

* Correspondence to: 830 Old Main Hill, Logan, UT 84342, USA.
E-mail address: jody.clarke@usu.edu (J. Clarke-Midura).

https://doi.org/10.1016/j.ijcci.2021.100250

bodies to enact the 'code' that the teacher stipulated. After this activity, the children engaged in robot programming tasks. In essence, the children began with physical enactment and ended with the more abstract robot programming task. Learning progressions such as this are found to be common in the early childhood programming literature, beginning with syntonic learning experiences and moving into more abstract programming.

Martinez et al. (2015) followed a similar theoretical perspective in their study with young children aged 3- to 6-years-old that examined participants' sequencing, looping, and conditional understanding of programs. The children first programmed each other and then programmed the robot. In this case, the younger children demonstrated proficiency in sequencing, but struggled to understand and apply looping and conditionals. Martínez et al. suggested that after the children completed the movement activity, they ''could clearly indicate the amount and orientations of arrows needed to make the robot get to target'' (Martinez et al., 2015), p. 162. This suggests that the embodiment was an important design element for the children to understand the later, more abstract programming concepts. This study emphasized syntonic learning and the benefits of the embodied experience of young programmers.

In addition to curriculum and tasks that employ this embodied framework, early childhood CT assessments are utilizing a similar embodied framework when constructing measurement tools (Città et al., 2019; Palmér, 2017). Palmér (2017) used physically enacted design elements in an assessment and intervention with eight preschool-aged children in a study on programming and spatial reasoning skills. Children engaged in tasks where they 'acted out' the robot and 'programmed' their peers. They then were given a Bee-bot robot coding toy that they then learned to program. Palmér found that children used spatial reasoning abilities as they engaged in the embodied tasks. Similarly, Città et al. (2019) used an embodied approach when evaluating the CT of young children aged 6- to 10-years old. The children acted out a program on a thematic carpet. Some of the children even wore a robot suit to help them visualize the path of the robot more realistically. Ultimately, they used this CT assessment data to compare the young children's mental rotation ability to their CT and found that the complex cognitive processes that underlie CT relate to young children's embodiment. Simply put, children who were better at mental rotations demonstrated better coding abilities as assessed through an embodied task.

Overall, we build on this work that involves physical movement and enactment, sensorimotor experiences, and embodied learning by understanding young children's frames of reference when they interact with a tangible coding toy that is supposedly body syntonic. Reference frames refer to the way that an individual perceives themselves in space and the connection between mental spatial mapping and physical actions. An egocentric or allocentric reference frame refers to the orientation and location of an object in space, either in terms of subject–object reference or object–object reference, respectively (Vogeley et al., 2004). Please note that this concept of egocentric reference frames is different from the Piagetian concept of egocentrism, which is often used in early childhood learning research (Piaget, 1927).

As broader context for our work, we are engaged in a multi-year project that involves designing tasks around robot coding toys for kindergarten-aged children (ages 5 to 6 in the U.S.) that are tangible and screen free. We designed a series of learning tasks and collected video data of children completing those tasks with a robot coding toy, the Fisher-Price Code-a-pillar. In the task implementations, we observed that the children engaged in shifts between egocentric and allocentric reference frames when they worked through the tasks. While past research has focused on syntonic learning and embodiment when learning to program, little is known about children's ability to use and shift reference frames between egocentric and allocentric frames. One exception is Smith, Berland, and Martin (2014), who created a programming environment called IPRO designed around first-person embodiment. They found that high school participants wrote more complex programs when they were able to shift back and forth between egocentric and allocentric frames of reference. Although these shifts helped high school youth in their programming, these results cannot be generalized to young children. It is unknown what reference frames young children embody and whether shifts in reference frames aid young children in programming. Nor is there research on how young children perceive the robot coding toys in physical space and how they orient themselves both mentally and physically to the coding toy and to the robot's path. Thus, the present study examines the ways young children reference, embody, and interact with the physical location of the coding toy. The following two research questions guide this work, (a) *What reference frames do young children adopt when learning to program,* and (b) *How do young children engage in reference frame shifts when playing with coding toys?*

2. Theoretical perspectives

This work is grounded in theories of embodied cognition that link action and perception and posit that abstract concepts need to be grounded in bodily experience (De Vega, Glenberg, & Graesser, 2008; Gibbs, 2005; Glenberg, 2008). What this means is that cognition is not abstract and amodal but is dependent on bodily experience (e.g., sensory motor systems, action) and grounded in context (Barsalou, 2008; Glenberg, 2008). The present study drew from the embodied cognition framework in the design of curricular tasks to teach kindergarten children how to program using screen-free coding toys (i.e., robots). Previous studies employing similar embodied approaches found that young children learn to program more effectively with higher levels of embodiment (Sung et al., 2017), and that higher levels of embodiment precipitates better retrieval and retention of knowledge (Johnson-Glenberg, Birchfield, Tolentino, & Koziupa, 2014).

2.1. Reference frames of young children

As a reminder, reference frames refer to the way that an individual perceives themselves in space, and the connection between mental spatial mapping and physical actions. Reference frames are typically divided into two main categories: egocentric and allocentric (Klatzky, 1998). Egocentric frames of reference are subject-to-object (Zaehle et al., 2007). They ''specify location and orientation with respect to the organism, and include eye, head, and body coordinates'' (p. 1, Ruggiero, Iachini, Ruotolo, & Paolo Senese, 2009). Klatzky (1998) highlights the importance of a shared heading of the object and the perceiver in order for an egocentric reference frame to be attained. This means that the angle of the object in space is the same as the angle of the perceiver in the same space, and they are both facing the same direction. To illustrate, imagine a work foreman gives you a personal tour around a new facility. Every room they show you and every hallway they walk you down, is described and referenced from their own direct spatial map and from a shared heading. This is an example of an egocentric frame of reference because the heading (physical alignment) would be shared between the spatial/physical map, the foreman, and yourself.

Allocentric frames of reference are object-to-object oriented and do not require a shared heading between the perceiver and the object (Klatzky, 1998; Zaehle et al., 2007). In the cognitive science literature, the terms world-centered and object-centered

Fig. 1. Code-a-pillar.

Fig. 2. Code-a-pillar's body segments that specify discrete codes.

have been used interchangeably, ''to refer to the representation of the spatial location of an object relative to that of *another* external object, independent of the ego's position or orientation'' (p. 2, Ruggiero et al., 2009). The term scene-centered has also been used in a similar manner (Foley, Whitwell, & Goodale, 2015). In essence, an allocentric frame of reference does not require the perceiver to share a heading. In the aforementioned example of the facility tour, instead of giving you a personal tour, the foreman may choose to describe the layout of the facility from a single, stationary position. In this case, the foreman would have to describe the location of the rooms and hallways in relationship to other areas of the building or to a map. This would be an example of an allocentric frame of reference because there would no longer be a shared heading (direct alignment) between the spatial/physical map, foreman, or yourself.

2.2. Imaginal updating

An important concept that arises in the reference frame literature is that of imaginal updating, which relates to the shifting between egocentric and allocentric frames of reference. Klatzky (1998) uses the term imaginal updating to describe the interplay between developing a new spatial map based on current or hypothesized locations of entities in space. Individuals have to draw upon egocentric relationships when developing spatial maps of areas in allocentric frames of reference (Gallistel, 1990). For example, for a perceiver to understand the distance from point A to point B, after point A has been rotated and translated, the perceiver must draw upon some original egocentric knowledge. The perceiver must access the original egocentric understanding of the distance to both point A and point B, and then apply that knowledge to the new distance from point A to point B after the rotation and translation have been made. In this sense, there is a constant interplay between referencing an egocentric frame of reference to develop an allocentric frame of reference. Humans use imaginal updating constantly throughout the day. Anytime an individual mentally relocates themselves or an object, they must create a new spatial map that accounts for the distances and spatial relationships that were known by the individual from an egocentric frame of reference.

3. Context of research: The coding toy and the computational thinking tasks

The tasks in the present study featured a commercially available tangible coding toy from Fisher Price called Code-a-pillar (Fig. 1). This coding toy is screen free and has removable body segments that each specify a code (Fig. 2). As the body segments (codes) are appended or removed from the Code-a-pillar body, the resulting program indicates the path that the Code-a-pillar will move. The research team created a large mat with a grid that aligned with the distance the Code-a-pillar moved when moving forward and backward, ensuring that one grid square measured exactly one forward movement and corresponded with exactly one unit of code. The mat allowed participants to perceive one code to one unit of movement, called action-instruction correspondence (García-Valcárcel-Muñoz Repiso & Caballero-González, 2019).

We designed 30-minute collaborative problem-solving tasks that involved programming the Code-a-pillar to move from one location to another. We opted for pair programming tasks based on research that suggests it supports learning (Sharma, Papavlasopoulou, & Giannakos, 2019). For the purpose of the present study, we focus on two tasks. However, these tasks are part of a larger research project focused on developing curricular resources and assessments for integrated computational thinking and mathematical thinking in early childhood (Shumway et al., 2020).

The general structure of each task was similar across sites. The structure started with eliciting prior knowledge (either existing knowledge or connected to prior tasks), then it involved engaging the children in a collaborative problem-solving task, and concluded with a reflection on learning. The primary role of the instructor in all the tasks and across both sites was to launch the task, probe for student justifications for their coding choices, and to redirect students back to the task goals if they veered away from the main goal. The present study focused on two consecutive tasks: *Code-a-pillar Introduction* and *Code-a-pillar Challenges*. We selected these tasks for data analysis because they were the first tasks in a five-task progression. Using the introductory tasks allowed us to investigate children's reference frames in the very early stages of learning to program.

3.1. Code-a-pillar introduction

In the first task, the teacher introduced each coding piece (Fig. 2) by asking children to predict what each code would command Code-a-pillar to do (i.e., *move forward, move backward, rotate left, rotate right*) and then test their predictions (Fig. 3). After testing the function of each code, the teacher asked the children to program the toy to move from one location to another location on the grid. One of these grid-based tasks was to have the students start on the leaf symbol (point A) and get to the flower symbol (point B). One correct solution for this task could have been *forward, rotate right, forward*. Another correct solution could have been *rotate right, forward, rotate left, forward*. At the end of the task, the teacher asked children what they learned during the activity.

3.2. Code-a-pillar challenges

The second task also asked children to program Code-a-pillar to and from various locations on the grid. However, the task was designed around ''challenges'' with constraints. In the first challenge, the children had the option of getting from point A to point B in any way they wanted (Fig. 4). In the second challenge, the children had to get from point A to point B, but the Code-a-pillar was required to pass through a designated

Fig. 3. Students testing each body segment (code) individually.

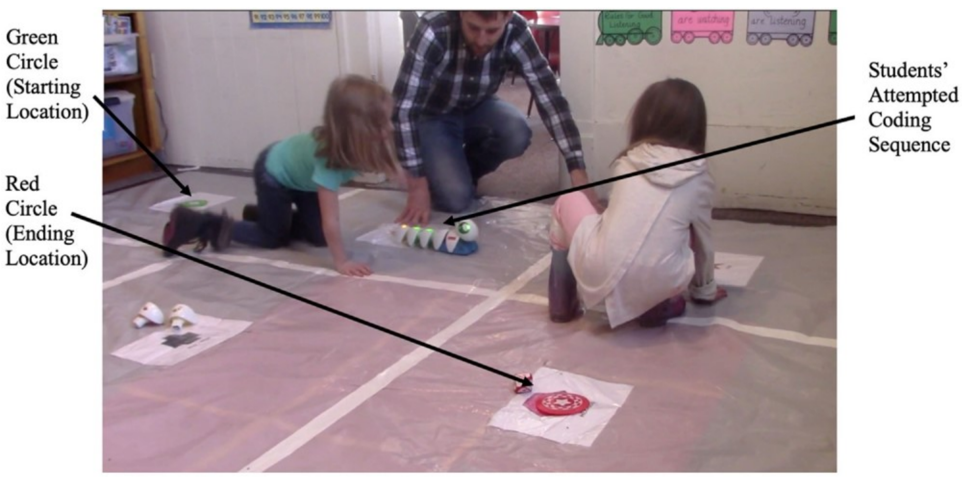


Fig. 4. Students watch code-a-pillar enact their program trying to get from the green circle to the red circle.

point C. Both challenges were designed to provide the children collaborative activities to engage in computer science concepts such as sequencing codes to build a program, decomposition, and debugging as well as mathematics concepts such as comparing, sequencing, spatial reasoning, and measurement.

Table 1
Gender composition of participant pairs.

Site	Group	Gender make-up
A	1	Female, Male
A	2	Female, Male
A	3	Female, Female
A	4	Female, Female
B	5	Female, Female
B	6	Male, Male
B	7	Female, Male
B	8	Female, Male

4. Method

4.1. Participants and setting

The sample consisted of 16 children (female = 10, male = 6, average age = 5) in a semi-rural area in the intermountain west in the United States. All of the children spoke English at home. Eight of the children attended a private pre-K/Kindergarten school (Site A) and eight attended a private afterschool learning center for children who attended half-day kindergarten in the public schools (Site B). Two separate school sites were selected to ensure results were not specific to one instructional context. At each site, the classroom teacher paired participating children for the activities. There were eight groups in total. Table 1 lists the gender make-up for each pairing.

In Site A, there was a separate room connected to the general education space. The researchers placed the large coding mat on the floor in this room for pairs of children to participate in the study activities. The researchers gathered one pair of children at a time and accompanied them to the separate room where the Code-a-pillar and large floor grid were set-up. Children in Site A participated in five separate tasks, one per week for five consecutive weeks.

In Site B, a private research room was located on the third floor of the education building. The large mat was placed on the floor in this rectangular room. The researchers gathered pairs of participating children from their classroom on the first floor of the building and accompanied them to the third floor to complete the coding tasks. Children in Site B participated in the same five tasks as the children in Site A, one per day for five consecutive days.

4.2. Procedure

One researcher, a former early childhood teacher, taught five 30-minute Code-a-pillar lessons to each pair of children. The role of the researcher was to introduce the tasks to the pairs and then support their problem-solving. Typically, the researcher would ask probing questions that would require the children to justify their thinking and to explain their solutions. The children worked collaboratively with one coding toy, sharing the responsibilities of putting on the codes, syntonically acting out what they thought the Code-a-pillar would do, and planning the path. In all, about seven minutes of the task were spent with discussion and reflection, and 23 min were spent with the children working collaboratively on the problem-solving scenario. All sessions were video recorded.

4.3. Data sources

The primary data source for the present study was video recordings of the children interacting with the Code-a-pillar tasks described above. We collected 240 min of video data in total. These data not only allowed the research team to hear the verbal comments and see the physical motions of the children, they allowed the research team to qualitatively analyze the data for themes through iterative viewing and coding processes (Miles, Huberman, & Saldana, 2014).

4.4. Data analysis

Two researchers conducted multiple rounds of coding on the two tasks for each of the eight pairs (240 min of video data) using qualitative data analysis methods (Miles et al., 2014). The first round of open coding was focused on understanding the reference frames the children adopted. The second round of coding was focused on reference frame shifts. The final round of coding investigated the context of the reference frame shifts.

In order to identify reference frames, shifts, and contexts of shifts, two researchers watched the videos and selected critical events. Described by Powell, Francisco, and Maher (2003), critical events are portions of the data that hold the specific elements of investigation. In this case, the researchers selected the critical events ($n = 127$) that captured children engaging in reference frames and shifts. To validate the findings, three researchers met on multiple occasions to discuss coding differences and to refine the indicators for specific codes. Once an agreement was met on the refined coding schema, an additional coder was brought in to secondarily code 20% of the data. A Cohen's k determined high agreement between the new coder and the original coders, $k = .90$. Once the coders established this high intercoder reliability, one researcher went back through all critical events ($n = 127$) to verify they were coded correctly. After this final verification of the codes, the total code frequency of shifts was reduced to 116 critical events ($n = 116$).

The previously analyzed reference frame shifts were discussed as a team and then a second cycle of subcoding (Miles et al., 2014) was conducted to evaluate the context of these shifts. The coders specifically focused on (a) who precipitated the shift (i.e., teacher, child, peer), and (b) the reason for the shift (i.e., justification for their code, observing the Code-a-pillar enact the code, embodying the code because researcher asked them to).

5. Results

5.1. Young children's reference frames when learning to program

Our first research question was about identifying the types of reference frames children adopt when learning to program with robot coding toys. When analyzing video data, we discovered that children's enactment of Allocentric and Egocentric reference frames were not always carried out with precision. The children engaged in reference frames that were imprecise. We identified two new reference frames that we refer to as ProtoEgocentric and ProtoAllocentric. A ProtoEgocentric reference frame is when a child shared a reference frame with the coding toy but was imprecise with demonstrating or carrying out the specific movements that corresponded to codes (lack of action-instruction correspondence) (García-Valcárcel-Muñoz Repiso & Caballero-González, 2019). An example of a child engaging in a ProtoEgocentric frame of reference is when the child walked around the grid as if they were the Code-a-pillar themselves, but their movements did not align with the spaces on the grid (and the movement of the coding toy). Similarly, a ProtoAllocentric reference frame is when the children did not share a reference frame with the coding toy and were imprecise with their gestures and verbalization. An example of a child engaging in a ProtoAllocentric frame of reference is when the child pointed around the grid using gestures, but their verbalizations or gestures indicated they were not attending to the specific incremental movements of the Code-a-pillar. For the toy used in the present study, Code-a-pillar, and most tangible coding toys we have encountered, one movement of the toy corresponds to one code. Hence, the ProtoEgocentric and ProtoAllocentric designations allowed us to capture that children did not always embody this direct one-to-one correspondence in their movements and their gestures did not always match the incremental and measured movements of the robot.

Fig. 5 highlights this distinction between the ''Proto'' reference frames and the standard reference frames. The four quadrants indicate the four blends between action-instruction correspondence and reference frame use demonstrated by the children. Ultimately, the children ranged from precise or imprecise action-instruction correspondence (along the y-axis), and ranged from using themselves or an external object as a frame of reference (along the x-axis). Because these new distinctions emerged in the open coding, they became valuable in the subsequent coding process, and specific definitions and identifiers were needed in order to code them in the data. Table 2 provides a description of each of the reference frames as well as indicators used in the subsequent coding process. The term *shared heading* is included in each description, and describes the alignment of the child's physical body (e.g., head, eyes, torso). When a child shared a heading with the coding toy, it indicated the child walked around the grid like the Code-a-pillar, crawled like the Code-a-pillar, or aligned their body in some way with that of the Code-a-pillar. When a child did not share a heading with the Code-a-pillar, it indicates instances when the child remained stationary and referred to the toy's movements without physically aligning their body, typically through gestures or verbalizations.

5.2. Young children's reference frame shifts when playing with coding toys

Our second research question explored how children engage in reference frame shifts when learning how to program with coding toys. To present these results, we first provide an example of a reference frame shift, then we report three cases that illustrate how these shifts occurred in various contexts. Finally, we present the frequency counts of the reference frame shifts.

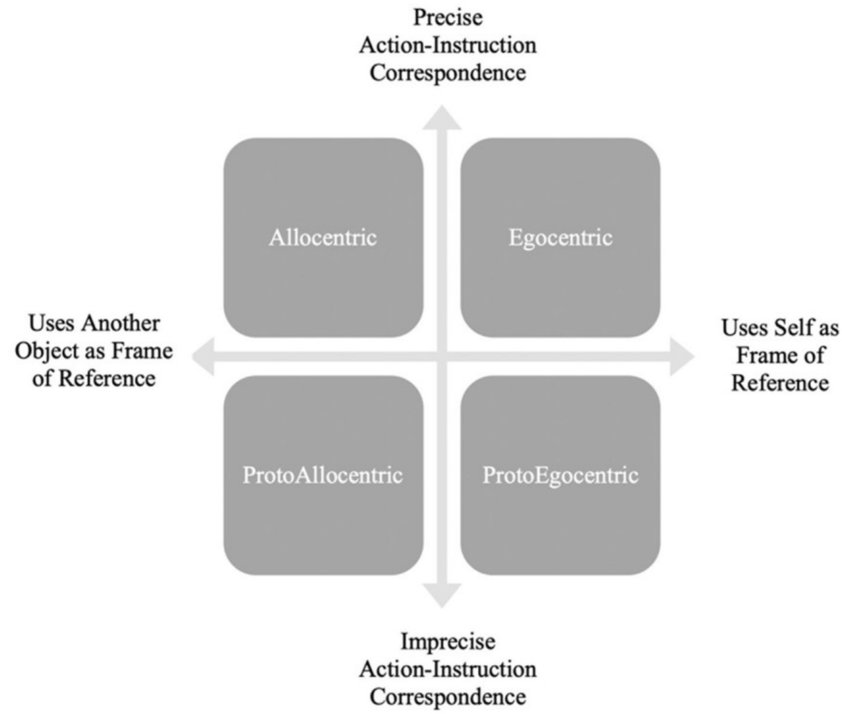


Fig. 5. Different ways young children engage in reference frames while using coding toys.

A reference frame shift occurred when the child originally engaged in one reference frame (e.g., ProtoAllocentric) and then shifted into another reference frame (e.g., ProtoEgocentric) when engaging in the task. The following example describes one of these shifts to illustrate the occurrence of a reference frame shift. When trying to show the path the Code-a-pillar should take, one child pointed around the grid with their finger to indicate the path, yet their gestures were imprecise and did not align with the toy's individual movements (did not have a one-to-one correspondence). At this point, the child engaged in a ProtoAllocentric reference frame. Then, to clarify their intended path, the child walked onto the grid. They enacted the path with their body, walking from point A to point B. They embodied the Code-a-pillar and shared its heading along the path, however, their steps and movement were too fast and did not correspond to the movement that Code-a-pillar makes. The child had switched into ProtoEgocentric reference frame.

5.2.1. *Cases demonstrating shifts occur in various contexts*

The following section provides three cases that illustrate reference frame shifts happening in a variety of contexts to qualitatively answer our question about how children engage in reference frame shifts when learning to program with coding toys.

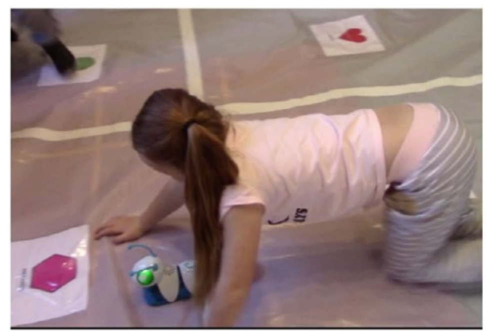
5.2.1.1. Case 1: Teacher-precipitated protoallocentric to protoegocentric shift where child justifies program to teacher. We visually illustrate Case 1, which is an example of a ProtoAllocentric to ProtoEgocentric reference frame shift. In this case, the shift was evidenced by the child beginning in an object–object reference frame while not attending to action-instruction correspondence. This shift occurred in a teacher-precipitated, child-justifying context. The children in Fig. 6 were trying to get Code-a-pillar from the green circle (point A) to the blue square (point B). Although there were many potential paths that could result in Code-a-pillar ending in this location, one of the most efficient paths would be *forward, forward, rotate left, forward*. There was no requirement to end in a specific orientation for either of the tasks.

As seen in Fig. 6(left), Jenny (a pseudonym, as are all names of children in this paper) sat on her knees on the mat and looked around the grid, trying to figure out the path for the Code-a-pillar. Jenny said, ''He needs to go this way [uses her arm and hand to make one big slanting motion from the green circle to the blue square]''. Two parts of her interactions indicated that she initially engaged in a ProtoAllocentric reference frame. Jenny's gesturing did not attend to specific action-instruction correspondence, which we coded as a Proto reference frame. The next component was that Jenny was not sharing a heading with the toy, which indicates an Allocentric reference frame. Taken together, she was coded as starting in a ProtoAllocentric reference frame. Then, the instructor asked Jenny to stand up and walk the path she thinks Code-a-pillar needed to go. When Jenny stood up to walk the path, she tiptoed forward a little and then started slanting toward the blue square (Fig. 6, right). This body motion indicated she was engaging in a ProtoEgocentric reference frame because she positioned her body to share a heading with the Code-a-pillar while not attending to precise action-instruction correspondence (indicated by the many tiptoes and continuous slanting motion). This case provides an example of a ProtoAllocentric to ProtoEgocentric reference frame shift in a context that was teacher precipitated and the children were asked to justify their thinking.

5.2.1.2. Case 2: Child-precipitated protoallocentric to protoegocentric shift where child builds and observes the program. Case 2 highlights another example of a ProtoAllocentric to ProtoEgocentric shift but in a different context. In this example, the shift is child-precipitated, which means that the child initiated the shift, and the shift happened when the child was observing the enactment of the program.

Fig. 7 depicts Tom engaging in a ProtoAllocentric reference frame as he watched Code-a-pillar carry out its program. Tom was observing the Code-a-pillar's movements from a standing and stationary position and looking down at the grid space. He incorrectly programmed Code-a-pillar to get from the green circle (point A) to the red circle (point B) using *forward, rotate right*. Fig. 7(left) shows Tom crouching in a stationary position as he

Table 2
Coding scheme with definition and example of reference frame.

	Uses another object as frame of reference	Uses self as frame of reference
Precise action-instruction correspondence	**Allocentric –** Engages with robot while not sharing a heading, making precise and distinct actions/language that correspond to specific code.  **Example –** A student watches from the side as the robot enacts the program and says, ''It needs two forwards because there are two squares!''	**Egocentric –** Engages robot with a shared heading, making precise and distinct actions that correspond to specific code.  **Example –** A student positions himself behind the robot and describes that one forward arrow gets the robot from the square he is crouching on, to the red circle.
Imprecise action-instruction correspondence	**ProtoAllocentric –** Engages with robot while not sharing a heading, does not make precise and distinct actions that correspond to specific code.  **Example –** A student observes the robot from the side and says, ''it needs to go over there!''	**ProtoEgocentric –** Engages robot with a shared heading, does not make precise and distinct actions that correspond to specific code.  **Example –** A student crawls continuously behind the robot as it enacts the program, not attending to specific incremental movements.

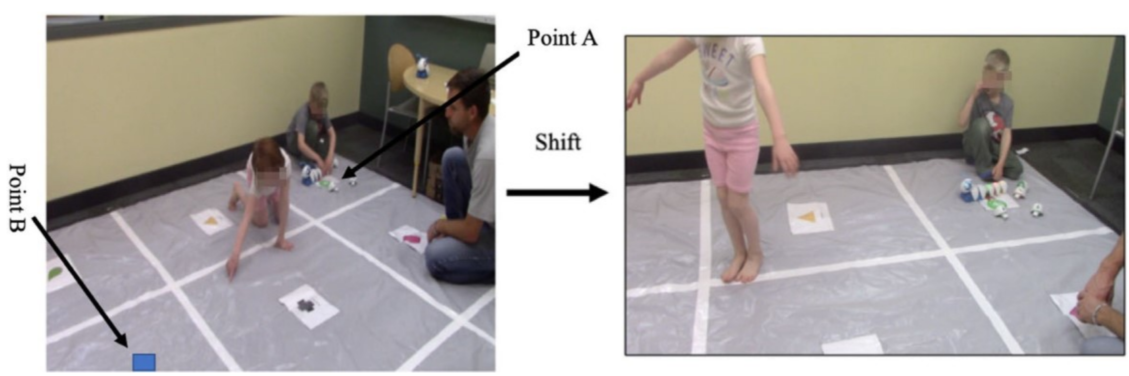


Fig. 6. Jenny shifts from a ProtoAllocentric to ProtoEgocentric reference frame in a teacher-precipitated, student-justified context.

watches the Code-a-pillar enact the program. We coded this as Proto because his gestures indicated a lack of precision of distance that corresponded to the coding pieces. We also coded it as Allocentric because he was not sharing a heading with the toy and observing the coding toy's movements from a standing position. While enacting the program, Code-a-pillar's wheels got stuck on the grid and Tom rushed to fix the problem and smooth out the grid (Fig. 7, right). At this point, Tom began crawling behind the Code-a-pillar and shifted reference frames into a ProtoEgocentric frame. We coded this new reference as Proto because he crawled behind the Code-a-pillar using movements that were continuous and imprecise. We coded it Egocentric because as he crawled, he would lift his head up and look behind Code-a-pillar's eyes and share its heading. He also changed his direction to match Code- a-pillar when it turned. As the Code-a-pillar continued its course, Tom continued to crawl behind it and maintained this ProtoEgocentric reference frame. When Code-a-pillar stopped without moving the final movement forward, Tom added a final *forward* code to Code-a-pillar's sequence which corrected the program. This example demonstrated how Tom both assembled the program and observed the program initially in a ProtoAllocentric

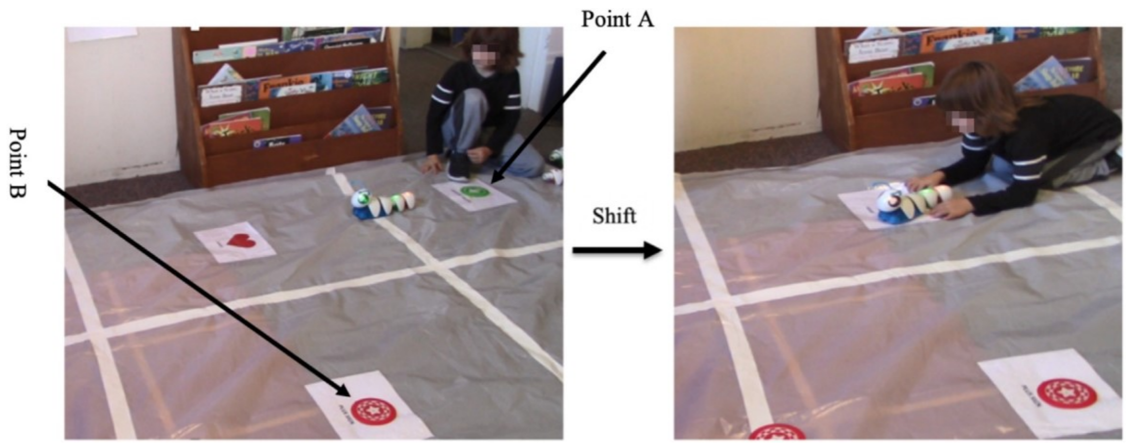


Fig. 7. Tom shifts from a ProtoAllocentric to a ProtoEgocentric reference frame in a student-precipitated, student-observation context.

frame of reference but shifted into a ProtoEgocentric frame part way through the enactment.

Cases 1 and 2 illustrate how children use the most common reference frame shift – ProtoAllocentric to ProtoEgocentric – in two different contexts. We did not find any themes or patterns in the data to indicate that shifts happened in a particular context. Sometimes the reference frame shifts were child-precipitated and sometimes they were teacher-precipitated. Similarly, there were different reasons for a reference frame shift, for example, sometimes a shift occurred because a child was asked to justify their thinking about a program and sometimes it occurred because they found a bug in their program. Sometimes shifting frames helped them build a program.

5.2.1.3. Case 3: Child-precipitated protoallocentric to egocentric reference frame shift. A ProtoAllocentric to Egocentric shift occurred when a child begins by using an imprecise object-to-object reference frame and shifted into a precise subject–object reference frame. This ProtoAllocentric to Egocentric shift was the third most common type of shift observed in the data. We provide an example of this as Case 3, with Alec making a reference frame shift to help him build a correct program.

Fig. 8 shows Alec and Sue working together to build a program to move the coding toy from point A to point B and pass through a specific point, point C. In this specific example, Code-a-pillar first wanted to visit Ladybug (point A), then move to the leaf for lunch (point C), and finally go back to Code-a-pillar's home for the evening (Point B). The children collaboratively determined which body segments should be appended to the Code-a-pillar to have it match the path in the story. Initially, Alec and Sue remained in the original starting position, visually looked around the grid, and did not indicate they were linking specific movements (action) to specific codes (instruction). These observations indicated that Alec and Sue engaged in a ProtoAllocentric frame of reference (Fig. 8, left). The first few codes did not pose a problem for them. However, the orientation of the coding toy in their imagined path needed a turn (i.e., a rotation), which challenged Alec and Sue. After trying to negotiate which turn was needed to make the correct rotation, Alec stood up, went to the location where he believed the Code-a-pillar would be with the current program, and used his body to figure out which code would be needed to yield the correct rotation (Fig. 8, right). As he stood up and moved into the position of the Code-a-pillar, he said, ''Let's see. It should go straight, turn right, turn left. . . '' As Alec said these commands, he aligned his body movements to the commands (action-correspondence). In that moment, Alec transitioned into a precise Egocentric frame of reference. He took on the reference frame of Code-a-pillar (Ego) and used precise movements. This shift from ProtoAllocentric to Egocentric frame of reference was child-precipitated, in a problem-solving context, and led to a successful outcome.

Table 3
Most common reference frame shift across sites.

Children (N = 16)			
Site	Total shifts	Total occurrence of ProtoAllo → ProtoEgo shifts	Frequency (%)
A	57	26	45.6
B	59	29	49.2
Total	116	55	47.4

5.2.2. *Frequency counts of reference frame shifts*

A ProtoAllocentric to ProtoEgocentric reference frame shift emerged as the most common reference frame shift (43%) in our analysis. The results were inconclusive in terms of a specific context driving these shifts. We did not find any themes or patterns in the data that could explain when or why a child shifted from ProtoAllocentric to ProtoEgocentric frames. At site A, the most common context was a child-initiated shift when children were building and then observing the enactment of their code (77%). Children were starting in a ProtoAllocentric reference frame when building their code and then shifting to ProtoEgocentric as they followed the Code-a-pillar's path. At site B, the most common context for this ProtoAllocentric to ProtoEgocentric shift was a teacher-precipitated shift or teacher-initiated reference frame shift. This happened when children were asked to justify their program for the Code-a-pillar's path (41%). The children were initially using a ProtoAllocentric frame of reference and then shifted into a ProtoEgocentric frame of reference when explaining their program to their teacher.

Table 3 summarizes the frequency of ProtoAllocentric to ProtoEgocentric shifts compared to the total number of reference frame shifts made in both sites. Table 4 summarizes the frequency counts of all other types of reference frame shifts.

Given the messiness of play and learning in a naturalistic environment, it is not surprising the data analysis did not reveal a specific context for reference frame shifts. However, the findings illustrate that children do switch back and forth between Egocentric and Allocentric frames of reference and that oftentimes, they engage in these reference frames in imprecise ways.

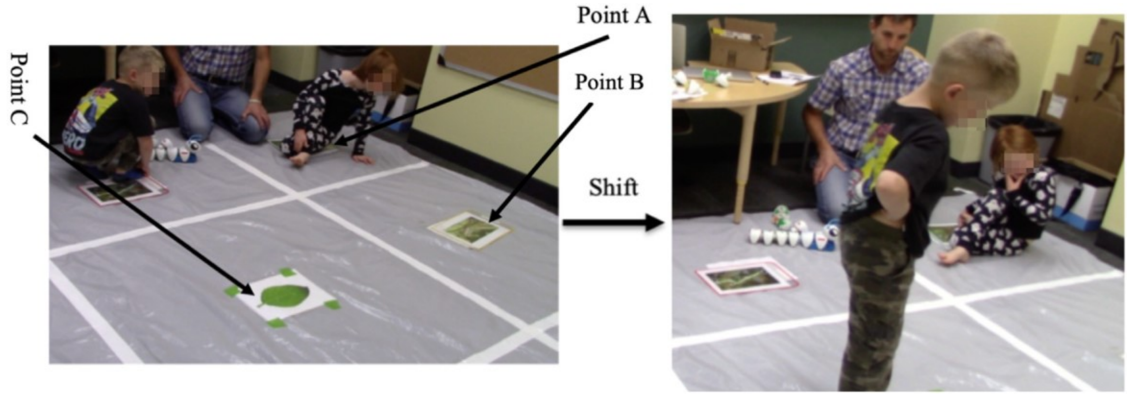


Fig. 8. Alex shifts reference frames from Less precise ProtoEgocentric to precise Egocentric.

Table 4
Frequencies of specific reference frame shifts across sites.

Children (N = 16)		
Shift	Occurrence	Frequency (%)
ProtoAllo → ProtoEgo	55	47.4
ProtoEgo → ProtoAllo	18	15.5
ProtoAllo → Ego	13	11.2
ProtoEgo → Ego	6	5.2
Allo → Ego	6	5.2
Ego → ProtoAllo	5	4.3
ProtoAllo → Allo	5	4.3
Allo → ProtoEgo	3	2.6
Ego → Allo	2	1.7
Ego → ProtoEgo	2	1.7
ProtoEgo → Allo	1	.9
Total	116	100

6. Discussion

Research suggests that unplugged and tangible-based coding opportunities are important in helping young children understand and demonstrate specific CT concepts (Gomes, 0000; Lavigne, Lewis-Presser, & Rosenfeld, 2020). While there has been a surge of research that investigates coding toys as ''objects to think with'' most of the research on how young children use their own body knowledge when learning to program focuses on embodiment (Città et al., 2019; Martinez et al., 2015; Palmér, 2017; Raabe et al., 2015) or designs comparing low and high embodiment (Fadjo et al., 2009; Sung et al., 2017). There is a lack of research on how young children perceive the coding toys in physical space and how they orient themselves both mentally and physically to the robot and to the robot's path. The present study explored how young children engage in reference frames and shift between reference frames when learning to program with a robot coding toy. Our findings are significant as they provide insight into what frames children adopt when programming and how they engage in shifts between reference frames as they learn to program. In particular, we identified three themes that have implications for design and implementation of coding toys for early childhood: (a) young children possess imprecise coordination systems that they blend with reference frames, (b) young children shift reference frames while learning to program, and (c) young children shift reference frames in a variety of contexts. We discuss these themes below.

6.1. *Young children's imprecise coordination systems*

The results indicated that young children may not have the precise coordination system that older children exhibit when using coding toys. Our findings of the ProtoEgocentric and ProtoAllocentric reference frames distinguished young children's developing use of reference frames from the standard Egocentric and Allocentric frames in the current literature. The ProtoEgocentric and ProtoAllocentric reference frames allowed the researchers to understand that young children do not always embody precise coordination between their movements and robot's movements. These findings may be similar to research on young children's early number sense (Dehaene, 2011). For example, young children's early number sense is characterized by imprecise magnitude and quantity comparisons. Young children have an innate, continuous sense of more or less, but have not yet developed an understanding of numerical representation that allows them to more precisely define and describe quantities. Similarly, young children in this study were more often able to engage in imprecise reference frames en route to developing coordination between the robot's precise amount of rotation or amount of forward movement and their own movements and gestures. The contributions of ProtoEgocentric and ProtoAllocentric indicators allow the consideration of developmental spatial imprecision when learning how to program with robot coding toys. The spatial nature of these coding toys could be a fruitful area for contributing to the existing literature base on young children's development of spatial reasoning and spatial literacy (Città et al., 2019; Verdine, Golinkoff, Hirsh-Pasek, & Newcombe, 2014).

The findings of the imprecise reference frames has implications for teachers who use coding toys in formal and informal settings. Research on the role of teachers when implementing robots and coding toys in preschool and early childhood classrooms highlight the need for professional support (Wang, Choi, Benson, Eggleston, & Weber, 2020) and the importance for teachers to understand how these foundational skills (i.e., spatial reasoning) affect children's learning to code in addition to knowledge of how to teach CT (Chalmers, 2018; Saxena, Lo, Hew, & Wong, 2020). This finding also has implications for design of future coding toys. Developers should consider including design features that can help direct children to this action-instruction correspondence. For example, simply adding a brief pause after each code is enacted by the robot allows children to see the one-to-one correspondence between movement and code, thereby helping students to become more precise in their use of spatial reference frames. Similarly, features such as visual or audio cues (e.g. flashing lights or sounds that can be turned off) that indicate when each code in a sequence is enacted may provide additional scaffolds to help students spatially see and understand the robots' precise movements. Such features have the potential to help young children attend to each code and its action, supporting their precise coordination of action-instruction.

6.2. Young children shift reference frames while learning to program

Previous research on reference frames focused on older children and suggested shifts from Egocentric to Allocentric helped students succeed in their programming activities (Smith et al., 2014). The present study provides evidence that young children also engage in shifting reference frames, which could affect their programming success. While many studies focused on how children used full embodiment while learning to program (Città et al., 2019; Martinez et al., 2015; Palmér, 2017; Raabe et al., 2015), the present study suggests that syntonic learning may not fully capture the reference frame shifting that young children exhibit while learning to program with coding toys. Additionally, the present study suggests that this shifting back and forth between reference frames may play an important role in learning how to program with a robot coding toy.

The results also indicate that the mental process of imaginal updating (Klatzky, 1998) may be observed in children as they interact with coding toys. The children in the present study sometimes used physical and verbal indicators to demonstrate how they were reorganizing spatial structures mentally. The present study indicated that the children most regularly shifted from ProtoAllocentric to ProtoEgocentric, which would align with the mental process of imaginal updating. Young children may need to connect the mental process of imaginal updating with the physical world through movement, gestures, and verbalizations.

This shift between reference frames may also be connected to the idea of increasingly sophisticated spatial thinking. Golbeck (2005) describes children's developmental understanding of how to anchor their ideas and themselves in space. Specifically, 5- and 6-year-old children are typically moving from a pre-axial system of spatial thinking (i.e., not yet able to coordinate objects in space with a stable reference point) to a uni-axial system of spatial thinking (i.e., singular dimensional thinking). Further, they begin understanding perspective as another reference system. Children's developing ideas about spatial thinking may explain some of the shifting between reference frames as they interacted with coding toys in this study. These ideas about anchors in space and reference frames develop from interactions in the spatial world, and children's interactions with coding toys could be yet another way to gain playful spatial experiences.

Finally, these shifts highlight an inextricable connection of mind and body that aligns with the embodied cognition literature and may hold links to Piaget's Theory of Cognitive Development. The children in the present study were in kindergarten and, according to Piaget's theory, may have still been in what he referred to as a preoperational stage. Although young children are beginning to apply symbolic thinking to reasoning, tangible materials are important to help solidify understanding of the symbols (Armoni, 2012; Golbeck, 2005; Ojose, 2008). The Code-a-pillar may act as a tangible interface to the child, which helps them in reasoning with the unique symbolic coding system. More than simply experiencing the symbols, these children dynamically interacted with the symbols by shifting in and out of reference frames when applying their knowledge of the symbols.

6.3. Young children's reference frame shifts occur in a variety of contexts

The present study found that shifts happen in a variety of contexts and have the potential to influence children's programming differently. Sometimes the teacher initiated a shift through questioning which may or may not have led to programming success. Other times, a child precipitated their own shift, which again had inconsistent effects on programming success. Sometimes there were repeated errors when teachers precipitated the shift and the child failed to self-precipitate their shifts in subsequent attempts. These variations in context of shifts made it challenging to determine if the shifts are developmental or if they are contextually driven. Other research has suggested that context characteristics such as open-ended play and collaboration provide a context optimal for learning to program (Bers et al., 2014; Bers, González-González, & Armas-Torres, 2019). More research is needed on the types of contexts that help or hinder children's learning to program with coding toys and whether or not the shifts occur due to differences in instruction or a child's cognitive development.

7. Limitations

As with any study seeking to better understand human learning and interaction with the environment, this study is limited by elements that are important to make transparent.

7.1. Limitations in data collection

Limitations in data collection included varied timing of tasks across sites and possible teacher effects, pairing effects, and age effects across sites. First, the children in site A completed the tasks in consecutive weeks whereas the participants in site B completed the same tasks in consecutive days. Although both time frames are relatively close together, there may be a difference in retention of the concepts which enabled one group to respond differently to the coding toys than the other. Similarly, two separate researchers acted as the teacher for these lessons across sites. Although the researchers were administering the same tasks, teacher effects and differences in task administration were inevitable. An additional confounding variable was the collaborative nature of the pairs. At any age, working collaboratively on a shared task can be challenging. This study was no exception. Sometimes, one child took the lead on the solution strategy. Finally, the average age of the children was 5 years old but students' actual age varied in months. At this young age, a few months can make a difference in terms of development.

7.2. Limitations in data analysis

Our initial selection of the critical events (Powell et al., 2003) from the larger data set (240 min of video) resulted in 127 shifts. It is possible that we missed some shifts in our analysis. Similarly, although children may have been engaging in mental shifts aligned with imaginal updating, we only coded physical indicators of shifts.

Finally, due to the limited sample size and narrow scope of contexts in which these data were collected, the findings of this exploratory study should not yet be generalized beyond the current participant population. This study lays the foundation for extended work that will allow researchers to look for patterns of reference frame engagement and shifts across a larger sample of participants and contexts, as well as analyze whether or not these reference frames and shifts affect programming success.

8. Conclusion

There is increased interest to introduce coding to children at an early age. As a result, there are a number of screen-free coding toys and games designed for preliterate children and emergent readers. Research on these toys has focused on operationalizing computational thinking (CT) in early childhood (Bers, 2019; Bers et al., 2019; Relkin, de Ruiter and Bers, 2020), toy designs and interfaces (Hamilton, Clarke-Midura, Shumway, & Lee, 2020; Shumway et al., 2020; Yu & Roque, 2019), the role of teachers in promoting CT (Wang et al., 2020), the role of parents on the design (Yu, Bai, & Roque, 2020), and use of toys at home (Relkin, Govind, Tsiang and Bers, 2020). However, missing from most of these studies is how preliterate children perceive the coding toys in physical space. A major contribution of the present study is the development of the theory around reference frames that accounts for imprecision of young children: ProtoAllocentric and ProtoEgocentric reference frames. Understanding how spatial precision affects how young children learn to code is important because the design of these ''objects to think with'' lend themselves to spatial literacy. For example, robot toys – such as Code-a-pillar – all involve an agent that moves through space. The programming language children use to move these robots is embodied in directional arrows that represent discrete units of linear movement (*forward* and *backward*) and rotations (typically 90° right or left rotations). Children's interactions with these toys press them to engage in using their own reference frames (Klatzky, 1998) and considering the robot's reference frames as well as consider the anchors in space in relation to themselves and the robot (Golbeck, 2005).

The present study found that children engage in reference frames while programming with coding toys, but not with the precision that is typically assumed. Their imprecision with action-instruction correspondence may mirror a similar type of imprecision we see in young children's early number sense (Dehaene, 2011). We also found that the children's most common reference frame was ProtoAllocentric to ProtoEgocentric. This suggests that children are remaining in a relatively imprecise state while shifting from object–object to subject–object reference frames.

Future research should investigate whether or not children's use of and shifts within reference frames leads to more successful programming. Researchers could also investigate how children engage in reference frame shifts when they have more coding experience or engage in more complex coding sequences or practices, such as loops or functions. This research agenda could help better inform the understanding of development of young children's use of reference frames as they engage in tasks as novices to more experienced programmers. This knowledge could have implications for ways educators can support young children as they learn how to code in their classrooms.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Science Foundation, USA grant #1842116. 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 or Utah State University.

References

Angeli, C., & Valanides, N. (2019). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, *105*, Article 105954. http://dx.doi.org/10.1016/j.chb.2019.03.018.

Armoni, M. (2012). Teaching CS in kindergarten: how early can the pipeline begin?. *ACM Inroads*, *3*, 18–19. http://dx.doi.org/10.1145/2381083.2381091.

Barsalou, L. W. (2008). Grounded cognition. *Annual Review of Psychology*, *59*, 617–645. http://dx.doi.org/10.1146/annurev.psych.59.103006.093639.

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

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computer Education*, *72*, 145–157. http://dx.doi.org/10.1016/j.compedu.2013.10.020.

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

Chalmers, C. (2018). Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction*, *17*, 93–100.

Città, G., Gentile, M., Allegra, M., Arrigo, M., Conti, D., Ottaviano, S., et al. (2019). The effects of mental rotation on computational thinking. *Computer Education*, *141*, Article 103613. http://dx.doi.org/10.1016/j.compedu.2019.103613.

De Vega, M., Glenberg, A. M., & Graesser, A. (2008). *Symbols and embodiment: Debates on meaning and cognition*. Oxford University Press.

Dehaene, S. (2011). *The number sense: How the mind creates mathematics*. OUP USA.

Fadjo, C. L., Lu, M.-T., & Black, J. B. (2009). Instructional embodiment and video game programming in an after school program. In *EdMedia innov. learn. association for the advancement of computing in education (AACE)* (pp. 4041–4046).

Foley, R. T., Whitwell, R. L., & Goodale, M. A. (2015). The two-visual-systems hypothesis and the perspectival features of visual experience. *Consciousness and Cognition*, *35*, 225–233. http://dx.doi.org/10.1016/j.concog.2015.03.005.

Gallistel, C. R. (1990). *The organization of learning*. Cambridge, MA: MIT Press.

Gibbs, R. W., Jr. (2005). *Embodiment and cognitive science*. Cambridge University Press.

Glenberg, A. M. (2008). Embodiment for education. In *Handb. Cogn. Sci.* (pp. 355–372). Elsevier, http://dx.doi.org/10.1016/B978-0-08-046616-3.00018-9.

Golbeck, S. L. (2005). Building foundations for spatial literacy in early childhood. *YC Young Children*, *60*, 72.

Gomes, T. C. S. Exploring an approach based on digital games for teaching programming concepts to young children. (n.d.) 2.

Hamilton, Megan, Clarke-Midura, Jody, Shumway, Jessica F., & Lee, Victor R. (2020). An emerging technology report on computational toys in early childhood. *Technology, Knowledge and Learning*, *25*(1), 213–224.

Johnson-Glenberg, M. C., Birchfield, D. A., Tolentino, L., & Koziupa, T. (2014). Collaborative embodied learning in mixed reality motion-capture environments: Two science studies.. *Journal of Educational Psychology*, *106*, 86–104. http://dx.doi.org/10.1037/a0034008.

Klatzky, R. L. (1998). Allocentric and egocentric spatial representations: Definitions, distinctions, and interconnections. In C. Freksa, C. Habel, & K. F. Wender (Eds.), *Spat. Cogn.* (pp. 1–17). Berlin, Heidelberg: Springer Berlin Heidelberg, http://dx.doi.org/10.1007/3-540-69342-4_1.

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*, 63–77. http://dx.doi.org/10.1080/21532974.2019.1693940.

Martinez, C., Gomez, M. J., & Benotti, L. (2015). A comparison of preschool and elementary school children learning computer science concepts through a multilanguage robot programming platform. In *Proc. 2015 ACM Conf. Innov. Technol. Comput. Sci. Educ. - ITiCSE 15* (pp. 159–164). Vilnius, Lithuania: ACM Press, http://dx.doi.org/10.1145/2729094.2742599.

Miles, M. B., Huberman, A. M., & Saldana, J. (2014). *Qualitative data analysis: A methods sourcebook*.

Ojose, B. (2008). Applying Piaget's Theory of Cognitive Development to mathematics instruction. *Mathematics Education*, *18*, 26–30.

Palmér, H. (2017). Programming in preschool - With a focus on learning mathematics. *International Journal of Early Childhood Education Research*, *8*, 75–87.

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

Piaget, J. (1927). *The child's conception of the world*. Routledge.

Powell, A. B., Francisco, J. M., & Maher, C. A. (2003). An analytical model for studying the development of learners' mathematical ideas and reasoning using videotape data. *The Journal of Mathematical Behavior*, *22*, 405–435. http://dx.doi.org/10.1016/j.jmathb.2003.09.002.

Raabe, A., Rodrigues, A. J., Santana, A. M., Vieira, M. V., Rosário, T., & Carneiro, A. C. R. (2015). Brinquedos de Programar na Educação Infantil: Um estudo de Caso. In: Maceió, Alagoas, Brasil. p. 42. http://dx.doi.org/10.5753/cbie.wie.2015.42.

Relkin, E., Govind, M., Tsiang, J., & Bers, M. (2020). How Parents Support Children's informal learning experiences with robots. *Journal for STEM Education Research*, *6*, 39–51.

Relkin, E., de Ruiter, L., & Bers, M. U. (2020). Techcheck: Development and validation of an unplugged assessment of Computational Thinking in Early Childhood Education. *Journal of Science Education and Technology*, *29*, 482–498. http://dx.doi.org/10.1007/s10956-020-09831-x.

García-Valcárcel-Muñoz Repiso, A., & Caballero-González, Y.-A. (2019). Robotics to develop computational thinking in early Childhood Education. *Comunicar*, *27*, 63–72. http://dx.doi.org/10.3916/C59-2019-06.

Ruggiero, G., Iachini, T., Ruotolo, F., & Paolo Senese, V. (2009). Spatial memory: The role of egocentric and allocentric frames of reference. In J. B. Thomas (Ed.), *Spat. Mem. visuospatial Process. Cogn. Perform. Dev. Eff.*. Nova Science Publishers.

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

Sharma, K., Papavlasopoulou, S., & Giannakos, M. (2019). Coding games and robots to enhance computational thinking: How collaboration and engagement moderate children's attitudes?. *International Journal of Child-Computer Interaction*, *21*, 65–76. http://dx.doi.org/10.1016/j.ijcci.2019.04.004.

Shumway, J., Clarke-Madura, J., Lee, V. R., Hamilton, M. M., & Baczuk, C. (2020). Putting spatial orientation and measurement skills to work: Coding toys in kindergarten. *Teaching Children Mathematics*, *25*.

Smith, C. P., Berland, M., & Martin, T. (2014). Exploring how students alternate perspectives in IPRO. In *Learn. Technol. Body* (pp. 132–148). Routledge.

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing computational thinking to Young learners: Practicing computational perspectives through embodiment in mathematics education. *Technology, Knowledge and Learning*, *22*, 443–463. http://dx.doi.org/10.1007/s10758-017-9328-x.

Verdine, B. N., Golinkoff, R. M., Hirsh-Pasek, K., & Newcombe, N. S. (2014). Finding the missing piece: Blocks, puzzles, and shapes fuel school readiness. *Trends in Neuroscience and Education*, *3*, 7–13. http://dx.doi.org/10.1016/j.tine.2014.02.005.

Vogeley, K., May, M., Ritzl, A., Falkai, P., Zilles, K., & Fink, G. R. (2004). Neural correlates of first-person perspective as one constituent of human self-consciousness. *Journal of Cognitive Neuroscience*, *16*, 817–827. http://dx.doi.org/10.1162/089892904970799.

Wang, X. C., Choi, Y., Benson, K., Eggleston, C., & Weber, D. (2020). Teacher's role in Fostering Preschoolers' Computational Thinking: An exploratory case study. *Early Education and Development*, 1–23. http://dx.doi.org/10.1080/10409289.2020.1759012.

Yu, J., Bai, C., & Roque, R. (2020) Considering parents in coding kit design: Understanding parents' perspectives and roles. In *Proc. 2020 CHI Conf. Hum. Factors Comput. Syst.* (pp. 1–14).

Yu, J., & Roque, R. (2019). A review of computational toys and kits for young children. *International Journal of Child-Computer Interaction*, *21*, 17–36.

Zaehle, T., Jordan, K., Wüstenberg, T., Baudewig, J., Dechent, P., & Mast, F. W. (2007). The neural basis of the egocentric and allocentric spatial frame of reference. *Brain Research*, *1137*, 92–103. http://dx.doi.org/10.1016/j.brainres.2006.12.044.