

# Geometric Coding Environment: Integration of Computational Thinking and Mathematics in Primary School

Nathalie Sinclair and Victoria Guyevskey  
Simon Fraser University

## ABSTRACT

The purpose of this study was to research the effectiveness of geometric coding environment. We conducted a two-month classroom intervention with grade 2/3 students, experimenting with computational thinking (CT) tasks, centrally involving geometry and spatial reasoning within a dynamic geometry environment (DGE). In our experiments, we were interested in how DGE/CT approach gives rise to concepts relevant to both mathematics and programming. We wanted to see (1) what the students would learn, and (2) what the effect of combining CT and geometry would be (how the concepts would change). Our finding is that, in addition to supporting the teaching of geometry, the DGE-based tasks effectively support teaching of many CT concepts, which seems to provide sufficient ground for Math/CT interdisciplinary integration.

## REFERENCES

- Bartolini Bussi, M.G. & Boni, M. (2003). Instruments for semiotic mediation in primary school classrooms. *For the Learning of Mathematics*, 23 (2), 12-19.
- Gadanidis, G. (2017). Computer coding in the K-8 Mathematics curriculum. In *What Works? Research into Practice*. ON: Queen's Printer for Ontario.
- Jackiw, N. (2012). *The geometer's sketchpad* [Computer software]. Berkeley, CA: Key Curriculum Press.
- Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I.K., Somanath, S., Weber, J., & Yiu, C. (2017). A pedagogical framework for computational thinking. *Digital Experiences in Mathematics Education*, 3(2), 154-171.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Sinclair, N., & Patterson, M. (in press). The dynamic geometrization of computer programming. *Mathematical Thinking and Learning*.
- Sinclair, N., Cirillo, M., & de Villiers, M. (2016). The learning and teaching of geometry. In J. Cai (Ed.), *Compendium for Research in Mathematics Education* (pp.457-489). Reston, VA: NCTM
- Sinclair, N., & Bruce, C. (2014). Spatial reasoning for young learners. In Liljedahl, P., Nicol, C., Oesterle, S., & Allan, D. (Eds.). *Proceedings of the 38th Conference of the International Group for the Psychology of Mathematics Education, Vol1* (pp. 173-203). Vancouver, Canada: PME.
- Sinclair, N., & Moss, J. (2012). The more it changes, the more it becomes the same: The development of the routine of shape identification in dynamic geometry environments. *International Journal of Education Research*, 51-52, 28-44.
- Sinclair, N., & Jackiw, N. (2009). Modelling Practices in Sketchpad. *International Community on Teaching Modelling and Applications*. New York: Springer
- Wing, J. (2006). Computational thinking and thinking about computing. *Communications of the ACM*, 49, 33-35.

## CONTACT

Victoria Guyevskey  
Simon Fraser University  
Victoria\_Guyevskey@sfu.ca  
+1 (778) 782-3395  
www.sfu.ca/education.html

## INTRODUCTION

While geometry has been an academic subject since Plato introduced his Quadrivium of Mathematical Sciences at the Athenian Academy in 387 B.C.E. (Campbell, 2004), the idea of computational thinking (CT) as a curriculum focus did not emerge until very recently, even though it is not brand new: in 1980, Seymour Papert (1980) used this term for the Logo programming language, which he argued could help improve the way children think and solve problems. More recently, CT is seen as a set of cognitive skills and problem-solving processes that Wing (2006) envisioned as part of every child's education. In the new BC curriculum, a CT module has been added for all middle grades. Gadanidis (2017) notes that "the trend of adding some form of computer coding to curriculum is an international phenomenon" (p.1). Kotsopoulos et al. (2017) emphasize that England, Finland, Estonia, and USA have all mandated CT curriculum. Our approach was to use dynamic geometry as our programming language, as a way to better promote students' geometric thinking and communicating (Sinclair, in press).

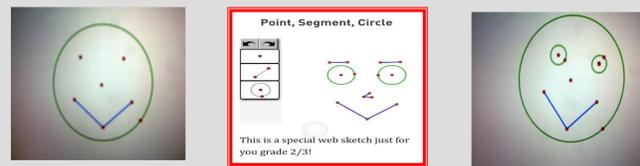
## METHODOLOGY AND RESEARCH QUESTIONS

This research project was carried out in an urban elementary school in British Columbia. We aimed to design tasks that would combine CT and geometry. We wanted to see (1) what the students would learn, and (2) what the effect of combining domains would be on how the concepts emerge.

We conducted a two-month classroom intervention with grade 2/3 students. In our lessons, we opted for a combination of whole-class Smartboard discussions, followed by students working in pairs on iPads. Each lesson was documented through note-taking, photographs, and examples of students' work, both in static and dynamic formats.

We conducted a series of seven lessons, each about hour-and-a-half in length, one lesson per week. We used *The Geometer's Sketchpad* – a dynamic geometry software found to facilitate mathematical modelling (Sinclair & Jackiw, 2009). We also used custom-built "Point, Segment, Circle" online web sketch (Sinclair, 2017) from the Geometry for Young Learners website to be used as exploration tool during this research project, along with a variety of other web sketches from the above-mentioned website.

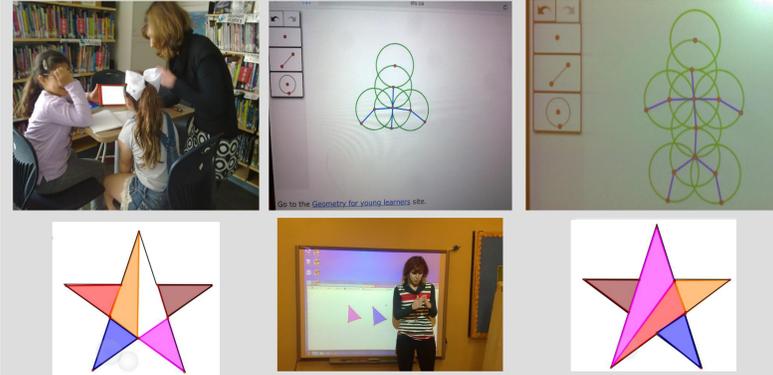
Sixteen students were interviewed afterwards, in order to enable researchers to gain understanding into the coding process that students undertook, and into the evolution of geometric concepts.



## RESULTS

Since the first step in using Sketchpad to build a model involves the circle tool (or construct parallel and perpendicular lines), our initial goal was to introduce students to the this tool. They used it to build a model of a stickman that "works": the stickman had to have arms and legs of equal length and be able to withstand the drag test, meaning his limbs would remain of the original length and attached to the body. Thus, students would first be developing a procedure (building a stickman), then testing the procedure (dragging test), and finally correcting (debugging) the procedure if it didn't work.

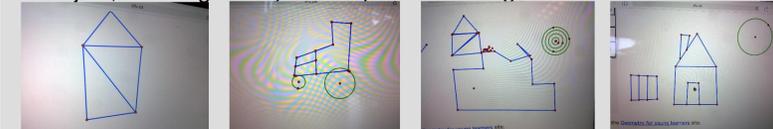
Participants have never worked with Sketchpad before, so the first couple of lessons were introductory, with students exploring the behaviour of triangles with pre-constructed sketches.



One of the tasks was to create their own design and copy them using pencil and paper. Students had more difficulty drawing the designs than actually making them:



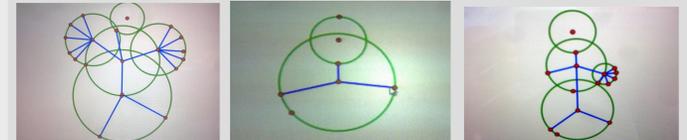
During the third session, a special web sketch that contained only three tools – segment, circle, point - was introduced, and students were encouraged to construct triangles from scratch and then assemble them into a house. Several students succeeded, although the houses came out somewhat crooked, and some would not pass the dragging test. This task was the first instance of developing a procedure, and carrying it out afterwards. In another task of making own designs, where students were asked to use each of the tools at least once, many used the circle tool as an object, focusing mainly on the points and segments:



In the fourth session, students were exploring properties of a circle step-by-step. A circle with two radii was constructed on the Smartboard, and two main questions were posed: are these two radii of the same length? How many more radii can we fit into one circle? At first, students thought the right radius was longer than the left one, but after a quick demonstration that it can move and approach the other radius, the students came to the consensus that they were the same length. The question about the potential number of radii caused hesitation, so students were invited to participate in a role play, where everyone was a point on a circumference, and at the exactly the same distance from one of the students who embodied the centre of the circle. As the circle became tight and another person tried to join in, one of the students protested: "There is no more space left". The researchers suggested to make the circle bigger by stepping back, which seems to have caused a shift in understanding, and to the question "how many radii can fit in a circle", students answered: "ten", "twenty", "a bunch", with someone adding: "it will take us three hundred years to find them all!"

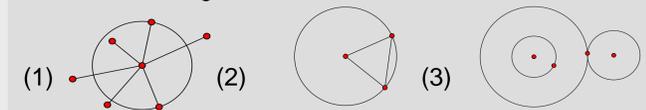


To see whether the students could apply their new understanding of the radii being the same length, we asked the class to make a person with two equal arms. Many students found this task challenging. Centres were off and everything was moving around. This was the second opportunity to make a procedure:



Students then were asked to create a stickman on their iPads that could grow larger or smaller. Task proved to be difficult for the same reason as before – the construction would often not stay together, so many adjustments had to be made to the design, meaning the procedure had to be corrected. This task was very rich in its potential to build fluency in such CT processes as abstraction, algorithmic thinking, decomposition, and debugging: before students could build a stickman, they had to create an algorithm to build two segments of equal length, iterate it to build arms, legs, and fingers, combine parts into a whole step by step, perform a dragging test, and debug if the design fell apart.

The interviews portion related to the circle tool, featured the following questions: (1) Which of these line segments are the same length? How do you know? (Fig.1); (2) Does this triangle have any equal sides? How do you know?(Fig.2); (3) What does the circle tool let you do? Can you use it to make this? (Fig.3); (4) Can you make two segments of the same size, without measuring them?



## DISCUSSION

Each geometric task that we presented to students was both geometrically and CT-rich. In addition to exploring geometric shapes, it required abstraction, automation, and analysis, using terminology introduced by Papert (1980). In the series of warm-up tasks, students were offered a particular design for examination (a square filled with triangles), and they had to extrapolate from that design an abstract procedure of how to fill any shape with any number of triangles. Then, in automation phase, students were creating an algorithm that would help solve any problem of the same type (e.g. drag triangles into the confinement of a shape, then adjust the size of the triangles by dragging vertices until the shape gets filled). Finally, in analysis phase, students were testing out their solution and debugging if necessary (e.g. realizing that there is no space left to fit the last triangle, and either re-doing the design, or shrinking it almost into a line to create an illusion of fitness). Using the Computational Thinking Pedagogical Framework proposed by Kotsopoulos et al. (2017), our two warm-up lessons featured *tinkering* and *making*, as students were exploring dynamic triangle, and the remaining lessons involved *making*, *remixing*, and *debugging*, when constructing congruent segments with a circle tool, and making more complex designs. Our finding is that, in addition to supporting the teaching of geometry, the DGE-based tasks effectively support teaching of many CT concepts, which seems to provide sufficient ground for Math/CT interdisciplinary integration.