STUDENTS' DIRECTIONAL LANGUAGE AND COUNTING ON A GRID

Sezai Kocabas
Purdue University
skocabas@purdue.edu

Laura Bofferding
Purdue University
Lbofferd@purdue.edu

Mahtob Aqazade
Purdue University
maqazade@purdue.edu

Ana-Maria Haiduc
Purdue University
ahaiduc@purdue.edu

Lizhen Chen
Purdue University
lizchen@purdue.edu

We explored 23 first and third graders' directional language and counting of spaces on a grid when debugging programming code after playing three and six sessions of Coding Awbie™. Students in both grades were more likely to describe a bunny as moving up using specific terms but more likely to describe going right using generic terms. Over half of the students made at least one counting error. Students had fewer counting errors after playing the programming game because many of them more efficiently corrected the programming code.

Keywords: Geometry and Geometrical and Spatial Thinking; Cognition; Elementary School Education

Spatial thinking is crucial for success in not only mathematics but also programming, chemistry, and medicine (Clements, 1999; Jones & Burnet, 2007; Krajewski & Ennemoser, 2009; Kyttälä et al., 2003; Lowrie & Logan, 2016; Sorby et al., 2013). Given the increased focus on students learning programming, mathematics teachers can leverage the finding that programming tasks can promote mathematical practices involving spatial thinking (Heghfield, 2003). For example, one of the Common Core Standards for Mathematical Practice is for students to reason abstractly and quantitatively; this practice involves representing situations symbolically and reasoning about quantities (National Governor's Association Center for Best Practices & Council of Chief State School Officers, 2010). Similar practices are incorporated in programming games for children, which often involve manipulating symbols that tell a character which direction to go in and how far. Therefore, the current study explored spatial thinking through students' explanations as they debugged a symbolic program written to move a bunny to a carrot on a grid.

Spatial Thinking

Spatial thinking involves perceiving and transforming the visual environment (Gardner, 1983), plays a strong role in students' ability to mentally represent and manipulate information or objects (Lowrie & Logan, 2016), and is highly correlated with academic achievement in programming, mathematics, and other fields (Clements, 1999; Jones & Burnet, 2007; Lowrie & Logan, 2016; Sorby et al., 2013). There are two dimensions of spatial thinking; spatial orientation and spatial visualization (Clements & Battista, 1992).

Spatial Orientation and Language

Spatial orientation refers to the ability to understand and interpret the relations among different positions of objects in space (Clements & Battista, 1992; Clements, 1999). Clements (1999) argued that students must have some mathematical ideas for reading and creating mental maps. Students should learn to manage abstraction, generalization, and symbolization in the mapping process. Furthermore, students should develop some ideas such as direction, distance and measurement, and location. Children naturally use vertical terms such as *up* and *down* at an early age, but *left* and *right* require more time (around age 6-8) for students to fully understand (Sarama & Clements, 2009). Teachers should help students move away from generic movement descriptions such as *over* because *over* does not extrapolate well to bidirectional spaces (Sarama & Clements, 2009). Students need to learn to use precise language, and computing programs can provide them with immediate feedback.

Spatial Visualization and Counting

On the other hand, spatial visualization involves imagining and making transformations, such as motions of an object (Sarama & Clements, 2009). Spatial visualization is strong predictor for success of counting achievement in earlier grades. For instance, spatial thinking was positively correlated with kindergarteners' ability to count 16 cubes in four rows and four columns (Kyttälä et al., 2003). Students may miscount (e.g., by double counting) if they do not find structure in the spaces they are counting (e.g., Battista et al., 1998).

Programming, Spatial Thinking, and Mathematics

Programming tasks might leverage spatial thinking because students have the opportunity to learn direction, orientation, measurement, and positional language, as with Logo turtle (e.g. forward 10 steps; Clements, 1999). For example, after using *Turtle Math*, a version of Logo for third to sixth graders, students' computation scores improved (Clements & Sarama, 1996). One current programming game for even younger elementary students (Coding Awbie™) involves tangible programming pieces that allow students to move a character along squares in a grid-like environment. This structured space could potentially support younger students' counting. Further, the programming pieces involve motions, directional arrows, and numbers that might better map to students' ideas about how to move the character. Therefore, we investigated the following research questions: How do first and third graders interpret symbolic location and movement representations and map them onto a 2-dimensional grid? How does this change after practice programming a character to move on a grid (a) in terms of the positional and directional language they use? and (b) in terms of their counting to determine the number of movements?

Method Participants and Setting

The study took place in a Midwestern public elementary school with about 46% students qualifying for free or reduced-price lunch. For this analysis, we focus on data from 23 participants who came from two first- and one third-grade class from our larger study. We selected these particular classes for the analysis in order to have an equal number of participants from each grade; however, we had to exclude one of the 12 first graders because of missing data.

Study Design

The study consisted of a pretest, three sessions playing a programming game, a presentation about programming, a midtest, three more game sessions, and a posttest. During game play, students played Coding Awbie™, a game using the Osmo™ tangible interface. Student pairs were randomly assigned to either engage in free-play for the first three sessions or explainedplay (i.e., students had to tell us their goal for each program and explain how their lines of code would help them meet their goal). During the second set of sessions after the midtest, they switched groups (those who had explained-play had free-play and vice versa).

Pretest, midtest, and posttest. Our analysis focuses on one common item from the three tests, a commenting and debugging problem. For this problem, students had to explain a pair of coding commands meant to move a bunny on a grid to get to a carrot. Students were told that the bunny could not move past or jump over houses, but they could jump over flowers. Students then identified the bug in the code and explained how they would fix the code (see Figure 1).

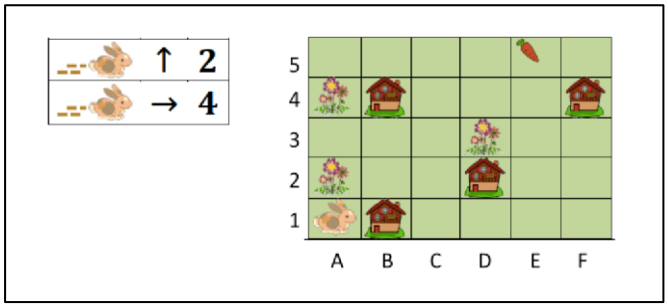

Figure 1: Programming Debugging Problem (Grid Labels Added for Ease of Reference)

Data Analysis

Positional and directional language. We analyzed the data in terms of spatial orientation for the midtest and posttest as we did not ask students to explain the program on the pretest. We expected the students to explain the first line of the program as “walk up two” (as used in the coding game) and the second line of the program as “walk right four.” Therefore, we coded students’ explanations of each line of code in three categories: movement, direction, and number. We focused on two categories for direction: generic or incorrect language and non-generic language. Non-generic language included the precise terms *up* and *right.* Generic language involved general statements such as *this way*, *that way*, *side*, *over*, and *forward* (incorrect).

Counting. We also analyzed the data in term of spatial visualization. When students corrected the code, they often counted spaces the bunny would move to determine if the number in the code should stay the same or change. We classified students based on whether they made a counting error on any of the three tests. For students who made any counting error, we identified on which test they made an error and whether they made an error on the *corrected* code (i.e., changed the first line of code to jump up 2) or when changing it to *new* code (i.e, changed the first line of code to jump up 1).

Results Positional and Directional Language

Table 1: Students’ Generic and Non-Generic Uses of *Up* and *Right*

Students		Midtest		Posttest	
		Generic	Non-generic	Generic	Non-generic
Up	1st (n=11)	9%	73%	9%	64%
	3rd (n=12)	8%	83%	---	83%

Otten, S., Candela, A. G., de Araujo, Z., Haines, C., & Munter, C. (2019). *Proceedings of the forty-first annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education*. St Louis, MO: University of Missouri.

Right				
1st (n=11)	82%	9%	55%	9%
3rd (n=12)	50%	42%	58%	25%
Up Explained-play (n=15)	13%	73%	7%	73%
Free-play (n=8)	---	88%	---	75%
Right				
Explained-play (n=15)	67%	27%	67%	13%
Free-play(n=8)	63%	25%	38%	25%

Overall, when describing the first line of code, students in both grades were more likely to describe the bunny as going *up* as opposed to describing thedirection with generic terms (see Table 1). On the other hand, the students were more likely to use generic terms, such as *go over* to describe the bunny's movements in the second line of code. Third graders were more likely than first graders to correctly describe this movement as going *right*, although across all students, fewer of them even referenced the direction on the posttest because some described and fixed the first line of code and then indicated the second line was okay without talking about it.

Interestingly, for students who played the game without explaining their code in the first three sessions, a higher percentage of them used the term *up* on the midtest than students who were forced to explain while they played. Further, students in this group also used less generic language for *right* on the posttest compared to the other group.

Counting

Overall, of the 23 students analyzed, 12 of them (five first graders and seven third graders; six free-play and six explained-play) made at least one counting error on this item. The majority of them, 80% of first graders who had errors and 71% of third graders who had errors, counted incorrectly on the pretest. In particular, these students changed the first line of code to have the bunny jump once from A1 to A3 and then had the bunny move right to C3 (see Figure 1). However, when counting from A1 to C3, they recounted the space A1 where the bunny would already be, suggesting the code should say *walk right 3* instead of *walk right 2*. Some students continued to double count the square the bunny occupied when adding on lines of code to walk up to C5 or over to E5. Fewer students miscounted on the midtest (none in first grade!) and posttest because they changed the first line of code to read *jump up 2*. Interestingly, they accepted the correct second line of code and did not double count in these situations (except for one third-grader who did this on the posttest). Only 33% of miscounters from the original freeplay group miscounted on the posttest compared to 67% of miscounters from the explained-play group.

Discussion

Similar to previous reports, we found that students were more likely to use the specific term *up* when describing vertical movement but less likely to use the term *right* (Sarama & Clements, 2009), choosing instead to use generic directions. Especially when giving directions to others, providing explicit directions is important. Therefore, the results suggest teachers should emphasize precise directional terminology as early as kindergarten (when the Common Core

Standards for Mathematics suggest students learn positional language), because playing the game without reinforcement of the specific language did not increase the use of *right* nor did being forced to explain their programming movements. On the other hand, students had fewer counting errors after playing the programming game because many of them more efficiently corrected the programming code and accepted the final direction to *move right 4* without experiencing the counting error they had when changing the code. This suggests that using worked examples could be used to scaffold their counting. There was a slight trend for students who initially had free-play to use more specific and less generic language as well as have fewer counting issues. If this pattern holds for the larger data set, it may suggest that having the opportunity to play before explaining helped students better attune to what they were explaining.

Acknowledgments

This research was supported by an NSF DRL ITEST Grant #1759254.

References

Battista, M. T., Clements, D. H., Arnoff, J., Battista, K., & Borrow, C. V. A. (1998). Students' spatial structuring of 2D arrays of squares. *Journal for Research in Mathematics Education, 29(5)*, 503-532.

Clements, D. H. (1999). Geometric and spatial thinking in young children. In J. V. Copley (Ed.), *Mathematics in the early years* (pp. 66–79). Reston, VA: National Council of Teachers of Mathematics.

Clements, D. H., & Battista, M. T. (1992). Geometry and spatial reasoning. *Handbook of research on mathematics teaching and learning*, 420-464.

Clements, D. H., & Sarama, J. (1996). Turtle Math-Redesigning Logo for Elementary Mathematics. *Learning and Leading with Technology*, *23*, 10-15.

Gardner, H. (2011). *Frames of mind the theory of multiple intelligences* (3rd ed.). New York: Basic Books.

Highfield, K. (2010). Robotic toys as a catalyst for mathematical problem solving. *Australian Primary Mathematics Classroom, 15(2), 22-27.*

Jones, S. J., & Burnett, G. E. (2007). Spatial skills and navigation of source code. *ACM SIGCSE Bulletin*, *39*(3), 231-235.

Krajewski, K., & Ennemoser, M. (2009). The impact of domain-specific precursors and visuospatial skills in kindergarten on mathematical school achievement in Grade 9. In *biennial meeting of the Society for Research in Child Development,* Denver, CO.

Kyttälä, M., Aunio, P., Lehto, J. E., Van Luit, J., & Hautamäki, J. (2003). Visuospatial working memory and early numeracy. *Educational and Child Psychology*, *20*(3), 65-76.

Lowrie, T., & Logan, T. (2007). Using spatial skills to interpret maps: Problem solving in realistic contexts. *Australian Primary Mathematics Classroom*, *12*(4), 14.

National Governors Association Center for Best Practices & Council of Chief State School Officers. (2010). Common Core State Standards. *National Governors Association Center for Best Practices, Council of Chief State School Officers, Washington D.C.*

Sarama, J., & Clements, D. H. (2009). *Early childhood mathematics education research learning trajectories for young children* (Studies in mathematical thinking and learning). New York: Routledge.

Sorby, S., Casey, B., Veurink, N., & Dulaney, A. (2013). The role of spatial training in improving spatial and calculus performance in engineering students. *Learning and Individual Differences*, *26*, 20-29.