Classroom Implementation Report (Grades 7 and 8): Sub-Programs and Dilations in *Scratch*!

Introduction

This is a report of the implementation of a math + coding activity taken from MathUP (Rubicon Publishing Inc.) in grade 7 and 8 classrooms by two pre-service teachers in collaboration with two in-service teachers. This report contains information about the activity, its implementation, and some reflections.

Learning Objectives

The main learning objectives of this activity are for students to:

- Write and execute code that includes sub-procedures to make similar shapes.
- Perform and understand dilations of two-dimensional shapes.
- Know the properties of similar two-dimensional shapes.

This activity addresses the following coding curriculum expectations:

- Grade 7 C3.1: Solve problems and create computational representations of mathematical situations by writing and executing *efficient* code, including code that involves events influenced by a defined count and/or *sub-program* and other control structures.
- Grade 7 C3.2: Read and alter existing code including code that involves events influenced by a defined count and/or *sub-program* and other control structures, and describe how changes to the code affect the outcomes and *efficiency* of the code.

Activity Summary

The activity includes 4 sections: Minds On, Action, Consolidate, and Your Turn. Students will first look at code that draws 2-D shapes, and predict which shape it will draw based on what they know about the shapes. Then they will use the code to make a set of similar shapes, and reflect about what they have learned about creating a program using sub-programs to draw a set of similar shapes. Finally, they will work through a series of questions pertaining to what they have learned about sub-programs, scale factors, and similar shapes.

For full activity details, see the MathUP website.

Grade 7/ Spatial Sense: Dilatations and Similarity. Activity 3: Coding: Using Sub-programs to Create Similar Shapes from MathUP.ca, Rubicon Publishing Inc.

Classroom Implementation

This activity can be implemented into the classroom over two to three 50 minute sessions. In order to maximize their learning, students should each have access to their own computer to work on (if possible). We also recommend incorporating a brief presentation with the example shapes and blocks into the lesson for students to refer back to while engaging in the activity.

Brief Reflection on Classroom Implementation

On day one, we entered our first session (grade 7) and started off by introducing ourselves, going through an outline of what we would be doing in our time with them, and then went through the learning goals for the day. From here, we guided students to the MathUP page where they could get started on the MindsOn Activity. As we walked around the class, we heard the students discussing the differences between shapes (e.g. why the first code couldn't be a rectangle). Some students asked if they could input the code into Scratch but we encouraged them to try and work through it without, i.e. engage in unplugged computational thinking. We let them know that we would be coding in the following activities and suggested some strategies for them to try, such as doing exactly what the code instructed them to do using a pencil and paper. There were some students who already had this strategy, but some needed encouragement. As mentioned, students took longer than anticipated on this portion and so it lasted about 30 minutes before we came back to discuss. If students were confident in their answers, we encouraged them to be even more specific (e.g. if they answered triangle, we would ask them to specify which kind and to elaborate). We then asked students if they had ever heard of sub-programs or blocks. Some of them indicated that they had but many did not. We then gave a mini lesson describing what blocks/sub-programs were and where/why they may be used in code. The students had been focusing on efficiency of code prior to our lesson and so this fit in nicely.

Next was the Action portion of the activity which lasted about 30 minutes. Most students only made it through the first 2/5 questions within this time, but we noticed that they were very engaged. For the first question, students were able to figure this out quickly and so in the second session we decided to go through this question together and take the opportunity to show the class how to create a block on Scratch (as the associate teacher had suggested). When students had to create their own shapes, we noticed a high level of excitement and engagement within students, showing the presence of agency. As we were walking around, there were some students who asked for ideas of what to draw. There were some instances that students even responded "No, that's too easy!" and we would offer an alternative that would provide more of a challenge. For example, if we suggested a triangle and the student felt it wasn't challenging enough, we may suggest an isosceles triangle. This was accepted well by students, and we could see their determination to meet this challenge. This level of confidence in coding was something that surprised us, as we expected students to feel more intimidated by coding. However, when discussing with some of the students, they did not even realize that coding had not always been a part of the curriculum. This may be why they did not feel too intimidated, as to them this was not something totally new or out of their ordinary. Something else that surprised us in this portion was how some students did not have the expected level of knowledge on shapes/polygons. For example, we had suggested that one student create a rectangle, thinking it was a "basic" shape. However, he did not know what a rectangle was and so therefore he did not know the properties (two sets of parallel sides, 90 degree angles, etc) and could then not code the shape.

When we arrived on day two, students had their Chromebooks open and were ready to continue working on the Action portion. For about 40 minutes, students continued to work on this section. By the second day, we noticed students asking for less help. Originally, we had thought that the students were less engaged, but after further thought and debriefing with the teachers, we also noted that the students seemed to be working together and collaborating more. As we walked around, we could see students pointing out errors to other students, etc. We worked to encourage this collaboration when going to check-in on the table groups. If a student had a question, then we would see if any of the tablemates had any ideas (often they all had the same questions and/or one student would have an idea of what to do) and we would work from there. In the first class (grade 7), we noticed many of the students were struggling with understanding what a scale factor was and how it worked and so we got the class's attention and took a few minutes to go through this concept all together. In the second class (grade 8), the students seemed to grasp this better and so we did not feel the need to do this. In both classes, when discussing with table-groups we would check-in on what they noticed with the scale factor and ask them if they could tell us what it did to which they would usually respond that it makes it bigger. We would say "Okay, now can you make it smaller?" or "Okay, now what numbers would make it smaller? The same?, etc." As mentioned, there were some students who had finished the Action portion early and so we needed to give them something to continue working on. With the first class, we suggested making more shapes with a higher level of difficulty. However, after taking time in between classes we thought that it would be a good idea to tell them to take a look at the consolidate questions and then move on to the Your Turn section as a sort of extension.

Following the Action portion, we moved onto the Consolidation section for the last 15 minutes or so. Here we noticed that the students did not all know what similar shapes were, which was something that we hadn't expected. However, we simply just took a few minutes to go over the concept and get different ideas from students. We went through each question and discussed why an answer was correct and if it was incorrect, we would work through the problem as a class so that the students would see why the answer may not be correct. The students were not very engaged in this portion, and we found the same students were answering the questions.

For the last 5 minutes of the lesson, we gave students time to discuss with their table-group what they had learned before sharing with the class. Most students said that they learned about how to use blocks, what they were, what scale factors were, or how blocks can make code efficient. This was nice to see, as all of the responses aligned nicely with the learning goals.

The teacher guidelines provided by MathUP were very useful in knowing what to expect. They provided us sample answers for student responses, giving us an idea of what answers students may come up with, or if there was a direction to guide them. It also gave us tips on how to explain sub-programs in a way that the students would be receptive of and gave us an answer key to the activity. Having this resource gave us confidence that we were giving the students the right information, as we ourselves did not have extensive experience coding in Scratch. This is something that teachers may also appreciate, as not only is coding a new requirement in the curriculum but it is also new to many teachers as well. These guidelines can be found on the MathUP website.

Computational Thinking Affordances in the Activity

- *Agency*[1] was very present throughout this activity, with the students getting very excited over their work and taking ownership. Once they finished creating their shapes, for example, they would call us over to show us what they did. While observing, we also noticed the students showing off their programs to their friends. This encompasses the idea of low floor, high ceiling. Students were able to create their own shapes with minimal direction or instruction, using what they had previously learned about polygons and Scratch.
- *Re-Using/Re-mixing*[1] was present in this activity, as students had to first look at codes and figure out what they did, before taking the given codes and modifying them to make their own shapes (by changing the side-lengths, angles, and number of repeats), eventually incorporating scale factors as well. This allows students to familiarize themselves with the properties of polygons and see how the shape changes based on the variables.
- *Dynamic Modelling*[1] was also present throughout the activity. As mentioned, students were able to modify the side-lengths and angles. At the click of a button, students could see how this would modify the shapes and could then make connections between what happens. For example, they were able to discover that when the scale factor was greater than one the shape increased in size and when between 0 and 1, the shape got smaller. They could also make the connection that when using a scale factor the side-lengths change but the angles do not, showing how in similar shapes, the side lengths are proportional and angles remain the same.

Notes for Teachers/Our Recommendations:

- Ensure students have the necessary Scratch background information prior to implementing this lesson. This includes being familiar with variables, events, sequences, using the pen, loops, etc. This was crucial in our activity as it allowed for the students to focus on learning about sub-programs and dilations rather than focusing entirely on these computational concepts.
- Review properties of shapes (number of sides, parallel sides, etc.), what a polygon is, regular vs irregular polygons, etc. We felt that the students' lack of knowledge surrounding polygon properties was a hindrance to their learning. For example, we noticed students saying "it cannot be a hexagon, all the sides are different"- while really it was just an irregular hexagon. Knowing the properties of polygons will help students make these connections between what the code is and what it means, thus allowing for a more meaningful investigation.
- The 'Your Turn' portion of the activity can be used as an extension activity depending on the level of your class. We chose to implement this activity over the span of two days and found that this was not enough for all students to go through the entire activity. If

[1] Gadanidis, G. (2017). Five affordances of computational thinking to support elementary mathematics education. Journal of Computers in Mathematics and Science Teaching 36(2), 143-151.

wanting to include the Your Turn component as planned rather than as an extension, the lesson may need to be planned for about three periods.

Two Major Takeaways

1. **Creating a friendly environment is crucial for students' success when implementing coding into the math classroom.** We observed that when students felt comfortable talking to their peers and asking questions, they were more engaged in the coding and mathematics. We did our best to allow students to be at the center of their own learning, thus making them feel like they are able to ask questions, leading to greater investigation. With this, it allows students to partake in more investigation because they feel as though they can explore the program and shapes.
2. **Students have a wide range of coding knowledge and confidence. Because of this, we must be prepared to work with students at all different points in their learning**. We need to ensure that students will not be left behind, while also ensuring that students who may be further along are not held back and/or do not become bored. This can be done by providing extension questions or opportunities. Again, as we mentioned in our classes there were students who were struggling with what certain shapes were, while others had completed the activity within the time. We decided that the final portion of our planned activity would make a good extension for those who had already finished so they were not just sitting there for the remaining time. When discussing with other groups, they had similar scenarios where some students finished the activity very quickly while others had barely begun.

The MKN is funded by the Ontario Ministry of Education. The MKN is a KNAER Project hosted by the Fields Institute for Research in Mathematical Sciences. The views expressed in this document belong to the authors and do not necessarily reflect the opinions of the Ministry of Education nor the Ontario government.

Classroom implementation report created by Jessica Sardella and Abbey Terry for MATH 3P41 at Brock University. Adapted by Jessica Sardella. June 2022.