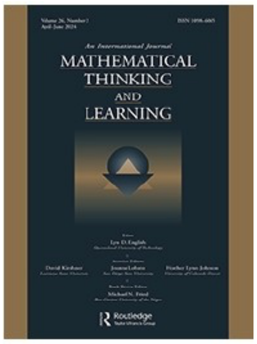


Mathematical Thinking and Learning

ISSN: (Print) (Online) Journal homepage: www.tandfon line.com/journ als/hmtl20

Vygotskian hybridizing of motion and mapping: Learning about geometric transformations in block-based programming environments

Jesus E. Hern andez-Zavaleta, Corey Brady, Sandra Becker & Douglas B. Clark

MATHEMATICAL THINKING AND LEARNING
https://doi.org/10.1080/10986065.2023.2191074

Routledge
Taylor & Francis Group

Check for updates

Vygotskian hybridizing of motion and mapping: Learning about geometric transformations in block-based programming environments

Jesus E. Hernandez-Zavaleta [a], Corey Brady [b], Sandra Becker [a], and Douglas B. Clark [a]

[a]Werklund School of Education, University of Calgary, Calgary, Alberta, Canada; [b]Peabody School of Education, Vanderbilt University, Nashville, Tennessee, USA

ABSTRACT

Research on geometric transformations suggests that early learners possess intuitive understandings grounded in motion metaphors, transitioning to mappings. The processes through which students transition between these two conceptions are not fully understood. We propose that Vygotskian hybridizing (related to Vygotsky's articulation of everyday and scientific concepts) may provide a lens for thinking about the relationship between these conceptions. Design features of block-based programming environments provide affordances to support hybridizing by providing a co-action space for learning. We conducted a comparative case study of four grade seven and eight students working in a Scratch task (Code the Quilts) and a game (Transformations Quest) to construct understandings of geometric transformations. Our findings suggest: (1) students hybridized their personal experience of motion and mathematical knowledge of mapping to build geometric transformations understandings and (2) the co-action space in which students worked promoted distributed interactions between students, block-based environments, and tasks to support hybridizing.

ARTICLE HISTORY
Received 7 June 2022
Revised 6 March 2023
Accepted 11 March 2023

KEYWORDS
Symmetry; geometric transformations; hybridizing; motion and mapping; Vygotsky; block-based programming

Across the transformational geometry research landscape there is consensus that learners initially understand geometric transformation through a motion conception and potentially later develop a mapping conception (Edwards & Zazkis, 1993; Edwards, 2003, 2009; Hollebrands, 2003; Yanik & Flores, 2009). Accordingly, authors of learning progressions centered on geometric transformations have advocated beginning in early grades with intuitive understandings that include moving shapes across the plane (with an implicit application of transformations) and finishing in high school with representations of transformations as functions from the plane to itself and matrix operations (Fife et al., 2019), making transformations and systems of transformations explicit. From early studies of geometric transformations to the present day, however, researchers have faced challenges in explaining how the two conceptions of motion and mapping are related and how students progress from one to the other (Edwards& Zazkis, 1993; Edwards, 2009; Moyer, 1978; Usiskin, 1972).

In this paper, we suggest that *Vygotskian hybridizing of motion and mapping* (a construct underpinned by ideas about the reflexive development of everyday and scientific concepts expressed by Vygotsky, 1986) in block-based programming environments (Weintrop, 2019) can provide a lens on how students can connect motion and mapping conceptions. In analyzing four 7th and 8th grade students' interactions in two different block-based programming environments - one a game and the other a microworld for artistic construction - we found that students hybridized motion and mapping conceptions to meet the demands of the environments and achieve their goals. Hybridization was

CONTACT Jesus E. Hernandez-Zavaleta hernandez.je@ufl.edu University of Florida, College of Education, Gainesville, FL 32601, IALT 520, USA

distinctive for different students and for a given student across environments. Yet our analysis also identified several common themes that provide insight into students' thinking and suggest directions for future research.

Vygotskian hybridizing as a frame for thinking about motion and mapping

Vygotsky {1986) outlines a theory of the developmental pathways of distinct forms of thought. In particular, he makes a distinction between "scientific" and "everyday" concepts, in their nature and in their developmental properties, while also emphasizing that they evolve and operate together in an *interfunctional system*. In their nature, the two forms of thoughts have contrasting and complementary strengths, Vygotsky writes: "The strength of scientific concepts lies in their conscious and deliberate character. Spontaneous concepts, on the contrary, are strong in what concerns the situational, empirical, and practical" (p .194).

In their growth, these two forms of knowledge are also complementary. On the onehand, scientific concepts develop from verbal definitions which, when applied systematically, enable them to grow "downward" to engage with concrete phenomena. On the other hand, everyday (or "spontaneous") concepts develop from concrete phenomena, building "upward" through generalization (p. 148). These developmental trajectories converge: "the child's scientific and his spontaneous concepts develop in reverse directions: Starting far apart, they move to meet each other" (p. 192). More emphatically, Vygotsky explains:

> One might say that *the developmentof the child's spontaneous concepts proceeds upward , and the development of his scientific concepts downward,* to a more elementary and concrete level. This is a consequence of the different ways in which the two kinds of concepts emerge. The inception of a spontaneous concept can usually be traced to a face-to-face meeting with a concrete situation, while a scientific concept involves from the first a 'mediated' attitude towards its object. *(p. 193-4, emphasis original)*

Given the contrasts between them, one might expect that everyday and scientific concepts represent forms of thought that are characteristic of *phases* in human development, and that as a learner matures, one form gives way to the other. Yet, Vygotsky insists that these two forms of thought participate in an *interfunctional system* (p. 167). Thus, they cannot be treated independently, and distinctive stages in human functioning must be understood as arising from distinctive patterns of coordination or "hybridizing" of the two. In this view, development "depends on changing relations between them" and the "development of each function, in turn, depends upon the progress in the development of the interfunctional system" (p. 167).

In spite of the importance of the intertwined nature of the development of everyday and scientific concepts, Vygotsky's account is light on the dynamic (and microgenetic) details of the hybridization processes that produce changes in learners' interfunctional systems of concepts. In the context of transformational geometry, the present study fills this gap, exploring how computational environments can support and illuminate Vygotskian hybridizing - in particular, the integration of everyday concepts of transformations-as-motion with scientific concepts of transformations-as-mappings in the interfunctional system of students' sense-making and actions.

A final word is necessary on Vygotsky's {1986) view of the integration of everyday and scientific concepts in "real" or "productive" thought as "thinking in concepts" (p. 204). Such thinking, he says, "is based on 'insight,' i.e., instant transfiguration of the field of thought." (p. 205). This transfiguration is produced by conceiving a "structure of generalization" that is appropriate to the problem at hand and that includes "its own class of logical operations" (p. 205). This *system* of objects and operations provides a frame in which the problem can be solved. The "absence of a system" of this type is, according to Vygotsky "the cardinal psychological difference distinguishingspontaneous from scientific concepts" (p. 205).Thus, in our study of students' Vygotskian hybridizingof motion and mapping concepts, we attend particularly to moments of such "insight," in which the students conceive of *systems* of computational operations that organize everyday motion concepts into a structure that enables them to reason about problems posed bya game level or by their own creative endeavors.

The description of "insight" above also indicates aspects of the role that can be played by computational environments in students' work. We describe this in terms of "co-action" (Moreno-Armella & Brady, 2018; Moreno-Armella & Hegedus, 2009), a process by which the properties of executable representations provoke and provide feedback for learners' actions. A key feature of co-action is the deeply interactive and emergent nature of the mediated action produced by learners-with-tools. As one consequence, different students can engage with the same learning environment to construct different insights, structures, and mediated actions.

Teaching and learning about geometric transformations

Research has underscored that a key challenge in teaching and learning about transformations is to understand how learners move from the motion to the mapping conception. Yet this movement need not be characterized as a one-way transition. Sinclair and Moss (2012) suggest *oscillations* between the everyday and formal conceptions that result "in intermediary hybrid forms of geometric communication" (p. 43). Based on Sinclair and Moss's statement, scholars in mathematics education have emphasized "the need to study the transition phases in the progress of geometrical concept formation" (Sinclair et al., 2016, p. 696).

Within this context, we are interested in the dynamic nature of students' idiosyncratic thinking about transformations as analyzed through the framework of Vygotskian hybridizing with a focus on how students integrate aspects of everyday and "scientific" (mathematical) conceptions using mediating environments to reveal the microgenesis of geometric transformations understandings that incorporate both motion and mapping perspectives.

With respect to motion, the mathematics of the position, direction, and visualization of two-dimensional shapes are related to our spatial experiences of the natural motion of our bodies and of things around us (Edwards, 2003; Lehrer & Romberg, 1998). Research on geometric transformations suggests that, in this "motion" conception of transformations, students and teachers act as if the Cartesian plane was a stage on which geometric shapes can be manipulated through physical properties (e.g., the application of a force, Yanik & Flores, 2009; Yanik, 2011, 2014). The motion conception of transformations also builds on our embodied experiences as reported by Edwards (2003, 2009). Thus, rotation builds from scenarios where centers of rotation belong to the shape to which the transformation is applied, following our bodily experiences of pivot motion (e.g., turning on one foot), while reflection grows from the experience of a mirror placed on a line of symmetry. According to some scholars (Edwards & Zazkis, 1993; Hoyles & Healy, 1997; Ng & Sinclair, 2015), students thus often relate transformations to their embodied experiences. For example, when translating, students may think of the shape taking on a continuous sliding motion. When reflecting, most students execute a flip along one edge of the shape (Edwards & Zazkis, 199,3) which relates to their own experience of flipping their hand. Finally, when rotating, students often connect to the turning of the hands on a clock.

In terms of the curriculum, teaching about geometric transformations is typically approached through a learning progression that begins in the early grades with a focus on describing transformations in terms of motion and then shifts in later grades toward exploring symbolic representations of them as functions (Fife et al., 2019). Thus, the learning progression typically begins in the early grades with the motion of physical objects that allow the analysis of their positions and movement employing slides, flips, and turns. The intermediate grades focus on the definitions of direction, the center of rotation, angle of rotation, and line of reflection, along with congruence as an invariant property under transformations. There are some differences between countries/jurisdictions in learning expectations for higher grades (e.g., the US National Council of Teacher of Mathematics [NCTM] (2000) recommends that students understand changes in representation, while Alberta Education (2016) focuses on the concepts of symmetry groups), but in general, higher grade levels emphasize formal aspects of geometric transformations (e.g., mappings of the plane onto itself).

While embodied and everyday understandings about motion thus pervade students' thinking about geometric transformations, the formalism of geometric transformations is typically also highlighted by the curriculum in two ways, as described by Edwards (2003, 2009): (1) the deductive logic that leads to formal proof and (2) the study of visual objects such as points, lines, planes, in a framing that foregrounds notions of invariance, group theory, and mappings. The second emphasis derives from the *Erlangen program* suggested by Felix Klein in the later 1870's, which supports the centrality of transformational geometry as the study of the groups and symmetries corresponding to Euclidean congruence. Edwards (2003, 2009) explains that this vision implies that students should eventually think about geometric transformations in the way that a mathematician would, "as mappings of the plane, rather than as simple motions of geometric figures" (Edwards, 2003, p. 6). Supporting research (e.g., Hollebrands, 2003; Yanik, 2014) suggests that developing a mapping conception is linked to attending to several geometric transformation attributes (e.g., translation vectors, lines of symmetry, centers of rotation), and conceiving of transformations as one-to-one mappings of the entire plane to itself. From these perspectives, the development of a mapping conception encompasses a learning progression that spans K-12 (Fife et al., 2019). In the seventh and eighth grade context, which are the grade levels of the students in the current study, the particular focus is on the attributes and relationships of geometric transformations that might support an evolving comprehension of a mapping view.

The processes through which students transition between motion and mapping conceptions, however, is not fully understood (Edwards, 2003, 2009; Hollebrands, 2003; Yanik, 2014). Hernandez-Zavaleta et al. (in preparation) have summarized three different research perspectives that explore this problem. The first perspective focuses on how students' and teacher's prior conceptions create opportunities for the development of the mapping vision (e.g., DeJarnette et al., 2016; Fan et al., 2017; Hollebrands, 2007; Mhlolo & Schafer, 2014; Son & Sinclair, 2010; Yanik, 2014), focusing attention on individual cognition. The second perspective deals with collective processes through which learning is constructed in its socio-cultural context, emphasizing the mediation of learning by the teacher-medium-student triad (e.g., Bartolini Bussi & Mariotti, 2008; Faggiano et al., 2018; Jiang et al., 2016; Ng & Sinclair, 2015). And the third perspective focuses on how the mapping vision is constructed by human beings engaged in a world of physical and embodied experiences that change in constant interaction with their environment (e.g., Edwards & Zazkis, 1993; Edwards, 2009; Gadanidis et al., 2018; Hoyles & Healy, 1997; Jacobson & Lehrer, 2000; Panorkou & Maloney, 2015).

Research conducted within each of these three perspectives has engaged digital media. In the first and second, the use of dynamic geometry software (e.g., GeoGebra, Cabri) predominates. Meanwhile, the use of programmable environments such as Logo or Scratch is primarily found in the third. Design-based research with programmable and dynamic geometry environments for the learning and teaching of geometric transformations have focused on the development of intuitive and visual experiences for learners (Borba & Villareal, 2005; Tall, 1986). Across the research, the growth of these experiences is typically explained in terms of the possibilities provided and limitations imposed by the mentioned environments (Hollebrands, 2007).

On one hand, the rise of Dynamic Geometry (e.g., Cabri https://cabri.com/en/ and GSP https://www.dynamicgeometry.com/) has created learning opportunities in which mouse-actions, particularly dragging (Arzarello et al., 2002; Hollebrands, 2007), allowed the learner and the dynamic representation to interact or "co-act" (Moreno-Armella & Brady, 2018). In this context, however, by introducing another salient motion and dynamism (the drag and drag-test - Laborde, 1993), the very power of dynamic geometry also potentially obscured foundational questions about the relations between motion and mapping metaphors for geometric transformations.

In contrast, with the development of LOGO as programmable environment, Seymour Papert emphasized a restructuration (Wilensky & Papert, 2010) of the Euclidean system, in which embodiment and embodied motion were foregrounded in the concept of *syntonic* learning (Papert, 1980). Notably, some scholars advocate that within these media it can be more intuitive to learn geometric transformations and to adopt new perspectives on how to conceptualize it (Edwards,2009; Gadanidis et al., 2018; Leron & Zazkis, 1992). However, it is still unclear whether and how these environments might aid in the development of a mapping conception of the plane (Edwards & Zazkis, 1993; Hoyles & Healy,1997). Regardless of the media context, mathematics educators agree that there is still much to learn about students' transition from motion to mapping conceptions of transformations (Avcu & <;:etinkaya, 2019; Yanik, 2011).

In this paper we argue that block-basedenvironments provide affordances that enable us to both support and delineate the Vygotskian hybridizing of motion and mapping. From a body-syntonic perspective (Papert, 1980) the utilized environments supported students in the integration of their everyday experiences with geometric transformations formal concepts. We will deepen these ideas in the following sections.

Block-based programming environments and their affordances for learning about motion and mapping

We further propose that particular design features of block-based programming environments can provide powerful affordances to support students as they engage in Vygotskian hybridizing of motion and mapping. By taking up these affordances in their own idiosyncratic ways, students in this study integrated lived and embodied understandings of motion with the formal mathematical conceptions of geometric transformations as mappings or correspondences of points in the plane.

A range of scholars have advocated for computer programming as a medium for expressing *embodied experiences* and connecting embodiment with core disciplinary ideas (diSessa, 2000; Edwards, 2009; Papert, 1980; Sengupta et al., 2018). The iterative and repeated changes involved in modifying and debugging code give a dual sense of the experience: it is *holistic* in itsevaluation of "full runs," and it is also *analytic* in that changes in the behavior "from run to run" are attributable to small changes in the code representation.

In this regard, the embodied experiences perspective also benefits from the removal of notational barriers in block-based programmable environments like Logo or Scratch, which provide a more direct interface to the on-screen world. Furthermore, the perspective of the *turtle* (e.g., the turtle in Logo or sprite in Scratch) permeates these environments, allowing learners to assign instructions to the turtle that cause it to move and interact with its environment (Papert, 1980). Papert proposes the notion of *body-syntonicity* wherein the attributes, perspectives, and behaviors of the turtle "can be explained and understood through simple embodied actions" (Sengupta et al., 2018, p. 55) of the people who interact with them. In this view, turtles are not something unfamiliar to individuals but something of their own, in which their intentions and their experiences of the physical world are reflected

In recent years, following the Logo tradition , research using block-based programming environments (e.g., Scratch, Starlogo, Blockly) has continued to explore mathematics learning. These investigations have shown that environments utilizing programmable blocks are conducive to the development of mathematical experiences that promote the integration of specific disciplinary topics, while also including possibilities for students to express themselves creatively (Gadanidis et al., 2018; Hoyles& Noss, 2020; Weintrop, 2019). Recently, Andersen et al. (2022) have shown that block-based programming provides "dynamic entities" that evolve in parallel with students' verbal (personal and shared) discourses. In this regard, block programming supports students in creating programs that allow them to control graphicelements. This feature, as articulated by Gadanidis et al. (2018), drawing on Papert {1980), provides a "low floor, allowing for mathematics engagement with minimal prerequisite knowledge, and a high ceiling, affording opportunities to extend concepts with more complex relationships and more varied representations" (p. 36). Gleasman and Kim's (2020) work with pre-service teachers also identified how teachers make connections between computational thinking and mathematics using block-based environments, demonstrating that manipulating conditions, data, and events to understand mathematical concepts tended to transform computational thinking into "a conceptual mathematics teaching tool" (p. 85). That said, researchers like Weintrop (2019) state there remains a need to understand the affordances and the best way to structure these environments to support effective computer science and mathematics learning.

For the purposes of this paper, from a body-syntonic perspective, we argue that block-based programming environments offer a variety of routes for supporting and studying learners' development across motion and mapping conceptions of geometric transformations (Edwards, 2003, 2009). On the one hand, such environments can foster projection into a first-person, intrinsic perspective, linking with the Logo tradition. On the other, by encoding transformations as computational representations (e.g., code blocks, parameters, procedures), these actions can afford a variety of conceptualizations and manipulations of geometric transformations.

Research questions

The current study explores four learners' separate interactions in two block-based programming environments : Code the Quilts, a constructionist quilt-pattern activity in Scratch and Transformations Quest, a digital game. Using the lens of Vygotskian hybridizing, we documented the ways learners used and shaped their everyday and scientific concepts as they constructed coding procedures to perform and create understandings of two-dimensional geometric transformations. In so doing, we considered two research questions:

(1) How does students' use of motion and mapping conceptions in block-based programmable environments reflect Vygotskian hybridizing of everyday and scientific concepts?
(2) How do students' interactions with the block-based environments support Vygotskian hybridizing of their everyday and scientific concepts of motion and mapping?

Method and data analysis

We engaged in a comparative case study (Merriam & Tisdell, 2015; Thomas, 2011) to explore how four students in grades seven and eight each engaged in the motion and mapping conceptions within Code the Quilts and Transformations Quest to create understandings of geometric transformations. Selecting a case study design allowed us to consider multiple students, each as a separate case, across two authentic settings (Harrison et al., 2017; Van Wynsberghe & Khan, 2007). We compared how the participants utilized the notation systems in the two environments and looked for pattern similarities and differences within and across students (Gaikwad, 2017).

Descriptions of environments

Both environments offer unique features that lead to distinctive experiences in geometric transformations learning. In the following two subsections, we will describe both block-based programming environments and their particularities.

Code the quilts activity

The *Code the Quilts* activity in Scratch, developed by Author 1, was inspired by and developed from the work of Lehrer et al. (1998). Their research draws on children' s informal knowledge and everyday activities for the development of intuitions about spatial structure. According to Lehrer et al., quilt design is an opportunity to explore symmetry and transformation and "to develop conjectures about how these mathematical ideas might contribute to their aesthetic experiences of the artistry of quilt designs" (p. 180). This activity has six different scenarios, each of them referring to particular transformations (the complete activity can be accessed in the link: https://scratch.mit.edu/projects/403789527/). For instance, the scene in Figure1 shows three 90°Clockwise rotations rotating about the bottom-right point of the core-square.

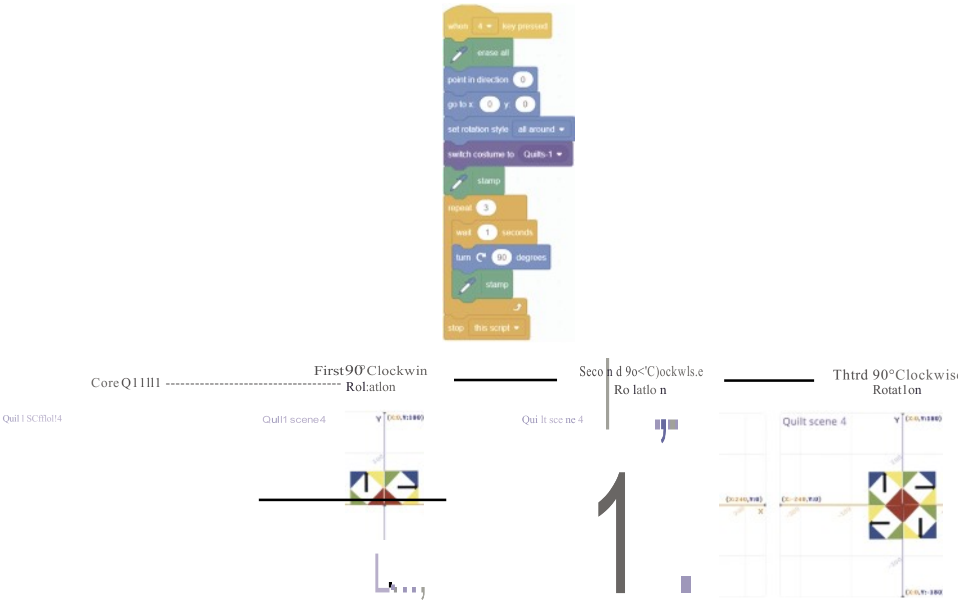


Figure 1. Quilt example with code from scene 4.

The Code the Quilts activity has two parts: (a) description of the quilt motion transformations and (b) quilt (re)design. In Part 1 of the activity, we asked students to describe how the individual quilt squares' colors and arrows "move" over the Cartesian plane (Figure 1, down) after executing the block sequence (Figure 1, top). Later in Part 1, students were asked to explain how each block affects the motion of each colored/arrowed square in the scene. In Part 2, students were invited to utilize the existing sequences of blocks from Part 1 or create their own sequence to make new quilt designs.

Transformations quest video game

The *Transformation Quest* game was developed by a team of researchers led by Author 4 (Clark et al., 2021, 2023). The game builds on ideas from our earlier work on conceptually integrated (Clark & Martinez-Garza, 2012) and disciplinarily-integrated (Clark & Sengupta, 2020; Clark et al., 2015) games, but Transformation Quest is also highly representative of a genre of math games that engage students with geometric transformations on Cartesian planes by specifying sequences of instructions and adjusting parameters. The National Council of Teachers of Mathematics' (n.d.) *Flip-N-Slide* game is a prime example, along with many others such as *Transformation Golf 2* (Hood amath, n.d.), and *Transtar* (Mangahigh, n.d.). In the Transformation Quest game, which consists of 11 levels, players use a limited number of code blocks to direct the movement of a triangular shape to positions on the plane, using geometric transformations (translations, reflections, and rotations). The [Loop] block is a basic block included in all levels. Code blocks with different transformations are included gradually as the level increases. Early levels use only translation blocks; middle levels include translations, then rotations; and advanced levels include a mixture of blocks corresponding to all of the transformations. The number of programming blocks in each level increases with its difficulty.

Regarding the types of levels, seven are *gems levels,* in which players must collect colored gems by landing on them before reaching the exit (red dotted triangle). Four are *Creative Levels,* where players must land on triangular "footprints," with different colors representing different point values (see Figure 2). Players can complete each level with bronze, silver, and/or gold shields, corresponding to different degrees of success, as measured in level-specific ways.

Figure 2. Transformations Quest (https://mathgame.ucalgary.ca). Gem Level Configuration (left), Creative Level configuration (right).

For the purposes of this study, we asked students to go through TQ levels in order. They started by reading the introductory slides that engaged them in the narrative of the game (space creatures invading the planet *Adanac)* and explained how to play. Authors 1 and 3 accompanied the students over all the levels in two 45-minute sessions, occasionally asking participants about their plans to achieve a shield and explaining the types of transformations blocks available to solve a particular level, explaining why they might choose a particular block for the solution.

Differences and commonalities across the two environments

Each activity maintains *its* own distinctive characteristics. Code the Quilts, created in a constructionist tradition (Kafai, 1996) invites learners to construct new ideas and conjectures, adopting an active perspective in designing and creating a quilt. On the other hand, Transformation Quest draws on the player's decisions and choices for the development of understandings of geometric transformations through meaningful play. Despite these differences, the two environments also share some commonalities. In particular, both environments are suitable spaces for the development of students' understandings about motion and mapping, as students explain their constructions of concepts, refer to the turtles (i.e., a quilt or the red triangle), and build code sequences.

Participants

The original plan for this research study was to observe grade seven and eight students playing Transformations Quest and engaging with the Code the Quilts activity collaboratively in a classroom environment. Given COVID-19, and adhering to the university ethics board requirements, the decision was made to recruit available students through our own adult networks (friends and teaching colleagues) and to adjust to the challenges of the pandemic by having Authors 1 and 3 work with four students individually and online in four separate sessions each. Of the four participants, three were of European and one was of Asian descent. E-mail invitations were sent to parents of three participants, with follow-up e-mails sent to determine participants' impressions of the experience and their attitudes toward mathematics. For the fourth participant, his teacher provided background and initiated contact with his mother. Each student brought a unique background to the research study. (All students are referred to using pseudonyms.)

Eric was completing grade eight. He indicated that friends had shared their work in Scratch with him but that he himself had never worked in or created anything in Scratch. He had had no exposure to any other block coding environment. After he participated in the study, his mother shared with us that he had often struggled in math and that participating in the Transformation Quest game had boosted his confidence.

Peter was also just completing grade eight at the time of the study. According to his father, he had no experience in Scratch or any other coding environment, but he did spend a considerable amount of time on the internet and playing video games. His father expressed some frustration that Peter, though very capable in mathematics and other subject areas, was satisfied with marks in the 70s and 80s and had even brought home failing marks.

Simon was completing grade eight at the time of the study. According to him, he did not have experience in the Scratch interface. Neither his mother nor his teacher provided information about his abilities or attitudes toward mathematics, but we found him to be highly engaged in the activities, particularly in Transformation Quest. Extremely animated and verbal, Simon was open and forthcoming in describing his thinking, as he explained to us how his "brain worked."

Zach was completing grade seven at the time of the study. According to his former grade six teacher, Zach was a Scratch "expert." He spent considerable time in the Scratch interface, and he particularly enjoyed creating games in Scratch. That same teacher described him as a "progressing/proficient student in math. I think he could be better, but his attention span and focus are not there in a formal setting."

Due to the constraints related to Covid-19, the video footage of student participation was recorded in a virtual setting (Zoom) as described above. Participants shared their screen with the researchers while participating in the activity in Scratch and during game play.

Data analysis

The data analysis process involved repeated observations of the videos to identify episodes of interest, followed by clarification, interpretation, and categorization of patterns, based on theoretical constructs drawn from the disciplines of mathematics and computational thinking (Gaikwad, 2017; Merriam & Tisdell, 2015). In order to ensure internal validity, we focused on attending to the patterns in the participants' explanation building, and we analyzed changes in their understandings over time (Gaikwad, 2017).

Authors 1 and 3 conducted the first cyclic process of coding, to identify in each participant specific "actions intertwined with the dynamics of time such as those things that emerge, change, occur in particular sequences or become strategically implemented through time" (Saldana, 2016, p. 111). The focal episodes included passages where students articulated their thinking in words and embodied gesture, (e.g., referring to a shape in motion as if it were themselves- "I go upwards") and where they explained their understanding of transformations and geometric properties while solving a level or programming a quilt. Based on Harrison et al. (2017) and Simons (2009), Authors 1 and 3 then engaged in a second cycle of analysis of the videos, focused on identifying and classifying thematic patterns in the students' work and gathering multi-modal evidence of their emergent ways of understanding. This analysis highlighted (a) students' spoken explanations of their understanding of transformations; (b) students' embodied explanations of transformations, interpreted with reference to the students' gestures and mouse movements in the screen; and (c) students' sequences of coding blocks (which may suggest aspects of student understanding of transformations) that the students created in order to solve a level or complete a quilt pattern.

This focus enabled us to observe how the students constructed hybrid conceptions of motion and mapping, in which neither motion nor mapping alone guided their thinking, but something new was enacted, which incorporated aspects from both. Moreover, as described below, we were also able to distinguish three crucial interactions that supported hybridizing of motion and mapping, afforded by the environments' computational components and design. Given the complex nature of the data and analysis process, our findings focus on a synopsis of significant incidents identified to us through a combination of students' actions and comments.

Findings

In this study, we found that the two block-based computational environments scaffolded students' building of geometric transformation understandings by promoting *hybridizing of motion and mapping*. Over an extended time period, we conjecture that ongoing engagement in hybridizing could contribute to a more durable and stabilized *hybridization* of motion and mapping in learners. Thus, we are not claiming that the interactions evidenced in the block-based environments by themselves represent permanent learning/hybridization of the concepts for the students, but rather that these interactions represent important seeds and foundations for learning/hybridization.

The findings section is divided into two main parts, each based on one research question. In part one, we provide illustrative examples of two students' hybridizing, one in Transformation Quest and one in Code the Quilts. In part two, we delve into more detail about three specific forms of interactions that the four students engaged in, which supported hybridizing and were afforded by both environments: discretizing, parametrizing, and composing-and-encapsulating.

Part 1: Students ' Vygotskian hybridizing of motion and mapping in transformations quest and code the quilts

Our analyses of Eric in Transformations Quest and of Zach in Code the Quilts illustrate how they each created hybrid conceptions of the mapping actions demanded by the computational environments, while still leveraging their motion experiences as lived in the real world. We address research question one (i.e., How does students' use of motion and mapping conceptions in block-based programmable environments reflect Vygotskian hybridizing of everyday and scientific concepts?) by recounting how the students leveraged their motion conceptions when they encountered challenges in the environments that favored elements of a mapping view.

Eric's hybridizing of motion and mapping in transformations quest

We observed that Eric hybridized motion and mapping as he constructed a system of rotation transformations by integrating (a) the continuous sense of a transformation, grounded in his experiences of motion in a physical/continuous world, with (b) the discrete nature of motion as enacted in the environment. Essentially, Eric represented the continuous embodied action of rotating his triangle about a vertex, in terms of a series of discrete 90° angle turns. Here, his motion conception supported a continuous image of circular motion, while his mapping conception allowed him to visualize before/after states of partial rotations along the circle, creating quarter-circle turn-units (see Figure 3).

The mapping view became salient when Eric represented the 90° angle turn as a unit, as he used a specific number of loop iterations to achieve transformations needed in Transformations Quest. His reasoning with these 90° turn iterations further provided a tool for thinking about an operational system of transformations. This example draws on Eric's gameplay in Transformations Quest Level 6 (See Figure 4). There, players must use transformations to move the red triangle to collect (land on) the blue and yellow gems; then reach the "doorway" to exit the level (the dotted triangle outline at lower right), all without landing on any of the monsters.

Eric chose to collect the yellow gems frrst. He used the code block [Rotate CW 90°][1] within a [Loop *n]* block (inputting [Loop repeat 3]), then he executed the code and observed how he landed on and collected the three yellow gems.

Immediately after this result, and without hesitation, Eric changed the number of repetitions to five and re-ran the sequence. Researcher 1 asked him why he used five repetitions. Eric's answer focused on the triangle's orientation: "So that I would end up back at the orientation, um, so that I would be able to fit in through the doorway." Figure 5 shows the red triangle positions after each repetition (see Excerpt A in the Appendix Collecting Yellow Gems).

Eric also accessed his personal experience to build understandings of transformations. Explaining his work to the researcher, he articulated his thinking about continuous rotation

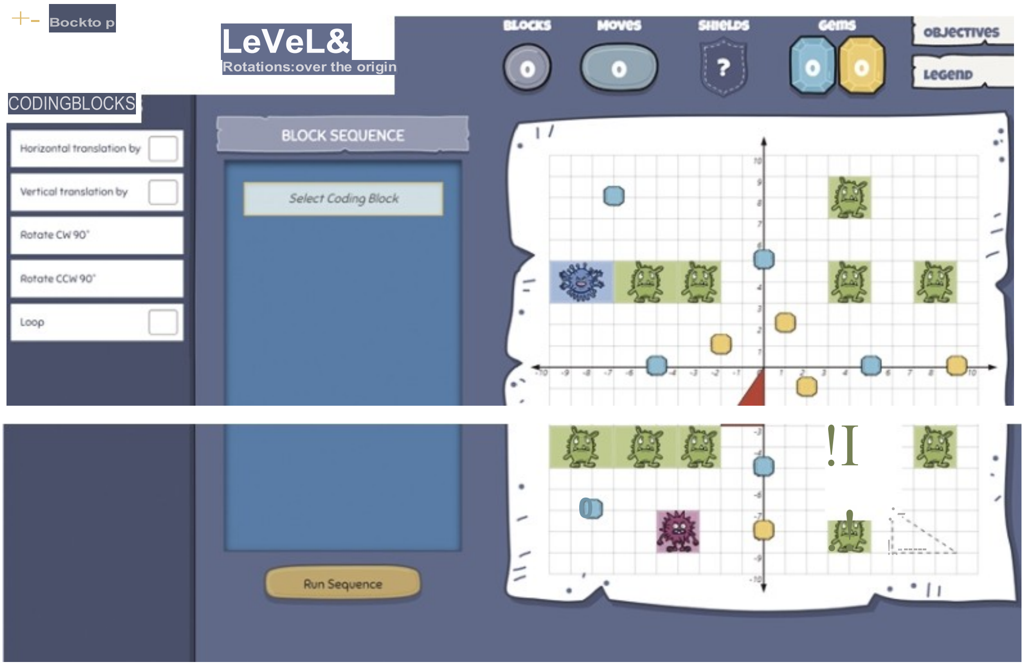


Figure 3. Initial set - up for transformat ions quest level 6.

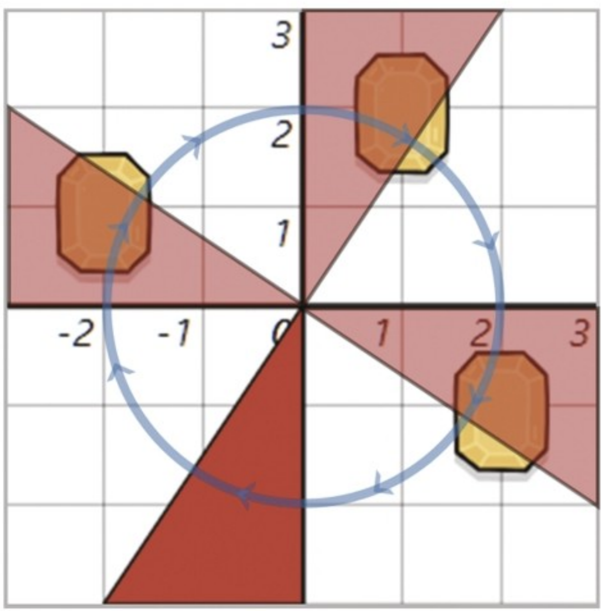


Figure 4. Triangles in red positions anchored the before/after states of partial rotat ions to co mplete a 360 ° T urn.

by using an uninterrupted circular movement of the mouse cursor over the gems in conjunction with the words "spinning, circle around." He went further by sharing a metaphor of drawing a circle and saying the "triangle would be the pencil," linking with its orientation when he

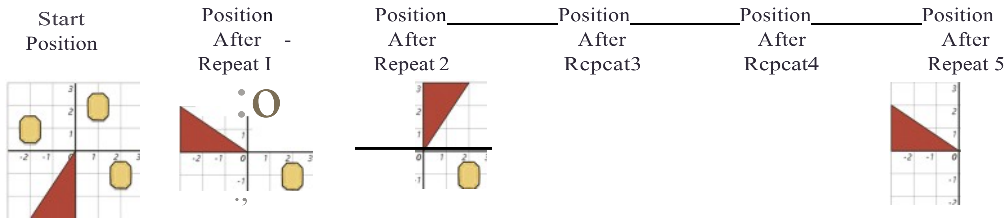


Figure 5. Eric's process to acquire the three yellow gems around the origin and orient the red triangle to match the level Exit.

stated, "It's kind of like it's cartwheeling around ." (See Excerpt A in the Appendix Collecting Blue Gems).

These methods of describing his plan suggest that Eric used his understand ing of a continuous rotation transformation, grounded in his motion embodied experiences, as a jumping off point in the game. Yet, the game incentivized him to think in discrete steps. His explanation "go clockwise three" suggested that he represented his solution with a series of definite actions that would lead him to land on specific gems to collect them. We came to call this kind of interaction with the computational environment *discretizing* (see Table 1 for definition). By this, we mean that students made sense of the before/after motion effects produced individually by each block of code. We will delve into this and other forms of interactions in more detail in part two of the findings. This is related to what Sengupta et al. (2012) found in terms of the affordances of block-based Logo style environments for supporting an understanding of continuous change from discrete movements through activities exploring aesthetics.

Also important to note is that Eric came to see the *unit 90°CW rotation* in relation to the system of rotations that it could generate. In particular, through use of the loop block, he could compose any number of 90°CW transformations; and with the recognition that (Loop 4 [90°CW]] returned the triangle to its initial position, he could infer that (Loop 5] would bring the triangle to the same position that a single application of the 90°CW transformation would achieve. Eric's ability to see that (Loop 5] produced the same position as the single (Rotate CW 90°] block when collecting the yellow gems demonstrates that he understood the correspondence in the operational system: the position after [Loop 1] is the same as position after (Loop 5] (see Figure 5). Eric's confidence and precision using the number of repetitions offered evidence about how he constructed and represented interactions that led him to grasp an operational system of rotations leading to group theoretic understandings. We came to call this kind of interaction *parametrizing* (see Table 1 for definition). Here, parametrizing is related to sense making about the "range of motion" of a chosen concrete "unit" transformation, by repeated applications of that transformation through the utilization of particular parameters across the plane.

Finally, we can say that the action of Vygotskian hybridizing arose in response to the interrupted continuous quarter circle turn that was refined as the before and after states were highlighted. By the end of the level, Eric's attention moved fluidly between envisioning the arc of a continuous rotation, and attending to the after-states (i.e., the position of the triangle after each discrete 90° turn). In the language of transformational geometry, he was able to envision the *image* of the triangle under each successive 90° rotation. This then made the mapping of pre-image to image an action that Eric could leverage when solving the game level problems. In this case, we can say that Eric engaged in Vygotskian hybridizing of motion and mapping when he blended his everyday experiences with the represented discrete nature of motion in the Transformations Quest game. This allowed him to integrate geometric transformation understandings into big ideas in mathematics, in the sense of Gadanidis et al. (2018) (e.g., the beginnings of a group theoretic understanding of a cyclic group generated by a rotation).

Table 1. Summary of the stude nts' observed types of interact ions in the two environments.

Terms	Definition	Linked Computationa l Rep resentation
Discretizing	*Discretizing* occurs when stude nts integrate (a) the qualitat ive and cont inuous sense of a transformation grounded in motion-based embodied experience with (b) the mapping-based idea of a quantified, discrete action, having before/ after states that differ in a measurable **way.** Stude nts engaging in discretizing trade off the simulation of a transformation through movement with the emphas is on before/ after states .	Code blocks that perform a particular geometric transformation, along with the instantaneous effecting of motion in the two environments .
Parametrizing	*Parametrizing* occ urs when students (a) ident ify the impact and implications of manipulating the parameters of a particular transformation and (b) explore how particular parameter settings **will** impact what can be achieved with repeated applications of that transformation. This allows the structuringof space and of transformations of t hat space in relation to particular arguments for this type of transformation. As learners structure and quantify the space in this **way,** they prepare the **way** for spec ific transformations to act as a "unit" of measure or as the generator of a system (an algebraic gro up) of transformations.	Parameters in the code blocks that control the geometr ic transformations arguments.
Composing-a nd-Encapsulat ing	*Composing-and-Encapsulating* consists of th ree crucial conceptual actions: (a) recognizing patterns in sequences of actions, (b) treating sequences of actions (possibly including multiple *types* of geometric transformations) as chunks that can be understood in terms of their total effect, and (c) reusing and combining these chunks as higher-level building blocks for procedural use in multiple situations . As lea rners engage in this interaction, they organize code into a single unit or procedure that can be used to create complex co mposit io ns of geometric transformations.	Procedures (block-stacks) that create complex compositions of geometric transformat ions.
Vygotskian Hybr idizing	A dynamic action is taken by students when interacting in the e nvironment, accessing both personal experience and formal knowledge of motion and mapping to build understandingsof transformations, to solve proble ms involving tran sformatio ns, or to predict the actio ns of transformations. This relates to Vygotsky's propositions about the relationship between "everyday" and "scientific" concepts , wherein "everyday" concepts can be extended, refined, and integrated with the associated• scient ific" concepts.	

Zach's hybridizing of motion and mapping in code the quilts

Zach's approach to hybridizing motion and mapping in Code the Quilts (Part 2) appeared in his construction of an extended symmetric pattern using a family of horizontal translation transformations. Zach's prior personal experience in Scratch programming allowed him to construct chunks of code that could be understood in terms of their total effect. In other words, Zach used the interactions afforded by Code the Quilts to focus on manipulations of code block chunks to construct personally meaningful patterns by composing transformations. This example draws on Zach's reuse, remixing, and extension of Quilt Scene 6, which focused on reflection across x and y axes using different quilt costumes. Figure 6 shows the pre-programmed pattern that Zach used as a starting point for his quilt design. The orange dashed square indicates the *core quilt.* The core quilt is the basis for the programming; in other words, every block in the sequence modified

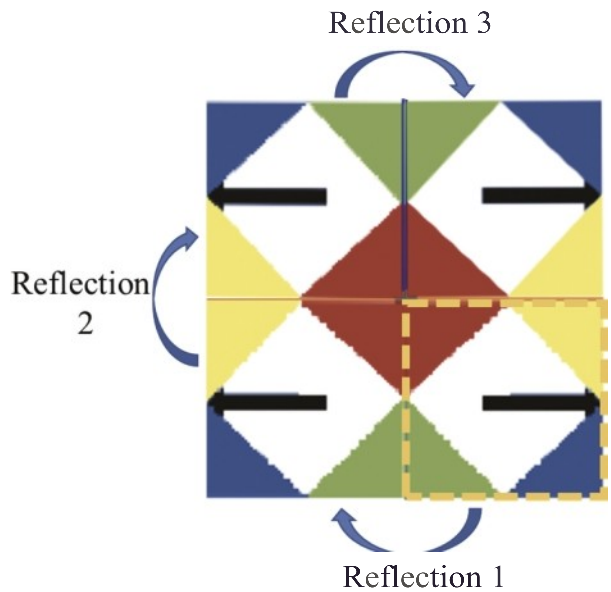


Figure 6. Quilt scene 6 four-square quilt pre-programmed pattern.

this quilt or made copies of it. The arrows indicate the construction order of the *four-square-quilt pattern*.

In this example, we describe how Zach reused a block sequence to create a finite figure by assembling four stamped images of the core quilt pattern around the origin (Figure 6), and then to use this composite object as a motif in a larger pattern (see Figure 7). As a key step in this process, Zach intuitively determined that due to the symmetries of the newly-constructed four-square-quilt pattern, a horiwntal translation was equivalent to a reflection over a vertical line positioned at its right-hand edge. The resulting image preserved the symmetries of the original four-square-quilt pattern, incorporating them into a larger extended family of symmetry patterns that included horiwntal translation (see Figure 7). It should be noted here that there were limitations within the Scratch environment in relation to the use of the code blocks as related to the representation of transformations (e.g., to create a reflection across *x* axis required use of the [Switch Costume] block). Even given those limitations, constructing this figure acted as the basis for the development of an infinite *frieze pattern* understanding. This evidence is the essence of a reflection symmetry property.

Zach's process (Figure 7) involved modifications from the original pre-programmed code affecting groups of quilt squares. In this process, Zach first moved the four-square-quilt pattern to the left by modifying the code. Then he affirmed he could use a repeat block to "stamp" it again, stating that this

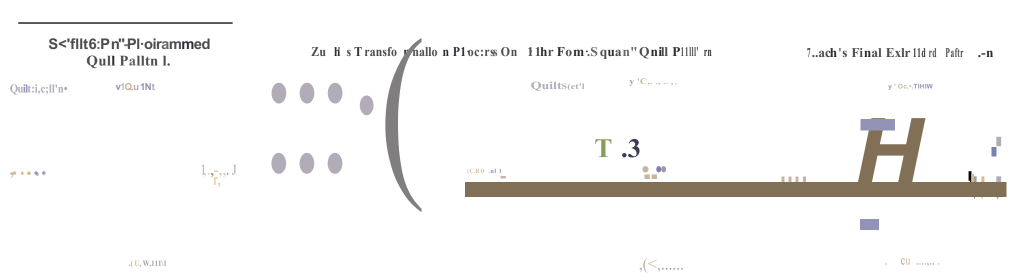


Figure 7. Zach's process of construction of the basis for the frieze pattern.

time it would be in a different place, pointing to the right side of the y axis in the plane. He explained he was going to build "an extended pattern" of the quilt, where "the right side and the left side are reflected off [i.e., 'over'] the y axis." (See a complementary transcript in Excerpt B in the Appendix.)

After he ran the code, a four-square-quilt pattern was produced, followed by the reflected image of the four-square-quilt pattern appearing on the opposite (right) side of the y axis (see Figure 7). Researcher 3 asked him if he thought he would be able to enlarge the quilt by adding more code. He answered, "Yes I could," indicating his confidence and understanding of the functions of individual blocks, chunks of code, and parameters.

We posit that two moments in particular contributed to Zach's hybridizing using chunks of code. The first moment was when Zach presented his plan, showing how he considered two chunks of code as composite units in designing the reflected extended pattern. This was evident when he said, "I just stamp this again ... and I get a repeat block." Additional evidence that Zach's thinking included encapsulation was when he stated clearly, "Ok that's one," referring to the first four-square-quilt pattern loop repetition as a unit that could itself be transported and replicated. Zach's statement here supplied strong evidence of conceptual chunking; he made a composite *unit* that consisted of four sub-units which provided the basis for his second repetition pattern, although in a different plane location. It is important to mention that the four-square-quilt pattern already had internal reflection symmetry (as the arrows point to opposite sides in Figure 6); this was a crucial feature in using translations to effect a reflection across the y axis. We came to call this kind of interaction *composing-and-encapsulating* (see Table 1 for definition). This interaction is linked to the computational representation of procedures and also to the creation of complex compositions of geometric transformations, viewed as functions.

A second hybridizing moment is highlighted by Zach's work with the translation blocks. The way in which he positioned the blocks inside and outside the (Repeat] block demonstrated his understanding of the parts of the sequence that needed to be repeated. These actions displayed a deep engagement in representation, where Zach visualized the reflection as a discrete operation, evidenced by the use of the translation and stamp block to create the reflected/translated pattern on the right side of the y- axis. Even though Zach was using translations, his visualization of the final extended pattern was a reflection. In this context, Zach's desire to make a reflection of an aggregate pattern drove him to see an equivalent mapping operation. That is, in mapping the four-square-quilt using translations he was able to achieve a quilt pattern with reflection symmetries. This moment revealed that Zach had constructed a deep understanding of the reflection, as he understood that the horizontal translation of a pattern with internal reflection symmetry would create a reflected total outcome.

Zach's interactions also exhibited parametrizing. He showed quantification of the space in order to move the quilt squares to the desired location. Central to this work was the conceptualization of a *unit of translation*. In this case, he used the number 67, equal to the length in pixels of the sides of one quilt square. The discrete motion became quantifiable as he was looking for multiples of 67, for instance, when he changed 134 to 67 in the x coordinate of the [Go to x y] block. Zach needed this unit to find the right spot where the second run would start.

Finally, we can say that the Vygotskian hybridizing of motion and mapping arose for Zach in the construction of an extended pattern and his intention to consider extending it forever. As Zach recognized chunks of code that were common to all the codes in the Code the Quilts activity, he mentally constructed a block procedure that could be repeated in different plane positions, showing a higher-level use of code chunks and correspondingly chunked visual units within an emerging frieze pattern. We considered Zach's statement "So, it's like an extended pattern" to be evidence that he recognized implicitly that translation and reflection have the same effect in this case. And while the "extended" pattern was not yet seen to be infinitely extensible, he was on the cusp.

Summary: commonalities across both examples of vygotskian hybridizing

In both examples, we could see indications of how the conceptual perspectives of motion and mapping materialized in the interactions between students and computational environments. An important fact was that the two perspectives did not appear dissociated; instead, the students showed a hybridizing that led them to generate understandings about geometric transformations based on the development of "big" mathematical ideas (Gadanidis et al., 2018), in this case, *group theory* and *the construction of frieze patterns.* These forms of hybridizing motion and mapping were expressed in the concrete interactions we identified in our analysis and labeled discretizing, parametrizing , and composing-and-encapsulating. Both students blended their embodied experiences with the discrete nature of motion provided by both environments. Distinctive features of the students' personal experience and of the computational environments make these interactions stand out in distinctive combinations within each episode analyzed, i.e., while Eric emphasized discretizing and parametrizing, Zach emphasized composing-and-encapsulating. We consider, however, that in both cases hybridizing of motion and mapping promoted understandings about geometric transformations.

We define and summarize the three observed types of interactions in Table 1 as a means of guiding the reader in understanding the illustrative examples presented in the remainder of the findings. Each interaction is linked to a particular computational representation that fosters it.

The characterization of the observed interactions in Table 1 provides a landscape of how the students engaged in building understandings of geometric transformations in Transformations Quest and Code the Quilts environments. It also organizes how we conceive, according to the data, the Vygotskian hybridizing of motion and mapping in these computational environments.

Part 2: How students' interactions afforded by the block-based environments supported Vygotskian hybridizing of motion and mapping

In this part, we provide specific examples of interactions within Transformations Quest (Simon and Peter) and Code the Quilts (Eric and Zach, using Zach's example described in part 1). We illustrate how these students hybridized motion and mapping conceptions by leveraging intrinsic features of the computational environments identified in the previous section: code blocks (discretizing), parameters, (parametrizing) and chunks of blocks as procedures (composing-and-encapsulating). Although we observed the students engaged in discretizing, parametrizing, and composing-and-encapsulating in complex and interconnected ways across the game and quilt design environments, for the purpose of providing clear explanations, we present each interaction as a separate section in the findings. In the interest of space and practicality, we limit to two new examples in Transformations Quest and one in Code the Quilts.

Students discretizing through the use of code blocks supported sensemaking about the images of transformations as before/after states of motion

We found that one feature of learners' hybridized thinking involved conceptualizing transformations by blending images of continuous motion from the physical world on one hand, with the before/after states of motions produced by code blocks on the other. An affordance of this perspective is that it not only allows the learner to envision before/after states for a single transformation, but it then also positions them to envision and iteratively apply that transformation in a *family* of transformations (i.e., the transformation *group* generated by single transformation). Even though the nature of motion in the presented computational environments is discrete, students tended to blend their personal experiences from life in a continuous world to make sense of the discrete motion on the screen. We came to call this kind of interaction *discretizing,* as defined in Table 1. The following examples will focus on the role of discretizing in students' hybridizing as they worked with transformations, which was foregrounded by the use of computational abstractions in Transformations Quest and Code the Quilts.

Discretizing in transformations quest. Simon animated his hybrid thinking in language by exclaiming "boom!" as he created and ran a program that ensured he landed only on yellow footprint triangles in Level 2 (Creative). (He received two points for landing on yellow instead of only one point for landing on purple). He constructed a block sequence, entered the parameters for all the moves, and then ran the code: "Let's test this. Boom, boom, boom, boom, boom, boom." Simon said "boom" each time the red triangle landed on a yellow footprint. We posit that with each "boom" he was connecting the motion of the triangle with steps in the discrete sequence of blocks in the program. In other words, Simon accurately predicted and verbally annotated the discretized motion sequence of positions of the triangle in the plane, while also maintaining a sense of it as a (continuously) moving object. The code blocks enhanced his hybridized view of motion and mapping, demonstrating an effective connection ("boom") between the predicted step-by-step block pattern and the resulting motion of the triangle.

Peter 's approach to discretizing with code blocks in Transformations Quest began initially with his exploration of block sequences to collect the gems using translations alone - he was making sense of this transformation's motion before/after states through observing the actions produced byeach block. Once he understood the problem from a more holistic perspective, however, he modified his block sequence to include different transformations (including reflections and rotations) as well as the loop block (Excerpt C in Appendix shows Peter's initial and final code). We interpret this incremental adoption of repeated block-representations as his leveraging of motion conceptions to understand mapping. His initial discretizing of step-by-step translations (as motion) supported him to later see how he could hybridize motion and mapping by entering a series of commands to perform the geometric transformations he was envisioning.

Discretizing in code the quilts. Eric's approach to discretizing with code blocks in Code the Quilts (part 2) enabled him to create a code sequence for a "tank" quilt pattern that was separated into four parts (the body, the turret, the gun,and the treads. Excerpt Din Appendix shows an Eric's tank Figure and transcription). The action ofseparating the design into components implied he was makingsense of the quilt design in terms of before/after states of available transformation blocks. He chose a translation block code and duplicated it in order to explore the creation of the distinct parts of the tank. We interpret this action as evidence of his understanding of the before/after states produced by the function of each block. This led him to construct a final quilt pattern based on the photo image of a real tank that he had located online.

In summary, both computational environments supported students in hybridizing motion and mapping through *discretizing* actions. These kinds of interactions enabled students to maintain a link to their embodied understandingsof transformations while reasoning about the effects of the blocks, whose actions were represented in terms ofdiscrete before/after states.

Students' parametrizing through the manipulation of parameters supported sensemaking about the reach of iterated transformations and about how they could structure and quantify the computational space

Another feature of learners' Vygotskian hybridizing appeared as they identified the impacts of manipulating the parameters of a particular transformation (e.g., the number of units to translate in a certain direction). As learners manipulated parameters, they were in a position to structure and quantify space in ways corresponding to the parameter and the transformation. In doing so, they prepared the way for specific transformations to act as a "unit of measure" and also to generate a system of transformations by *iterating* it. Space could be structured and quantified in terms of the locations that could be reached by iterating the transformation. We came to call this kind of interaction with the environment *parametrizing,* as defined in Table 1. This interaction turns the qualitative, continuous sense of transformations' motion into an iterable, measurable, and quantified sense of the repeated application of a unit action. In the following paragraphs, we illustrate how the students in our study parametrized in the environments.

Parametrizing in transformations quest. Simon's work in Transformations Quest Level 4 illustrated his use of parametrizing, involving reflections combined with translations (i.e., glide-reflections), which enabled him to navigate the plane horizontally and vertically. He used the [Reflect across y-axis] block to move from a location a given distance to the left of the *y-axis* to a location the same number of units to the right of the axis. He then used [Vertical translation by-6] to move downwards. Using this combination of reflection and translation, Simon structured an iterable composite transformation as the image of a *glide-reflection,* which he then repeated three times (see Excerpt E in the Appendix for a transcription and code). The positions of the blue gems in the levelscaffolded this construction because the distance between the original and reflected image was the same, thereby making the glide-reflection a useful "unit" out of which to trace a path to reach the exit. Simon's hybridizing of motion and mapping occurred when he considered the number of iterations, of horizontal and vertical movements together, as parameters to complete the level.

Peter's approach to parametrizing in Transformations Quest Level 7 used [Rotate CCW 90°] in combination with [Verticaltranslation by 2] to circle around the origin *three times,* collecting blue gems. In this case, [Verticaltranslation by2] and the size of the smaller leg of the triangle allowed Peter to quantify the separation between pairs of blue gems; while rotation in units of 90° allowed him to move long distances to land on locations 90° from the prior spot. This use of the parameter in the repetitions (three times) showed that Peter was able to identify the impact of each repetition to predict the final position of the transformations, which demonstrated that he had an accurate sense of the motion space in the Cartesian plane (see Excerpt F in the Appendix for a transcription and code).

Parametrizing in code the quilts. Eric's approach to parametrizing in Code the Quilts was evidenced (part 2) when constructing the tank's gun. Here, he considered the use of the translation's parameters (direction and distance). He discovered that the default quilt square's side length was 67 pixels, therefore, translating by 67 would place a copy adjacent to the original. This allowed him to think of the quilt squares' desired positions in terms of an iterated translation from a starting point by multiples of 67. When his tank code had a "bug," Eric observed that he left a blank space between the tank's body and its gun, measured at 67 pixels. Upon seeing this, Eric determined that he needed to change the parameters of the two gun quilt squares, thereby eliminating the blank space and connecting the gun to the body. In doing so, Eric used 67 as a unit to quantify how far in the Cartesian plane he should move a quilt (see Excerpt D in the Appendix for a transcription and code).

In summary, by parametrizing discrete motions, students used the **[Loop]** block to repeat single or composite transformation as a "unit." This had the effect of iterating the unit transformation of motion and moving their sprite a specific number of times to achieve desired end results. Another sign of parametrizing was the manipulation of parameters to structure and quantify the space of the computational environment in terms of the discrete unit transformation.

Students' composing-and-encapsulating through the reuse of chunks of code blocks supported sensemaking about compositions of geometric transformations

A final interaction that we identified in students' hybridizing involved the use and reuse of chunks of code blocks as procedures, which allowed students to solve multiple situations. For students, focusing on chunks of code blocks has the affordance that it allows them to treat complex compositions of geometric transformations as a single runnable unit of code. This creates the opportunity to understand chunks of code in terms of their total effect and to use them as higher-level building blocks that can be applicable in multiple situations. We came to call this kind of interaction *composing-and-encapsulating,* as defined in Table 1. In the following paragraphs, we illustrate how the students in our study composed-and-encapsulated in Transformations Quest and Code the Quilts.

Composing -and-encapsulating in transformations quest. Simon's work in TQ Levels 3 and 4 illustrated his use of composing-and-encapsulating involving reflections combined with translations (see Excerpts E and Gin the Appendix for a transcription and code). In Level 3, Simon constructed a step-by-step six-block sequence alternating between [Reflect *x]* and [Horizontal translation by 5]. After collecting a bronze shield in Level 3, Simon used the same strategy to solve Level 4 (we analyzed this episode in "Parametrization in Transformations Quest" above), creating a six-block sequence repeatedly alternating between [Reflect y] and [Vertical translation by - 5], this time inside a [Loop] with *one repetition* (he did not immediately realize that looping once was useless). Here, Researcher 1 prompted him to go back to his code sequence and analyze what he did. Simon demonstrated that he was composing-and-encapsulating by recognizing the pattern and then removing all but one copy of the core sequence from Level 3, [Reflect y] (Vertical translation by - 5]. Finally, he changed the Loop argument to *three repetitions*. Simon's hybridizing of motion and mapping was directed in this case toward his improved use of the loop block from level to level to repeat an encapsulated action. This also gave Simon insight about the *glide-reflection* as a composite transformation.

Peter's approach to composing-and-encapsulating in TQ was developed across levels by reusing strategies and improving them using loop structures. Peter solved Level 7 for a gold shield, using the structure [Loop 3, [Rotate CCW 90°], (Vertical translation by 2]]. Peter created this chunk of code by recognizing a pattern in how the blue gems were distributed in the level and treating the composition of two different transformations inside the loop block in terms of their total effect. Specifically, he needed the combined outcome of the two transformations in order to fit through the level's exit. In other settings, Peter also exhibited composing-and-encapsulatingas he reused old structures from previous levels. For instance, the (Loop *n,* (Horizontal translation *x]]* structure was a chunk of code that he frequently employed across levels, reusing it in increasingly complex settings and in combination with other transformation blocks (see Excerpt Fin the Appendix for a transcription and code).

Composing-and-encapsulatingin code the quilts. Eric's approach to composing-and-encapsulating in Code the Quilts islinked with the example we used in the subsection "Discretizing in Code the Quilts" about his "tank" design, where he divided the desired image into four main components: the body, the turret, the gun, and the treads. He grouped the blocks of code corresponding to each part of the tank. In order to complete his design, Eric created an extra costume to represent a reusable component that represented the tank's treads. He used the two costumes for the different tank components. In doing so,Eric showed awareness of the construction of chunks of code blocks that hecould reuse in order to produce different components of the tank design. When Eric visualized the links between the motion of objects in the plane and their matching chunk of code inside a complete block sequence, he displayed a Vygotskian hybridizing of motion and mapping.

In summary, we can say that composing-and-encapsulating emerged as students interacted with "stacks" of code blocks in both environments. These kinds of interactions supported students in recognizing patterns, in understanding code sequences' total effects, and in combining and reusing structures of code blocks in different and increasingly specialized situations. As a result, these situations led the students to engage with big ideas in the mathematics of transformational geometry (e.g., Simon's invention and reuse of glide reflections). Therefore, composing-and-encapsulating supported them in a hybridizing of motion and mapping.

Summary:commonalities across the three interactions

Across the presented examples we found indicators of the affordances that block-basedprogramming environments can provide, to support Vygotskian hybridizing of motion and mapping conceptions about geometric transformations. In the analyzed episodes, a co-action space was created by the participants, in which they integrated their everyday concepts while using the computational abstractions embedded in the environments (code-blocks,parameters, and procedures). Learners reasoned about, planned, predicted the impact of, and executed geometric transformations in ways that leveraged their everyday concepts. In analyzing these episodes, we documented evidence of the emergence of three types of human/tool interactions: discretizing, parametrizing, and composing-and-encapsulating.

Discretizing linked the before/after effects of the code-blocks, which is a crucial interaction to consider when working with visual programming environments. Focusing on participants' discretizing highlighted the learning effects of coding and executing the code for geometric transformations. Focusing on participants' parametrizing showed the importanceofthe construction of unit measures when structuring a digital space. Through their parametrizing work, participants were able to navigate the plane and construct formal understandingsof transformations. Finally, through composing-and-encapsulating, participants created block sequences, appreciated the effects of multiple code-blocks taken together, and identified stacks of blocks that could function as procedures to perform complex compositions of transformations.

Discussion

In this paper, we have suggested that two block-based programming environments designed for geometric transformations provided a number of affordances for supporting learners in hybridizing understandings that were neither purely motion nor purely mapping conceptions. New hybrid forms of thinking emerged, which shared elements of both. To this end, based on our observations, these environments afforded scenarios for the creation of a *co-action*space for learning in which everyday and formal concepts met in ways that supported the hybridizing of motion and mapping. We showed how working in these environments led students to hybridize mathematical concepts through discretizing, parametrizing, and composing-and-encapsulating.

A co-action space for distributed interaction between student, the environment, and the task

Across the study, a co-action space (Moreno-Armella & Brady, 2018; Moreno-Armella & Hegedus, 2009) expanded learning opportunities, by enabling students to utilize the diversity of their embodied and personal experiences of motion to guide their actions in the two programming environments. Our analysis identified aspects of the environments' design that supported the growth of students' constructions toward ideas that integrated key aspects of a mapping view of geometric transformations. Three crucial actors contributed to the creation of this co-action space: the students, the block-based environments, and thespecific task designs created to position students asactive hybridizers of motion and mapping conceptions.

Students as active hybridizers

Considering the students as hybridizers means the students were primary actors in the construction of their understandings. Hybridity emerged as students used their intuitive, motion-based understandings to guide their interactions and sense-making within the block-based environments and on the tasks designed within the environment.

In the literature, motion and mapping conceptions have often been thought as binary entities (i.e., a student's conceptions align with *either* motion *or* mapping), demanding the progressive development of mathematical skills to allow students to move *from* motion *to* mapping (Edwards,2003, 2009; Hollebrands, 2007; Yanik, 2014). However, the analysis and results of this paper suggest that students put both conceptions in play in a hybrid manner, in which neither everyday concepts of motion nor more formal understanding of mapping took absolute precedence, but where elements of both were incorporated in a convergent trajectory (Vygotsky, 1986). These hybrid ways of conceptualizing meant that motion and mapping were dynamically active at the same time (e.g., Simon and Peter visualized continuous motions as they designed sequences of translations that were enacted in discrete steps), supporting Sinclair & Moss' (2012) image oflearners' retaining both everyday and formal conceptions in their more mature thinking.

Whether in a constrained activity or an open exploration, the challenge to move an object around the Cartesian plane using coding blocks involved complex interactions, in which students utilized elements of both mathematics and computer science to produce coordinated "turtle" movements (diSessa, 2000; Papert, 1980) and respond to the constraints and affordances of the challenges. In this co-action space, forms of hybridizing emerged, where elements of everyday and disciplinary understandings were incorporated, leading to a productive interplay between tool and human, incentivized by bridging of blocks and planar motions.

Block-based environments fostered Vygotskian hybridizing

The block-based coding environments pressed students to leverage their previous understandings of transformations to invent new ways of describing movement. Capturing these motion-descriptions in code opened up opportunities for their constructions to take on key attributes of a mapping view, as structured by the design of the activities and challenges. This is consistent with recent research on block-based programming environments (Andersen et al., 2022) where the verbal discourses evolve with the "dynamic entities" provided by these environments. For instance, in the students' use of the loop block, whether they possessed previous experience or not, the students quickly switched from simple repeated translations to groups of transformations parametrized by the number of repetitions. This led students to discover new units of actions given by the before/after states of the blocks to produce familiar sequences of discrete motion that they linked to their embodied experience (e.g., explaining a group of four 90° rotations as "cartwheeling around"). Following Gadanidis et al. (2018) ideas, this hybridizing led toward understanding of big mathematical ideas such as group theory foundational concepts (e.g., symmetry and transformations).

The student descriptions and executed plans illustrated an important aspect of this space: the understandings that can be built are neither the formal mathematics that mathematicians conceptualize (Edwards, 2009) nor the "everyday" embodied experience of motion. The students' ensemble of phrases, coding strategies, predictions, and code debugging created hybrid mathematical and computational concepts that gave new meaning to the transformation's performances.

Task design promoted students' meaningful explorations

By focusing on patterns in how the students interacted with the block-based environments, we observed that the task designs supported distributed, dialogic relations between the students' conceptions and the representationalsupports of block-based programming (Pierson et al., 2020). We recall Pea (1993), citing Simon (1981), in suggesting that solving a task or a problem may be looked at from a distributed perspective "between mind and the mediational structures the world offers." (Pea, 1993, p. 51). A fundamental aspect of this perspective was a change in students' understandings, reshaped through "dialectical reciprocal influences" (p. 57) given by the design of a productive task that mediated students' interactions while acknowledging their contextual contributions. This is a key feature of a co-action space.

It was not just the students exploring freely, but utilizing the affordances that designers built into the tasks to guide students in the exploration of the challenges in the environments. These design affordances supported learning about geometric transformations in meaningful and personalized ways, while keeping mathematical ideas intact. According to Gadanidis et al. (2018) the design of meaningful mathematical experiences considers agency and access as crucial elements for the development of big mathematical ideas. In this regard, our designs also supported students as active learners, positioned to make choices, to investigate, and to discover ideas. They offered a "low-floor, " providing mathematical engagement with minimal prerequisites of knowledge while at the same time giving opportunities for a "high ceiling" where students could explore more complex relationships (Papert, 1980).

Different block-based programming environments support students' Vygotskian hybridizing

Comparable processes of Vygotskian hybridizing of motion and mapping took place in two different block-based programming environments - one a game (process-oriented), and the other a construction environment (product-oriented).This suggests that building understandings of geometric transformations can be developed in a range of computational environments that share key features of turtle geometry and block-based programming. When determining key features of the environments that supported the creation of hybrid concepts, we identified how computational thinking abstractions (cf. Sengupta et al., 2018)- such as code blocks, parameters, and procedures - served in both environments to foster three conceptual forms of interactions - discretizing, parametrizing, and composing-and-encapsulating.These interactions allowed students to hybridize motion and mapping conceptions through practical activity in immediate interaction with the environments.

These three interactions may happen separately and in distinctive ways for different students, but the power for learning isenhanced when the interactions combine with and build upon each other. In different combinations, discretizing, parametrizing, and composing-and-encapsulating supported geometrically sophisticated constructions and sensemaking in which students combined everyday language and experiences with big ideas in the discipline. Their constructions used code blocks for domain-specific purposes (Sengupta et al., 2012, 2013; Wilkerson-Jerde et al., 2015) that leveraged geometric transformations concepts and language (e.g., [Rotate 90°CW] in TQ or [Turn] in Code the Quilts) to further personally meaningful activity. The combination of code blocks for domain specific purposes and the use of interactions in geometric transformations task designs meant that students could build on their deeply embedded everyday knowledge while engaging in the development of more formal articulations of geometric transformations understandings.

Conclusion

This study suggests a new perspective on students' reasoning about geometric transformations. Much research on geometric transformations has conceptualized learning as a unidirectional learning progression from motion to mapping. However, more recent work has problematized and enriched this picture. For example, Sinclair and Moss (2012) have talked about the oscillation everyday and formal conceptions and have indicated the importance of both for learners. This study extends this research direction further, highlighting that not only are students maintaining both motion and mapping as important resources, but they are also actually hybridizing them in new and powerful ways. Across two computational environments, the Vygotskian hybridizing of motion and mapping promoted by the distributed interaction between student, the environment, and task design led to a meaningful and agentive co-action space fostering the development of geometric transformations understandings. We identified three interactions that students engaged in (discretizing, parametrizing, and composing-and-encapsulating)which illuminated common design features that promoted hybridizing of motion and mapping, independent of whether the activity was process-oriented (the Transformations Quest game) or product-oriented (the Code the Quilts activities). This suggests the need for future research to explore designs that can foster similarly generative hybridizing of concepts.

Thus, this study contributes insights into students' ways of thinking about geometric transformations and how they can interact with affordances of a range of computational environments. These insights are essential for research that aims to support and study learners' integration of images of motion and of mapping in reasoning about transformations across multiple contexts. Further, this study reveals important synergies between emergent and hybridized transformational geometry reasoning on one hand and the kinds of computational thinking that can be fostered in visual programming environments on the other. These insights could providea foundation for coordinating students' short-term hybridizing of ideas across multiple environments and contexts to support students in developing longer-term understandings and learning of transformations grounded in a more durable hybridization of everyday and academic ideas of motion and mapping. A more durable hybridized understanding that integrates both motion and mapping should provide a stronger foundation for ongoing mathematical understanding and practice.

Although we found evidence through this small-scale study of a range of different patterns of student interaction suggesting that block programming environments could promote Vygotskian hybridizing of motion and mapping, we recognize that further research is needed in this area, given the limited size and scope of this study. As indicated earlier in the paper, exploring how students hybridize motion and mapping underscores the need for further study and perhaps a reconsideration of "the transition phases in the progress of geometrical concept formation" (Sinclair et al., 2016, p. 696). It would also be beneficial to compare the analysis here with student work in other computational environments, to explore how Vygotskian hybridizing may be occurring there. Furthermore, we propose that the descriptive examples provided here may be useful in the context of professional development for pre-service and in-service teachers, as it can serve to concretely typify the notion of Vygotskian hybridizing. Given the small sample size and restrictions related to the COVID-19 pandemic, however, the students' interactions that emerged from our study need to be explored across a broader student population, including a more diverse set of students, by gender and other core demographic considerations.

As this research evolves, and as we explore new computational and educational settings, we anticipate that we will encounter new and rich forms of student thinking that will lead us to modify and extend computational environments and create new task designs based on learning from our interactions with students and teachers. Our overarching goal is to create spaces in which computational thinking and transformational geometry offer the opportunity to bring students' everyday experiences into the mathematics classroom in powerful ways, democratizing access to big mathematical ideas both within and beyond the standard curriculum.

Note

1. Note: We use [) and bold font to indicate a block procedure belonging to the computational environments in the text. We use only bold font in block procedures in a transcription.

Disclosure statement

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

Funding

This material is based upon work supported by the National Science Foundation under Grant No. 1742257. Any opinions, findings, and conclusions or recommendationsexpressed in thismaterial are thoseof the author(s) and do not necessarily reflect the views of the National Science Founda tion. We are grateful to Natalie Major, teacher from the Canadian Rockies Public Schools for her thoughtful comments and collaboration to interview some participants for this study.

Notes on contributors

*Jesus E. Henuindez-Zavaleta*is a Postdoctoral Scholar at the University of Florida's College of Education focusing on Equitable and Interactive Mathematical Modeling. His previous appointment at the Werklund School of Education in the University of Calgary focused on the development of mathematical and computational thinking, with particular emphasis on K-12 students. He has also been a classroom teacher at the high school level and a mathemat ics teacher trainer in Mexico and South America.

Corey Brady is an Assistant Professor of the Learning Sciences at Vanderbilt University. His research focuses on mathematical and computational modeling, with a particular emphasis on supporting and understanding the collective learning of classroom groups. Prior to entering academia he worked in industry, leading the design and development of Texas Instruments' TI-Navigator system (2001-2006). He has also been a classroom teacher, at middle school, high school, and community college levels. Brady holds degrees in Pure Mathematics (MS), English Literature (MA), and Mathematics Education (PhD).

Sandra Becker is Postdoctoral Scholar at the University of Calgary's Werklund School of Education. Her recently completed PhD research focuses on the affordances of elementary school makerspace environments as teacher professional learning studios when designing for interdisciplinary curricular study. Prior to entering the PhD program, Sandra taught elementary school, both as a classroom teacher and teacher librarian.

Doug Clark is the Werklund Research Chair of Design-Based Learning at the University of Calgary's Werklund School of Education. His research focuses on the design of digital learning environments, students' learning processes, and the opportunities of design for teachers and students. Doug has been awarded multiple funded grants from the US National Science Foundation, Department of Education, and the US National Academy of Education/Spencer Foundation.

ORCID

Jesus E. Hernandez-Zavaleta http:/lorc id.org/0000 -0003-2937-1932
Corey Brady http://orcid.org/0000-0002-4086-9638
Sandra Becker http://orcid.org/0000-0002-0478-045X
Douglas B. Clark http:/lorcid.org/0000-0002-4757-8446

Ethical approval

Ethics Certificate number REB18- 0304 covered the experimental interviews reported in this research work, issued by the Conjoint Faculties Research Ethics Board (CFREB) at the University of Calgary for the "Students as mathematical and scientific thinkers and modelers" project on April 23, 2018, with expiry date April 23, 2023.

Reference s

Alberta Education. (2016). *The guiding framework for the design and development of kindergarten to grade 12 provincial curriculum (programsof study) (the guiding framework).* Alberta Education Student Learning Standards Division.
Andersen, R., M0rch, A., & Torine Litherland, K. (2022). Collaborative learn ing with block-based programming : Investigating hum an-centered artificial intelligence in education. *Behaviour & Information Technology, 41(9),* 1830- 1847. https://doi.org/10.l080/0144929X.2022.2083981
Arzarello, F., Olivero, F., Paola, D., Robutti,0 . (2002). A cognitive analysis of dragging practices in Cabri environments. *ZDM,* 34(2), 66- 72.
Avcu, ,S. & <;:etinkaya, B. (2019). An ins t ruct ional unit for prospective teachers' conceptualization of geometric transformations as functions. *International Journal of Mathematical Education in Science and Technology,* 1- 30. https://doi.org/ 1 0.1080/0020739X.2019.1699966
Bartolini Bussi, M., & Mariotti, M. A. (2008). Semiotic mediation in the mathematics classroom: Artifacts and signs after a Vygotskian perspective. In L.English, M. Bartolini Bussi, G. Jones, R. Lesh, B. Sriraman,& D. Tirosh (Eds.), *Handbook of international research in mathematics education.* Routledge. https://doi.org/10.4324/9780203930236.ch28
Borba, M., & Villareal, M. (2005). *Humans -with-media and the reorganization of mathematical thinking.* Springer.
Clark, D. B., Hernandez-Zavaleta, J. E., & Becker, S. (2023). Academically meaningful play: Designing digital games for the classroom to support meaningful gameplay, meaningful learning, and meaningful access. *Computers & Education, 194,* 104704. https://doi.org/10.1016/j.compedu.2022.104704
Clark, D. B., Hernandez-Zavaleta, J. E., Rothschuh, S., Becker, S., Mattingly, P., Ostrowdun, C., Chu, M., Sandhu, K., Vryenhoek, J., Cao, L., & Plosz, J. (2021). *Designing games for geometric transformations in the classroom: Mathematically meaningful play.* Paper presented at the Canadian Society for the Study of Education (CSSE 2021), Edmonton, AB/Online.
Clark, D. B., & Martinez-Garza, M. (2012). Prediction and explanation as design mechanics in conceptually-integrated digital games to help players articulate the tacit understandings they build through gameplay. In C. Steinkuhler, K. Squire, & S. Barab (Eds.), *Games, learning, and society: Learning and meaning in the digital age* (pp. 279- 305). Cambridge University Press. https://doi.org/10.l017/CB09781139031127.023
Clark, D. B., & Sengupta, P. (2020). Reconceptualizing games for integrating computational thinking and science as practice: Collaborative agent -based disciplinarily-integrated games. *Interactive Learning Environments,* 28(3), 328- 346. https:/Idoi.org/l 0.1080/ 1 0494820.2019.1636071
Clark, D. B., Sengupta, P., Brady, C., Martinez-Garza, M., & Killingsworth, S. (2015). Disciplinary integration in digital games for science learning. *International STEM Education Journa, l* 2(2), 1- 21. https://doi.org/10.1186/s40594-014-0014-4
DeJarnette, A., Gonzalez, G., Deal, J. T., & Rosado Lausell, S. L. (2016). Students' conceptions of reflection: Opportunities for making connections with perpendicular bisector. *The Journal of Mathematical Behavior,* 35- 52. https://doi.org/ 1 0.1016/j.jmathb.2016.05.001
diSessa, A. (2000). *Changing minds: Compute rs, learning, and literacy.* The MIT Press. https://doi.org/10.7551/mitpress/1786.001.0001
Edwards, L. (2003). The nature of mathematics as viewed from cognitive science. Third Conference of European Research in Mathematics Education, Bellaria, Italy, February (pp. 1- 10). Retrieved 03 20, 2023, https: //www.researchgate.net / publi cation /2385577 55_T HE_ NATU RE_O F_ MATHEMATICS_AS_VIEWED _FROM_COGNJTI VE_SCIENCE
Edwards, L. (2009). Transforma tion geometry from an embodied perspective. In W. M. Roth (Ed.), *Mathematical representation at the inte,face of body and culture* (pp. 27- 44). Information Age Publishing Inc.
Edwards, L., & Zazkis, R. (1993). Transformation geometry: Naive ideas and formal embodiments. *Journal of Computers in Mathematics and Science Teaching,* 12(2), 121- 145.

Faggiano, E., Montone, A., & Mariotti, M. A. (2018). Synergy between manipula tive and digital artefacts: A teaching experiment on axial symmetry at primary school. *International Journal of Mathematical Education in Science and Technology,* 49(8), 1165- 1180. https://doi.org/10.1080/0020739X.2018.1449908

Fan, L., Qi, C., Liu, X., Wang, Y., & Lin, M . (2017). Does a transformation approach improve students' ability in constructing auxiliary lines for solving geometric problems?: An intervention -based study with two Chinese classrooms. *Educational Studies in Mathematics,* 96(2), 229- 248. https://doi.org/10.1007/sl0649-017-9772-5

Fife, J. H., James, K.,& Bauer, M. (2019). *A learning progression forgeometric transformations (research report No. RR-19-01).* Educational Testing Service. https://doi.org/10.1002/ets2.12236

Gadanidis, G., Clements, E., & Yiu, C. (2018). Group theory, computational thinking, and young mathematicians. *Mathematical Thinking and Learning,* 20(1), 32- 53. https://doi.org/10.1080/10986065.2018.1403542

Gaikwad, P. (2017). Including rigor and artistry in case study as a strategic qualitative methodology. *Qualitative Report,* 22(13), 3431. https://doi.org/10.46743/2160-3715/2017.3436

Gleasman, C., & Kim, C. (2020). Pre-service teacher's use of block-based programming and computational thinking to teach elementary mathemat ics. *Digital Experiences in Mathematics Education,* 6(2020), 52- 90. https://doi.or g/10.1007/s40751-019-00056-1

Harrison, H., Birks, M., Franklin, R., & Mills, J. (2017, January). Case study research: Foundations and methodological orientations. *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research, 18(1),*1- 17.

Hernandez-Zavaleta, J. E., Becker, S., Clark, D., Corey, B., & Cao, L. (in preparation). *Learning about symmetries and transformations in 2Dgeometry: A review of theoretical threads and themes.*

Hollebrands, K. (2003). High school students' understandings of geometric transformations in the context of a technologicalenvironment. *The Journal of Mathematical Behavior,* 22(1), 55- 72. https://doi.org/10.1016/S0732-3123(03)00004-X

Hollebrands, K. (2007). The role of a dynamic software program for geometry in the strategies high school mathematics students employ. *Journal for Research in Mathematics Education,* 38(2), 164- 192. https://www.jstor.org/stable/30034955

Hoodama th. (n.d.). https: //www.hoodamath.com/games/transfo rmationgolf2.html

Hoyles, C., & Healy, L. (1997). Unfolding meanings for reflective symmetry. *International Journal of Computers for Mathematical Learning, 2 ,* 27- 59. https://doi.org/10.1023/a:1009721414546

Hoyles, C., & Noss, R. (2020, June 19). Mapping a way forward for comput ing and mathematics: Reflections on the UCL ScratchMaths Project. In C. Bateau & G. GadanidisEds., *Online seminar series on programming in mathematics education.* Mathematics Knowledge Network . http://mkn -rcm.ca/online-seminar -series-onp rogramm ing-in -mathematics-education/

Jaco bson, C., & Lehrer, R. (2000). Teacher appropr iation and student learning of geometry through design. *Journal for Research in Mathematics Education,* 31(1), 71- 88. https://do i.org/10.2307/749820

Jiang, Z., Dickey, E., White, A., McBroom, E., Sorto, M. A., & Rosenwasser, A. (2016). A dynamic geometry-centered teacher professional development program and its impact. In E. Dickey, Z. Jiang, A. White, & B. Webre (Eds.), *Findings of the dynamic geometry in the classroom project at texas state university* (pp. 39-46). Texas State University, NFS.

Kafai, Y. B. (1996). Learning design by making games: Children's development of strategies in the creation of a complex computational artifact. In Y. B. Kafai & M. ResnickEds., *Constructionism in Practice: Designing, Thinking and Learning in a Digital World.* (pp.71- 96). https://doi.org/10.4324/9780203053492

Laborde, C. (1993). The computer as part of the learning environment: The case of geometry. In C. Keitel & K. Ruthven (Eds.), *Learning from computers: Mathematics education and technology. NATO ASI series* (Vol. 121, pp. 48- 67). Springer-Verlag.

Lehrer, R., Jacobson, C., Thoyre, G., Kemeny, V., Strom, D., Horvath, J., Gance, S., & Koehl,er M. (1998). Developing understand ing of geometry and space in primary grades. In R. Lehrer & D. Chazan (Eds.), *Designing learning environments for developing understanding of geometry and space.* Lawrence Erlbaum Associates. https://doi.org/ 10.4324/9780203053461-13

Lehrer, R., & Romberg, T. (1998). Springboards to geometry. In C. Mammana & V. Villani (Eds.), *An ICM[study* (pp. 62- 71). Springer Science+ Business Media, B. V. https://doi.org/10.1007/978-94-0ll -5226-6

Leron, U., & Zazkis, R. (1992). Of geometry, turtles, and groups. In C. Hoyles & R. Noss (Eds.), *Learning mathematics and logo* (p p. 319- 352). The MIT Press. https://doi.org/10.7551/m itpress/4171.003.0022

Mangahigh. (n.d.). Transtar. https://www.mangahigh.com/en-us/games/transtar

Merriam, S. B., & Tisdell, E. J. (2015). *Qualitative research: A guide to design and implementation.* John Wiley & Sons.

Mhlolo, K. M., & Schafer, M. (2014). Potential gaps during the transition from the embodied through symbolic to formal worldsofreflectivesymmetry. *AfricanJournalofResearchin Mathematics, Science and Technology Education,* 18(2), 125- 138. https://doi.org/10.1080/10288457.2014.925269

Moreno-Armella,L., & Brady, C. (2018). Technolog ical supports for mathematical thinking and learning: Co-action and designing to democratize access to powerful ideas. In L. Ball, P. Drijvers, S. Lade!, H. S. Siller, M. Tabach, & C. Vale (Eds.), *Uses of technology in primary and secondary mathematics education* (pp. 339- 350). Springer. https://doi.org/10.1007/978-3-319-76575-4_19.

Moreno-Armella, L., & Hegedus, S. J. (2009). Co - action with digital technologies. *ZDM Mathematics Education, 41,* 505- 519. https://doi.org/10.1007/sl 1858-009-0200-x

Moyer, J. C. (1978). The relationship between the mathematical structure of Euclidean transformations and the spontaneously developed cognitive structures of young children. *Journal for Research in Mathematics Education,* 9 (2), 83-92. https://doi.org/10.2307/748871

National Council of Teachers of Mathematics. (2000). *Principles and standards for school mathematics.* The National Council of Teachers of Mathematics.

National Council of Teachers of Mathematics. (n.d.). Flip-N-Slide. https:// www.youtube.com/watch?v=KJrmywqKAEs

Ng,0 . L., & Sinclair, N. (2015). Young children reasoning about symmetry in a dynamic geometry environment. *ZDM Mathematics Education,* 47(3), 421- 434. https:// doi.org/10.1007/s11858-0140660-5

Panorkou, N., & Maloney, A. (2015). Elementary studen ts' construction of geometric transformation reasoning in a dynamic animation environment. *Constructivist Foundations,* 10(3), 338- 347.

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

Pea, R D. (1993). Practices of distributed intelligence and designs for education. In G. Salomon (Ed.), *Distributed cognitions: Psychological and educational considerations* (pp. 47- 87). Cam bridge University Press.

Pierson, A. E., Brady, C., & Clark, D. (2020). Balancing the environment : Compu tational models as interactive participants in a STEM classroom. *Journal of Science Education and Technology,* 29(1), 101- 119. https://doi .org/10.1007/s10956-019-09797-5

Saldana, J. (2016). *The coding manual for qualitative researchers* SAGE.

Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a phenomenology of computational thinking in STEM education . In M. Khine (Ed.), *Computational thinking in the STEM disciplines* (pp. 49- 72). Springer. https:// doi.org/10.1007/978-3-319-93566-94

Sengupta, P., Farris, A. V., &Wright, M. (2012). From agents to continuous change via aesthetics: Learning mechanics with visual agent-based computational modeling. *Technology, Knowledge and Learning,* 17(1), 23-42. https://doi.org/10.1007/s1075-8012-9190-9

Sengupta, P., Kinnebrew,,J. Basu, S., Gautam, B., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-basedcomputation: A theoretical framework. *Education and Information Technologies,* 18(2), 351- 380. https://doi.org/10.1007/s10639-012-9240-x

Simon, H. (1981). The *science of the artificial* (2nd ed.). MIT Press.

Simons, H. (2009). *Case study: Research in practice.* (https://doi.org/10.4135/9781446268322). SAGE Publications.

Sinclair, N., Bartolini Bussi, M., de Villiers, M., Jones, K., Kortenkamp, U., Leung, A. & Owens, K. (2016). Recent research on geometry education: An ICME-13 survey team report. *ZDM Mathematics Education,* 48(5), 691- 719. https://doi.org/10.1007/sl 1858-016-0796-6

Sinclair, N., & Moss, J. (2012). The more it changes, the more it becomes the same: The development of the routine of shape identificationin dynamic geometry environment. *International Journal of Educational Research, 51-52,* 28-44. https://doi.org/10.1016/j.ijer.2011.12.009

Son, J. W., & Sincl air, N.(1986). How preservice teachers interpret and respond to student geometric errors. *School Science and Mathematics,* 110(1), 31- 46. https://doi.org/10.llll /j.1949-8594.2009.00005..x

Tall, D. (1986). Visualising calculus concepts using a computer. In R. F. Churchhouse, B. Cornu, A. G. Howson, J. -P. Kahane, J. H. van Lint, F. Pluvinage, A. Ralston, M. Yamaguti (Eds.), *The Influence of Computers and Informatics on Mathematics and its Teaching: Proceedings From a Symposium Held in Strasbourg, France in March 1985 and Sponsored by the International Commission on Mathematical Instruction* (pp. 303- 308). Strasbourg, France, Institut de Recherches sur l'Enseignement des Mathematiques .

Thomas, G. (2011). A typology for the case study in social science following a review of definition, discourse, and structure. *Qualitative Inquiry, 17(6),* 511- 521. https://doi.org/10.1177/1077800411409884

Usiskin, Z. P. (1972). The effects of teaching Euclidean geometry via transforma tions on student achievement and attitudes in tenth -grade geometry. *Jou rnal for Research in Mathematics, Education* (4), 249-259. https://doi.org/10.2307/748492

Van Wynsberghe, R, & Khan, S. (2007). Redefining case study. *International Journal of Qualitative Methods,* 6(2), 80- 94. https://doi.org/10.1177/160940690700600208

Vygotsky, L. S. (1986). Thought and language. In A. Kozulin (Ed.), *revised edition,* The MIT Press.

Weintrop, D. (2019, August). Block-based programming in computer science education. *Communicationsof the ACM,* 62(8), 22- 25. https://doi.org/10.1145/3341221

Wilensky, U., & Papert, S. (2010). Restructura tions: Reformulations of knowledge disciplines through new representa-tional forms. *Constructionism,* Paris, August (pp. 1- 15). Retrieved 03 20, 2023, https://ccl.northwestern.edu/2010/wilensky_restr ucturations_Constructionism%202010-latest.pdf

Wilkerson-Jerde,M. H., Wagh, A., & Wilensky, U. (2015). Balancing curricular and pedagogical needs in computational construction kits: Lessons from the DeltaTick project. *Science Education,* 99(3), 465-499. https://doi.org/10.1002/sce.21157

Yanik, H. B. (2011). Prospective middle school mathematics teachers' preconceptions of geometric translations. *Educational Studies in Mathematics, 78,* 231- 260. https://doi.org/10.1007/s10649-011-9324-3

Yanik, H. B. (2014). Middle-school students' concept images of geometric translations. *The Journal of Mathematical Behavior, 36,* 33- 50. htt ps://doi.org/10.1016/j.jmath b.2014.08.001
Yanik, H.B., & Flores, A. (2009). Understanding rigid geometric transformations: Jeff's learning path for translation. *The Journal of Mathematical Behavior,* 28(1), 41- 57. https://do i.org/ 10.1016/j.jmathb.2009.04.003

Appendix

Text between the brackets [] explains actions taken by the students.

Excerpt A. Eric' s Collecting Yellow and Blue Gems in Transformations Quest Level 6

Collecting the Yellow Gems

Eric: well, since I'll be rotating around the center, clockwise is going, spinning, so that this block will end up here, I'llshow you. [Eric executed the code block Rotate CW 90° to demonstrate how to collect the left yellow gem.)

Eric: so, then I'll go clockwise three and collect those ones.

[As he explained his plan, Eric moved his mouse cursor in a clockwise circular direction over the yellow gems to indicate the movement of the triangle and the location of the three yellow gems he would collect.]

[Im mediately after this result, without hesitation, Eric changed the number of repetitions to five and re-ran the sequence.]

Researcher 1: why did you use five?

Eric: so that I would end up back at the orientation, um, so that I would be able to fit in through the doorway.

Collecting the Blue Gems

Researcher 1: Could you explain what happens with the triangle rotation when you run the sequence?

Eric: Rotating is when you've got a certain point you're rotating around. In this case, it's the very middle of this chart. And it's like you're drawing a circle around it and your triangle would be the pencil.

[As he explained "drawing a circle" he used his cursor to demonstrate a clockwise circular motion passing the pointer over the four blue gems around the origin.]

Eric: And it ends up here, it's going to be facing a certain way.

[He pointed out the top blue gem over the positive y axis.]

Eric: And then over here, it's gonna flip again. It's kind of like it's cartwheeling around .

[He pointed out the right blue gem over the positive x axis.]

Researcher 1: Could you explain why the triangle changes its direction?

Excerpt B. Zach' s Extended Pattern in Code the Quilts

Zach: I have one block, ... two blocks facing the right way.

Researcher 2: You have four built squares there. Could you build on that with other four block squares?

Zach: I'm pretty sure I can, cause if I just stamp this again ... and I get a **Repeat** block.

[He moved his mouse over the code sequence from the **Stamp** blockat the top to the button at the end of it.)

Zach: Except for this time, I'm going to move it to a different place, then it's going to do it again.

[He pointed out to the right side of *the y* axis in the plane.)

[He used his mouseto indicatethe quilt at the bottom right position in Figure 1, Zach' s Adding a Stamp Block Column.]

[He used his mouse to indicate the positions that the quilts had in the left side of *the y* axis then executed the program.)

[He pointed block byblock from top to bottom with his mou,se while referring to previous explanations of the use of sequences of blocks in the activity.]

Zach: First it's going to take a loop from that point

Zach: Then it's going to go do that pattern. Then it's going to go to the right side of the axis. Then it's going to do the same thing again. So, the right side and the left side are reflected off they axis. So, it's like an extended pattern.

Researcher **1:** Zach, could you tell us what did you do there?

Zach: First, I put this **Go to** code block outside the repeat two variable, repeating loop, because after this is done, I go to here. [He pointed to the top of the repeat block.]

Zach: And then switches its costume, and then, point in direction, I've explained this before.

Zach: So, it makes the square, except for this time it goes to x 67. But when it started, it says **Go to** -67. So, it starts making the square at -67 x, except for this is outside the **Repeat** so once this happens you can't really use this anymore. So then at the end of this **Repeat,** I have **Go to** 67 which is two blocks away and it goes by two blocks wide, so **it** works, so this time it's starting at 67 instead and then repeats this again.

Excerpt C. Peter's Solving Transformations Quest Level 6

Researcher I: What's your plan Peter?

Peter: I think I going to move to this one

[He pointed his cursor to the left blue gem]

Peter: And then back to this one

[He pointed out his cursor to the right blue gem]

Peter: Then go up here

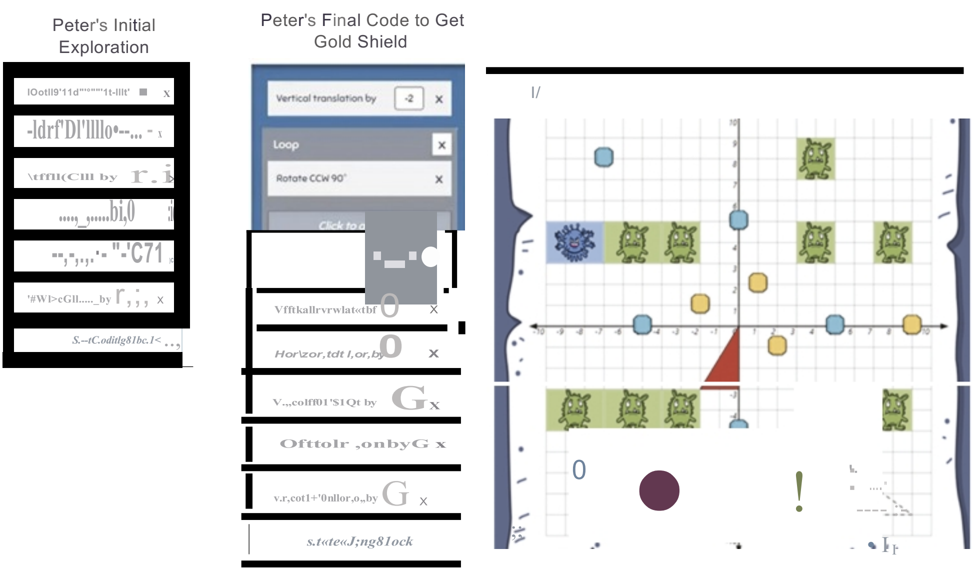
[He pointed his cursor to the blue gem at the top-left]

Peter: Then down to this one

[He pointed his cursor to the blue gem at the bottom left]

Figure Excerpt C

Peter's Code for Initial Exploration and Final Code to Solve Transformations Quest Level 6 and Level 6 Configuration



Excerpt D. Eric's Tank Con s truction in Code the Quilts

Eric: I'm going to keep this fairly simple.

[He entered - 134 in the glide block]

Eric: And then I'm going to duplicate.

[He grabbed a string of code, duplicated, it and placed it at the end of the sequence].

Eric: So, I'm going to keep this one below it so I'm going to keep it at negative 134

[He referred to x]

Eric: but instead, I'm going to keep it at negative 67

[He changed y from O to - 67]

[Eric used the glide block again but changed the parameters. He entered x - 67, y 67 to build the turret]

Eric: I am going to use the glide block again keeping it at x -67 and y 67 to build a turret at the top

Eric: For the gun, I'm going to try and put it in between them, so I'm going to make it a positive, the tanks going to be facing to the right, so let's make it a positive, hmmm, 134 and then we'll make it half of 67 so it willbe, we'll try 33. And this one'sgonna be ahead of that [to the right] so ... we'll add 67 ... hmmm.

Researcher 2: Well, you've got your piece [gun) set up, it's just placing it on the tank.

[He executed the code]

Eric: It's going too far on x. This is going to be 134

[He inputted y 134 and moved to change they parameter in the second quilt square ...]

Eric: Okay, I see. This is going to be 67

[He inputted x 67 in the code for the first quilt square]

Eric: and this is going to be 134

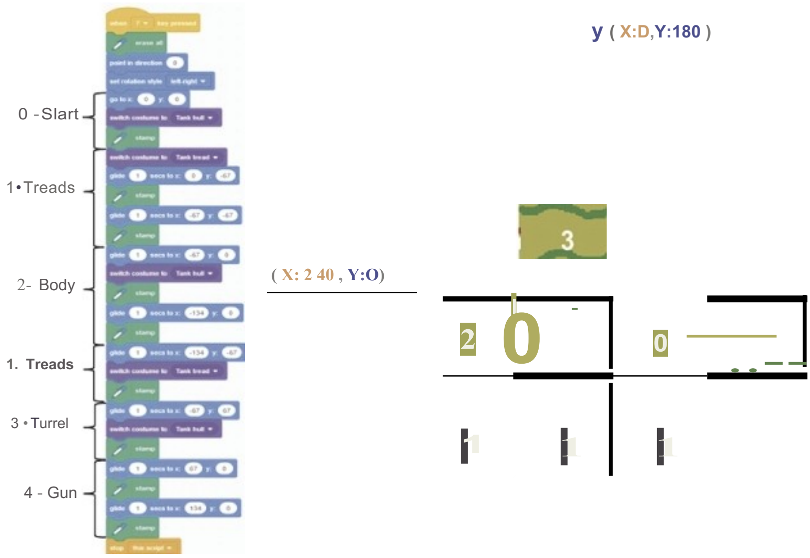
[He inputted x 134 in the code for the second quilt square].

Eric: and this is going to be zero and zero

[He inputted y O for both quilt squares].

Figure Excerpt D

Eric's Scratch Code and Tank.



Note: We include numbers to identify the four parts Erick used to build the tank.

Excerpt E. Simon's Glide Reflections in Transformation Quest Level 4

Researcher 2: I noticeyou were moving your mousearound there. Can you tell us what you were doing with your mouse?

Simon: Well, my brain is, I don't know why I keep mentioning my brain, I mean, I'm thinking that I can go here, here, here, here, and here

[He indicated with the mouse a repeating pattern of reflection over y, then translation down]

Simon: and then there to get all of the gems, because it's going to be a lot of reflection or jumping. It will probably work, that's my guess. Oh, wait, there isn't a . . . oh I see, it doesn't want you to reflect x.

[He was lookin g to the block section]

[He was moving his mouse from the top to the button. After he constructed a solution program].

Simon: Very, very, very smart.

Researcher **1:** So, Simon before you go any further, how do you understand reflection about they axis?

Simon: Well, if it was about the x axis, you couldn't collect all the gems that you need to get the shield.

Researcher 1: Let's go to the block sequence you have there. Could you find some kind of pattern?

Simon: The pattern I found is vertical translation minus five, well, I could probably take away these here

[He removed the blocks that repeat the pattern],

Simon: and just make it simpler, not having to put in as many coding blocks

[He entered a 3 in the repeat block].

Researcher 1: What is going to happen? You changed a lot of things.

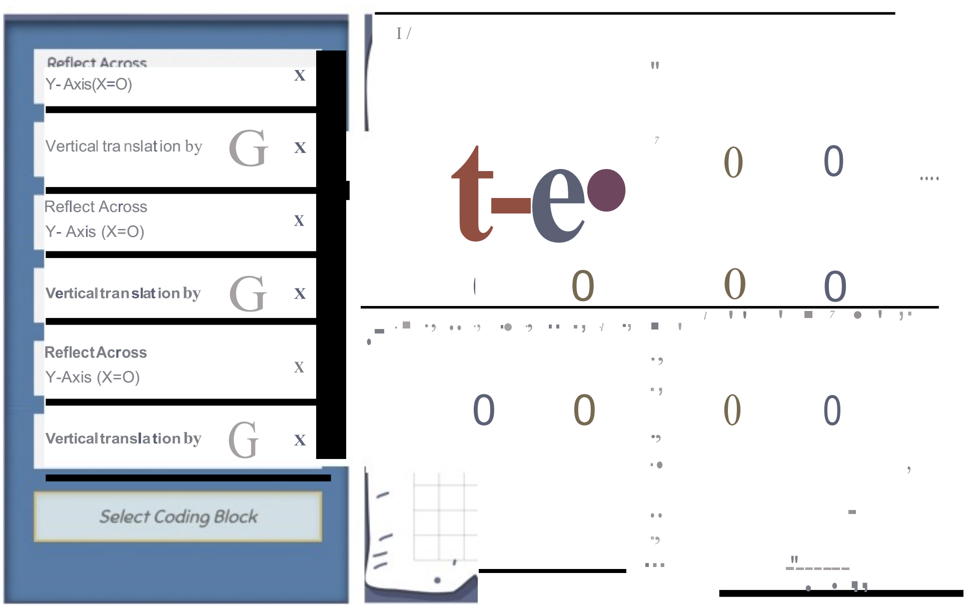
Simon: It's going to repeat the sequence that I have here by three. I'm expecting it to do the same exact thing that the other one did with less steps.

[He referred to his last run sequence]

Simon: Well, it has the same amount of steps, with less coding blocks in the loop block.

Figure Excerpt E

Simon's Solution to Solve the Level 4 and Level 4 Configuration



Excerpt F. Peter ' s Solving Transformations Quest Level 7

Researcher I: what do you think is a good strategy, Peter?

Peter: mmm ... moving along rotating

[He pointed to the blue gems at the top left],

Peter: then rotating again

[He pointed to the blue gems at the bottom left].

[He constructed a block sequence including a loop block with [Rotate CCW 90°] in combination with [Vertical translation by 2] and he choose to repeat 3 times]

Peter: Then move up.

Researcher I: why did you choose three times?

Peter: mmm . . . to get to here, that's one

[He pointed to the blue gems at the top left again],

Peter: to get to here, that's two

[He pointed to the blue gems at the bottom left].

Peter: Then to get these two, that's three

[He pointed to the blue gems at the right, then he ran the sequence twice].

Researcher I: what do you think is missing?

Peter: It's to get to here

[He pointed to the exit]

Researcher I: what kind of instructions do you need to get that?

Peter: I need to rotate it. I think just one rotation will be enough, because the coordinate over this corner is 6 by 2

[He pointed out the right angle on the red triangle]

Peter: and this is 6 by 2

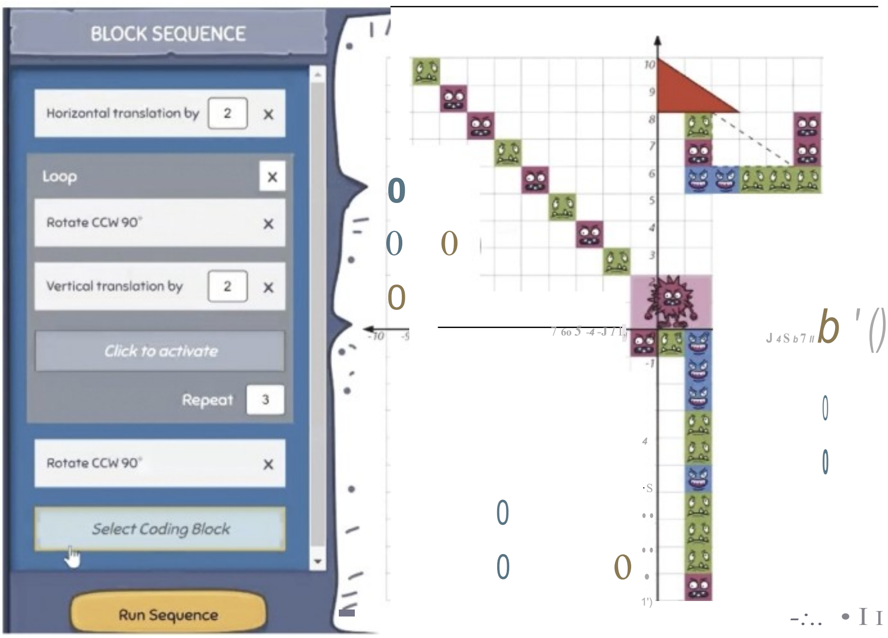
[He pointed out the coordinate (6, - 2)].

Peter: It is the same spot, just in differen t quadr ants. So, if you rotate it ends up there

[He pointed out the exit, then Simon added an extra [Rotate CCW 90°] ou tside the loop and ran the sequence]

Figure Excerpt F

Peter's Final Code to Solve the TransformationsQuest Level 7 and Level 7 Configuration



Excerpt G. Simon Solving Transformations Quest Level 3

After Simon constructed a sequence including [Vertical tra nslatio n by 4] [Reflect xl[Horizont al translation by 5] [Reflect x]

Simon: Then I have to go here

[He pointed to the blue gems at the bottom left].

Simon: Then I am here right now

[He pointed to the blue gem at the top left].

Simon: I haveto go horizontal by 5 again and then I haveto go another x reflect. Ma n, I really like these things. Then, I am just going to run the sequence right now. Then I have to do a t ranslatio n here by one, two . . .

[He counted from the last position of the red triangle].

Simon: Yes, I am so close. Then I have to do a horizontal translation by 5.

Simon: I think I can just times it by three or something.

Researcher **1:** Do you think is possible to find some kind of pattern?

Simon: Yeah, it's reflect five, reflect five, reflect five. I know how to finish it all. By vertical translation by4

[He inserted a vertical translation block]

Figure Excerpt G

Simon's Solution to Solve the Level 3 and Level 3 Configuration