Shih, JL. et al. (Eds.) (2023). Proceedings of the 31st International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

# Developing Computational Thinking: Using TurtleStitch and Physical Computing

**Peter SEOW [a*], Chin-Lee KER[a] & Haoran XUE[b]**
[a]*National Institute of Education, Nanyang Technological University, Singapore*
[b]*New Life Community Services, Singapore*
*peter.seow@nie.edu.sg

**Abstract:** In this paper, we describe how the use of physical computing devices like an embroidery machine and TurtleStitch software can be used to engage learners in coding and developing their Computational Thinking (CT) skills. Two lessons are described on how novice learners can create geometric patterns with code while applying CT skills. In the first lesson, students learn to create polygon shapes such as square, triangle, pentagon, and hexagon. Through abstraction, pattern recognition, and algorithmic thinking, they must develop a modularized code block to create a polygon with sides and length of the polygon as input parameters. In the second lesson, they must create pinwheel geometric patterns through decomposition, pattern recognition and algorithmic thinking. The accompanying approaches such as tinkering, creating, debugging persevering, and collaborating were used to develop CT skills as learners generate geometric patterns with block-based codes in TurtleStitch. The use of embroidery machines and coding on TurtleStitch can provide opportunities for novice learners to develop coding and computation thinking skills as they creatively generate patterns with codes that can be embroidered in a tangible textile form.

**Keywords:** Coding, Computational Thinking, Physical Computing, Embroidery, Novice Leaners

## 1. Introduction

Embroidery machines have been used for centuries to create intricate and beautiful designs on fabric. However, with the advent of physical computing and software, these machines can now be used to design patterns with code. We explore the use of embroidery machines as a physical computing device with TurtleStitch to develop computational thinking skills such as abstraction, decomposition, pattern recognition, and algorithmic thinking. We present two lessons on how novice learners of coding can develop Computational Thinking skills using the TurtleStitch to create geometric patterns. In the first lesson, learners can learn about algorithmic thinking, pattern recognition and abstraction to create a function for creating a n-sided polygon. In the second lesson, learners use decomposition, pattern recognition and algorithmic thinking to create pinwheel patterns. We used the approaches of having learners explore and experiment the codes through tinkering, creating their own pattern designs, find out errors in the code through debugging, preserving through their experienced failures, and collaborating with others in the process of learning coding. The generated pattern designed by the learner can be used to create an embroidery on a textile with the embroidery machine. The purpose of this paper is to share the potential of using embroidery machines as a tool with software like TurtleStitch to develop computational thinking in novice learners.

## 2. Background

### 2.1 Physical Computing

Physical computing is a recent growth area in computing education that involves the use of the hardware and software to build interactive physical systems that sense and respond to the real world (O'Sullivan & Igoe, 2004). It has been shown to result in the broad engagement of users in learning and address shortcomings in established approaches (Sentence, Waite, Hodges, MacLeod & Yeomans, 2017). An example of a tool used for physical computing in the classroom is the popular BBC micro:bit (Cápay, & Klimová, 2019). Physical computing can provide an engaging and hands-on experience to learning computing, and can help simulate digital creativity, collaboration, and broader technology skills such as the use of sensors alongside coding and computational thinking.

A physical computing system uses sensors and microcontroller to translate analogy signals to a software system, and control electro-mechanical devices such as motors, lighting, servos, and other hardware devices. Many of the technologies that we experienced today are examples of physical computing such as the automatic lighting systems, voice-activated home control, or robot vacuum cleaner. Another class of physical computing examples are those that enable the creation of physical or tangible objects using the hardware and software. This is commonly found in computer-aided manufacturing (CAM) systems where pre-programmed computer software dictates the movement of factory tools and machinery such as robotic arm on an assembly line or Computer Numerical Control (CNC) machines. Closer to home, some sewing embroidery machines can function as embroidery machines by reading embroidery designs. A software can be used to generate embroidery patterns that can be used by the machine to create the embroidery. This approach could offer students a creative opportunity to learn coding and apply computational skills to create embroidery on sewing machines. Also, learners can see the application of codes and computers to creation of objects in real-world contexts.

### *2.2 TurtleStitch*

TurtleStitch is a free and open-source platform for generating and sharing patterns for embroidery machines (Wolz, Auschauer, Mayr-Stadler, 2019). It is derived from the educational block-based programming platform Snap!(Harvey et al., 2013). While TurtleStitch itself is not physical computing, it can be used in conjunction with physical computing devices such as embroidery machines or electronic paper cutters to create physical objects based on the designs generated by the program. In this sense, TurtleStitch can be considered a part of the physical computing process. TurtleStitch bridges the programming with textiles, enabling the creation of stitched patterns of materials, for novice and experienced coders. TurtleStitch is built upon the ideas of Seymour Papert's (1999) Logo where the microworld is the controlling the turtle. In TurtleStitch, the turtle represents the needle of the embroidery machine where it can be given instructions to create novel patterns for embroidery. By using TurtleStitch, novice learners can develop computational thinking (Wing, 2008) skills by working through the process of creating designs for embroidery machines. This involves breaking down complex designs into smaller, more manageable parts (decomposition), looking for similarities and patterns in the designs (pattern recognition), focusing on the important information to make generalisations (abstraction), and developing step-by-step instructions for the embroidery machine to follow (algorithms). Through this process, learners can develop their computational thinking skills while creating objects that are tangible and beautiful. After creating the patterns on TurtleStitch, the design can be exported to the embroidering sewing machine to create the embroidery on textile.

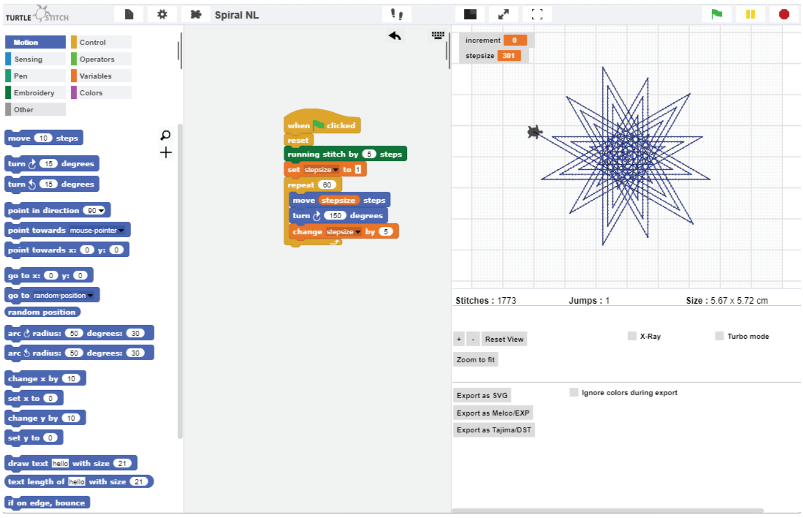


Fig 1. Shows the interface of the TurtleStitch programming platform.

## 3. TurtleStitch Lesson

In the following sub-sections, we describe the lessons that we designed and enacted to introduce novice learners to coding with the TurtleStitch and applying Computational Thinking in the process of generating their geometric patterns with codes.

### *3.1 Creating simple polygons to N-sided polygons*

After introduction to the interface and basics of using TurtleStitch, learners code to create simple polygons such as a square, triangle, pentagon, and hexagon. We used Parson's puzzle approach by providing them the blocks to use to create a square. Along with the blocks, we provided prompts for them to think about how the turtle will move to create a square (see fig 2). Learners would need to put the blocks together and fill in the appropriate values such as the repeat loops and degrees to turn to create a square.

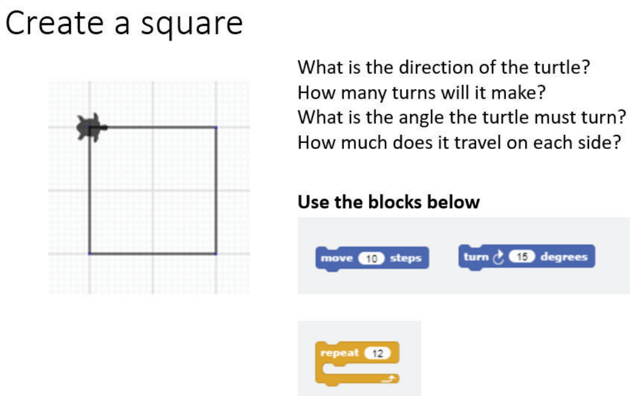


Fig 2. Creating a square on TurtleStitch

Next, learners modify the code to create the triangle, pentagon, and hexagon. To guide the learners, prompts were provided on how many turns the turtle will make or angle the turtle needs to turn to create the polygons. After creating the polygons, learners are to observe the patterns in the codes and created polygons (see fig. 3). They would notice the number of repeats corresponded to the number of turns or sides of the polygon and turn degrees was 360/number of turns. Also, they observed the number of sides was the important in creating the polygon in the codes. This is a form of abstraction by removing the details and focusing on the essential information.

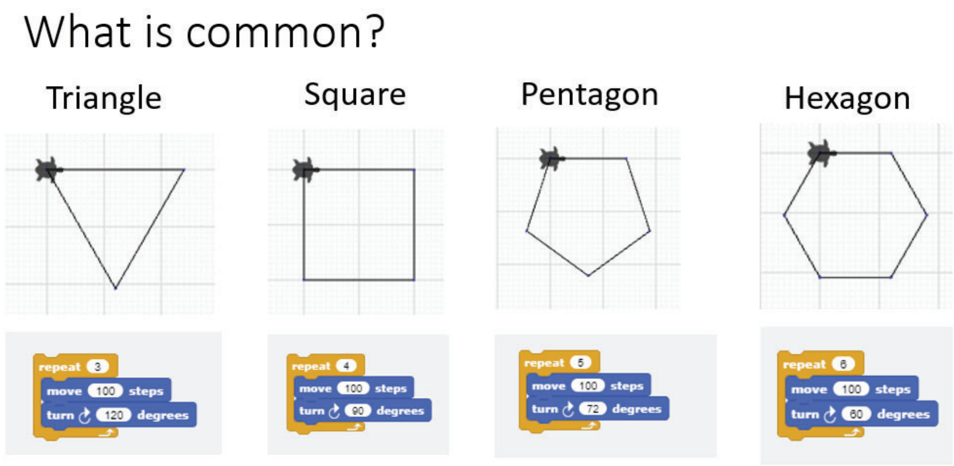


Fig 3. Creating different polygons

Based on their observations of patterns in the codes and the polygon sides, learners must create an n-sided polygon given the number of sides using the Block feature in TurtleStitch. Creating the block encapsulates the details of creating the polygon which is a form of abstraction of hiding the details. Learners are guided to create a Block with input arguments on the number of the sides and length of polygon (See Fig. 4).

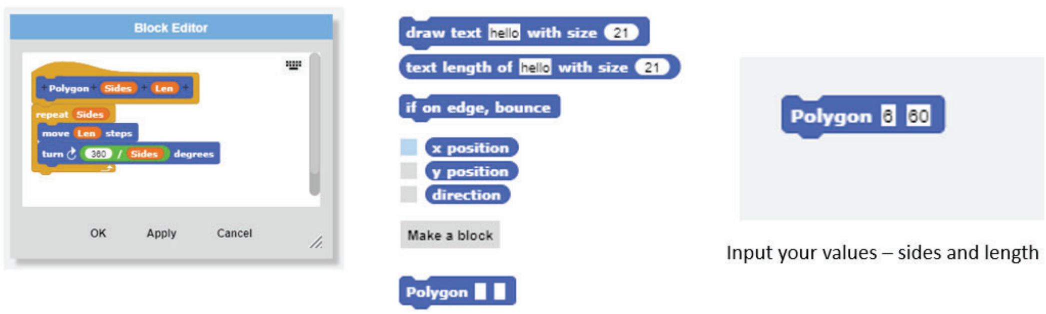


Fig 3. Creating a block for creating N-sided polygons

From the process of applying Computational Thinking in pattern recognition, learners can see the relationship between the codes, created shapes and values to identify the important information by abstraction to develop a generalized code to create polygons. In generating the polygons, they would have to use algorithmic thinking in sequencing the blocks, using the loop statements and appropriate values such as the angles to turn.

### *3.2 Pinwheel Patterns*

The pinwheel is commonly used pattern in TurtleStitch to create embroidery. Based on the previous lesson on creating polygons with codes, learners are introduced to the creation of pinwheel patterns. Given several pinwheel patterns, learners are asked to decompose and recognize polygon patterns (Fig 5).

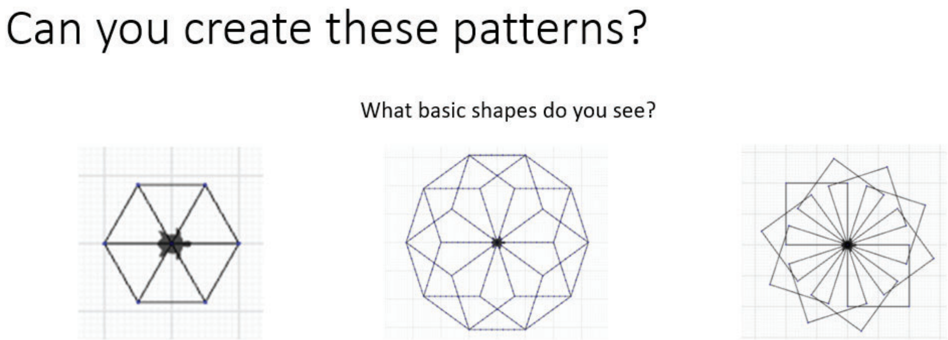


Fig 5. Decomposing and pattern recognition in pinwheels

After they can identify the basic polygon, the learners think about how the pinwheel pattern is formed with the polygon shape and the codes required to create. Applying decomposition and algorithmic thinking the learners code to create the pinwheel patterns.

## 4. Approaches to developing Computational Thinking

In developing novice leaners' Computational Thinking skills in algorithmic thinking, pattern recognition, abstraction, and decomposition, it is important to consider the approaches to support the development. In our lessons, we adopted the approaches of tinkering, creating, debugging, persevering, and collaborating advocated by UK's Barefoot Computing (UK Stem Learning, 2023) Through tinkering, we encouraged learners to develop computational thinking and coding skills by exploring and experimenting codes. They could change parameter values and sequenced the blocks differently to see how their designs are changed. We encouraged creativity in our learners to create the designs that are meaningful to them and giving them choices. On many occasions, the codes do not produce the intended design and produce errors. Learners will debug their codes to find and rectify their errors. If not, they would have to create a solution as a work around. Debugging codes or creating complex designs require the learners to be persevere and keep going with their projects. We support the learners by working with them to solve problems in collaborative approach. Students experience challenges when their designs are embroidered on the textiles with the sewing machines. For example, their designs may have too many stitches tearing the textile or appear to be sparse with the wrong stitching. They would need to revise their codes for the design to be embroidered again. Fig 6a-c shows the processes where learners are engaged in learning coding, collaborating with peers, and creating the embroidery. From our lessons with novice coders, the associated processes of coding coupled with the use of physical computing plays an important role in the development of computational thinking.

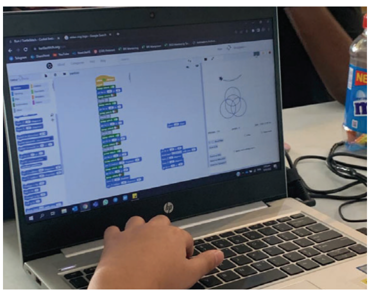

Fig 6a. Coding on TurtleStitch

Fig 6b. Learner collaborating on the machine


Fig 6c. Creating embroidery on the machine

## 5. Conclusion

In this paper we describe the use of TurtleStitch with an embroidery machine as a physical machine to develop computational thinking skills of pattern recognition, decomposition, algorithmic thinking, and abstraction. Through the design of lessons, a novice learner of coding can apply computational thinking to design and create geometric-based patterns such as polygons and pinwheels that can be embroidered on textiles on a machine. The approaches of tinkering, creating, debugging, persevering, and collaborating used in our lessons support the development of computational thinking in learners. Thus, it is important to consider the processes and experiences of learners that would enable to develop computational thinking.

## Acknowledgements

The study was funded by SMF project “It takes a village 2: Mentoring through Tinkering.” The authors would like to thank the TurtleStitch users’ group for the many ideas in enabling creativity through coding and embroidery.

## References

Cápay, M., & Klimová, N. (2019). Engage Your Students via Physical Computing!. In *2019 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1216-1223). IEEE.

Harvey, B., Garcia, D. D., Barnes, T., Titterton, N., Armendariz, D., Segars, L., Lemon, E., Morris, S., & Paley, J. (2013). Snap!(build your own blocks). In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 759-759).

O'Sullivan, D., & Igoe, T. (2004). *Physical computing: sensing and controlling the physical world with computers*. Course Technology Press.

Papert, S. (1999). What is Logo? Who needs it. *Logo philosophy and implementation*, 4-16.

Sentance, S., Waite, J., Hodges, S., MacLeod, E., & Yeomans, L. (2017). " Creating Cool Stuff" Pupils' Experience of the BBC micro: bit. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education* (pp. 531-536).

Papert, S. (1999). What is Logo? Who needs it. *Logo philosophy and implementation*, 4-16.

UK Stem Learning (2013). *The Computational Thinkers.* https://www.stem.org.uk/resources/elibrary/resource/465617/barefoot-computing-information-and-posters#&gid=undefined&pid=5

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717-3725.

Wolz, U., Auschauer, M., & Mayr-Stalder, A. (2019). Code crafting with turtlestitch. In *ACM SIGGRAPH 2019 Studio* (pp. 1-2).Mackay, W. E. (1995). Ethics, lies and videotape. In I. R. Katz, R. Mark, L. Marks, M. B. Rosson, & J. Nielsen (Eds), *Proceedings of CHI’95* (pp. 138-145). ACM Press.