LUMAT
INTERNATIONAL JOURNAL ON MATH, SCIENCE AND TECHNOLOGY EDUCATION

Research article

Primary school students' problem-solving strategies in creating artworks with GeoGebra: Integrating computational thinking skills into mathematics and visual arts lessons

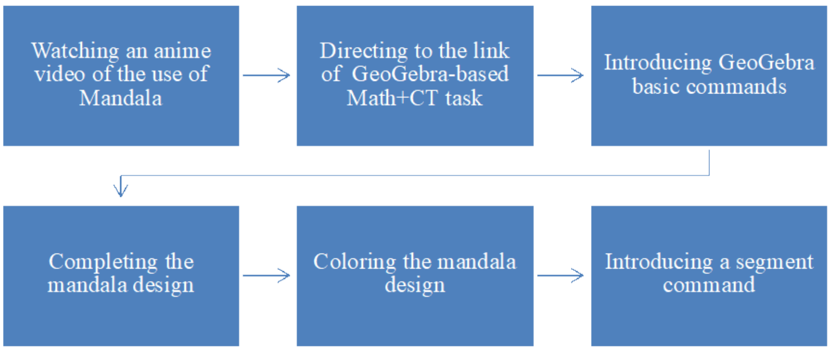
Wahid Yunianto[1], Daniel Jarvis[2], Zsolt Lavicza[1], Zetra Hainul Putra[3] and Shereen El-Bedewy[1]

[1] Johannes Kepler University Linz, Austria
[2] Nipissing University, Canada
[3] University of Riau, Indonesia

Abstract: Computational thinking (CT) as a problem-solving skill has been argued to be an essential skill for all learners. Accordingly, there have been efforts to formalize and operationalize CT within school curricula in various countries. In primary schools, students often develop CT through unplugged activities and visual programming activities. However, in this study, we investigated the use of mathematical software with which students typed in commands (codes) to construct artistic artifacts. Educational Design Research (EDR) has guided the development of our task. We attempted to utilize technology to support students' problem-solving skills and creativity by developing a GeoGebra-based Math+CT task infusing arts. Fifteen Grade 5 primary school students worked on a task to construct a mandala (Hinduism-Buddhism sacred geometrical figures) involving mathematical concepts. Data, in the form of students' GeoGebra (i.e., "ggb") files and screen video recordings, were collected and then analyzed using a content analysis method. Findings revealed that our designed task had promoted students' different problem-solving strategies while working with technology. Additionally, most students did not encounter serious problems in working with GeoGebra commands, and students' computational thinking skills were supported as a result of engagement with our activities.

Keywords: computational thinking, mathematics, textual programming, GeoGebra, visual arts

Correspondence: yunianto.wah@gmail.com



Received 23 November 2024 / Accepted 23 May 2025 / Published 13 June 2025
Published by the University of Helsinki, Finland and LUMA Centre Finland.

1 Introduction

Wing (2006) defined computational thinking (CT) as problem-solving skills used to develop solutions, with or without the help of an information-processing agent. Furthermore, CT is regarded as an essential skill for all learners (Wing, 2006), and some countries and institutions have thus attempted to formalize and operationalize CT in their school curricula (Bocconi et al., 2016, 2022). Since this skill is argued to be relevant and essential for 21st century living, it must therefore be explored deeply and continually throughout the different levels of one's formal education (Mohaghegh & Mccauley, 2016).

Bocconi et al. (2016) have highlighted several European countries in which education policymakers have tried to incorporate CT as a compulsory element in primary school learning. This is in line with what Wing (2006) argued that CT should be taught as early as possible. A recent study by Ye et al. (2023) found that research on the integration of CT in primary schools, especially in mathematics lessons, constituted a significant portion of current studies. Additionally, Chan et al. (2022) in their literature studies have provided some insights on how tools and approaches have been utilized, which could help to consider the suitable methods for primary school students. However, Nordby et al. (2022) with their literature review found the necessity to focus on a process-oriented approach, and this approach needs more investigations. We propose that a process-oriented approach allows researchers to operate in either a plugged or unplugged mode of delivery.

Polat and Yilmaz (2022) compared primary school students engaging with plugged (i.e., digital, software-based) activities versus comparable unplugged (non-digital-based) CT activities and found that academic achievement differed among groups, with higher results achieved by those students who participated in the unplugged activities. They further concluded that unplugged and plugged activities can both enhance students' CT skills and academic achievement. However, working with plugged activities, especially textual programming, can involve unique challenges for both teachers and young learners (Resnick, 2012; Resnick et al., 2009), which underscores the need of carefully selecting and designing activities appropriate for primary school students.

Using both plugged and unplugged delivery mode methods, teachers could integrate CT into various school subjects (Wing, 2006), specifically within mathematics lessons. Ye et al. (2023) conducted a systematic literature review on the integration of CT in mathematics education and reported that CT and mathematics can be successfully co-developed. Based on this finding, we developed Math+CT lessons to support students' CT development while learning mathematics (Yunianto, Sami El-Kasti, et al., 2024). The lessons were first implemented with junior high school students (aged 12–15 years), incorporating GeoGebra commands to facilitate the construction of visual arts and develop creativity (Yunianto, Cahyono, et al., 2024).

Encouraged by these results, we adapted one of Math+CT tasks for primary school students, hypothesizing that this task could support their problem-solving skills, creativity, and CT development. GeoGebra offers a dual role in this context—not only as an instructional tool but also as a means model and to visualize mathematical ideas

(Ziatdinov & Valles, 2022). We were intrigued to know the results of primary school students working on our Math+CT task to construct a visual art. The following research questions guide our investigation:

- RQ1: How do primary school students approach the plugged CT activities utilizing GeoGebra commands for the first time?
- RQ2: How do CT skills emerge in a Math+CT task involving primary school students?

2 Literature review

Initially, Papert (1980) introduced the term "computational thinking" (CT) as a way to access knowledge by interacting with computers. Several decades later, Wing (2006) redefined CT as problem-solving skills that are required to formulate solutions, with or without the help of an information-processing agent. Wing's (2006) CT definition is frequently cited by researchers (Irawan et al., 2024). However, Broley et al. (2023) argued that scholars might neglect the essence of the initial CT definition by Papert (1980). Therefore, Lodi & Martini (2021) proposed that researchers maintain core ideas from both Papert's and Wing's CT definitions.

In line with Wing (2006), Grover and Pea (2013) also formulated CT as a mental process to support students to formulate and express solutions that are executable by computational methods. This definition is also in line with the problem-solving conceptualization presented by Polya (1945), and by Schoenfeld (1985), to solve problems in a systematic manner. A recent study by Wu et al. (2024) revealed that CT components are employed in problem-solving. Moreover, Maharani et al. (2019) have noted how CT components support Polya's problem-solving steps (i.e., 1. Understanding the Problem, 2. Devising a Plan, 3 Carrying Out the Plan, and 4. Looking Back) within a mathematics lesson. It seems it is not surprising, since Sneider et al. (2014) had previously presented the intersection of CT and mathematical thinking, which share some common skills such as problem-solving. CT components that support problem-solving can be carried out with or without digital tools (see Wu et al., 2024). Given our working definition of CT as 'problem-solving within a digital environment,' we further examine how problem-solving is specifically supported with technology.

The use of technology has influenced students' problem-solving skills (Alsarayreh, 2023). Olsher et al. (2023) have pointed out that problem-solving has been used with technology specifically for the purpose of conjecturing. They utilized dynamic geometry software (DGS) for fostering students' conjecturing skills by investigating draggable points on a quadrilateral that formed a unique arrangement. Sinclair (2001) investigated the color calculator for students to manipulate fractions represented in a grid. This activity was shown to promote problem-solving skills and to encourage students to pose questions.

Similarly, Dakers (2024) argues for a form of problem-solving in which the solution does not yet exist so that students become more fully engaged, creative, and experimental.

Based on literature review, and for the sake of consistency throughout our study, we have considered CT as involving both the problem-solving skills that are required to formulate solutions—with or without the assistance of information-processing agents—as well as the accessing of knowledge through interaction with processing agents. Furthermore, a CT definition needs to be operationalized in order to make it more fully implementable within school settings. Several scholars have described this operationalization in terms of specific CT components relating to their own work (Bocconi et al., 2016). For example, Brennan and Resnick (2012) proposed the following CT components: computational concepts, computational practices, and computational perspectives. Zhong et al. (2016) and Shute et al. (2017) further adapted the Brennan and Resnick (2012) CT framework by adding flexibility to the grade levels and for non-computer science subjects. Another CT framework for science and mathematics education by Weintrop et al. (2016) was built upon by Shute et al. (2017). Likewise, by following the development of a previous framework, our Math+CT lessons were guided by the CT framework of Shute et al. (2017).

The Shute et al's. (2017) CT framework consists of six facets, namely, decomposition, abstraction, algorithm, debugging, iteration, and generalization. Decomposition deals with decomposing the problem into smaller-related problems to be solved. Abstraction focuses on accessing essential information and eliminating any redundant information. The algorithm requires that one develop steps to solve the problem. Debugging involves the identification of errors within the algorithm and, hence, fixing them. Iteration denotes the process whereby actions need to be repeated in order to arrive at the ideal solution(s). Finally, generalization involves the application of CT skills within different contexts or areas of investigation. This framework was incorporated into the Math+CT lessons by Yunianto, Sami El-Kasti, et al. (2024) and Yunianto, Cahyono, et al. (2024) and their results showed that some students could successfully develop and implement CT skills relating to mathematics concepts.

Integrating CT into mathematics lessons in primary schools has become more predominant in recent years (Ye et al., 2023). This is in line with Wings's (2006) proposal to introduce CT as early as possible to children. Working with young learners, Papert (1980) introduced "Turtle Geometry" for the development of CT skills in schools. He found that certain challenges relating to textual programming, or syntax, were encountered by teachers and students who ultimately felt uncomfortable with the tool. Resnick et al. (2009) developed a block programming software entitled Scratch designed to support student coding via a fun and easy-to-use interface.

The approach where the CT is introduced through computer programming belongs to the plugged CT mode of delivery. This mode uses computers and similar digital devices for students to learn CT skills (Hermans & Aivaloglou, 2017). Using "Turtle Geometry" that has been used by Papert (1980) or similar tools belongs to plugged-mode CT activities. Dakers (2024) defined two different problems for students while engaging technology, namely, problems of embodied applied problem solving and problems of virtual creative problem solving. The latter is like Papert's (1980) idea to make students construct knowledge by interacting with a computer where the interference of the adults is limited. The teaching and learning with problems of virtual creative problem-solving poses challenges to implement (Dakers, 2024) and we believe this approach needs more examples and investigations.

Students can benefit from plugged, unplugged, or a combination of both modes of learning, as we can see from studies conducted by Rijke et al. (2018), van Borkulo et al. (2021), and Chytas et al. (2024). Primary school teachers who supported students' CT skills by using a plugged approach saw similar benefits to students using an unplugged approach (Polat & Yilmaz, 2022). In this paper, we reported on students who engaged with plugged-mode activities in the Math+CT lesson using GeoGebra software to work with textual programming language (i.e., GeoGebra commands).

GeoGebra is free mathematics software that has powerful features for geometry and algebra topics (www.geogebra.org). van Borkulo et al. (2021) and Chytas et al. (2024) have utilized GeoGebra for supporting students' CT skills (i.e., algorithmic thinking and data practices) while learning mathematics. Yunianto, Sami El-Kasti, et al. (2024) also utilized GeoGebra-based Math+CT lessons, which were shown to support students' algorithmic, debugging, and other related skills. These previous studies were conducted with junior and high school students, and thus more research is arguably needed to explore the implementation of such strategies in primary school. In this paper, we investigated the implementation of a GeoGebra-based Math+CT task within a primary school context wherein students experienced a rich integration of mathematics, technology, and visual arts. The use of arts and cultures as contexts in CT has been developed in some previous studies (e.g., Putra et al., 2022), and it has the potential to increase students' engagement in learning CT.

3 Theoretical framework

This study is informed by four related theorical perspectives: constructionism (Papert, 1980), "half-baked" artifacts (Kynigos, 2007), technological problem-solving (Morrison-Love, 2021), and problem-solving in digital environments (Dakers, 2024).

Papert (1980) expanded Piaget's constructivism theory into constructionism, emphasizing learning through creations or constructions. This theory posits that computing tools function as objects-to-think-with (Papert, 1980), allowing students to enhance their ideas by engaging in designs and manipulations. Papert (1980) proposed the constructionism learning theory while developing his "Turtle Geometry" software designed to enhance computational thinking skills. His ideas involved students interacting with a computer to construct artifacts, thereby learning and mastering key computational thinking concepts. Additionally, he proposed three principles in constructionist learning, namely, engagement in the activity, ownership of one's ideas and learning style, and exposure. The engagement principle means that students are actively constructing or reconstructing digital artifacts. The second principle, ownership, means that students use their own ideas and strategies to construct these artifacts. The exposure principle involves students presenting their ideas to their peers, working with others, and demonstrating how they created their objects.

In our lessons, students worked with GeoGebra to construct geometrical objects through various GeoGebra commands. This resembles the constructions of mathematical objects carried out by Papert (1980) with “Turtle Geometry.” Students were shown a brief video regarding the history and use of mandalas, but no sample mandala creations were shared with them prior to the design activity. Thus, with their own unique ideas and imagination, students were directed to create individual mandala designs using GeoGebra commands. The use of different colors, sizes, and number of objects were permitted. Students did not formally present their works to peers, however, they were given the opportunity to walk around at the end of the session to see their peers’ creations.

Kynigos (2007) introduced the concept of a “half-baked artifact,” which served as an additional guiding principle in this investigation. His concept was to allow students to modify and enhance incomplete artifacts in a manner similar to that of engineers. The “half-baked” task is designed to enable students to take ownership of the techniques and concepts associated with the artifacts’ construction (Kynigos, 2015) and eventually refine the artifact, integrating the knowledge they have acquired from previous learning. This approach seems to fall between the two approaches suggested by Dakers (2024), i.e., problems of embodied applied problem-solving and problems involving virtual creative problem-solving.

Morrison-Love (2021) proposed a technological problem-solving (TPS) framework consisting of the forms of the task and the related knowledge resources. In this framework, the task can be categorized as well-defined, ill-defined, troubleshooting, or emergent. The well-defined task depicts a constrained problem with convergent solutions and can be solved through a limited number of strategies within well-defined parameters (Jonassen, 1997). Dakers (2024) also added that a well-defined task often involves pre-existing methods to solve it, which follows the idea of embodied applied problem-solving. In contrast, Jonassen (1997) defined the ill-defined task as a task possessing various solutions and containing uncertainty about which concepts and/or strategies are necessary for the solution. The troubleshooting task involves technical or technological problems relating to the tools or artifacts (McCade, 1990; MacPherson, 1998; Schaafstal et al., 2000). Lastly, the emergent task means that a task is potentially a new/emergent problem while solving it (McCormick, 1994). Additionally, Dakers (2024) argued that this type of task could lead students to virtual creative problem-solving, enabling the creation of something original, new, and never-before-created. Considering these different types of tasks, the mandala design task that was highlighted in this study can arguably be described as a combination of the troubleshooting and emergent forms.

In attempting a task, Morrison-Love (2021) elaborated on the types of knowledge that can be used, namely, conceptual knowledge, procedural (explicit) knowledge, or tacitprocedural (implicit) knowledge. This is also in line with the knowledge required when problem-solving with technology as described by Dakers (2024). Students can use one or more pieces of knowledge in solving the problem with technology. The technological problem-solving construct can thus help us to see how students utilized these forms of knowledge when designing mandalas using GeoGebra commands.

4 Method

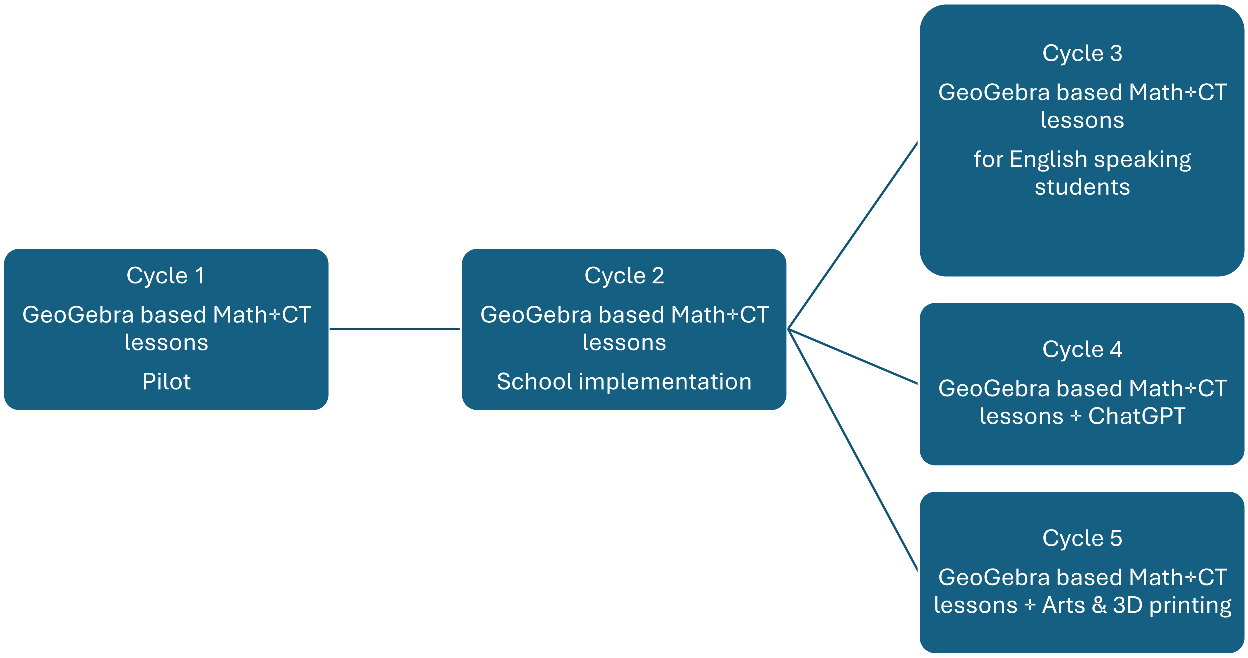
In this section, we describe the research approach that was utilized in our study, the participants, the learning scenario, and the data collection and analysis methods.

4.1 Study design

This study forms part of a larger research program focusing on the development and understanding of the integration of computational thinking (CT) into mathematics lessons. The educational design research (EDR) by McKenney and Reeves (2018) was suitable for our large study as we developed and investigated specific educational innovations. EDR involves iterative cycles of design, enactment, analysis, and reflection, with the aim of both improving educational practice and contributing to theoretical knowledge (Bakker, 2018; McKenney & Reeves, 2018, 2021; van den Akker et al., 2006). The EDR approach has been widely used by several scholars researching at the Freudenthal Institute for innovating the learning of mathematics (FISME, 2015). Guided by this research approach, we investigated the curriculum and literature on integrating CT into mathematics lessons, and later we drafted an initial design of Math+CT lessons consisting of several tasks.

We piloted the lessons and improved them for consecutive implementations. The Math+CT lessons that were used in this study have undergone several iterations (Yunianto, Cahyono, et al., 2024; Yunianto, Sami El-Kasti, et al., 2024), and we specifically selected and adapted one of the tasks by incorporating elements of visual arts and simplifying the coding instructions for the primary school learners (Figure 1). Visual arts elements such as line, shape, and colour, combined with geometrical objects such as lines, circles, and polygons, become intricate parts of the overall mandala designs.

Figure 1. Iterative cycle of related studies employing Educational Design Research (EDR)



Briefly, we will explain the cycles in our study. Cycle 1 began by 1) exploring literature and curriculum, 2) designing and developing the tasks or lessons, and 3) piloting these tasks with a few students. Students (aged 12-15 years old) learned GeoGebra commands relating to the creation of points, angles, several regular polygons, circles, and an inscribed polygon within a circle. This cycle can be found in Yunianto, Sami El-Kasti, et al. (2024). Based on the revision and improvement of the tasks and lessons in Cycle 1, we then conducted Cycle 2 by implementing the revised lessons in several schools with more students (aged 12-15 years old). This cycle featured the same artifacts as in Cycle 1, but we were further able to record students' log attempts. In these revised lessons, we could therefore record all of the GeoGebra commands/codes as inputted or deleted by students. Thus, we could gather more information about students' strategies in solving our tasks. In this cycle, we also focused our analysis on the effects of gender and grades on students' computational thinking performances and learning analytics relating to how students struggled with GeoGebra commands in creating the inscribed polygon in a circle. We carried out the next cycles (Cycles 3-5) in parallel.

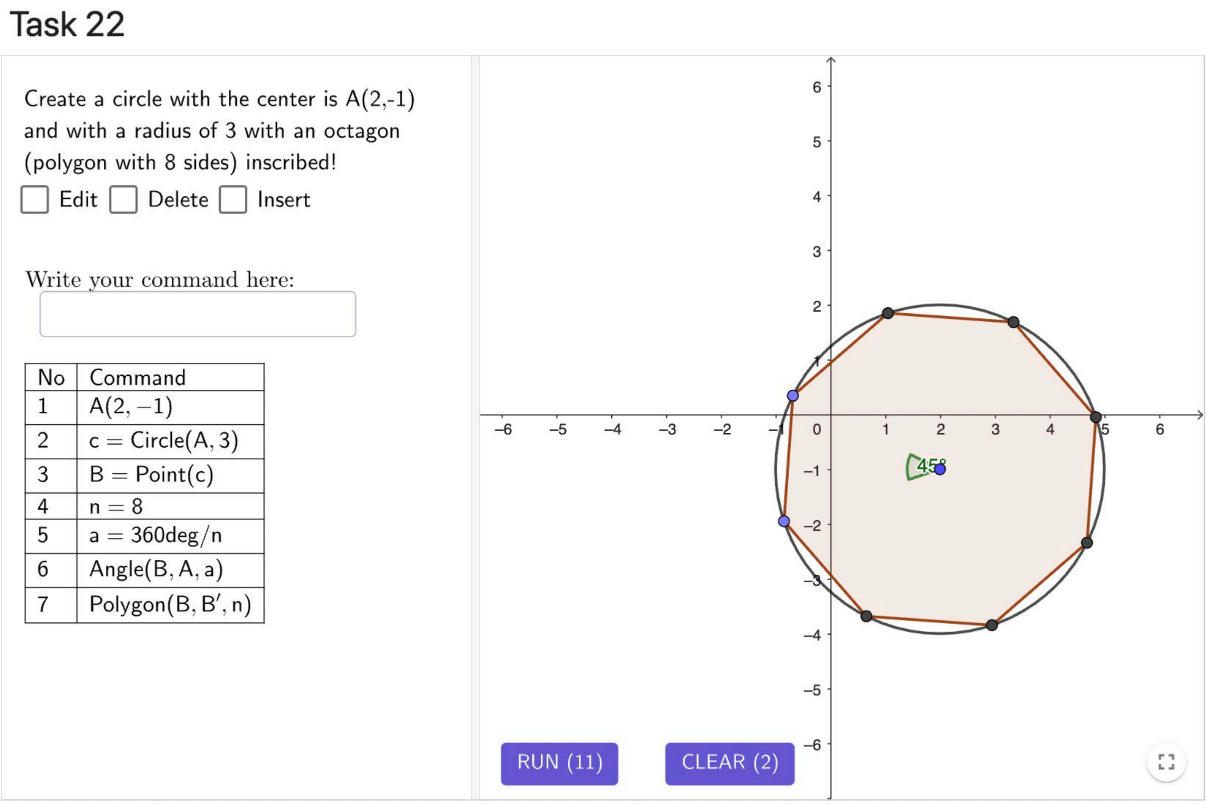
GeoGebra pop-up notifications are available in English, and it seemed to be a hindrance for Indonesian students. Therefore, to understand the effect of English pop-ups on students' learning, in Cycle 3 we implemented the English-translated Math+CT lessons in English-speaking Philippines classrooms (aged 12-15 years old). The artifacts that the students created were the same as those in Cycle 2.

ChatGPT had just been introduced at this time. Considering the importance of this emerging tool, in Cycle 4 we embraced ChatGPT in our Math+CT task. We selected and adapted one of our tasks by asking our participants to create an inscribed hexagon of a certain size and location with the help of ChatGPT. We carried out this cycle with both adults (Yunianto, Galic, et al., 2024) and with Junior High school students (aged 12-13 years old) (Yunianto, Lavicza, et al., 2024). We attempted to see how ChatGPT could support students' CT skills while learning mathematics.

In Cycle 2, we had let students learn mathematical constructions through dedicated assignments and applicable hints. The emphasis on learning to use GeoGebra for these tasks seemed to limit students' creativity. In Cycle 5, with a new group of students (aged 18-19 years old), we carried out Cycle 5 to investigate students' creativity by infusing culture and the arts. We introduced students to a short instructional video on how to use GeoGebra commands to create inscribed polygons, allowing them to pay more attention to the mathematics involved. To challenge their creativity, we also asked them to develop Batik stamp designs and to create them using 3D modelling and printing. For more details on this study, see Yunianto, Cahyono, et al. (2024).

In this paper, we specifically report on a second part of Cycle 5 in which we had somewhat younger students (aged 9-10 years old) likewise explore the GeoGebra commands necessary for creating visual arts designs. As already stated, in Cycles 1 to 4, we mainly focused on the creation of inscribed regular polygons within a circle (Figure 2) but without considering the visual arts aspect. Now, we challenged young students to make these mathematical objects into mandalas. We wondered if or how young students would be able to utilize GeoGebra commands to create these complicated visual artworks. Additionally, how, with limited mastery of GeoGebra commands, students would be able to accomplish the mandala designs while engaging with technological problem-solving.

Figure 2. The inscribed hexagon creation in previous cycles



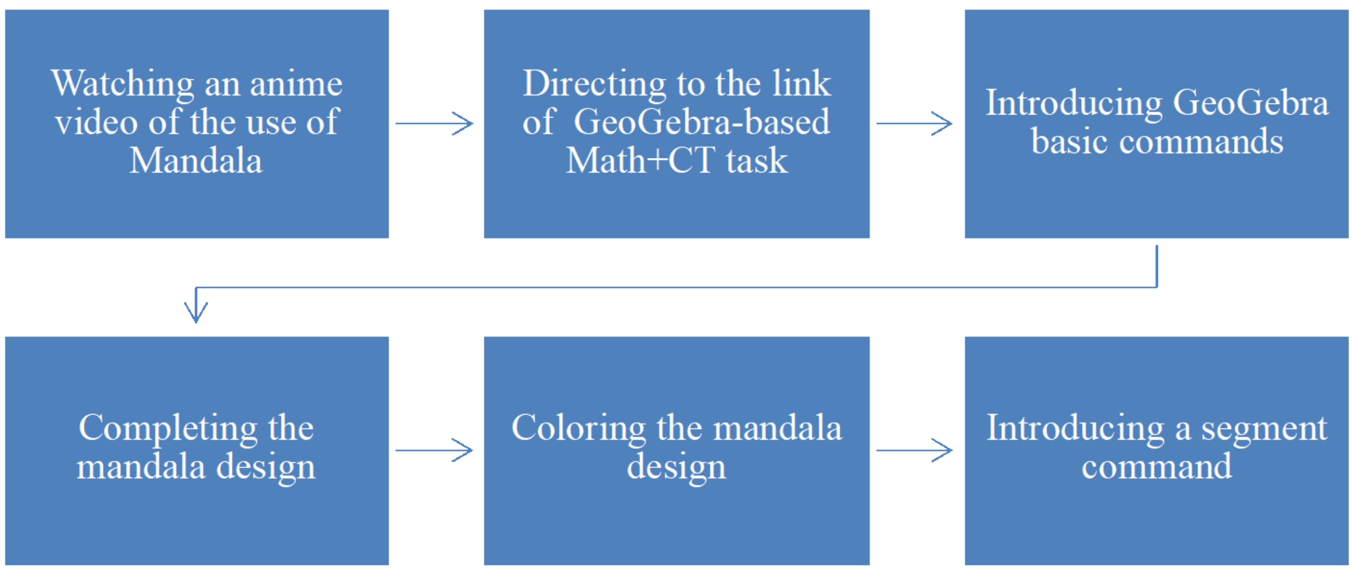
The visual arts aspect was intended to provide students with freedom to create artistic works from the unfinished artifacts. The goal of this study was to explore how primary school students would approach our task and how the CT skills would emerge from their engagement with our task.

4.2 Learning scenario

Learning about circles and their properties through the direct presentation of formulas and with recall-type tasks has led to students' poor performance on this topic (Mifetu, 2023). In Indonesia, this was also common practice to "teach mathematics by telling" (Fauzan, 2002), or direct instruction, and this practice is still commonly observable (Muhammad et al., 2023). To prepare students with good mathematics problem-solving skills, Mifetu (2023) proposed activity-based learning, allowing students to actively engage in constructing circles and improving their problem-solving skills. Similarly, this study also encouraged students to actively construct circles within a digital environment and supported students' problem-solving skills through computational thinking. The following is how we structured the lesson.

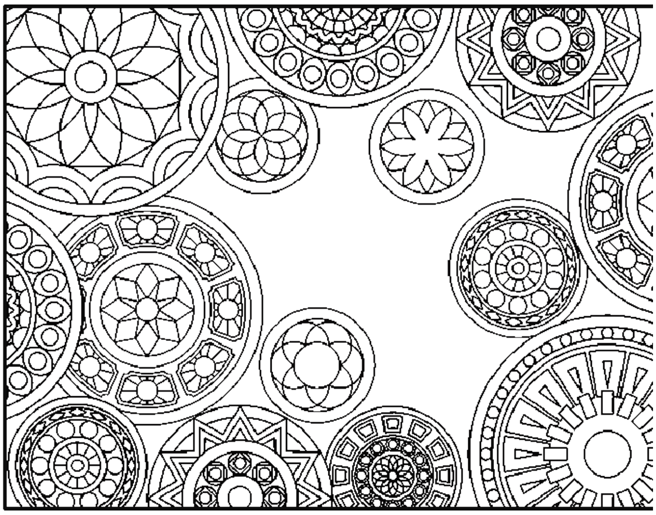
We began by introducing students to an anime video of the use of mandalas, directing students to the GeoGebra-based Math+CT task, introducing basic GeoGebra commands (points and circles) and tools, asking students to continue the mandala design, coloring the objects, and lastly introducing the segment command (a GeoGebra command to create a line segment) (Figure 1).

Figure 3. The learning sequence of GeoGebra-based Math+CT task



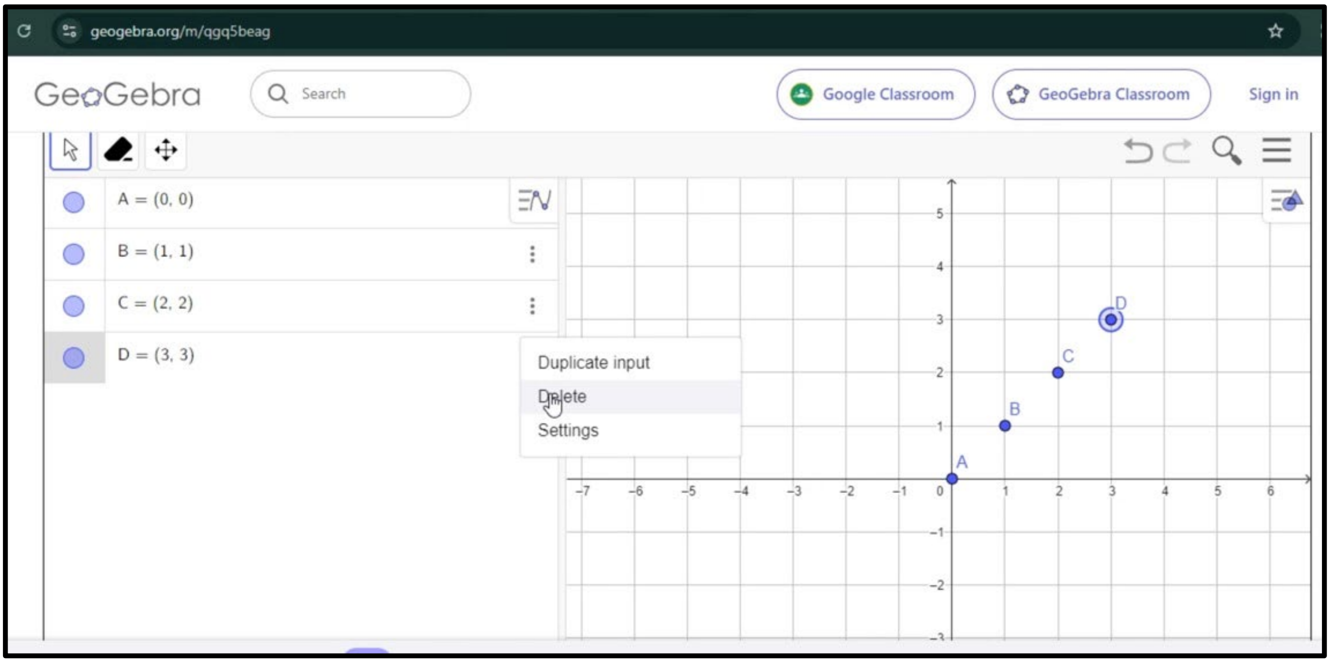
In this paper, we selected and adapted a task from the previous study (Yunianto, Sami El-Kasti, et al., 2024). This task involved the construction of a circle and an understanding of several related concepts, namely, the centre point and the radius. The lesson itself lasted for approximately 1.5 hours. In order to introduce this topic to primary students in an engaging and understandable way, we provided them with a short anime video (https://www.youtube.com/watch?v=HGS2__sck24) that presented the use of circles in the creation of various artworks. We connected this with mandalas (Figure 4), which are sacred geometrical shapes from the Hindu-Buddhism tradition, and which involve circles and polygons (Britannica, 2023). In this way, students encountered clear examples of the use of circles in creating art.

Figure 4. Examples of Mandalas (Free copyright image by Pixabay)



After introducing students to the mandalas, they each worked at a desk with a computer (in rare cases where there was a problem with technology, students worked with a friend). Each computer was connected to the internet, and students were directed to the GeoGebra-based Math+CT task that was available on the GeoGebra website (https://www.geogebra.org/m/qgq5beag). After visiting the webpage, students were instructed by the researcher on how to use GeoGebra commands to create a point. For instance, to create a point A with the Cartesian coordinates (1,1), students were instructed to enter A = (1,1). This was continued by the creation of point B at coordinates (2,2). The researcher then challenged students to create new points C and D using their own selected coordinates. Students created points D and E on the coordinates (3,3) and (4,4), respectively. Afterwards, students were asked to delete all the constructions by using the "Delete" tool for each inputted point (Figure 5).

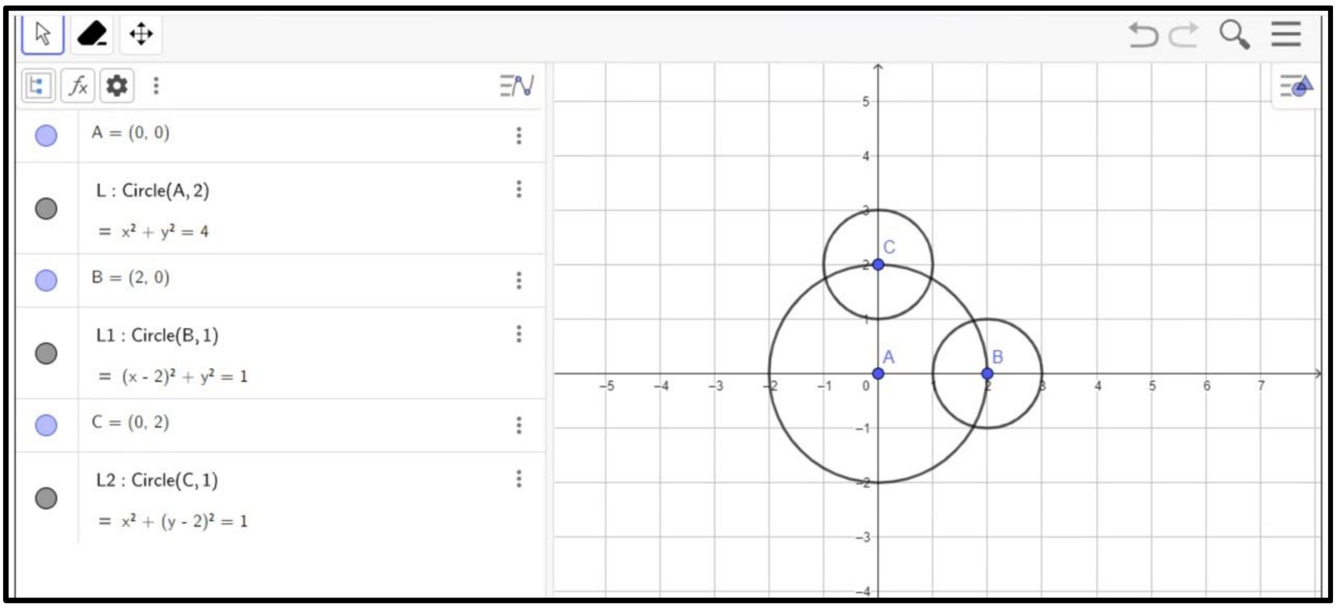
Figure 5. Constructing points using GeoGebra commands and deleting them



After becoming familiarized with the Point command, the researcher presented the GeoGebra command used to construct a circle. First, students were required to create point A at (0,0) and then enter L=Circle(A,5) which created a circle with a centre at (0,0) and a radius of 5 units. Note that "Lingkaran" means "circle" in the Indonesian language; thus, "L" is used to denote a circle construction. To change the size of the radius, students could modify the number of the second coordinate.

Next, students were directed to create a point B on the circle L, and from this new point B they were then asked to construct another circle named "L1." To make it easier, L1, L2, L3, etc. were used for the naming of all consecutive circles (Figure 6).

Figure 6. Constructing more circles



Students were then asked to continue the construction of multiple circle objects within their drawing using the commands highlighted above. After they finished constructing their objects, the researcher instructed students on how to change the color of an object and encouraged them to select and use any combination of colors for their circular designs. Lastly, the researcher introduced the segment command.

4.3 Participants

This study involved a homeroom teacher and primary school students. Fifteen Grade 5 Indonesian students (aged 10-11 years), one month into the school year, participated in this study. The homeroom teacher informed the parents and guardians that there was an additional session under research with their children. If they permitted their children to participate, they could respond on the Parents-Teacher WhatsApp group. Parents' and guardians' consents have been obtained for their children to participate in this study. Therefore, students who attended the sessions have their parents' consents.

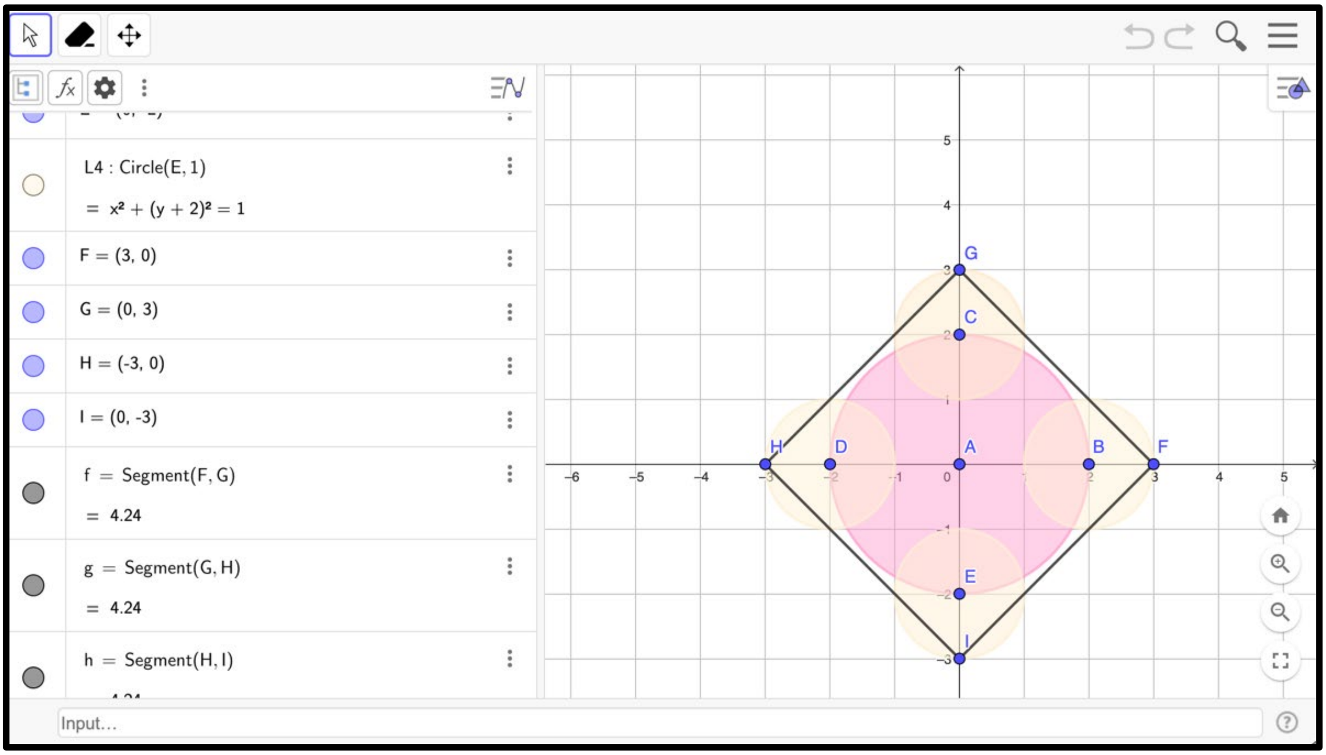
Each student worked on one personal computer (PC) that was located in their school's computer laboratory. Having been introduced to the use of computers since Grade 3, participants were familiar with the keyboard and general functionality of the machines being used in the study. This fact benefited our study insofar as students were easily able to input GeoGebra commands that required certain symbols, such as opened and closed brackets. To type an open bracket, for example, students had to press the 'Shift' button and the '9' button simultaneously.

The researcher (first author) delivered the task and instructions while the homeroom teacher, who was not yet familiar with the GeoGebra software, observed the sessions and also helped the researcher to motivate students to share their thoughts during the CT activity.

4.3 Data collection and analysis

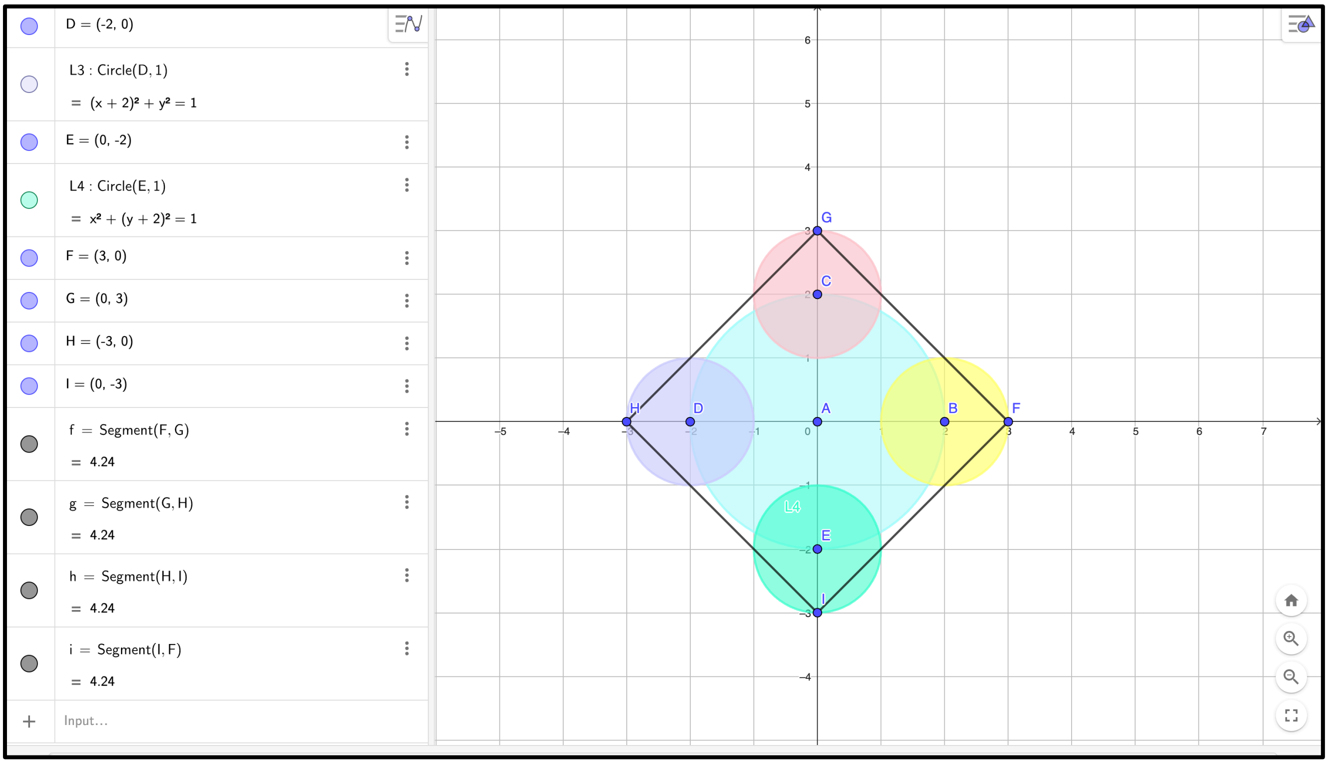
We collected data (i.e., ".ggb" files) from all of the students' work on the GeoGebra task (Figure 7), including three video-screen recordings (Screen 1, Screen 2, and Screen 3) from three different students (Student A:male, StudentB:male, and StudentC:female). The selection of the three students was random as we first set up on the computers that could be installed with screen video recording add-ons. The start buttons were clicked to record the students' screens before they worked on the task. These videos offered real-time insights into students' task engagement and utilization of GeoGebra. The video recordings were particularly beneficial for capturing non-verbal communication, problem-solving strategies, and events where students encountered problems in completing the task.

Figure 7. One of the students' works in the form of a GeoGebra (ggb) file



From the students' GeoGebra files and screen-recording videos, the researchers analyzed the data using the content analysis method proposed by Krippendorff (2004). This particular content analysis method belongs to directed content analysis, referring to Hsieh and Shannon's work (2005), as our approach featured the analysis of results from an existing framework or theory. This content analysis involved three steps, namely 1. making, 2. categorizing, and 3. concluding the codes from the ggb files and screen video recordings. Step 1 involved the familiarization with data in which each ggb file would be identified and categorized based on the emerging traits or characteristics of the students' unique creations. For example, it might be the case that the creations were distinctly based on the colors used or the sizes of the constructed shapes. Figure 8 shows an example of student work that uses different and relatively bright colors for each circle within the overall design. Then we assigned initial codes for the numbers of objects, commands, and colors, such as SC for small circle, BG for big circle, GC for GeoGebra commands, and TCol for the total color. The particular CT facet that emerged (i.e., StrugBut, DebgDel, DebgRev, NAlgo, Algo, and PR) was also coded. The colors can be coded or categorized into primary colors (P), secondary colors (S), and tertiary colors (T). In Step 2, we quantified the objects and categorized them. We categorized the colors used for the big circle only because it used only one color. We also categorized if the line segments were used, or not, to connect the small circles. We also combined some codes into the same category, such as placing both DebgEdt and DebgRev under the category of Debugging Facet; NAlgo and Algo both under the category of Algorithm; and PR under the category of Abstraction. In Step 3, we communicated these codes related to participants' strategies and involvement of CT skills and connected them with relevant studies in the Results and Discussion sections.

Figure 8. A student uses distinct colors for the circles



Researchers were curious to know if students would use different approaches in creating their mandalas. From the screen recording video footage, we would specifically focus on the various elements of computational thinking found within the framework proposed by Shute et al. (2017).

5 Results

This section presents the results of the analysis of the students' artifacts and video recordings, focusing on how primary school students approached the plugged CT activities involving GeoGebra and how computational thinking (CT) skills emerged during the task. The findings are organized around the two primary research questions.

5.1 Students' strategies

We began to provide the analysis of students' artifacts derived from ggb files regarding the number of colors and commands. Table 1 presents the descriptions of students' creations in terms of the number of the big circle (BC), small circle (SC), number of GeoGebra commands (GC), and total colors used (TCol). Most students used eighteen GeoGebra commands and created five circles (one big circle and four smaller circles). Further, students tended to use at least two different colors within their geometric creations. It seems that the way students colored the objects is unique.

Table 1. Descriptions of students' creations

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
Big Circle	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Small Circle	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
GeoGebra Command	18	18	18	18	15	18	18	18	18	18	18	18	18	18	18
Total Color	2	5	5	3	5	5	5	2	2	5	2	5	5	5	5

All students constructed only one big circle in the centre of their mandalas (Table 1). Additionally, all students had the big circle size with a radius of 2 units and the smaller circle size with a radius of 1 unit (Figure 9). It seems that students' creations are uniform in terms of the number of objects. However, we could not find the same final artifacts from students with the same colors. It seems that in our teaching, we did not really encourage students to create more objects, or we did not deliver the message to students that they could create as many objects as possible. It might result in various objects and the number of commands if we have done so. Therefore, this finding could help us in future implementations.

Figure 9. The student used distinct colors for the circles

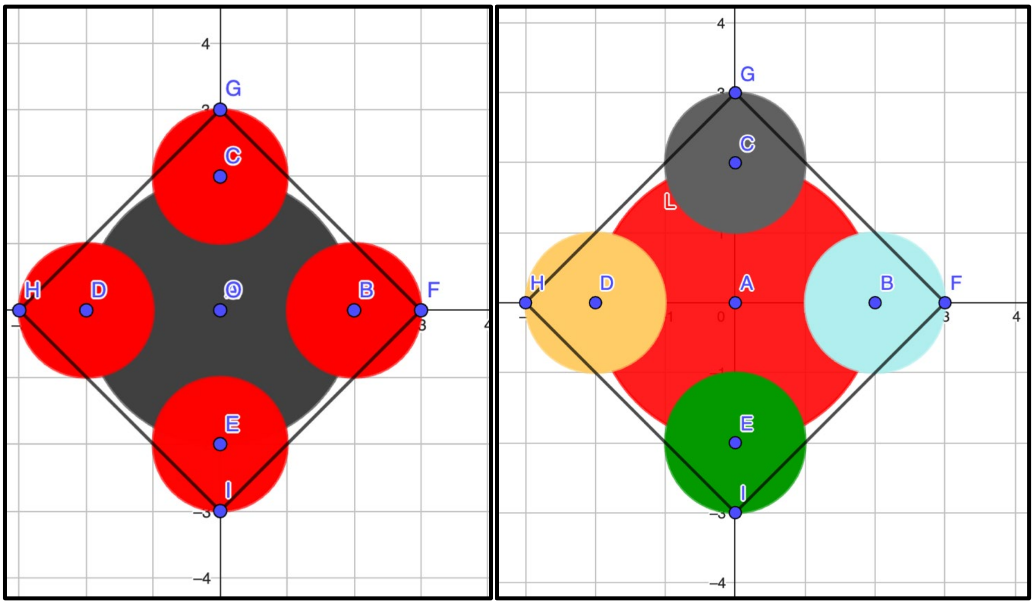


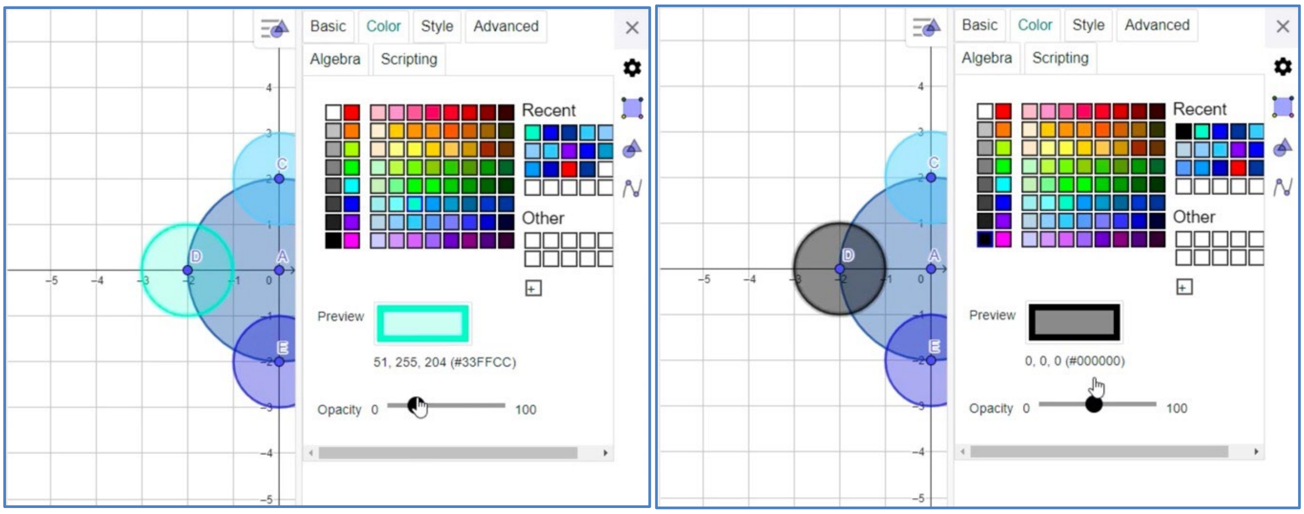
Figure 9 depicts two students' creations and demonstrates how they each chose to color their mandalas. Some students used a dark color for the big circle and a lighter color for the smaller circles (Table 2), while others did the opposite. Some students varied the colors of the small circles using four different colors. Students primarily connected the smaller circles using line segments. Most students connected the smaller circles from points that intersected the x-axis and y-axis, and only rarely did students connect circles using line segments from other parts of the small circles (Table 2).

Table 2. The colors and the segments used by students

Student	Central Circle's Color	Four smaller circles' colors	Line segments
1	Pink (Tertiary)	Light Brown/ Yellow	4 small circles connected
2	Light Blue (Tertiary)	Yellow, Pink/Magenta, Purple, Green	4 small circles connected
3	Purple (Secondary)	Light Blue, Pink/Magenta, Light Yellow, Light Green	4 small circles connected
4	Purple (Secondary)	Light purple, Pink Magenta, Light purple, Pink/Magenta	4 small circles connected but not uniform
5	Green (Primary)	Pink/Magenta, Purple, Blue, Yellow	2 small circles connected
6	Blue (Primary)	Light Purple, Light Blue, Magenta, Green	4 small circles connected but not uniform
7	Yellow (Secondary)	Light brown, Light Blue, Pink, Purple	4 small circles connected
8	Black (Primary)	Red, Red, Red, Red	4 small circles connected
9	Blue (Primary)	Purple, Purple, Purple, Purple	4 small circles connected
10	Black (Primary)	Blue, Red, Green, Yellow-Orange	4 small circles connected
11	Blue (Primary)	Green, Green, Green, Green	4 small circles connected
12	Red (Primary)	Light Blue, Dark, Orange, Green	4 small circles connected
13	Purple (Secondary)	Light Blue, Lighter Blue, Blue-Green, Purple	4 small circles connected
14	Light Green (Tertiary)	Orange, Cobalt, Magenta, Yellow	4 small circles connected
15	Red (Primary)	Blue, Green, Pink, Yellow	4 small circles connected

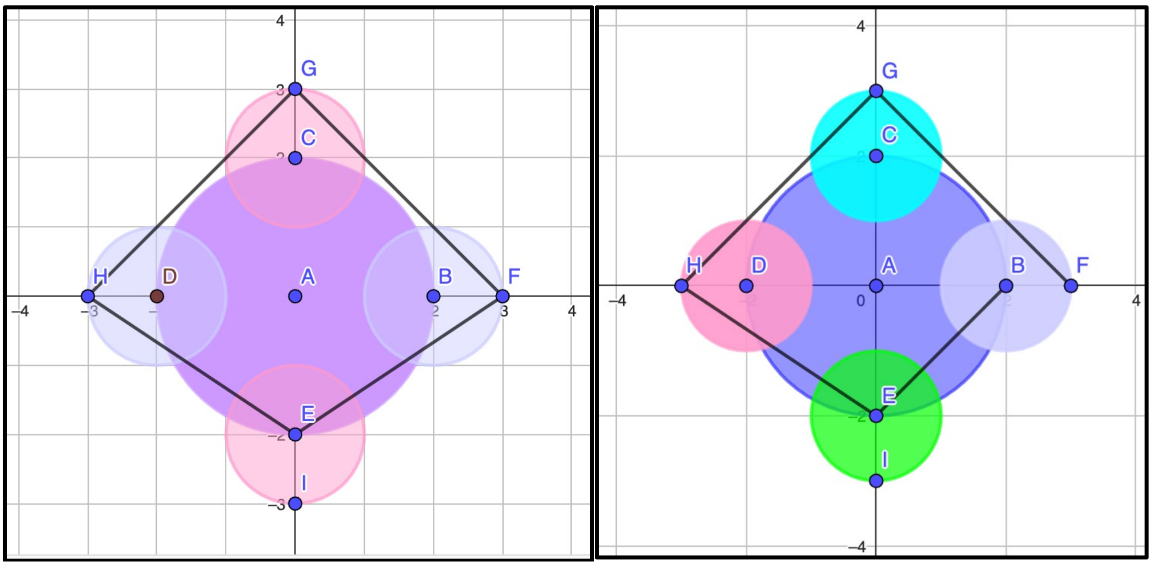
From the three videos, we could see how students colored the objects. In Screen 1, this student used a one-time coloring strategy, meaning that after using a color, this student did not change it. Meanwhile, in Screen 2, the student often changed the colors and adjusted the opacity of the colors (Figure 10). For example, this student selected green, adjusted its opacity, and later changed the color to black. In Screen 3, the student changed one time the color for the big circle and then used a one-time coloring strategy. In the anime video and the example, the mandalas were not colored, and this coloring activity had not given students any examples of colored mandalas. Students had to find their own colors, and this seems to have given students the opportunity to use problem-solving skills by iterating their colors.

Figure 10. A student changed and adjusted the colors overtime



After students colored their objects, they continued to use the segment command. The use of the segment command appeared to be relatively easy for students, as they just had to enter "Segment(Point1,Point2)." For instance, a student entered the command "Segment(F,G)" and the line segment FG would be created. Most students created segments FG, GH, HI, and IF. Only two students constructed the segment differently (see Figure 11), creating a segment labeled HE. It seemed that the student who created the mandala on the right tried to construct a line segment EF but perhaps failed, and this resulted in the creation of line segment EB. To some extent, the use of segments was limited to connecting only the small circles. We should have encouraged students to use the segment command to connect anything.

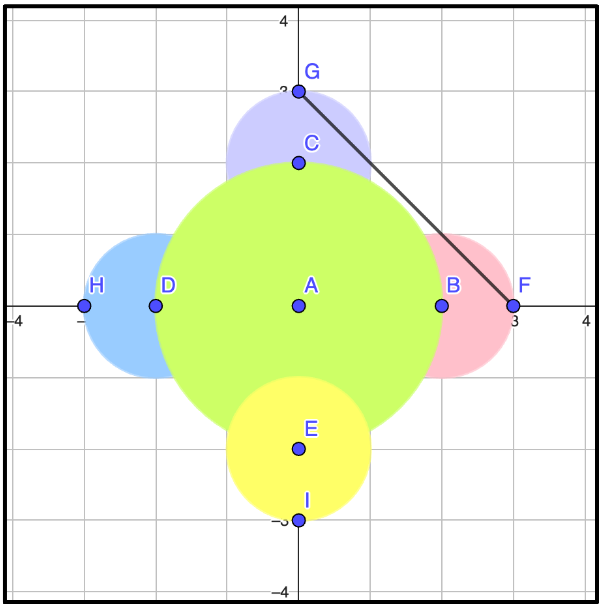
Figure 11. Two students constructed rather different line segments (not uniform)



One student's work featured the creation of only a single line segment (see Figure 12). It is possible that this student was busy with the coloring feature and thus did not realize that they should proceed with line segment creations.

We further noticed that some students were busy arranging the colors in such a way as to position the smaller circles either in front of or behind the larger circle. For example, in Figure 12, we see that a student tried to have the smaller circles appear behind the larger circle. However, the small yellow circle appears to be not yet set to appear behind the large green circle. In GeoGebra, the preceding object's color would automatically appear behind the most recently created object. Therefore, it appears likely that this student first created three smaller circles, then the larger circle, and finally the fourth smaller circle, hence the yellow shape appearing in front of the group.

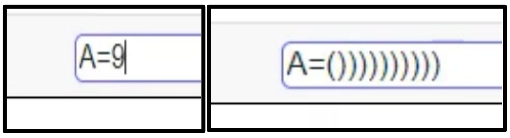
Figure 12. A student's incomplete segment creations



5.2 The emergence of CT skills

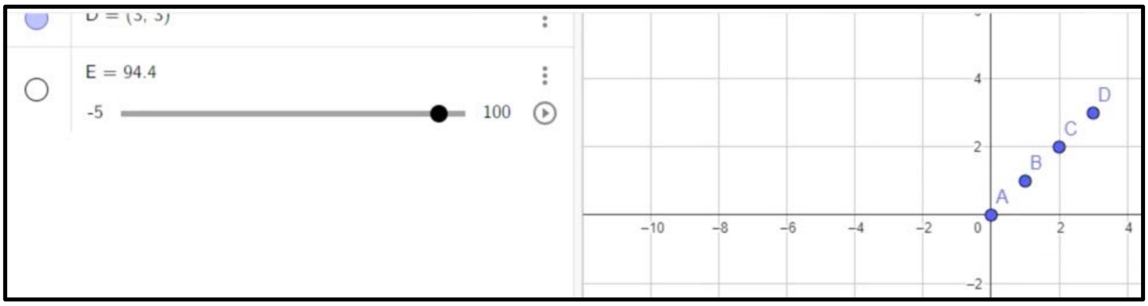
We analyzed three screen-recording videos to look at how students CT skills emerged while students engaged with our task. In Screen 1, within less than two minutes, the male participant (StudentA) was able to input the GeoGebra command for creating a point. Initially, this student made a mistake while entering the open bracket by hitting the 9 key without the Shift button also being depressed (Figure 13). We coded this as an example of struggling with a symbol which required pressing the Shift button (i.e., code StrugBut). Later, he easily inputted the brackets for constructing objects.

Figure 13. The progress of inputting the correct command



Meanwhile, in Screen 2, another male student (StudentB) made a similar mistake in entering "9" instead of "9 + Shift key" for the open bracket creation, and this resulted in the creation of point E and a slider instead of a singular point (see Figure 14). While this case was likewise coded as a "struggle involving a button," or (StugBut), we also coded it as an example of "debugging by deleting," or (DebgDel) since he had deleted the faulty command and then inputted the correct command.

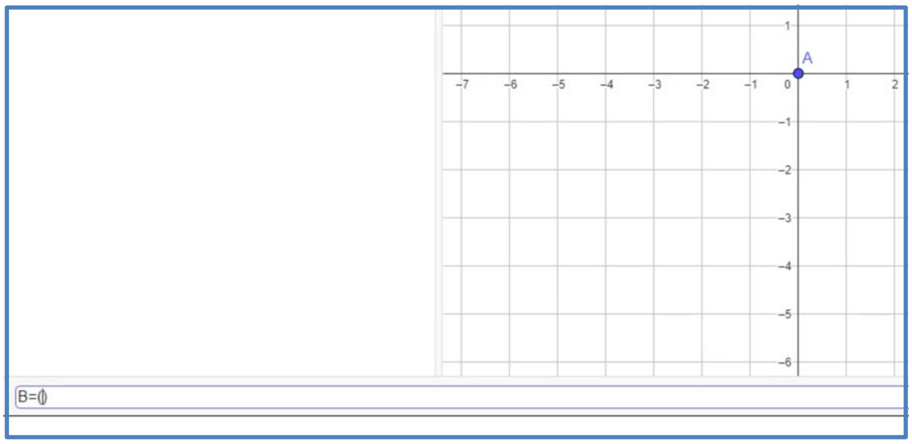
Figure 14. The student mistakenly uses 9 instead of an open bracket, resulting in a slider



In Screen 3, a female participant (StudentC) did not make any apparent mistakes when creating objects during the activity due to her ability to properly use commands.

Clearly, mastering the use of the various keys and keystroke combinations used to input commands in GeoGebra is important for students in terms of successfully working with the syntax of textual programming. In the case of StudentA, he was able to quickly adjust his ability to find and use the combination of the Shift and 9 keys to obtain the symbol for brackets, even when he did not at first succeed (Figure 15). Overall, students used correct input and coordinates for creating their points.

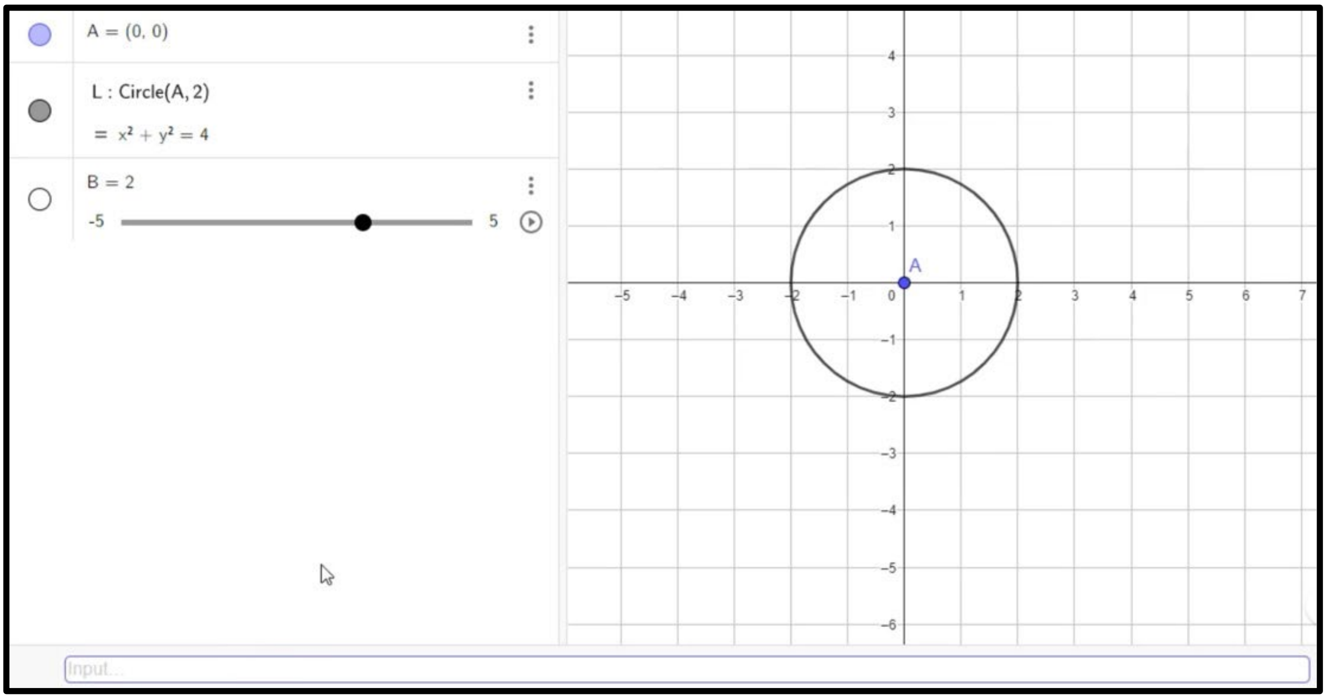
Figure 15. StudentA learned quickly on how to get brackets



On another occasion, when StudentA had learned about the GeoGebra command for creating a circle, he made a mistake in creating point B by using the period ('.') as the separator rather than the comma (",") and thus created a slider instead of a point. StudentA realized his mistake and revised the command to read "B=(2,0)" instead of "B=(2.0)." It is possible that the proximity of the period and comma keys on a keyboard led to the unintentional selection of the period key. Regardless of the reason for the mistake, he successfully revised his keystroke input by deleting the incorrect command and replacing it with the correct one. This instance was also coded as "debugging by deleting" or (DebgDel).

StudentB made a mistake by forgetting the separator when creating point D by inputting D=(-2) resulting in the creation of a slider (see Figure 16). Later, he deleted the incorrect command, replacing it with the correct command.

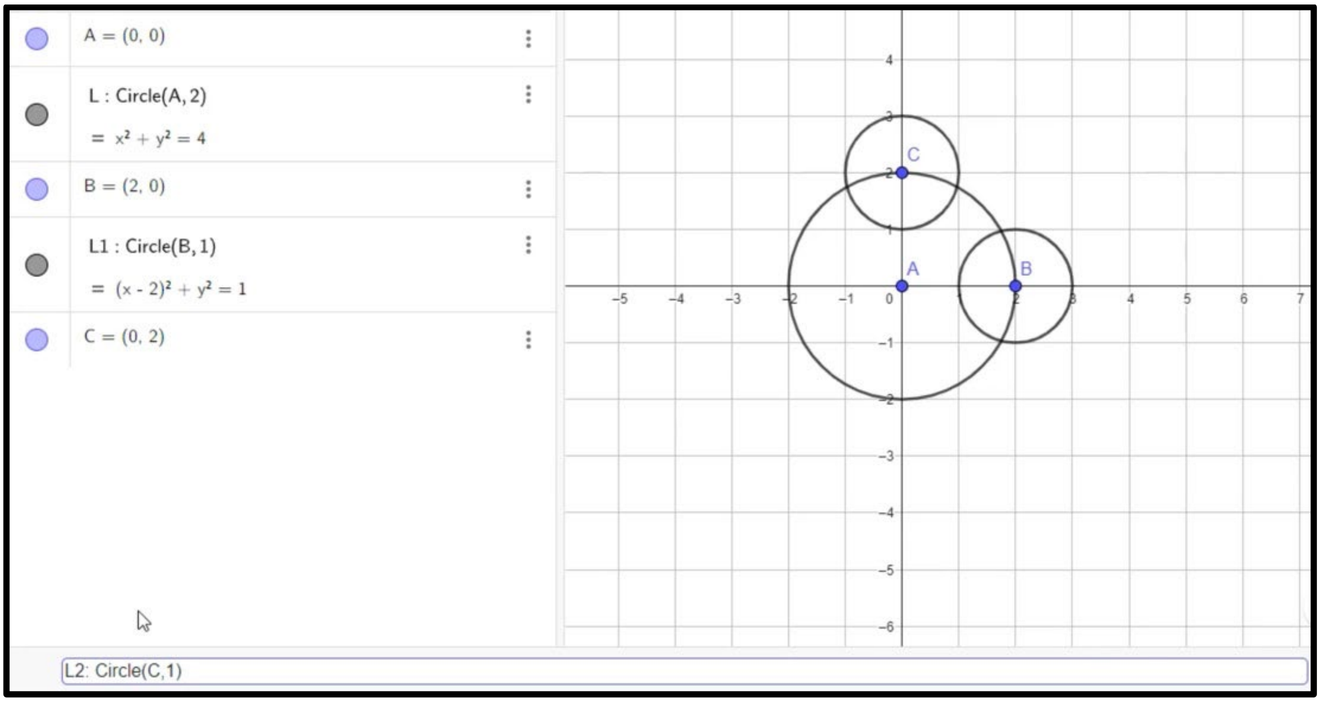
Figure 16. A student enters incomplete syntax, resulting in a slider creation



The previous examples of student mistakes provide us with some clear samples of the critical thinking component "debugging by revising mistakes."

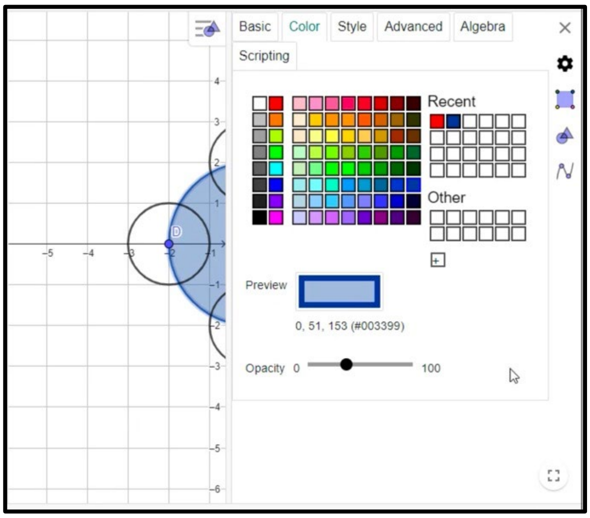
Another interesting finding is that StudentA used a different command than the one that was introduced by the researcher and was thus able to successfully create the object (Figure 17). More specifically, instead of inputting "L2=Circle(B,1)," he entered the command "L2:Circle(B,1)" which had not been presented in class. We coded this item as "using a new algorithm," or (NAlgo). This new method was then used by StudentA on the consecutive circles L3 and L4. StudentB and StudentC both followed the command introduced by the researcher (i.e., using the "equal" sign, "=").

Figure 17. A new, non-demonstrated command was used by StudentA



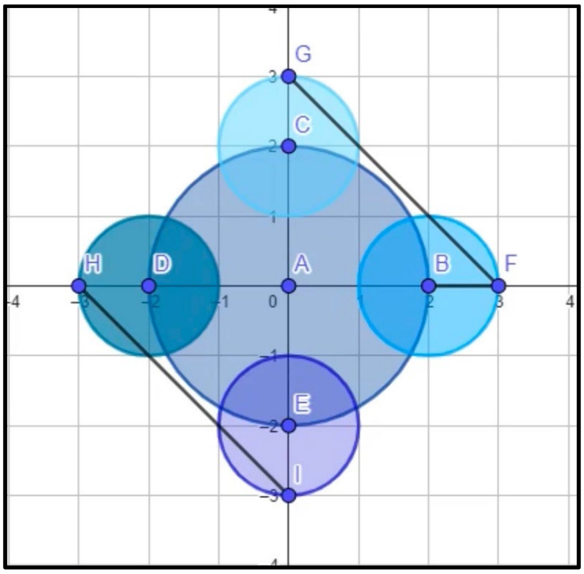
The next activity was to color the objects (Figure 18). StudentA adjusted the color so that it would appear more transparent. StudentB and StudentC also preferred transparent colors for their mandala constructions.

Figure 18. Setting up the object's color on GeoGebra



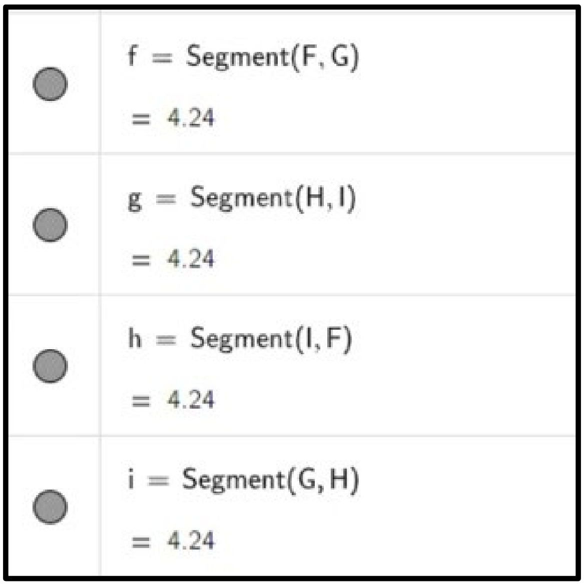
After students colored the objects, another GeoGebra command for creating a segment was introduced. Students could connect two points to create a line segment by inputting "Segment(Point1,Point2)." Once this command was introduced, StudentA appeared to be able to use it immediately. Note that this student did not create the line segments sequentially but rather in a random order by inputting Segment(F,G) and then Segment(H,I) (Figure 19). In other words, it seems that this student realized that the order does not matter in terms of creating the line segments as long as each pair of two points is connected. We thus coded this example as "Pattern Recognition," or (PR).

Figure 19. Creating line segments in a more apparently random order



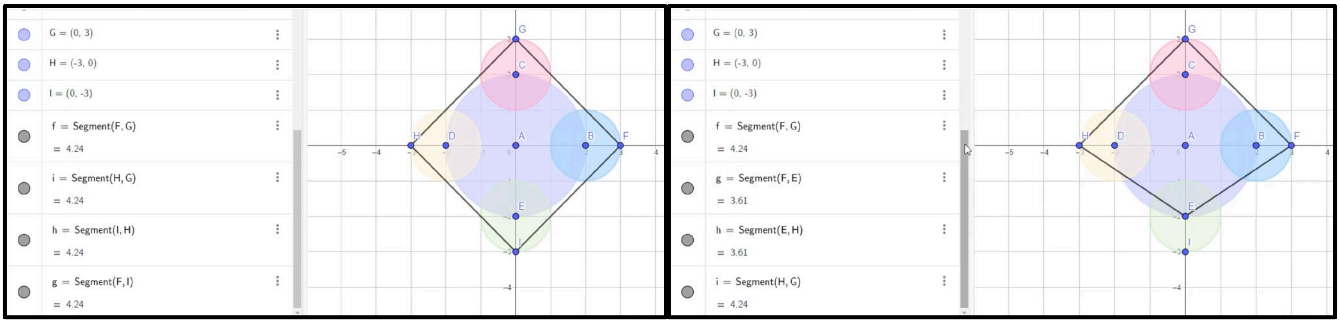
From students inputted commands, we could trace the way students inputted the segments (Figure 20). In the case of StudentA, he entered the commands Segment(F,G), Segment(H,I), Segment(I,F), and then Segment(G,H); he did not choose to follow a clockwise or a counter-clockwise pattern as other students had done.

Figure 20. The order of line segment commands entered by StudentA



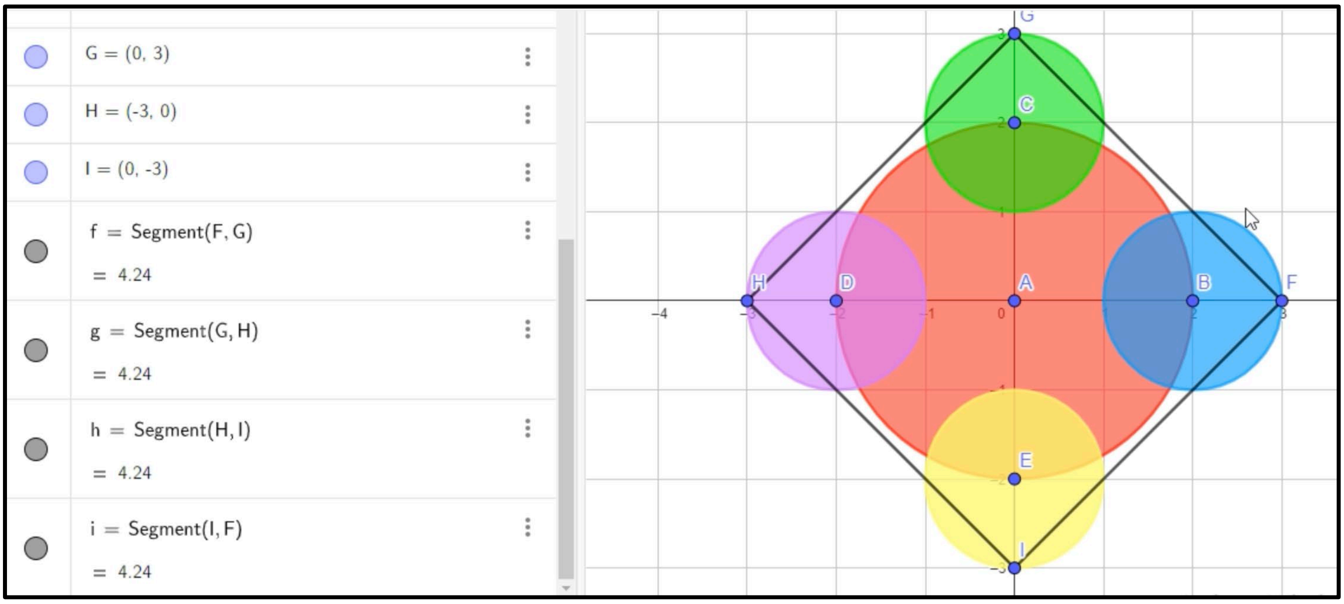
StudentB used another approach by following a clockwise pattern and entering the commands in the order Segment(F,G), Segment(F,E), Segment(E,H), and Segment(H,G). Interestingly, the line segment creations were different from what most other students did. He did not connect point I, and this resulted in a unique mandala design (Figure 21 left). However, he later revised the segments and connected the point I (Figure 21 right). We coded this as "Debugging by Revising the Commands," or (DebgRev).

Figure 21. StudentB revised the line segments by editing the commands



In contrast, StudentC used a counter-clockwise pattern (see Figure 22) by entering the line segment comments in the order Segment(F,G), Segment(G,H), Segment(H,I), and Segment(I,F). Therefore, we assume that StudentB and StudentC apply a procedure to create the segment, and it is coded as an "algorithm" or (Algo).

Figure 22. StudentC entered the line segments using a counter-clockwise pattern



6 Discussion

As we utilized GeoGebra commands, this approach bears resemblance to Papert's (1980) pioneering work with "Logo Turtle" and later evolved into "Turtle Geometry" wherein students were required to command the turtle to move and trace lines in order to create mathematical objects. In this paper, we encouraged students to create artistic, digital mandala objects through the use of specific GeoGebra commands. Students were required to type in commands to construct points, circles, and line segments and to later adjust the color of their objects. We found that a few students struggled with the use of brackets as it required students to press two keys (Shift and 9) simultaneously. In sometimes forgetting to press the Shift key, this resulted in a "9" being entered instead of the required open bracket ("("). Even in these instances, students often were able to self-correct their mistakes through ongoing inquiry and visual feedback.

In primary schools, it is common to introduce computational thinking through unplugged activities and block-based programming (Polat & Yilmaz, 2022). It seems that there is, at least more recently, a preference to introduce CT in primary schools through visual programming software such as Scratch or similar programs (Chan et al., 2022; Erdem & Kalelioğlu, 2024). Similarly, Tsukamoto et al. (2016) argued that visual programming is highly suitable for primary school students. Resnick et al. (2009) developed the software entitled "Scratch" in order to support young students and teachers in learning how to code, or master simple programming skills, via a more visual, click-and-drag approach. While young students may struggle with syntax and textual programming, proper classroom support can assist them in also developing these important skills. Tsukamoto et al. (2015) provided a successful example of utilizing text programming for students in primary schools. In our study, we utilized mathematics software, GeoGebra, wherein students are required to enter specific commands to construct objects. To some extent, primary school students in this study did not encounter serious technical problems with the GeoGebra commands and utilization of the computer's button keys.

Additionally, in terms of mathematical concepts, Grade 5 students in our study correctly used the point creation feature by successfully entering coordinate information. Tsukamoto et al. (2015) found that young primary school learners can indeed handle the coordinate system understanding and related skills. While the National Curriculum of Indonesia currently does not include the learning of the Cartesian coordinate system in Grade 5, our findings could be useful in reconsidering the placement and timing of these important mathematical skills. Moreover, related to the mathematics concepts such as points, line segments, and circles, students in Indonesia learn these math content items under geometry and measurement. Therefore, students should be able to differentiate 2D objects. This corresponds to Level 2 of van Hiele's (1984) geometry thinking, related to the property of geometrical objects. Research by Skordialos and Baralis (2017) revealed that Grade 2 students were motivated in learning geometry with technology and could arrive at Level 2 of van Hiele's geometry thinking. This is in line with our finding that students were able to create circles with the intended coordinates and sizes, a skill which belongs to Level 2 of van Hiele's model. If they have not grasped the idea of the size of the circle (the radius), it would be difficult for students to create four smaller circles with the same radius but with different centers in GeoGebra. Understanding the mathematics concepts and the use of GeoGebra commands made it possible for students to create the circles with different sizes and centers thus clearly depicting Level 2 aspects of van Hiele's geometry thinking.

In terms of students' artistic mandala creations, we found that students used different colors and strategies. While most of the designs were fairly similar in nature, there was evidence of some creativity in terms of both colors and coding input. In adopting the "half-baked artifact" approach proposed by Kynigos (2007), this led participating students to continue patterns and produce similar shapes. It appeared that most students followed the pattern to complete the mandala so that their mandala became uniform (i.e., one large circle and four connected smaller circles). It may be the case that students needed more encouragement to go beyond the simple examples and/or that this expectation was not clearly communicated. Some follow-up questions need to be developed to support students' initiative to create more circles and unique mandalas. However, when it came to adding colors, there were definitely some aspects of creativity in terms of the type of objects, the number of colors used, and the lines constructed. This kind of activity could be considered as students developing problem-solving skills utilizing technology in a creative manner, as proposed by Dakers (2024). Additionally, coloring objects digitally allowed students to freely and flexibly change or replace the colors, unlike options that would be available to them in a comparable paper-based activity. Further research relating to the digital coloring could engage students even more significantly with creative processes and products (Angeli et al., 2023; Tokuihsa & Kamiyama, 2010).

The first research question has been answered from the previous explanations, which show that students did not encounter serious problems with GeoGebra commands and swiftly used the commands and tool to create and creatively color the mathematical objects. They also used different colors and coloring strategies for their mandalas.

Understanding that problem-solving skills are critically important, in this study we also investigated CT skills emerging from our Math+CT task. Students engaged in Polya's problem-solving steps, i.e., (1) they understood what they had to solve (creating the mandala); (2) they devised the plan (selecting centers and radii for the circles, the line segment connections, and colors); (3) they inputted GeoGebra commands to create the objects; and (4) the looked back at the coordinates, sizes, and colors of the objects to complete their mandalas. The task provided students with adequate freedom so that the plans and execution aspects could vary for each student, resulting in different outcomes. Moreover, technological problem-solving by Morrison-Love (2021) was observable when students debugged the commands, operated the Shift key button, and organized the colors to be in contrast to the big circle. However, an emergent task was not yet appearing, such as constructing other objects or investigating other commands. Students also created something unique and personalized that had never been created before, having only briefly seen one example and thus not able to mimic it. This resonates with the virtual creative problem-solving presented by Dakers (2024). Moreover, students apparently engaged in computational activities when creating their mandalas. We observed students' looking for a pattern when constructing the consecutive points and circles. Without further instruction, students continued the pattern thereby creating objects required to finish the mandala. As in Shute et al. (2017), pattern recognition is an important sub-set of the abstraction facet.

Shute et al. (2017) defined an algorithm as creating a set of coherent and systematic instructions for the effective execution of a task by either a human or a machine. In our study, students were introduced to simple examples of how to use the Point command, Circle command, and Line Segment command in the software GeoGebra in order to complete their mandalas. Some students created segments by following the clockwise or counter-clockwise manner, and it is such a procedure followed by them. While these commands were readily implemented by students, there was at least one instance where a participant developed their own variation of a command entry, which served to equally construct a valid circle. Whether this was an example of a random keystroke entry error that inadvertently resulted in a functional code, or whether the student repeatedly tried different command variations until one worked, or whether the student may have had prior experience with GeoGebra coding options remains to be seen. Notwithstanding the student's method or background knowledge, this instance does serve to highlight the potential for creative space and algorithmic exploration while using such software.

Another CT facet that is apparent in this study is debugging. Students revised their incorrect commands by deleting and editing the command that they had previously entered. This is in line with Yunianto, Sami El-Kasti, et al's. (2024) claim that GeoGebra-based Math+CT lessons can assist students in developing their debugging skills. Likewise, in this current study, some students were able to detect their mistakes and to revise their commands as per the definition of debugging proposed by Shute et al. (2017).

The second research question has been answered that students have been involved in recognizing patterns when constructing consecutive objects depicting a sub-skill of the abstraction; inputting GeoGebra commands for creating mathematical objects such as points, circles, and segments, depicting algorithm design; and revising and improving their GeoGebra commands, depicting the debugging. Therefore, this study presented and supported three CT skills that emerged in our Math+CT task involving primary school students.

7 Conclusions

Our investigation clearly demonstrated that primary school students can indeed enjoy and be successful mathematically while working with textual programming within a software such as GeoGebra. Students were engaged in technological problem-solving and were capable of understanding and extending their learning around simple coding commands while constructing mathematical objects. Additionally, students were able to enhance their creativity through the completion of the "half-baked task" set before them in terms of object size, complexity, and color patterns. Moreover, lessons such as the one implemented here can easily be modified to allow for even more creativity in the task completion in terms of added commands or more complex geometric and algorithmic expectations. Factors affecting the type and degree of students' computational thinking, such as debugging, algorithm design, and abstraction, were addressed in the study and could be further explored in future research.

While this study was obviously limited in terms of its participant sample size, grade level, and the relatively narrow geometric focus (i.e., circle and line segment construction), it nonetheless sheds significant light on the potential benefits of plugged, programming-based, arts-integrated learning strategies for primary school students. Further research with even younger students in different educational contexts and with more elaborate geometric and algorithmic goals is indeed encouraged and warranted.

Research ethics

Author contributions

W.Y.: conceptualization, investigation, methodology, project administration, validation, visualization, writing—original draft preparation, writing—review and editing
B.B.: conceptualization, writing—review and editing
Z.L.: conceptualization, supervision, writing—review and editing
Z.H.: writing—review and editing
S.E.: writing—review and editing.

All authors have read and agreed to the published version of the manuscript.

Artificial intelligence

While drafting this manuscript, the Grammarly add-ons on MS. Word was used to check and correct the grammatical errors.

Institutional review board statement

This study has received the Certificate of Integrity of the JKU Ethics Committee.

Informed consent statement

The parents and guardians of the children participating in this study have consented to their children by communicating with the homeroom teacher in the Parents-Teacher WhatsApp group. Therefore, only children who were permitted by their parents or guardians attended our session.

Data availability statement

Data is unavailable due to privacy or ethical restrictions.

Acknowledgements

The first author would thank the OeAD scholarship for awarding him the EMG scholarship to pursue a PhD study at JKU Linz.

Conflicts of interest

The authors declare no conflicts of interest.

References

Alsarayreh, R. S. (2023). The effect of technological skills on developing problem solving skills: The moderating role of academic achievement. *International Journal of Instruction*, *16*(2), 269–388. https://doi.org/10.29333/iji.2023.16221a

Angeli, E., Cholevas, C., Mavrikos, E., Chatzistamatis, S., & Tsekouras, G. E. (2023). Computational creativity under the framework of recoloring art paintings for color-blindness. 14th International Conference on Information, Intelligence, Systems and Applications, IISA 2023, 1–6. https://doi.org/10.1109/IISA59645.2023.10345967

Bakker, A. (2018). Design research in education: A practical guide for early career researchers. In A. Bakker (Ed.), Design research in education: A practical guide for early career researchers (1st ed.). Routledge. https://doi.org/10.4324/9780203701010

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). Developing computational thinking in compulsory education - Implications for policy and practice (P. Kampylis & Y. Punie, Eds.). Publications Office of the European Union. https://doi.org/10.2791/792158

Bocconi, S., Chioccariello, Augusto Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V., & Stupurienė, G. (2022). Reviewing computational thinking in compulsory education (A. I. dos Santos, R. Cachia, N. Giannoutsou, & Y. Punie, Eds.). Publications Office of the European Union. https://doi.org/10.2760/126955

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 Annual Meeting of the American Educational Research Association, 1–25.

Britannica. (2023). Mandala. In The Editors of Encyclopaedia. Encyclopaedia Britannica. https://www.britannica.com/topic/mandala-diagram

Broley, L., Buteau, C., Modeste, S., Rafalska, M., & Stephens, M. (2024). Computational thinking and mathematics. In B. Pepin, G. Gueudet, & J. Choppin (Eds.), Handbook of Digital Resources in Mathematics Education (pp. 323–360). Springer Nature. https://doi.org/10.1007/978-3-031-45667-1_12

Chan, S. W., Looi, C. K., Ho, W. K., & Kim, M. S. (2022). Tools and approaches for integrating computational thinking and mathematics: A scoping review of current empirical studies. *Journal of Educational Computing Research, 60*(8), 2036–2080. https://doi.org/10.1177/07356331221098793

Chytas, C., van Borkulo, S. P., Drijvers, P., Barendsen, E., & Tolboom, J. L. J. (2024). Computational thinking in secondary mathematics education with GeoGebra: Insights from an intervention in calculus lessons. *Digital Experiences in Mathematics Education, 10*, 228–259. https://doi.org/10.1007/s40751-024-00141-0

Dakers, J. R. (2024). What is technology education? *LUMAT: International Journal on Math, Science and Technology Education, 11*(4), 1–10. https://doi.org/10.31129/LUMAT.11.4.2207

Erdem, E., & Kalelioğlu, F. (2024). Comparing traditional and flipped classrooms in teaching block-based programming: Effects on fifth-graders' self-efficacy towards computational thinking skills, conceptual learning, and their perspectives. *Education 3-13*, 1–23. https://doi.org/10.1080/03004279.2024.2400933

Fauzan, A. (2002). Traditional mathematics education vs . realistic mathematics education: Hoping for changes. In P. Valero & O. Skovsmose (Eds.), P. Valero &The 3rd International Mathematics Education and Society Conference (pp. 1–4). Danmarks Pædagogiske Universitet.

FISME. (2015, August). IMPOME (International Master Programme on Mathematics Education): master theses of the group 2010-2014. http://www.fisme.science.uu.nl/en/impome/

Grover, S., & Pea, R. (2013). Computational thinking in K-12: a review of the state of the field. *Educational Researcher, 42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Hermans, F., & Aivaloglou, E. (2017). A controlled experiment comparing plugged first and unplugged first programming lessons. Proceedings of the 12th Workshop on Primary and Secondary Computing Education, 49–56. https://doi.org/10.1145/3137065.3137072

Hsieh, H. F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative Health Research, 15*(9), 1277–1288. https://doi.org/10.1177/1049732305276687

Irawan, E., Rosjanuardi, R., & Prabawanto, S. (2024). Research trends of computational thinking in mathematics learning: A bibliometric analysis from 2009 to 2023. *Eurasia Journal of Mathematics, Science and Technology Education, 20*(3), 1–16. https://doi.org/10.29333/ejmste/14343

Jonassen, D. H. (1997). Instructional design models for well-structured and ill-structured problem-solving learning outcomes. *Educational Technology Research and Development, 45*(1), 65–94. https://doi.org/10.1007/bf02299613

Krippendorff, K. (2004). Content analysis: An introduction to its methodology. In Sage Publications (2nd ed.). Sage Publishers Ltd.

Kynigos, C. (2007). Using half-baked microworlds to challenge teacher educators' knowing. *International Journal of Computers for Mathematical Learning, 12*, 87–111. https://doi.org/10.1007/s10758-007-9114-2

Kynigos, C. (2015). Constructionism: Theory of learning or theory of design? In S. J. Cho (Ed.), Selected Regular Lectures from the 12th International Congress on Mathematical Education (pp. 417–438). Springer Cham. https://doi.org/10.1007/978-3-319-17187-6_24

Lodi, M., & Martini, S. (2021). Computational thinking, between Papert and Wing. *Science & Education, 30*(4), 883–908. https://doi.org/10.1007/s11191-021-00202-5

MacPherson, R. T. (1998). Factors affecting technological trouble shooting skills. *Journal of Industrial Teacher Education, 35*(4), 1–29.

Maharani, S., Kholid, M. N., Pradana, L. N., & Nusantara, T. (2019). Problem solving in the context of computational thinking. *Infinity Journal, 8*(2), 109–116. https://doi.org/10.22460/infinity.v8i2.p109-116

McCade, J. (1990). Problem Solving: Much More Than Just Design. Journal of Technology Education, 2(1), 1–13. https://doi.org/10.21061/jte.v2i1.a.5

McCormick, R. (2004). Issues of learning and knowledge in technology education. *International Journal of Technology and Design Education, 14*(1), 21–44. https://doi.org/10.1023/B:ITDE.0000007359.81781.7c

McKenney, S., & Reeves, T. C. (2018). Conducting educational design research (2nd ed.). Routledge. https://doi.org/10.4324/9781315105642

McKenney, S., & Reeves, T. C. (2021). Educational design research: Portraying, conducting, and enhancing productive scholarship. *Medical Education, 55*(1), 82–92. https://doi.org/10.1111/medu.14280

Mifetu, R. K. (2023). Using activity method to address students' problem-solving difficulties in circle geometry. *Contemporary Mathematics and Science Education, 4*(1), 1–8. https://doi.org/10.30935/conmaths/13079

Mohaghegh, M., & Mccauley, M. (2016). Computational Thinking: The skill set of the 21st century. (IJCSIT) *International Journal of Computer Science and Information Technologies, 7*(3).

Morrison-Love, D. (2022). Technological problem solving: an investigation of differences associated with levels of task success. *International Journal of Technology and Design Education, 32*(3), 1725–1753. https://doi.org/10.1007/s10798-021-09675-5

Muhammad, R. R., Lawson, D., Aslam, F., & Crawford, M. (2023). Indonesian curriculum 2013 ten years on: Impact on mathematics teaching. *Journal of Research in Science, Mathematics and Technology Education, 6*(SI), 109–136. https://doi.org/10.31756/jrsmte.616si

Nordby, S. K., Bjerke, A. H., & Mifsud, L. (2022). Computational thinking in the primary mathematics classroom: A systematic review. *Digital Experiences in Mathematics Education, 8*(1), 27–49. https://doi.org/10.1007/s40751-022-00102-5

Olsher, S., Abrahamson, D., Arcavi, A., Arzarello, F., Chazan, D., Clark-Wilson, A., Leikin, R., Sinclair, N., & Yerushalmy, M. (2023). Problem solving with technology: Multiple perspectives on mathematical conjecturing. In M. Ayalon, B. Koichu, R. Leikin, L. Rubel, & M. Tabach (Eds.), Proceedings of the 46th Conference of the International Group for the Psychology of Mathematics Education (Vol. 1, pp. 134–138). PME.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas (1st ed.). Basic Books.

Polat, E., & Yilmaz, R. M. (2022). Unplugged versus plugged-in: Examining basic programming achievement and computational thinking of 6th-grade students. *Education and Information Technologies, 27*(7), 9145–9179. https://doi.org/10.1007/s10639-022-10992-y

Polya, G. (1945). How to solve it. Princeton University Press.

Putra, Z. H., Ramiati, Zufriady, Hidayat, R., Jismulatif, Hermita, N., & Sulistiyo, U. (2022). Development of computational thinking tasks based on Riau Malay culture: A study of fifth-grade public school students in Pekanbaru, Indonesia. *Education 3-13*, 1–11. https://doi.org/10.1080/03004279.2022.2150063

Resnick, M. (2012). Reviving Papert's Dream. *Educational Technology, 52*(4), 42–46.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. Communications of the ACM, 52(11), 60–67. https://doi.org/10.1145/1592761.1592779

Rijke, W. J., Bollen, L., Eysink, T. H. S., & Tolboom, J. L. J. (2018). Computational thinking in primary school: An examination of abstraction and decomposition in different age groups. *Informatics in Education, 17*(1), 77–92. https://doi.org/10.15388/infedu.2018.05

Schaafstal, A., Schraagen, J. M., & van Berl, M. (2000). Cognitive task analysis and innovation of training: The case of structured troubleshooting. Human Factors: *The Journal of the Human Factors and Ergonomics Society, 42*(1), 75–86. https://doi.org/10.1518/001872000779656570

Schoenfeld, A. H. (1985). Mathematical problem solving. Academic Press.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Sinclair, N. (2001). The Aesthetic Is Relevant. *For the Learning of Mathematics, 21*(1), 25–32.

Skordialos, E., & Baralis, G. (2017). A teaching approach of geometric shapes' properties with the use of online educational tools in Greek primary school. *New Trends and Issues Proceedings on Humanities and Social Sciences, 4*(9), 101–109. https://doi.org/10.18844/prosoc.v4i9.3044

Sneider, C., Stephenson, C., Schafer, B., & Flick, L. (2014). Computational thinking in high school science classrooms. *The Science Teacher, 81*(05), 53–59. https://doi.org/10.2505/4/tst14_081_05_53

Tokuihsa, S., & Kamiyama, Y. (2010). The World is Canvas: A coloring application for children based on physical interaction. The 9th International Conference on Interaction Design and Children, 315–318. https://doi.org/10.1145/1810543.1810601

Tsukamoto, H., Takemura, Y., Nagumo, H., Ikeda, I., Monden, A., & Matsumoto, K. I. (2015). Programming education for primary school children using a textual programming language. Proceedings - Frontiers in Education Conference, FIE, 1–7. https://doi.org/10.1109/FIE.2015.7344187

Tsukamoto, H., Takemura, Y., Oomori, Y., Ikeda, I., Nagumo, H., Monden, A., & Matsumoto, K. I. (2016). Textual vs. visual programming languages in programming education for primary schoolchildren. Proceedings - Frontiers in Education Conference, FIE, 1–7. https://doi.org/10.1109/FIE.2016.7757571

van Borkulo, S. P., Chytas, C., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Computational thinking in the mathematics classroom: Fostering algorithmic thinking and generalization skills using dynamic mathematics software. ACM International Conference Proceeding Series, 1–9. https://doi.org/10.1145/3481312.3481319

van den Akker, J., Gravemeijer, K., McKenney, S., & Nieveen, N. (2006). Introducing educational design research. In J. van den Akker, K. Gravemeijer, S. McKenney, & N. Nieveen (Eds.), Educational design research. Routledge.

van Hiele, P. M. (1984). The child's thought and geometry. In D. Fuys, D. Geddes, & R. Tischler (Eds.), English Translation of Selected Writings of Diana van Hiele-Geldof and Pierre M. van Hiele (pp. 243–252). City University of New York.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wu, T.-T., Asmara, A., Huang, Y.-M., & Permata Hapsari, I. (2024). Identification of problem-solving techniques in computational thinking studies: Systematic literature review. *Sage Open, 14*(2), 1–20. https://doi.org/10.1177/21582440241249897

Ye, H., Liang, B., Ng, O.-L., & Chai, C. S. (2023). Integration of computational thinking in K-12 mathematics education: A systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education, 10*(3), 1–26. https://doi.org/10.1186/s40594-023-00396-w

Yunianto, W., Cahyono, A. N., Prodromou, T., El-Bedewy, S., & Lavicza, Z. (2024). CT integration in STEAM learning: Fostering students' creativity by making Batik stamp pattern. *Science Activities, 62*(1), 26–52. https://doi.org/10.1080/00368121.2024.2378860

Yunianto, W., Sami El-Kasti, H., Charitas Indra Prahmana, R., & Lavicza, Z. (2024). A constructionist approach to learning computational thinking in mathematics lessons. *Journal of Information Technology Education: Innovations in Practice, 23*, 1–33. https://doi.org/10.28945/5365

Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research, 53*(4), 562–590. https://doi.org/10.1177/0735633115608444

Ziatdinov, R., & Valles, J. R. (2022). Synthesis of modeling, visualization, and programming in GeoGebra as an effective approach for teaching and learning STEM topics. *Mathematics, 10*(3), 1–16. https://doi.org/10.3390/math10030398