Chapter 4
Geometry for Computer Graphics in K-12 Education

Armando Paulino Preciado Babb
University of Calgary, Canada

ABSTRACT

Computing education and computational thinking have gained increasing attention in education both as means to support the learning of other subjects, such as mathematics, science, and humanities, and as outcomes by themselves. This chapter proposes a focus on teacher knowledge of geometry for computer graphics used for virtual image manipulation and coding in the context of an online graduate course for teachers. Teachers were required to design tasks for their classrooms that incorporated the content of the course and to participate in an online discussion forum. These tasks, along with the discussion entries, are analyzed, and suggestions are provided for how to incorporate relevant geometrical content used in computer graphics. Teacher challenges to learn and incorporate this content in the classroom are addressed, along with recommendations for teacher education. The findings of this study are discussed in terms of teachers' knowledge for teaching geometry for computer graphics.

INTRODUCTION

Programs of studies in mathematics at K to 12 levels around the world are slowly introducing new topics such as cryptography and network (or graph) theory to their curriculum. However, teachers are not likely to be familiar with these new topics. Addressing this issue, the University of Calgary has developed a graduate certificate for teachers to explore mathematics beyond what is traditionally included in such programs of studies, but which can still be taught at school level. One of these courses, *Geometry in Art, Nature and Computer Graphics*, specifically addresses mathematical content for virtual image manipulation. This content is relevant for teachers to incorporate computer education in their classrooms.

The content of this course has not yet been commonly addressed as part of computer education. For instance, Hubbard (2018), in a review of the literature, noted that scholars included specific topics as part of teachers' required knowledge for computing education, namely: arrays, control structures, data structure, decomposition, direct and indirect referencing, formal language grammar and syntax, functions,

DOI: 10.4018/978-1-7998-1479-5.ch004

generalization, input and output, logical thinking, parameters, problem-solving skills, procedures, reusability, thinking in modules, user interface and variables, algorithms, bubble sort, metaphor, programming, recursion, and unified modelling language. While some of these topics, such as arrays, are very specific, others, such as problem-solving skills, are more general, with specific skills varying depending on the subject. In contrast, the content of the course described in this chapter comprises specific geometrical topics such as proportion, trigonometry, vectors, fractals, transformations, tessellations, and symmetry in the contexts of their use for image manipulation and animation.

This course also takes a different approach to what has been the predominant focus in the literature on computer education and computational thinking; namely a focus on teaching either these independently, or in support to other subjects. For instance, Weintrop and colleagues (2016) draw from the literature, from interviews with mathematicians and scientists, and from instructional materials to propose a framework for integrating computational thinking for mathematics and science. The framework consists of four categories: data practices, modeling practices, computational problem-solving practices, and system-thinking practices. While some elements that might be related to computer graphics are mentioned, such as computer simulations and graphical interfaces, the specific mathematical knowledge is not addressed. The main approach of the framework is to incorporate computational thinking in mathematics and science instruction, which contrast with the approach of this chapter that focuses on mathematical knowledge that can support computer education.

The purpose of this chapter is to investigate how geometry for computer sciences can be introduced in K to 12 education through an analysis of teachers' task designs for their classrooms and the online conversations that were part of the graduate course. Teachers' challenges to learn and incorporate the content of the course in their classroom offer insights relevant for both the implementation of geometry for computer graphics at school levels and teacher education in this area. The chapter also discusses the specialized knowledge required for teaching geometry for computer sciences.

BACKGROUND

Geometry for computer graphics can be loosely defined as the relationships between geometry and programing, in particular with respect to the manipulation of virtual images. This also includes programing for robotics, with involves spatial elements. Knowledge for teaching geometry for computer graphics deserves particular attention, as offered in the next subsection.

Knowledge for Teaching Geometry for Computer Graphics

This study, conducted in the context of mathematics teacher education, is intrinsically related to extensive scholarly work on mathematics knowledge for teachers that dates back more than four decades. Such work continues to be a prominent focus in the literature. Early results of this work repeatedly showed little or no correlation between teachers who possess extensive teachers' college credits in mathematics and the performance of their students on standardized tests (Begle, 1972, 1979; Monk, 1994), prompting the realization that the knowledge required for teaching mathematics should be specialized and go beyond mere mathematical knowledge.

Shulman's (1986) seminal work articulated this point distinguishing between content knowledge and pedagogical content knowledge. While the former refers to the specific knowledge of a subject or of professions such as mathematics or computer sciences, the later concerns specialized knowledge for teaching, including useful forms or representations of concepts or ideas as well as powerful analogies, illustrations, and examples that can help to make the content accessible to others. Other elements of this knowledge include common misconceptions and challenges related to learning particular topics.

Since then, the idea of specialized knowledge for teaching mathematics has been reinforced by growing evidence, as summarized by Baumert and colleagues (2010) in a comprehensive review of empirical research. They noted that: "Findings show that [teachers' content knowledge of mathematics] remains inert in the classroom unless accompanied by a rich repertoire of mathematical knowledge and skills relating directly to the curriculum, instruction, and student learning" (p. 139). More fulsome accounts of this research are presented elsewhere (e.g., Ball, Lubienski, & Mewborn, 2001; Blömeke & Delaney, 2012; Hill, Rowan, & Ball, 2005).

A more recent development among researchers has been to attend to the necessarily situated nature of teachers' disciplinary knowledge. Ball, Thames, & Phelps (2008), for example, noted that teachers' mathematical knowledge is not static and argued it should be thought of as knowledge-in-action. More recently, Blömeke, Gustafsson, and Shavelson (2015) conceptualized *competence* as a continuum that includes teachers' dispositions, situated-specific skills, and performance in class. This pragmatic perspective on knowledge for teaching is relevant to this study that looks into teachers' task designs and conversations about the implementation of geometrical knowledge related to computer graphics in K to 12 education.

Manipulating computer graphics not only involves geometry, but also its connections to human perception of the world. For the computer, all the geometric objects are data. The computer is programed to process these data to render pixel-generated images or frames that change several times per second. Humans perceive such virtual objects as moving continuously. Similarly, while memory capacity and process speed limit what a computer can handle, geometry objects, such as lines and planes, are conceptualized as infinitely extended. In this sense, virtual worlds do not exist in the computer or internet. We perceive a coherent image from the computer's monitor resembling some aspects of our physical world. Similarly, our world does not exist as a particular geometric entity; rather, we model our world using geometry. This perspective represents a particular conception of geometry, or mathematics in general terms, based on human experience, as elaborated below.

An Embodied Perspective on Mathematics

A growing number of scholars share an embodied cognition perspective on mathematics and mathematics education (e.g., Braithwaite & Siegler, 2018; Davis & Renert, 2014; Davydov, 1990; Duijzer, den Heuvel-Panhuizen, Veldhuis, Doorman, & Leseman, 2019; Fischer, 2017; Gerofsky, 2016; Lamon, 2012; Ni & Zhou, 2005; Thom, D'Amour, & Preciado-Babb, 2015). While there are several perspectives on embodiment in mathematics education as identified by Gerofsky (2016), such perspectives share a focus on the body with respect to learning. From an embodied perspective of cognition, the role of the human experience in the environment shapes the way we learn — the environment here not only includes physical elements such as the space, gravity and biology, but also other elements such as culture and language, including the role of metaphors in cognition.

Two particular mathematical concepts are relevant examples of the role of the embodied perspective on learning for this chapter: number and geometric transformations. Lakoff and Núñez (2000) elaborated on how mathematics can be constructed from grounding metaphors based on bodily experience of humans in the world. They offered four different grounding metaphors for arithmetic operations regarding particular meanings of number, namely: object collection, object construction, measuring stick, and motion along a path. The later metaphor for arithmetic corresponds to the number line, which had historical relevance in the conception of number, as Lakoff and Núñez posited: "Conceptualizing all (real) numbers metaphorically as point-locations on the same line was crucial to providing a uniform understanding of number" (p. 73). Regarding geometric transformations, Thom, et al. (2015) described how students made sense of transformation through experiences with their body, gesturing, and manipulating furniture to describe rolling, staking, and sliding.

As most of the images in computer graphics refer to some aspect of the physical world, even in their representation in two dimensions, the geometry involved in graphic manipulation has a direct connection to human experience in the world. An embodied perspective on mathematical concepts, therefore, can inform a teaching approach that incorporates computing education in K to 12. Notice that under this perspective, geometry for computer graphics can be introduced to students through bodily experiences without need for a computer. For instance, angles can be introduced through movement and direction at low grade levels, instead of via the fixed definition using two intersecting lines. This meaning for angle is particularly relevant in agent-based computation (Francis, Khan, & Davis, 2016; Sengupta, Kinnebrew, Basu, Biswas, & Clark, 2013), in which people program the behaviour of one or more agents, such as a character in a video game or a robot's movement. In agent-based programing, the references for direction are intrinsic in the sense that they depend on the position and orientation of the agent, as demonstrated, for instance, in the directive "turn 45 degrees to the right." This command depends on the position of the agent, as opposed to other external frames of reference conveyed in commands such as "face north."

TEACHING GEOMETRY FOR COMPUTER GRAPHICS

Teaching geometry for computer graphics at K to 12 school levels has a number of potential benefits. First, teachers can take advantage of the already existing connections to the program of studies. Second, the explicit application to computers and programing shows the relevance of mathematics in contemporary society, thus making mathematics content more authentic and relevant to students. Third, this mathematical knowledge can support classrooms oriented to the use of digital technologies for learning purposes, such as video game design as a learning activity (e.g., Ke, 2014). Finally, a knowledge of geometry for computer graphics can support self-directed students in learning about programing. For instance, while free software for video game design, such as Scratch and Unity, often include tutorials and examples to guide the learner step-by-step, a knowledge of geometry is essential to understand and engage in provided examples. This section describes the graduate course for teachers from which the study was conducted, elaborates on the research methodology of this case study, and the reports the corresponding findings.

Description of the Course

The course *Geometry in Nature, Art, and Computer Graphics* is the third component for the four-course graduate program *Contemporary, Emergent Mathematics*, co-developed and co-taught by the Werklund School of Education and the Department of Mathematics and Statistics of the University of Calgary. This program is based on the premise that contemporary branches in mathematics have a variety of applications that, with the aid of digital technology, are increasingly common in our society. Some of these branches and applications have started to appear in different programs of studies in Canada and around the world — for example, graph theory, cryptography, and programing.

This graduate program introduces teachers to contemporary mathematics and a range of current applications not traditionally included in the program of studies. The purpose is to explore mathematical topics, applications, and implications to society that could complement and enrich mathematics education at K to 12 levels. Each course in the program involves open-ended explorations beneficial for teachers with different levels of mathematical knowledge and who are interested in learning about the latest applications of mathematics. Digital technologies play an important role by supporting the exploration and application of mathematics.

The course *Geometry in Nature, Art, and Computer Graphics* focuses on the mathematical ideas around shape and symmetry in nature, in contemporary and classical art, and in computer graphics, including algorithms for image representation and manipulation used for diverse applications such as virtual reality and media communication. It was first taught during the winter 2019 term (thirteen weeks). The following topics were covered in its first version: proportion in nature and arts; agent-based programing; trigonometric constructions; image generation through iteration; fractals; symmetry and transformations; tessellations; and coordinates and vectors. Teachers enrolled in this course were expected to:

- Develop an understanding of the mathematical principles involved in different applications of geometry to arts, modeling nature, and computer graphics.
- Apply geometric concepts in the generation of images using diverse software.
- Write recursive codes to generate fractal images using specific computer languages.
- Design learning activities for K to 12 levels related to applications of geometry.

This fully online course was delivered through the Desire2Learn (D2L) platform for online courses and the Zoom tool for synchronous video conferences. D2L was used to introduce course content, share classroom resources, including links to readings and video, and as a site for ongoing dialogue. Teachers enrolled in the course were required to work on a programming language for image generation. As the course does not assume a pre-requisite in programming, versions of Logo and Scratch were used as programming languages. Additionally, GeoGebra was used for image generation and manipulation.

The course involved three learning tasks for teachers. The first learning task comprised weekly discussions and presentations in discussion forums and three two-hour synchronous video sessions. Teachers were expected to complete the readings for each week, engage in additional tasks, and actively participate in group discussions. The discussions focused on learning the content of the course, searching and sharing teaching resources, discussing the implementation of the content in the classroom, and sharing explorations with the content, including generated images and animations. Teachers were expected to post content attending to the directions for each week and to respond to at least one of the others' posts within the corresponding week. They were also asked to comment on any difficulty, frustration, and success while engaging in the activities of the week. The synchronous sessions were used to provide general information about the course, introduce new topics, and receive feedback from students.

The second learning task in the course consisted of the design of three learning activities for the classroom, commonly reported as lesson plans. Teachers could work individually or with a partner to design the learning tasks and illustrate key topics and ideas introduced in the course. They were asked to submit drafts of their designs for feedback before final submission. The designed tasks should fulfill the following requirements: be appropriate for grades K to 12; facilitate a deep and rich understanding of the mathematical topics addressed in the course; include a rationale for the learning task, as supported by the curriculum; elaborate on the practicalities of its implementation in the classroom; and describe formative-assessment strategies.

The third learning task of the course consisted of an online laboratory. The purpose was to introduce new mathematical and computational content and to assess teachers' understanding of the key mathematics concepts and procedures used throughout the course. This laboratory consisted of introductions to the content, including images and video, and quick questions used as a means of formative assessment to provide feedback to instructors and material that could be adapted and used according to students' needs. The quiz tool of D2L was used for this purpose; it allowed teachers to see the correct answers after submission and to re-submit an unlimited number of times.

Purpose and Research Questions

The general purpose of the study reported here is to identify how elements of geometry for computer graphics can be incorporated in K to 12 education. For this purpose, a qualitative case study approach (Merriam & Tisdell, 2016; Stake, 2003; Yin, 2015) was followed with a focus on the first cohort of the *Geometry in Nature, Art and Computer Graphics* course. This first cohort (winter term, 2019) consisted of eight teachers from elementary to upper high school. Teachers had diverse backgrounds regarding mathematics experience: while some had majors in engineering or science, others specialized in subjects such as arts and physical education.

The results of the study reported in this chapter address the following research questions:

- How did teachers incorporate elements of the course into learning activities they designed for the classroom?
- What challenges and suggestions for incorporating this content in the classroom can be identified in this implementation of the course?

The answers to these questions also provide ground to identify the particular knowledge required to teach geometry for computer graphics at school levels.

Case Study as Research Methodology

Although there are multiple perspectives on case study as a research methodology (Yin, 2015), there is a common agreement that this approach focuses on a phenomenon bounded by a specific unit of analysis and defined by contextual elements — for example, the cohort of teachers involved in this study. Diverse authors also agree that case study can draw from a variety of research methods, including quantitative, qualitative, and mixed methods, depending on epistemological perspectives. The approach taken for this research is a qualitative case study drawing from a constructivist perspective, as elaborated by Merriam and Tisdell (2016). In contrast to the realistic perspective (Yin, 2015), the constructivist perspective does not assume one single reality; instead, this approach acknowledges different realities, including the researcher's perspective (Charmaz, 2006). Scholars such as Stake (2003), Charmaz (2006), and Strauss and Corbin (1990), agree that methods from grounded theory, such as open and axial coding, are also suitable for qualitative research, even if the purpose is not the generation of theory.

This research can be considered as an instrumental case study (Stake, 2003), as the purpose is to gain insight on a particular phenomenon, specifically, how teachers can implement the content of the graduate course into their own classrooms. Contextual information is also key in case study (Merriam & Tisdell, 2016; Stake, 2003; Yin, 2015). Consequently, the chapter provides specific descriptions of the course and its implementation, including quotes and excerpts of tasks designed by teachers.

Although results from a qualitative study are not meant to be generalized in the same way that quantitative studies do, the results here have the potential to inform other teacher professional learning initiatives targeting computer education and computational thinking.

A common hallmark of case study research is the use of multiple sources of data (Yin, 2015). This study draws from diverse sources of data, such as assignments submitted by teachers, discussions blogs, teachers' feedback, and course design notes from the instructors. These sources of data were used to support the finding in the study — a process called triangulation.

Six out of the eight teachers enrolled in the course provided consent to use their assignments, posts, and comments for this study. They requested to use their names when presenting quotes or parts of their work. The assignments included design of classroom tasks (9 learning tasks), participation in discussion boards (859 entries), and feedback provided during synchronous, online meetings (3 sessions).

Entries from the discussion board that related to the implementation of the content of the course into the classroom and to the tasks for the classroom were subjected to open coding (Charmaz, 2006). This coding process is based on a constant comparative analysis in which codes are compared, revisited, refined, and blended or split as data are being analyzed. While there was no pre-determined set of codes before analyzing the data, the specific purpose of the study informed this process, namely, the focus on how the content of the course could be incorporated into K to 12 education. These codes then informed the emergence of broader categories, including their relationships between each other — axial coding. The categories were contrasted with teachers' comments during the synchronous sessions of the online course and the tasks designed by teachers. Documents from the course, such as the syllabus, the design of the online learning platform, and instructor's notes during design meetings, were used to provide the contextual description of the course. Finally, findings were shared with participants for their confirmation and feedback, a process called member-checking in qualitative research (Charmaz, 2006).

The analysis of the learning tasks and the posts of the discussion forum resulted in two broad categories: the role of computer graphics in the designed tasks; and teachers' learning experience during the course. The first category can be described as a product, while the second can be described as a process. Before teachers started to design their tasks for the classroom, they had to engage in the content of the course. The next two subsections correspond to these two categories and provide an answer to the first research question. The subsequent subsection focuses on the second research question; the posts in the discussion forum provided further information about challenges and advice for implementing computer graphics at school level.

The Role of Computer Graphics in the Designed Tasks

The nine tasks analysed in this chapter can be classified into three categories depending on the role of computer graphics in each task: indirect, enrichment, and instrumental. The tasks in which computer graphics played an indirect role addressed topics related to computer graphics but without any type of coding or virtual image manipulation. One example of this task is a Grade 6 lesson plan in which students went to the gym and were required to work with angles in the contexts of ball passes in basketball. Three students formed a configuration corresponding to a particular position and followed different commands to pass the ball. Commands included teacher's calls, such as: "Chest Pass to 90 degrees"; "Bounce Pass to 270 degrees"; and "Vertex Trade Places with 180 degrees."

The intention of this lesson was to address angles in a more dynamic way than via the static definition of angle using two lines. Students' actions required a conception of angle as movement as they would have had to turn the corresponding angle to pass the ball. This notion of angle is suitable for agent-based programing in which the programmer provides direction to the agent (the turtle in Logo or a robot moving along the floor) to navigate the environment. It is important to notice that teachers were deliberate in the design of the tasks regarding the connection to computer graphics. Other tasks in which computer graphics played an indirect role included transformations in the plane, matrices, and symmetry in the form of self-similarity.

In other designs, computer graphics were used for enrichment after students engaged in the content of a lesson. For instance, one of the designed learning tasks, a lesson on Golden Ratio, focused on how to calculate this ratio by deducing a rational equation that could be solved by simplifying it to an equivalent quadratic equation. The enrichment piece of the lesson consisted of creating routines in Logo to draw Golden Rectangles and Golden Spirals. For this purpose, the designed lesson included some pedagogical decisions, such as:

- Allocating time to learn some basic Logo commands through the use of the Logo cheat sheet provided.
- Allowing students to play with the commands and experience Logo for the first time.
- Discussing what would it take to make a Golden Rectangle on the computer screen.
- Drawing the image, paying attention to every detail of movement (forward, backward, turn and by how much).
- Developing a step-by-step program to create a Golden Rectangle with $a = 10$ and $b = 6.180$ (the first value in the previous table).

Some of these pedagogical decisions were implemented in the course and addressed in the discussion forum, as elaborated in the next subsection.

Finally, in some tasks, the role of computer graphics was instrumental in the sense that it was a main component of the lesson. For example, one teacher designed a task in which students had to construct right triangles in Logo. The following excerpt of the lesson plan indicates the purpose and rationale for the task:

The point of this activity is to connect the use of code as vehicle for drawing out triangles. It forces the student to visualize the path in terms of magnitude and direction (connection to physics). Students that understand how to code in Logo give the teacher another avenue to apply knowledge and hopefully opens doors for students to explore a new skill. (Dominic)

The lesson begins by asking students to create different right-angle triangles, having defined sides and angles. Students might have to change the scale of the Logo field to display the shapes in a reasonable size. Then, the teacher asks students to discuss any difficulties experienced as a class — some students would likely have difficulty with the rotations — and to show the process in front of the class. Then, students are asked to create right-angle triangles with specific information and missing angles and/or sides, such as the right triangle with sides 60, 30 and 40 units. Here, students need to use trigonometric ratios to find the missing information.

The lesson plan indicates that the teacher should model examples in front of the class, as indicated below:

Continue with additional examples if necessary to ensure students are comfortable with the code. Show the process in front of the class. The point of this activity is to connect the use of code as vehicle for drawing out triangles. It forces the student to visualize the path in terms of magnitude and direction (connection to Physics). (Dominic)

The lesson plan also includes some elements of debugging, requiring students to modify one of the following codes to generate an oblique triangle:

Code #1: cs fd 170 rt 180-40.7 fd 23 rt 180-47.3 fd 150

Code #2: cs fd 170 rt 180-40.7 fd 230 rt 47.3 fd 150

Note that the code already considers the supplementary angles instead of the interior angles, such as 180 – 40.7. This is an example of specific geometric knowledge related to computer graphics: Agent-based programing requires to work with the exterior, instead of the interior angle, which is used when drawing the triangles by other means.

The lesson plan also incorporates suggestions to support students who struggle to figure out the mistake in each code:

Emphasize that there is only one side or angle that needs to be fixed to create a proper triangle. Look for the following strategies:

- Students checking to see if all [interior] angles add up to 180°
- If angles all add to 180°, students can sketch the triangle with the angles and be able to tell which side is too short.
- If angles do not add to 180°, students can sketch the triangle with the sides to see which angle doesn’t add up.
- Advanced students might discover that you can line up all three proportions using the Sine Law to see which one was not equal to the others. (Dominic)

The suggestions for supporting students in this task are specifically related to trigonometry. Yet, the implication for manipulating computer graphics is relevant as the advice involves verification of geometric elements in the construction of the triangles in agent-based programing.

Teacher's Learning and Task Design Process

Teachers engaged in learning the corresponding content of the course before they started designing the learning tasks. Content was provided through the laboratory activities, some readings, and the discussion forum. The laboratory was intended for formative assessment, rather than for assessing teachers' understanding of the content. The quiz tool in D2L was used for this purpose and teachers received immediate feedback after each quiz was submitted; they could also resubmit the quiz as many times as they wanted. Sometimes, teachers were required to generate images from the content provided in the laboratory.

Teachers often considered pedagogical decisions used in the course for their own task designs, as prompted in the previous subsection. One example was the use of the cheat sheet, a short glossary of the specific commands needed for engaging in the activities of the course. Teachers appreciated this resource, as shown in the following excerpt:

Wow, I was away from programming for a few days to mark diploma exams, and when I returned, I felt I had forgotten everything. I was super grateful for my summary sheet [cheat sheet]. I didn't have to start from the beginning. Such a great resource! (Roxanne)

The following example of the laboratory is presented to show several features of the course, some of which teachers included in their own task designs. The laboratory, *2D Vectors and Tessellations,* was assigned by the end of the course. It included two videos showing how to create a row of equal figures from a given figure that could be used to tesselate the plane, and then how to copy several rows to create the tessellation in GeoGebra.

Figure 1 shows the use of the *Polygonal* command to create the initial shape, named *poly1*, which was selected to generate the tessellation. This process can be easily done with the graphic tool instead of the script, as shown in the video; however, it is presented here as a reference for the reader. Then, the *Sequence* command was used to create a row of five shapes with the aid of the *Translate* command, which uses a vector for the translation. GeoGebra generated the images in a list named *l1*. The command Sequence is then used again to copy several rows of the image. In this case, the shape has to move four units to the right every time it moves two units downwards. So, the command uses the vector (4,-2) multiplied by the scalar *i*, that runs from -2 to 2. The command Vector has to be used to create the vectors by multiplying by a scalar.

The design of the course implemented two of the pedagogical practices identified by Weintrop and colleagues (2016), namely: understanding and modifying others' scripts; and troubleshooting and debugging. Teachers had access to the script used to generate the tessellations shown in the videos. The purpose was to help them create their own tessellations by copying the scripts. The laboratory also included quiz questions intended to identify common issues in a script. Figure 2 shows an example of such questions that focused on the iteration needed to generate a row of five copies of the original shape. Answering this question requires attention to both the parameter of the iteration in the command Sequence and the

Figure 1a. Scripts in Geogebra used to generate a tessellation. The initial shape was generated as a polygon with given set of points as vertices.
poly1 = Polygon(A,B,C,D,E,F,G,H,I,J)

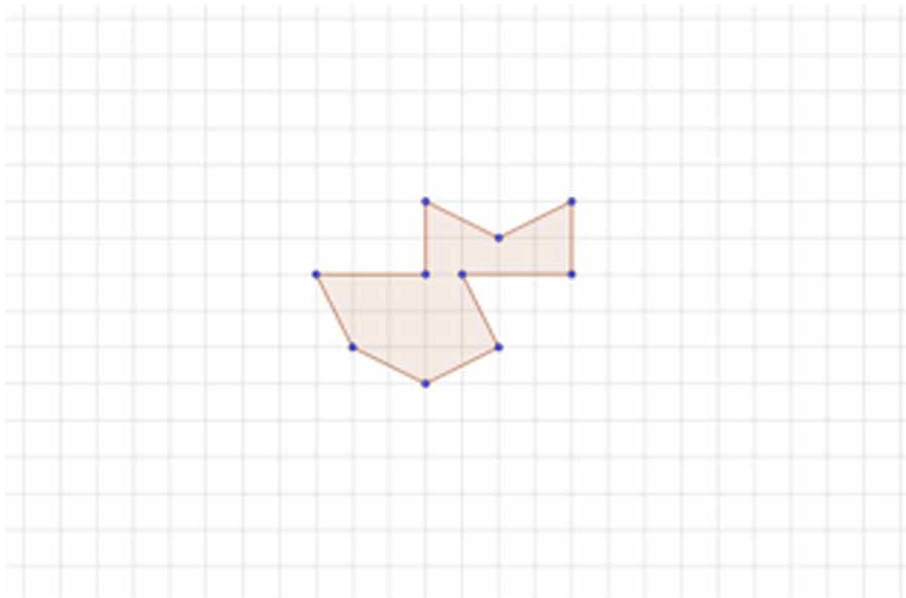


Figure 1b. Scripts in Geogebra used to generate a tessellation. The initial shape was generated as a polygon with given set of points as vertices.
(Images created in GeoGebra by Paulino Preciado, 2019 under CC BY NC SA 3.0 https://www.geogebra.org/m/g5avh2js).
*l1 = Sequence(Translate(poly1,(4*i, 0)),i,-2,2)*
l2 = Sequence(Translate(l1,Vector(i (2,-4))),i,-2,2)

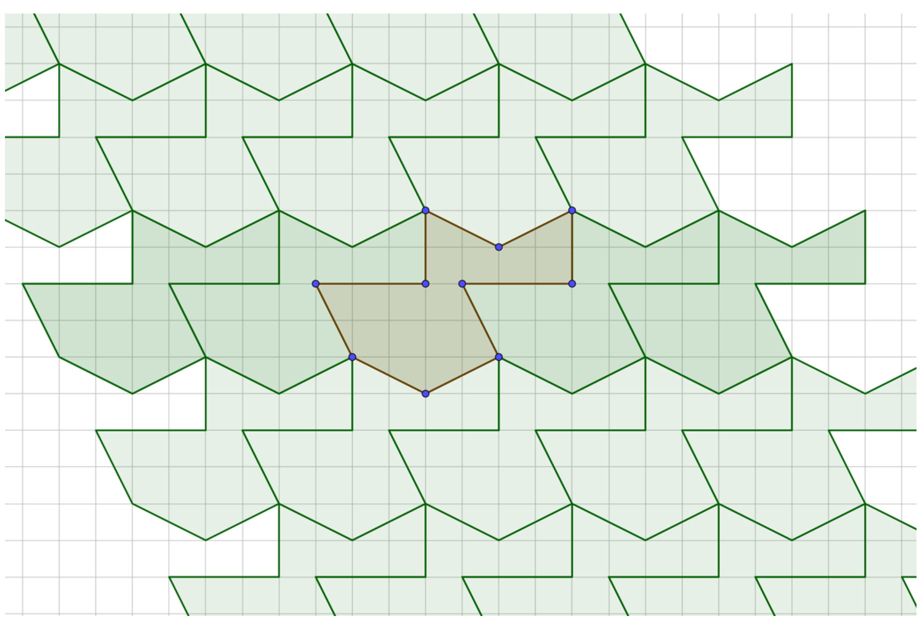
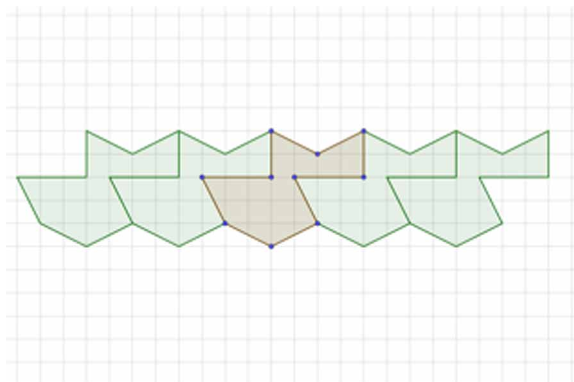


Figure 2. Multiple choice question from the laboratory. (Image created in GeoGebra by Paulino Preciado, 2019 under CC BY NC SA 3.0 https://www.geogebra.org/m/g5avh2js).



distance between each copy, corresponding to the scalar by which the vector (1,0) has to be multiplied — in this case it is 4.

Which of the following scripts can be used to create the row of five shapes in the figure?

a) Sequence(Translate(poly1, (4*i, 0)), i, -2, 2)
b) Sequence(Translate(poly1, (i, 0)), i, -5, 5)
c) Sequence(Translate(poly1, (2*i, 0)), i, -2, 2)
d) Sequence(Translate(poly1, (i, 0)), i, -2, 2)

Examples of the images generated by teachers are shown in Figure 3. Their initial attempts often consisted in changing the initial shape, as in part (a) of Figure 3. However, they also generated more sophisticated examples, as shown in parts (b) and (c).

Figure 3a. Examples of tessellations generated by teachers
(2019 ©, Alisa Cooper, 2019. Used with permission.)

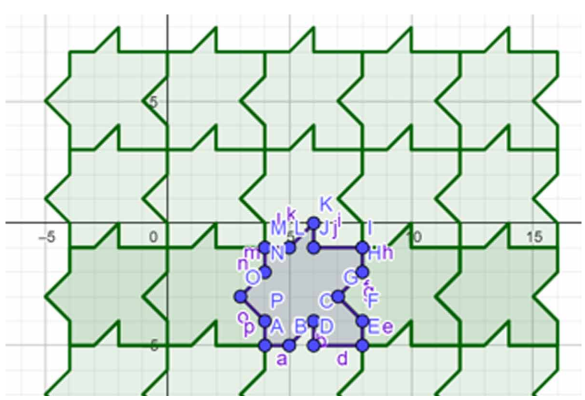


Figure 3b. Examples of tessellations generated by teachers
Image created in GeoGebra by Krishani Starnes, 2019 under CC BY NC SA 3.0, https://www.geogebra.org/m/dkmmtrma

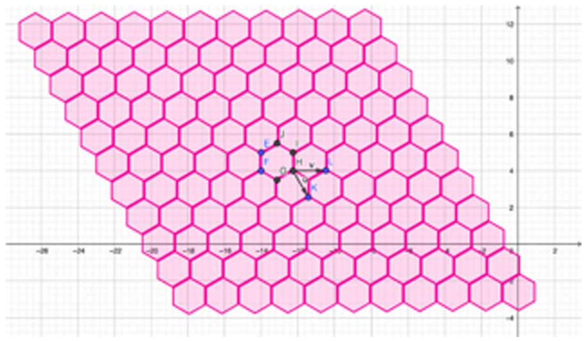
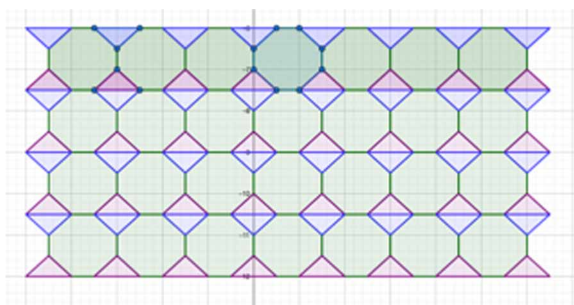


Figure 3c. Examples of tessellations generated by teachers
(2019 ©, Carissa Hanson, 2019. Used with permission.)



The tessellation in Figure 3, part (b) was generated using a double sequence and the vectors shown in the image. The following script was used to generate the tessellation:

Sequence(Sequence(Translate(poly1,Vector(u*i+v*j)),i,-5,5),j,-5,5)

Finally, the tessellation in Figure 3, part (c) comprised three shapes instead of one: one octagon and two triangles.

Some posts in the discussion forum also included descriptions of teachers' attempts to engage in the corresponding activities. These comments provide some accounts of the teachers' learning process. The following script is an example of what teachers shared in the forum:

Paulino's initial set of instructions for replicating a polygon Sequence(Translate (q1, (i, 0)), i, -3, 3) was just something I was copying with different shapes at first. But I was frustrated when the shapes would not do what I wanted them to! I slowly learned by manipulating each variable what the outcomes would be and was feeling pretty confident by the end that I could create tessellations with this sequence. I'm still working on vectors and will post more on those soon. A big breakthrough for me was when I realized that I could use (Sequence(Translate (l8, 1.5 (0, i))), i, -3, 0) to move a group of shapes that were individually larger than one unit by several spaces so they didn't overlap (my first ones did) and to change so I could move the shapes on the x or y axis. (Carissa)

Three features can be highlighted from this script. Teachers benefit from the initial approach of copying the script and applying it to a different shape. Second, there was a process of debugging, applying systematic variation to the variables involved in the script. And third, a realization of the effect that the scalar — 1.5 in this case — has when multiplied by the vector.

Similar strategies were used for the generation of fractals and other shapes in Logo. Teachers were requested to create a Logo cheat sheet to record descriptions of the commands they were learning about. In other cases, teachers were given a particular code and were requested to modify it. As discussed below, some of these strategies were used for their designing of learning tasks for their own classrooms.

The discussion forum provided the environment for learning as a community. Teachers shared and discussed online resources that could be used to incorporate the content of the course in the classroom. They also engaged in conversation regarding implementing the elements of computer graphics in their classroom, as well as their own experience in the engagement with the content of the course. Regarding their own experience in the content of the course, two specific strategies discussed in the forum were salient and can inform the design of learning tasks related to this topic. One refers to sketch and conduct calculations before coding, and the other is about coding step-by-step before implementing an iteration resulting from using commands such as Repeat in Logo or Sequence in GeoGebra, or using recursion in the code. Excerpts of the conversations in the forums are offered as follows as a means to illustrate how teachers talked about these strategies. The first excerpts refer to sketching and drawing before coding:

I found the program pretty easy to follow and realised quite early on that if I want to draw something, I would need to draw it out by hand and calculate the dimensions of my drawing beforehand and translate it to the instructions for the little turtle. So, using a bit of trig, I plotted out my angles and dimensions on a separate piece of paper and drew out my piece of conceited art. (Dominic)

This comment was seconded by another teacher.

I completely agree with this statement. I really don't love the educated guess and check method I tend to use when trying to complete numerous exercises. I prefer to spend my time on paper with sketches and measurements prior to beginning the coding. Once I have all the details worked out, I can delve into the code and trouble shoot that. (Roxanne)

The following excerpt demonstrates the strategy of writing a long code before using an iterative command or a procedure and also provides a sense of how teachers felt in the online community:

Thank you for your response. It made me feel better reading your response. Yes, I too write out the long code before I add the shorter commands and [parameters].

…

It's interesting to reread everyone's posts at the end of the week and see some common themes. I am noticing that many of us still proceed by writing everything out in the "long" way before finding more efficient methods. … I don't know about everyone else, but I find it reassuring that we're all having those common experiences as we go through this process. (Krishani)

Finally, the online conversation also included discussion about the curriculum. One particular topic that permeated through several grade levels is vector, as indicated in the following excerpt:

In Math, we hardly ever explicitly talk about vectors, but we use them all the time. In transformations, students describe translations with a combination of direction and magnitude – your h and k values are vectors. In lower [Grade] level transformations, students describe translations in phrases such as "4 units down and 3 units left." For rates of change, we also calculate slope to find things like velocity, just like it's done in Physics, but we tend to stay on our realm of academia rather than stepping on the toes of the Physics teacher. Our word problems are just one step shy of being physics questions because we shy away from emphasizing too much on direction and we focus instead on mostly magnitude.

I believe there is a ton of material in High School Math that deals with vectors, but for some reason, we don't explicitly use the word vector in discussion. (Dominic)

The previous excerpt also prompts to a potential multidisciplinary approach by stressing the connection of vectors to physics. Other example of interdisciplinarity included connections to art and physical education, as elaborated in the lesson plan discussed in the previous subsection related to angle.

Challenges and Suggestions

From the posts in the discussion forum, it was possible to identify teachers' struggles with parts of the course; this could provide further guidance for the implementation of computer graphics in the classroom. A recurrent issue in the course was the different versions of Logo that could be used in the classroom. The versions vary considerably, and this caused teachers to become frustrated with and distraction from the content of the course. Here is one example of teachers' comments on this issue:

Round 2 was definitely a bit more challenging, but still a lot of fun! Since I have been working with ACSLogo, the programming language is a bit different. I felt pretty deflated the more I tried to program commands, as I was getting nowhere. Even after searching for different tutorials and videos, I kept hitting some major roadblocks. So, then I jumped ship to TurtleAcademy, in hopes that maybe something would click there. Unfortunately, continuing to strike out, I was still unable to get that turtle to follow any of the programmed commands. I couldn't help but think, I bet my students wouldn't be having these issues figuring this out, after blowing me away with their creations last week! (Carlene)

As teachers had different operating systems on their computer, they were able to use different versions of Logo. We tried to provide different resources and re-write scripts for different versions of Logo for the teachers, but this was dauting and inefficient. The differences in the versions of Logo go beyond syntax structure; some commands do not have an equivalent version from one version to the other. The advice here is to make sure that everyone has access to the same version of the software. Online versions that work in any platform — for example, Turtle Academy and Logo Interpreter — could be used when people work with different operating systems.

Lack of experience programing can also be a source of struggle for both teachers and students. Teachers often reported being afraid of coding as well as feelings of anxiety and frustration while engaging in coding tasks in the course. It would be important to consider a learning curve for both teachers and students when using a programing language in the classroom.

The other challenges that were evident in teachers' posts relate to mathematical knowledge itself and mathematics in the school curriculum. Some teachers indicated that lacking knowledge about mathematics prevented them from seeing connections across mathematical topics and grade levels. The following is an excerpt from an exchange between two teachers:

I also agree with you that students wouldn't likely make any connections to that image/example independently. I am not 100% sure about the Alberta curriculum, but I know that in Saskatchewan tessellations are included in Math 8, and then don't show up again. But many teachers also see that as a superfluous unit and it is the one that is most likely to get cut if teachers are running short on time. So, it's no wonder that students wouldn't recognize the tessellating pattern if they only have very limited exposure to the concept. (Alisa)

Another teacher replied to this post:

You nailed it Alisa. I am currently sitting in a PD next to a junior high math teacher. He said it is part of the Alberta Math 8 curriculum, but it is often disregarded by the teacher because it doesn't seem to have much of a connection to prior skills nor to skills yet to come. It may be brushed over quickly because teachers don't have enough exposure to resources that allow for a deeper understanding of tessellations at all. Time is always a factor when covering curriculum, and sometimes teachers need to make decisions to delve deeply into some concepts while others are covered superficially. Talking to the department head of a junior high school, he said tessellations are only covered superficially. (Roxanne)

The point made in this excerpt was further supported by a comment regarding the textbooks used for teaching:

Today I was going back to the grade 8 textbook to double check something to help me with my own work in Geogebra. I thought it would be interesting to share the relevant sections of the textbook here. Turns out that not only does the textbook talks about characteristics of tessellations and how to make them, it even includes an enrichment activity for creating them with technology. (Alisa)

The previous exchanges suggest that a deeper understanding of the connections between tessellations and geometric transformations can help teachers identify the relevance of this topic across the program of studies. For instance, transformations and symmetry are not exclusive to geometry; they also appear when studying transformations of functions. It seems that a compartmentalized perspective on the program of studies prevents teachers from identifying these connections.

Finally, another challenge reported in teachers' conversations was the lack of time to learn to use new technologies. This was combined with a lack of confidence and knowledge with the program of studies for one particular teachers, as illustrated in the following quotes.

I completely agree with you, Krishani. The time necessary to learn the technology is just not available to us. In addition, a teacher would have to feel confident enough about the technology to bring it to a classroom environment. That again takes a lot of time and practice. (Roxanne)

I know from my own experience that I am in the same boat as Krishani. Between time constraints and my own lack of confidence in the programs I too skipped the enrichment. I was just excited of years when I got to teach tessellations at all (it was always my last unit, and depending on the group of students sometimes we took longer in other topics and didn't get there). Because I was also the Arts Ed teacher I could sometimes do some tessellation related stuff in that class, but it was never to the same depth of knowledge as when I taught it in Maths class. (Alisa)

These previous quotes suggest a need for teacher training programs to incorporate some of these skills to support teachers' confidence before they try to implement elements of geometry for computer graphics in their own classroom.

DISCUSSION

This case study sheds light on how teachers can implement elements of computational graphics in K to 12 education. Shulman's (1986) classification can be used to identify specific knowledge related to geometry for computer graphics. The *specific content* of the course in terms of geometry for computer graphics corresponds to the interactions of geometry and computer programing. The geometric content of the course included: proportion and ratios; angles; trigonometry; fractals and tessellations; vectors and matrices; and transformations and symmetries. The computational content included: general commands and syntax for Logo, Scratch, and GeoGebra; procedures and variables; Repeat and Sequence commands; and recursion. The content knowledge in this case included intrinsic connections between the two subjects. For instance, virtual worlds involved in movies and video games are structured by some type of coordinate system, such as the cartesian plane or the cartesian space for 2D and 3D environments. Virtual objects in these worlds correspond to geometrical objects, such as points, segments, polygons and fractals. Geometrical transformations, including translations and rotations, are used to animate these worlds.

In addition to this knowledge of the content, teachers need pedagogical, specialized, and curricular knowledge for teaching computer graphics. The specialized knowledge for teaching differs from the content knowledge in the sense that it is specialized for teaching. While expert programmers might not be aware of this knowledge, teachers should be. In this sense, the focus on the difference between meanings related to specific mathematical concepts is relevant for teaching.

This focus allows for the introduction of meanings of mathematical objects relevant for computer graphics even without the need for start coding. One specific example is the concept of angle, which is introduced in the Alberta Education (2008) program of studies through the intersection of two lines. However, in image manipulation, the concept of angle has a more dynamic perspective as it relates to movements and transformations. Notice that these two meanings for angles are very different. While an expert (the teacher or a professional programmer) has already blended these two meanings, the learner, as a novice, has to discern the connection between the two. In this line of thought, the integration of computer graphics in K to 12 education can be done by addressing elements already present in the program of studies but with different representations of meanings.

Other examples, such as the concept of vector as a magnitude with a direction and geometric transformations, can be enacted by children using manipulatives and their body. Proportion also has a relevant role when a vector or an image is multiplied by a scalar to shrink or stretch it; this differs from the regular meaning of multiplication that only involves numbers.

A relevant element of the specialized knowledge for teaching geometry for computer graphics is the connections between mathematical concepts. For instance, the topic of tessellations is not only relevant for geometry, but also for algebra when students study transformations and symmetries of functions.

There are also specific elements of programing related to this specialized knowledge, such as the implementation of recursive algorithms for generating fractals or the use of vectors in different commands: These require interaction between geometry and programing. The way that the coordinate systems are used in computer graphics and the need to consider exterior angles when generating polygons in agent-based programing represent other examples of this interaction.

The pedagogical content knowledge identified in this case study corresponds to the specific decisions made for teaching. Examples of these decisions are the use of the cheat sheet as a quick reference for commands while students learn them, or the provision of scripts that students can modify to explore and build on them. In this case study, systematic variation of specific elements of parameters was a useful strategy that teachers followed in the course.

One particular difficulty identified by teachers during their learning process was the implementations of sub-routines or the use of iteration through recursion or through specific commands such as Repeat and Sequence. Teachers reported that writing the "long" version of the code was useful to later understand a simplified version. Finally, teachers also reported the need to sketch the drawings and conduct computations before starting coding. This was later implemented in some of the tasks designed by teachers in the course.

The data in this case study also prompted knowledge of the curriculum. This was reflected in two different features. On one hand, a knowledge of the program of studies at different grade levels would help teachers identify the relevance of some topics, such as tessellations and fractals, to algebra. On the other hand, being aware of the connexions among different topics would help teachers to pay more attention to content already included in textbooks, such as tessellations, which, as indicated by one teacher, is often excluded from teaching; furthermore, some teachers might not even be aware of this content.

RECOMMENDATIONS

The focus of this chapter is already a reminder to pay attention to how subjects such as geometry can support computer education. The results of this case study can inform education for students and teachers and the development of curricular materials. The examples provided in the findings can be replicated by other teachers, who can also search for different resources that could involve computer graphics indirectly as enrichment or as an instrument to teach other content.

Shulman's (1986) classification of knowledge for teaching was useful to identify venues for teacher education programs. Teachers' learning experiences, reported in the findings of this study, can also be transformed in recommendations for teacher education. For instance, there is a need to pay particular attention to topics that were difficult, such as recursion, sub-routines, and commands for iteration.

Finally, developers of curricular materials such as textbooks and online resources could stress the connection between different topics and across grade levels so teacher are aware of the relevance of these topics, as well as the potential opportunities to approach subjects such as algebra from a geometrical perspective.

FUTURE RESEARCH DIRECTIONS

This case study already shed light into how elements of geometry related to manipulation of computer graphics can be integrated into the classroom at K to 12 levels; however, similar studies are required to extend these results. Future venues for research in this direction could include: the communities of learning that develop in this online course; evaluation of the impact of this course, and other interventions, on teachers ‘classrooms; and an extended conversation on the knowledge for teaching geometry for computer graphics.

CONCLUSION

An analysis of the online discussions and tasks involved in the course showed how mathematical content across K to 12 could be addressed in relation to the different geometrical topics of the course, even though the topics themselves might not be a part of the program of studies at the corresponding grade levels. These implementations did not necessarily involve adding new content to the program of studies, but rather addressed the content in a different way. For instance, in the mathematics program of studies in Alberta, the topic of angles is introduced in terms of the intersection of two lines. However, for agent-based programing (Sengupta et al., 2013), such as Logo or programing a robot's movement, angles are more dynamic with respect to rotations and direction. Similarly, vectors are important in image manipulation; however, they are not usually presented in the program of studies from the mathematical perspective. Yet, the main ideas of a vector can be introduced at a lower grade level as actions represented by magnitude and direction, such as: "walk three steps forward." An understanding of mathematics as embodied can help teachers identify these connections.

The findings of this chapter contribute to a debate on the role of mathematics, in particular geometry, in computer education. The research also contributes to the literature by demonstrating how mathematics can support computer education as well as how mathematics and computer education can mutually support each other. Such an approach is currently uncommon in the literature, as discussed at the beginning of the chapter.

Results from this study also begin to shed light on the articulation of specialized and pedagogical content knowledge specific to geometry for computer graphics. The study has identified some of the specialized knowledge for teaching in this case, namely, considering meanings of geometrical objects in a more dynamic way, such as with angles and vectors for transformations. Similarly, pedagogical knowledge is presented, such as in providing students with a pre-given script to analyze and modify, and in introducing step-by-step examples before using abstract iterative routines as in repetition or recursion. This initial articulation of knowledge for teaching geometry for computer graphics is a theoretical contribution that can inform teacher education programs and curricular resources oriented to computer education.

ACKNOWLEDGEMENT

I appreciate teachers' willingness to participate in this study and to share their insight, frustrations and creations.

REFERENCES

Alberta Education. (2008). *Program of studies – Mathematics*. Government of Alberta. Retrieved from https://www.alberta.ca/programs-of-study.aspx#toc-1

Ball, D. L., Lubienski, S. T., & Mewborn, D. S. (2001). Research on teaching mathematics: The unsolved problem of teachers' mathematical knowledge. In V. Richardson (Ed.), *Handbook of research on teaching* (4th ed., pp. 433–456). Washington, DC: AERA.

Ball, D. L., Thames, M. H., & Phelps, G. (2008). Content knowledge for teaching: What makes it special? *Journal of Teacher Education*, *59*(5), 389–407. doi:10.1177/0022487108324554

Baumert, J., Kunter, M., Blum, W., Brunner, M., Voss, T., Jordan, A., ... Tsai, Y. M. (2010). Teachers' mathematical knowledge, cognitive activation in the classroom, and student progress. *American Educational Research Journal*, *47*(1), 133–180. doi:10.3102/0002831209345157

Begle, E. G. (1972). *Teacher knowledge and student achievement in algebra* (Vol. 9). Palo Alto, CA: School Mathematics Study Group.

Begle, E. G. (1979). Critical variables in mathematics education: Findings from a survey of the empirical literature. Washington, DC: Mathematical Association of America; National Council of Teachers of Mathematics.

Blömeke, S., & Delaney, S. (2012). Assessment of teacher knowledge across countries: A review of the state of research. *ZDM*, *44*(3), 223–247. doi:10.100711858-012-0429-7

Blömeke, S., Gustafsson, J. E., & Shavelson, R. J. (2015). Beyond dichotomies: Competence viewed as a continuum. *Zeitschrift fur Psychologie mit Zeitschrift fur Angewandte Psychologie*, *223*(1), 3–13. doi:10.1027/2151-2604/a000194

Braithwaite, D. W., & Siegler, R. S. (2018). Developmental changes in the whole number bias. *Developmental Science*, *21*(2), e12541. doi:10.1111/desc.12541 PMID:28229555

Charmaz, C. (2006). *Constructing grounded theory: A practical guide through qualitative analysis*. Thousand Oaks, CA: SAGE.

Davis, B., & Renert, M. (2014). *The math teachers know: Profound understanding of emergent mathematics*. New York, NY: Routledge.

Davydov, V. V. (1990). Types of generalization in instruction: Logical and psychological problems in the structuring of school curricula (J. Teller, Trans.). Reston, VA: National Council of Teachers of Mathematics.

Duijzer, C., den Heuvel-Panhuizen, M., Veldhuis, M., Doorman, M., & Leseman, P. (2019). Embodied learning environments for graphing motion: A systematic literature review. *Educational Psychology Review*, 1–33. doi:10.100710648-019-09471-7

Fischer, M. H. (2017). Why numbers are embodied concepts. *Frontiers in Psychology*, *8*, 2347. doi:10.3389/fpsyg.2017.02347 PMID:29379459

Francis, K., Khan, S., & Davis, B. (2016). Enactivism, spatial reasoning and coding. *Digital Experiences in Mathematics Education*, *2*(1), 1–20. doi:10.100740751-015-0010-4

Gerofsky, S. (2016). Approaches to embodied learning in mathematics. In L. English & D. Kirschner (Eds.), *Handbook of International Research in Mathematics Education* (3rd ed., pp. 60–97). Abingdon, UK: Taylor & Francis.

Hill, H. C., Rowan, B., & Ball, D. L. (2005). Effects of teachers' mathematical knowledge for teaching on student achievement. *American Educational Research Journal*, *42*(2), 371–406. doi:10.3102/00028312042002371

Hubbard, A. (2018). Pedagogical content knowledge in computing education: A review of the research literature. *Computer Science Education*, *28*(2), 117–135. doi:10.1080/08993408.2018.1509580

Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*, *73*, 26–39. doi:10.1016/j.compedu.2013.12.010

Lakoff, G., & Núñez, R. E. (2000). *Where mathematics comes from: How the embodied mind brings mathematics into being*. New York, NY: Basic Books.

Lamon, S. J. (2012). *Teaching fractions and ratios for understanding: Essential content knowledge and instructional strategies for teachers*. New York, NY: Routledge. doi:10.4324/9780203803165

Merriam, S. B., & Tisdell, E. J. (2016). *Qualitative research: A guide to design and implementation*. San Francisco, CA: John Wiley & Sons.

Monk, D. H. (1994). Subject area preparation of secondary mathematics and science teachers and student achievement. *Economics of Education Review*, *13*(2), 125–145. doi:10.1016/0272-7757(94)90003-5

Ni, Y., & Zhou, Y.-D. (2005). Teaching and learning fraction and rational numbers: The origins and implications of whole number bias. *Educational Psychologist*, *40*(1), 27–52. doi:10.120715326985ep4001_3

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, *18*(2), 351–380. doi:10.100710639-012-9240-x

Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher*, *15*(2), 4–14. doi:10.3102/0013189X015002004

Stake, R. E. (2003). Case studies. In N. K. Denzin & Y. S. Lincoln (Eds.), *Strategies of qualitative inquiry* (2nd ed., pp. 134–164). Thousand Oaks, CA: Sage.

Strauss, A., & Corbin, J. (1990). *Basics of qualitative research: Grounded theory procedures and techniques*. Newbury Park, CA: Sage.

Thom, J., D'Amour, L., & Preciado Babb, A. P. (2015). Spatial knowing, doing, and being. In B. Davis et al. (Ed.), *Spatial reasoning in the early years: Principles, assertions, and speculations* (pp. 63–82). New York, NY: Routledge.

Weintrop, D., Beheshti, E., Horn, E., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and sciences classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. doi:10.100710956-015-9581-5

Yin, R. K. (2015). *Case study research and applications: Designs and methods* (6th ed.). Thousand Oaks, CA: Sage.

ADDITIONAL READING

Akcaoglu, M. (2014). Learning problem-solving through making games at the game design and learning summer program. *Educational Technology Research and Development*, *62*(5), 583–600. doi:10.100711423-014-9347-4

Boyd, S., & Vandenberghe, L. (2018). *Introduction to applied linear algebra – Vectors, matrices, and least squares*. Cambridge, UK: Cambridge University Press. Retrieved from http://web.stanford.edu/~boyd/vmls/

Calao, L., Correa, J., Moreno-León, H., & Robles, G. (2015). Developing mathematical thinking with Scratch: An experiment with 6th grade students. In G. Conole, T. Klobučar, C. Rensing, J. Konert, & E. Lavoué (Eds.), *Design for teaching and learning in a networked world* (pp. 17–27). New York, NY: Springer, Cham. doi:10.1007/978-3-319-24258-3_2

Casselman, B. (n.d.). Geometrical symmetry and the fine structure of regular polyhedral. Retrieved from https://www.math.ubc.ca/~cass/graphics/manual/pdf/symmetry.pdf

Davis, B.Spatial Reasoning Study Group. (2015). *Spatial reasoning in the early years: Principles, assertions, and speculations*. New York, NY: Routledge.

de Freitas, E., & Sinclair, N. (2014). *Mathematics and the body: Material entanglements in the classroom*. Cambridge, UK: Cambridge University Press. doi:10.1017/CBO9781139600378

Dubinsky, E. (1999). Mathematical reasoning: Analogies, metaphors, and images. *Notices of the American Mathematical Society*, *46*(5), 555–559.

Edwards, L. (1995). Microworlds as representations. In A. A. diSessa, C. Hoyles, & R. Noss (Eds.), *Computer and exploratory learning* (pp. 127–154). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-642-57799-4_8

Falconer, K. (2013). *Fractals: A very short introduction*. New York, NY: Oxford University Press. doi:10.1093/actrade/9780199675982.001.0001

Fauconnier, G., & Turner, M. (2008). *The way we think: Conceptual blending and the mind's hidden complexities*. New York, NY: Basic Books.

Frame, M. L., & Mandelbrot, B. B. (2002). *Fractals, graphics, and mathematics education (Mathematical Association of America Notes, Book 58)*. Washington, DC: Mathematical Association of America.

Gerofsky, S., & Goble, J. (2007). A conversation on embodiment at the heart of abstraction in mathematics education and music education. *Journal of the Canadian Association for Curriculum Studies*, *5*(1), 49–68.

Goldman, R. (2009). *An integrated introduction to computer graphics and geometric modeling*. New York, NY: CRC Press. doi:10.1201/9781439803356

Harrison, H., Birks, M., Franklin, R., & Mills, J. (2017). Case study research: Foundations and methodological orientations. *Forum Qualitative Sozialforschung [Forum: Qualitative Social Research]*, *18*(1). doi:10.17169/fqs-18.1.2655

Ke, F. (2016). Designing and integrating purposeful learning in game play: A systematic review. *Educational Technology Research and Development*, *64*(2), 219–244. doi:10.100711423-015-9418-1

Laborde, C. (1990). Language and mathematics. In J. Kilpatrick & P. Nesher (Eds.), *Mathematics cognition* (pp. 4–19). Cambridge, UK: Cambridge University Press. doi:10.1017/CBO9781139013499.005

Lakoff, G., & Johnson, M. (1999). *Philosophy in the flesh: The embodied mind and its challenge to Western thought*. New York, NY: Basic Books.

Landy, D., Trninic, D., Soylu, F., Kehoe, J., & Fishwick, P. (2014). The implications of embodiment for Mathematics and Computing Education. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (p. 36). Academic Press. Retrieved from https://escholarship.org/uc/item/5hs6g3rz

Moreno-León, J., & Robles, G. (2016). Code to learn with Scratch? In *Proceedings of the 2016 IEEE Global Engineering Education Conference (EDUCON 2016)* (pp. 150–156). Institute of Electrical and Electronics Engineers. 10.1109/EDUCON.2016.7474546

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

Papert, S. (1991). Situating constructionism. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 1–12). Westport, CT: Ablex.

Pickover, C. (1990). *Computers, pattern, chaos, and beauty: Graphics from an unseen world*. New York, NY: St. Martin's Press.

Presmeg, N. (1992). Prototypes, metaphors, metonymies and imaginative rationality in high school mathematics. *Educational Studies in Mathematics*, *23*(6), 595–610. doi:10.1007/BF00540062

Presmeg, N. (2005). Metaphor and metonymy in processes of semiosis in mathematics education. In M. H. G. Hoffmann, J. Lenhard, & F. Seeger (Eds.), *Activity and signs — Grounding mathematics education* (pp. 105–115). New York, NY: Springer. doi:10.1007/0-387-24270-8_10

Schiralli, M., & Sinclair, N. (2003). A constructive response to "Where mathematics comes From." *Educational Studies in Mathematics*, *52*(1), 79–91. doi:10.1023/A:1023673520853

Sengupta, P., Krishnan, G., Wright, M., & Ghassoul, C. (2015). Mathematical machines and integrated stem: An intersubjective constructionist approach. In S. Zvacek, M. Restivo, J. Uhomoibhi, & M. Helfert (Eds.), *Computer supported education. CSEDU 2014* (pp. 272–288). Cham, Switzerland: Springer. doi:10.1007/978-3-319-25768-6_18

Varela, F., Thompson, E., & Rosch, E. (1991). *The embodied mind: Cognitive science and human experience*. Cambridge, MA: MIT Press. doi:10.7551/mitpress/6730.001.0001

KEY TERMS AND DEFINITIONS

Axial Coding: A process of qualitative data analysis in which previously identified categories and themes are connected together theoretically into broader categories.

Case Study: An umbrella of approaches to research that have in common their focus on a phenomenon bounded by a specific unit of analysis and defined by contextual elements, such as space, time, and demography.

Computational Thinking: An analytic approach to problem solving, designing systems, and understanding human behaviours, including practices such as abstraction, decomposition, prediction, problem representation, simulation and verification.

Embodiment: A perspective on learning that considers the role of the body from different perspectives, including biology, culture, and language.

Geometry for Computer Graphics: Refers to the relationships between geometry and programing, in particular with the manipulation of virtual images. This includes programing for robotics as spatial elements have to be considered.

Open Coding: A method for qualitative data analysis consisting of segmentation of pieces of data and the assignment of labels or codes to each one. This is usually done as a first step in data analysis and can be informed by a previous framework; it can also emerge during the analysis (Charmaz, 2006).

Recursion: A mathematical tool that allows algorithms to be defined iteratively. This tool is commonly used for programing.

Triangulation: A form of validation for qualitative research consisting of contrasting multiple forms of data against each other to support the validity of the findings.