Classroom Implementation Report (Grade 9): Investigation of Interior Angles of Regular Polygons in *Scratch*

Introduction:
This is a report on the implementation of a newly created *Scratch* activity in a Grade 9 de-streamed math classroom by two pre-service teachers in collaboration with three in-service teachers. This report includes information about the activity, its implementation, and some reflections. In addition, it includes the student worksheet with solutions and the necessary link to a *Scratch* file.

Learning Objectives:
This activity has the main learning objective of adjusting and building code for the mathematical purpose of learning about interior angles of regular polygons.

- Gr 9 - C2.3: read code to predict its outcome, and alter code to adjust constraints, parameters, and outcomes to represent a similar or new mathematical situation

For the extension:

- Gr. 9 - C2.2: create code by decomposing situations into computational steps to represent mathematical concepts and relationships, and to solve problems

Activity Instructions:
The activity begins with a brief presentation (5-7 minutes) introducing *Scratch*. Students are then provided with a Student Guideline that they will progress through and explore (either uploaded to their school site or by using a printed copy). Students will modify values in the code and see how these values create various shapes (changing how many times a loop repeats and the turn angle associated with that specific polygon). The interior angle of each regular polygon will be investigated and explored in hopes of students being able to observe patterns based on the number of sides of a shape and their interior angles. There will be an opportunity for students to get creative with their program as well, allowing them to change the Sprite, background, add sound effects, etc. In the case that there is time leftover, students will be encouraged to work through several extension questions provided. The goal of the extended questions is for students to be able to use the extra-time they have and further investigate more challenging properties of the topic of interior angles.

Discussion Points and Recommended Questions for the Class:

- What did you notice when you changed the numeric values in the existing code provided?
- Did you notice patterns between the inputs and outputs (i.e., the turn angle)?
- How would you use what you have learned today in a real-life setting?

- Throughout the activity, what did you find to be the most challenging part in your learning and why?

Classroom Implementation:

The lesson itself is designed to last for one 75-minute period. Integration involves access to computers. The plan is to have one computer per student but if there is not enough then another possible scenario could be that students can work in pairs or group settings. This was not an issue in our implementation as there were adequate resources but something to be mindful of when incorporating the lesson in future classrooms.

Computational Thinking Affordances in the Activity:

The lesson makes use of the affordance "audience" to reuse and remix and is a critical part of this activity.

"Agency" is incorporated by a low-floor approach introduced at the beginning of the activity. Students are introduced to programming and are tasked with exploring a math concept through a guideline document (easy to know what to do). There is a later point in the activity where agency is used again in the form of a high-ceiling approach, which is portrayed through the extension questions (what math students can reach).

Things to Note:

Overall, the implementation went very well, and we believe it was largely in part to the support of the teachers, both being able to tell when students were struggling even without asking for assistance and with quick thinking/problem solving skills.

The teachers were attentive of our ideas and the overall implementation of the activity. They assisted in troubleshooting with the Chromebooks when certain computers were not working or were too slow.

This activity was developed to be used in a Grade 9 (De-Streamed) mathematics course. However, the content being covered for a first-time programmer could be open to Grades 6-9, varying on the programming and math skill level the students have.

Brief Reflection on Classroom Implementation:

The implementation of the activity was successful and went just as planned. This was our first time implementing an activity into a classroom and being able to do this for six classes provided us with great experience. The only surprise we encountered was the addition of another teacher having a math class added at the last minute, so we were asked to split up again.

The associate teachers wanted to show their class that programming is a tool that can be used in a math classroom to enhance one's learning as this was our goal through our implementation period. Students were challenged to construct shapes and modify their existing code to rotate, flip, or modify their baseline code.

Prior to students starting the activity, we began introducing *Scratch* to the class as many students did not have any prior experience programming. Students were briefly taught various features *Scratch* has to offer and familiarizing themselves with the coding blocks. As taught in MATH 3P41, we were advised to create an activity that was driven towards a hands-on

approach with minimal instruction provided. Thus, with this in mind our activity was primarily a student driven approach. The implementation led us to modify our activity summary very minimally as we had to add a couple more extension questions as some students were very experienced using *Scratch* and had an extra 10-15 minutes to work, thus we incorporated some more challenging problems they could attempt.

Two Major Take-aways:

1. **When designing a lesson, it is important to ensure that all students are challenged.** When dealing with a topic such as introducing coding, where some students have seen code before and some are brand new to the topic. It is especially crucial to create a lesson that is low floor for those who have never seen programming before, and to create a lesson that challenges those who are well versed in programming. I believe our lesson accomplished this well, as regardless of the level of coding skill of the students, they received a challenge.

2. **Helping students is more than just looking for raised hands, sometimes students will not ask for help when they are struggling.** It is important as a teacher to recognize when students are having trouble. This was one of the aspects that the associate teacher helped much with, as they knew their students better than we did, and their experience teaching has allowed them to detect with greater efficiency when students were struggling internally. Working together allowed both us student teachers and the associate teachers to utilize our skillset. As the teachers were new to programming, we were able to help with that aspect, whereas we were new to implementing lessons and working in a classroom, we benefited by learning strategies the associate teachers used in the classroom.

Resources:

- Student Worksheet and Solutions (see next pages)
- Presentation (see next pages and separate pptx file)
- Programming Files:
    - Scratch Activity: https://scratch.mit.edu/projects/648339339/editor/

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.

Activity created by Noor Mahmood and Allan Hickey for MATH 3P41 at Brock University in collaboration with three teachers from the Niagara Catholic DSB under the Math Knowledge Network. Coordinated by Laura Cronshaw and Jeffery Martin (Niagara Catholic DSB) and Dr. Chantal Buteau and Dr. Laura Broley (Brock University). Adapted by Chantal Lof. June 2022.

Mathematics Knowledge Network
Réseau de connaissances en mathématiques

Name: _____. Date: _____.

Beginning to Code with Scratch - Session 1

Set Up Scratch Account:
You will need a *Scratch* account to create, save, and share your *Scratch* projects. Follow the steps below to create a new account. This is important so we can continue our work during our next session.

If you already have an account, feel free to use it.

1. Go to the website: scratch.mit.edu

1. Top right corner click 'Join Scratch' where you will create a 'username' and 'password'

Username: _____.

Password: _____.

Explore Scratch:
Today's session we will familiarise ourselves with what we can do on the programming software called *Scratch*. Use the code that has already been provided to you to explore and modify.

Create
Step 1:
To begin your first *Scratch* program, click on 'Create New Project', and then using the 'Building Blocks' located on the left of your screen create the code below by dragging the blocks over onto the grey area of your screen.

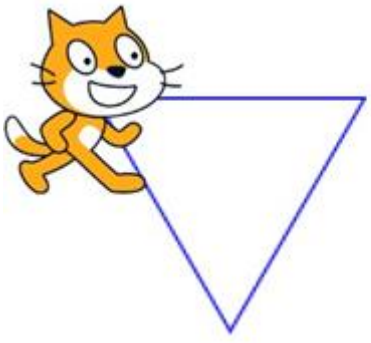
After the code has been copied, click the GREEN FLAG on the right side of your screen to run your code.

What shape did this code create: _____.

After the shape has been created, try exploring *Scratch* by:

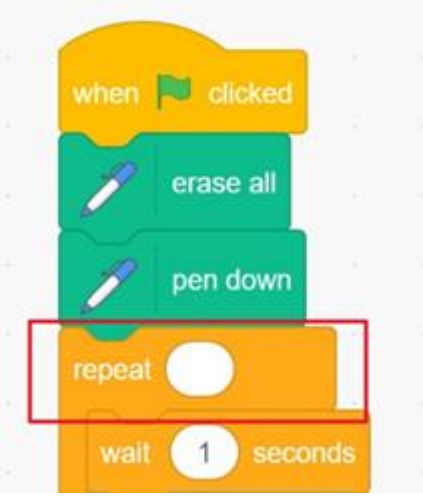
- Changing the 'Sprite' which is the orange cat to a different character of your choosing
- Adding a cool background to your project
- Adding some sort of sound effect to your project

Step 2: Construct an Equilateral Triangle
Using the code that you made in the "Create" part of this activity, modify the code to now construct an equilateral triangle.

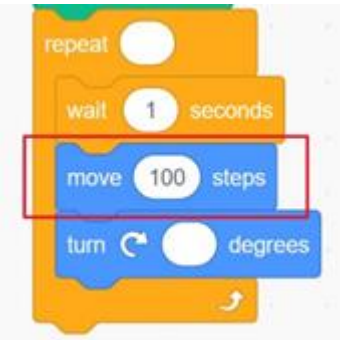


Step 3: Construct regular polygons (Use the following chart below)

Polygon	Turn Angle	Interior Angle	Sum of Interior Angles	Difference in Sums
Triangle				(In the box below calculate the difference between the square and triangle)
Square				
Pentagon				
Hexagon				
Octagon				

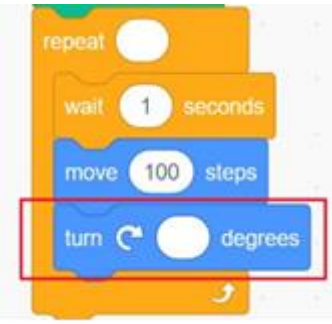


a) What does the selected part (in red) of the code do when creating the shape?



b) What do you notice when you change the value in the red box?

c) Do you notice any patterns between the turn angle and the interior angle columns from the chart?



d) How does the turn angle change when altering the number of sides? Do you notice any patterns?

_____

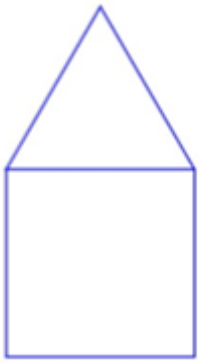
_____

_____

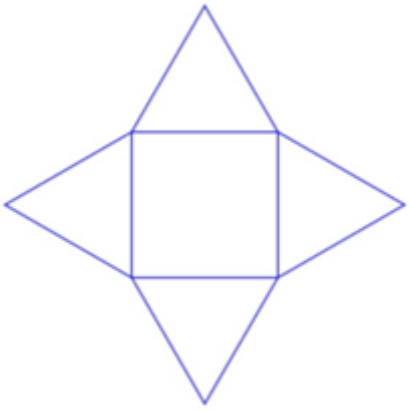
_____

Extended Question (BONUS):
If you have extra time, try these problems out!

a. **Code a program for the drawing below:**



b. **Based on your code from a), try to draw this shape using repeats:**



c. Using your code, try to create a shape that is NOT a regular polygon by adjusting the *repeat* and the *turn* blocks? If you find any, draw it below and write the numbers in the *repeat* and the *turn* blocks.

Activity created by Noor Mahmood and Allan Hickey for MATH 3P41 at Brock University in collaboration with three teachers from the Niagara Catholic DSB under the Math Knowledge Network. Coordinated by Laura Cronshaw and Jeffery Martin (Niagara Catholic DSB) and Dr. Chantal Buteau and Dr. Laura Broley (Brock University). Adapted by Chantal Lof. June

Solutions:
Name: *Insert Student Name* Date: Tuesday, March 29, 2022

Beginning to Code with Scratch - Session 1

Set Up Scratch Account:
You will need a *Scratch* account to create, save, and share your *Scratch* projects. Follow the steps below to create a new account. This is important so we can continue our work during our next session.

If you already have an account, feel free to use it.

1. Go to the website: scratch.mit.edu

1. Top right corner click 'Join Scratch' where you will create a 'username' and 'password'

Username: *Student Username*.

Password: *Student Password*.

Explore Scratch:
Today's session we will familiarise ourselves with what we can do on the programming software called *Scratch*. Use the code that has already been provided to you to explore and modify.

Create
Step 1:
To begin your first *Scratch* program, click on 'Create New Project', and then using the 'Building Blocks' located on the left of your screen create the code below by dragging the blocks over onto the grey area of your screen. After the code has been copied, click the GREEN FLAG on the right side of your screen to run your code.

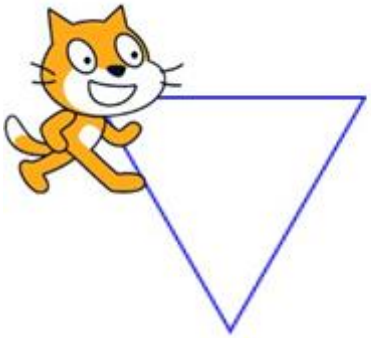
**We provided the existing code above for students to be able to get familiar with the coding blocks (i.e. learning how to drag and connect them like a puzzle piece) - used to get students familiar with Scratch and explore certain commands. **

What shape did this code create: **Square** .

After the shape has been created, try exploring *Scratch* by:

- Changing the 'Sprite' which is the orange cat to a different character of your choosing
- Adding a cool background to your project
- Adding some sort of sound effect to your project

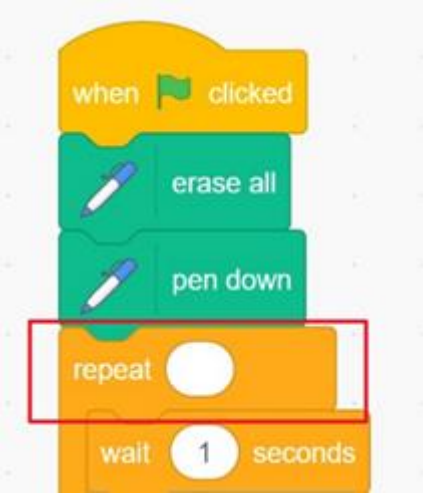
Step 2: Construct an Equilateral Triangle
Using the code that you made in the "Create" part of this activity, modify the code to now construct an equilateral triangle.



Step 3: Construct regular polygons (Use the following chart below) (AGENCY: LOW FLOOR, AUDIENCE: REUSE)

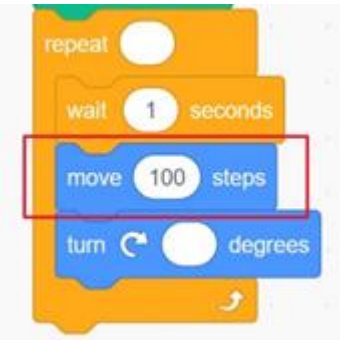
Polygon	Turn Angle	Interior Angle	Sum of Interior Angles	Difference in Sums
Triangle	120	60	180	(In the box below calculate the difference between the square and triangle)
Square	90	90	360	180
Pentagon	72	108	540	180
Hexagon	60	120	720	180
Octagon	45	135	1080	360

**The goal of the chart above is to allow students to see patterns numerically, i.e., turn angle + interior angle = 180, increasing sides by 1 increases the interior angle sum by 180. **



a) What does the selected part (in red) of the code do when creating the shape?

It loops the commands that are inside the "mouth" as many times as the number beside the repeat tells it to. This part of the code informs us of how many "sides" the code will draw.

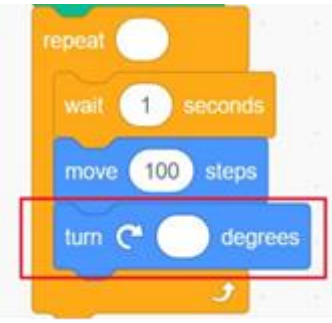


b) What do you notice when you change the value in the red box?

The side lengths of the shapes will be increased/decreased depending on how much you change the value.

c) Do you notice any patterns between the turn angle and the interior angle columns from the chart?

The interior angle is always equal to 180 - the turn angle (and vice versa).



d) How does the turn angle change when altering the number of sides? Do you notice any patterns?

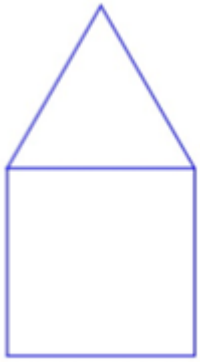
Students can also mention how the more sides the shape has, the lower the turn angle and the higher the interior angle.

Extended Question (BONUS): (AGENCY: HIGH FLOOR, AUDIENCE: REMIX)

**Some students have had experience programming with a software like Scratch before and may be finished earlier than those who are novices to programming. To add a "high-ceiling" aspect to the lesson, the following extension questions provide students with an additional challenge, two of increasing difficulty, and one exploration question. **

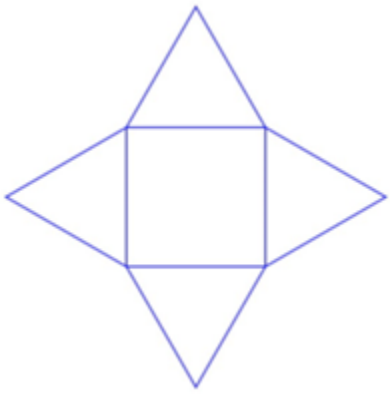
If you have extra time, try these problems out!

a. **Code a program for the drawing below:**



**Through applying the "drawing square" and "drawing triangle" loops from earlier in the lesson, students can try to create a new shape (use-modify-create model). This is a slightly more challenging problem that allows students to think outside of purely editing the code. **

b. Based on your code from a), try to draw this shape using repeats:



**This is a more complex problem meant for those who have already experienced coding before and wish to be challenged. **

Note, this code is not the only solution just one possibility

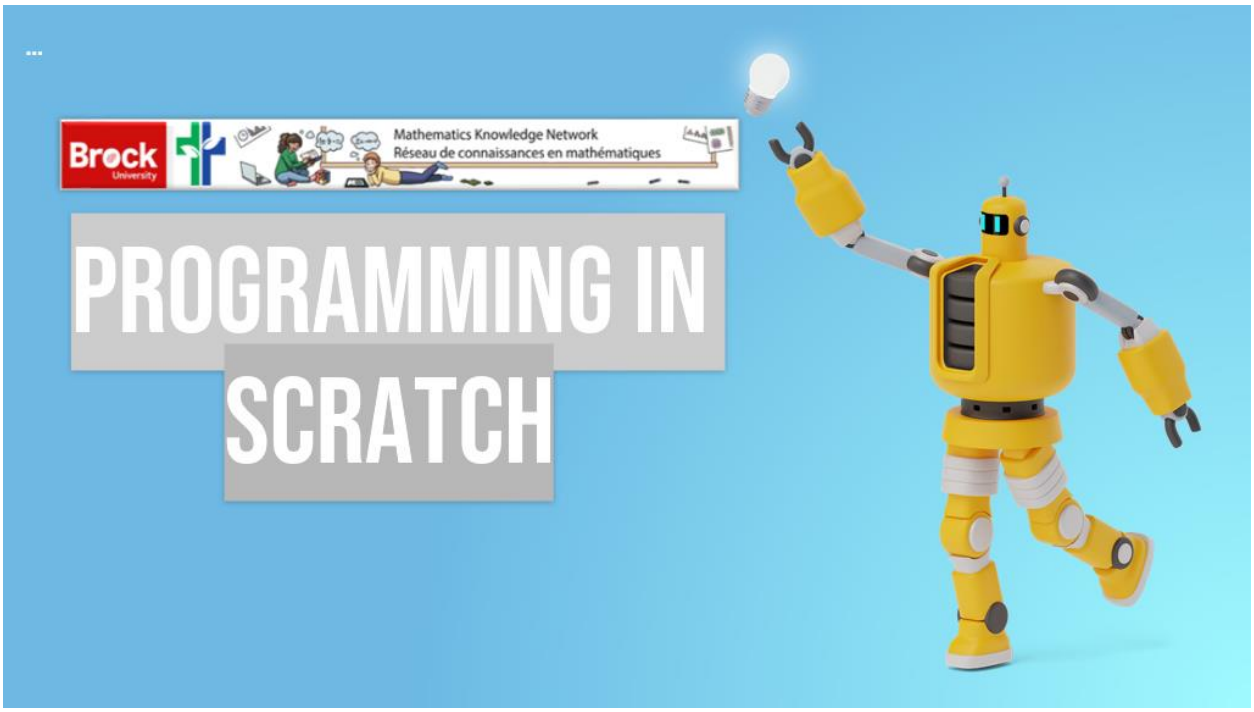
c. Using your code, try to create another shape that is NOT a regular polygon by adjusting the *repeat* and the *turn* blocks? Draw what you created below.

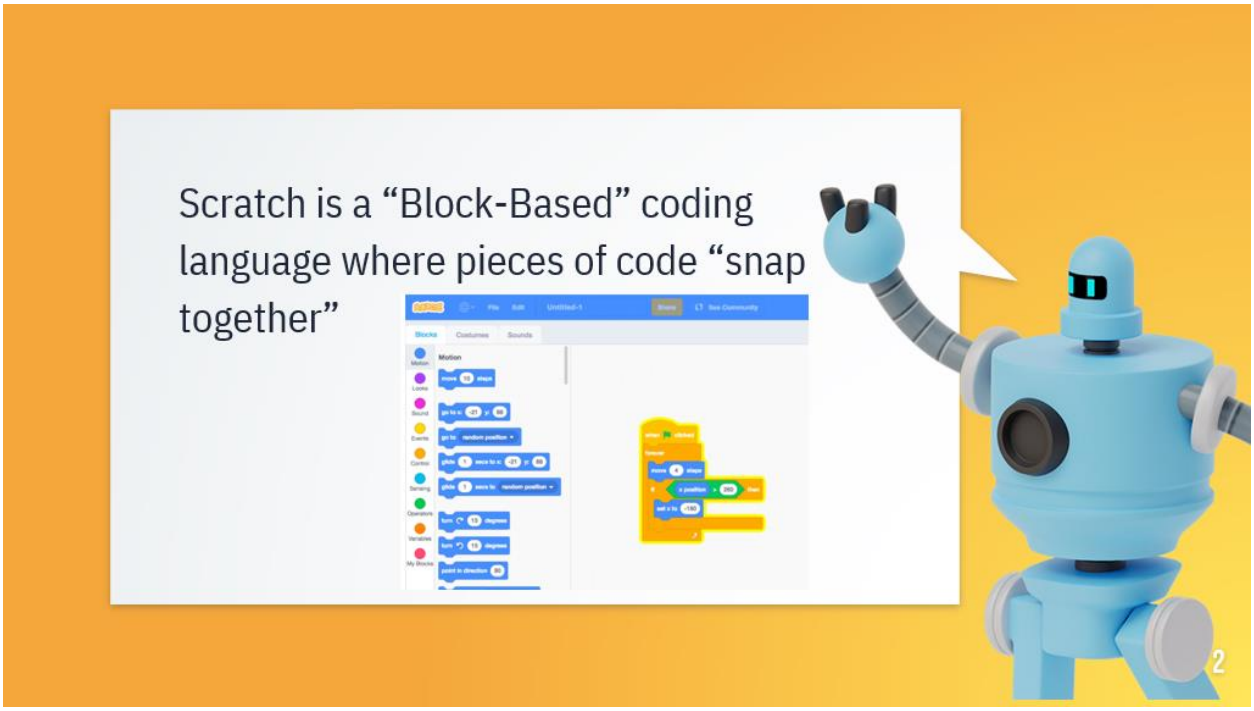
There can be many solutions to this problem. Any number that is a factor of 360, 720, …, 360*n is a possible turn angle for a shape, (i.e., 135 - 8 pointed "star-like" shape, 216 - 5 pointed star) and then however many sides the shape has is what the "repeat" will be.

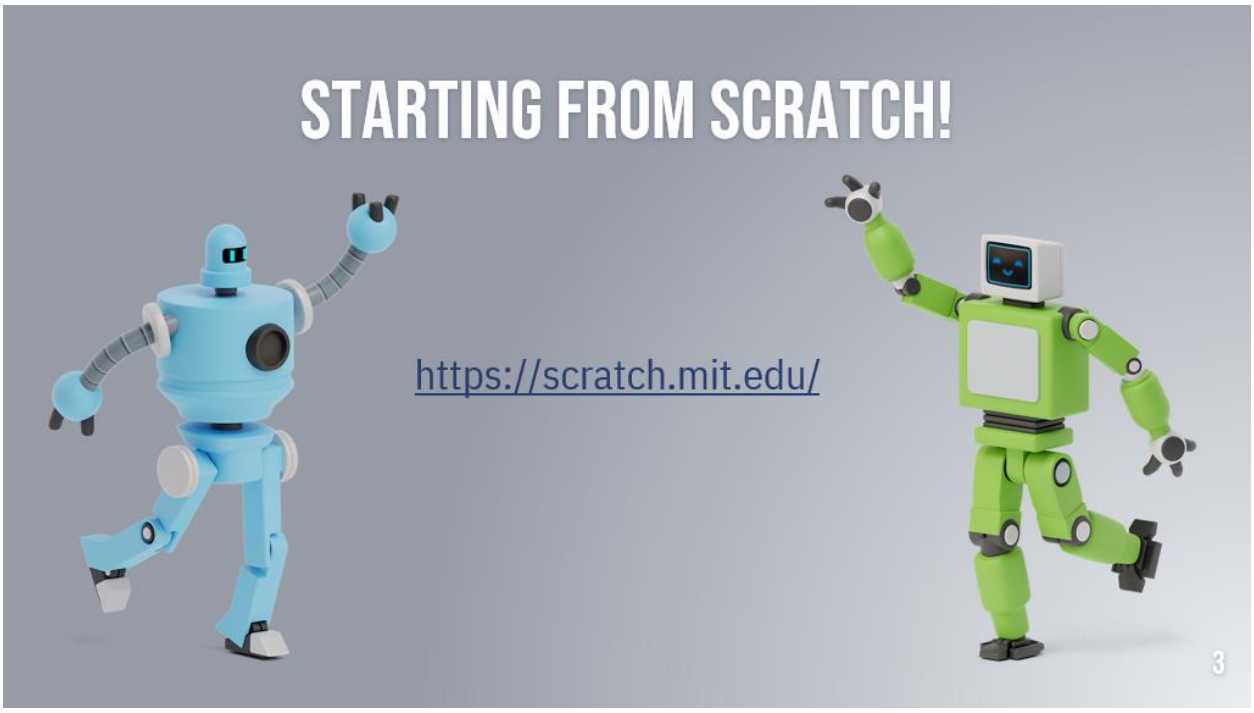
They could also construct shapes like the drawings above if they so wish.

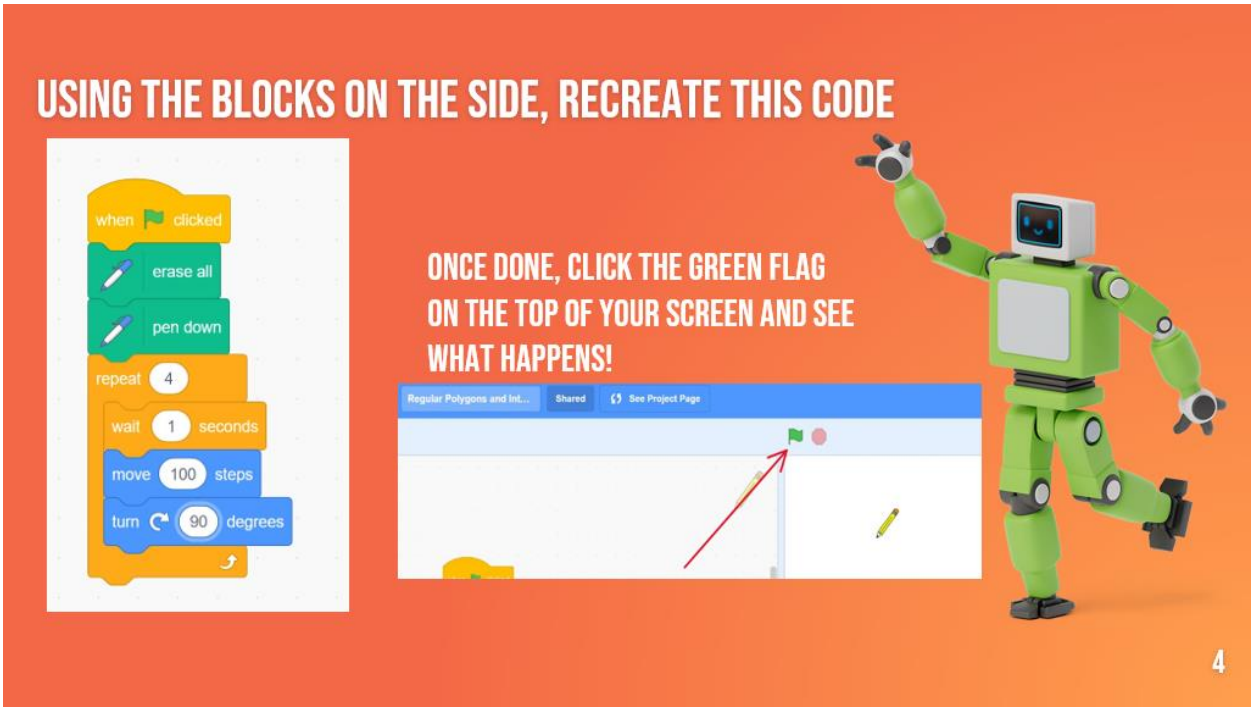
This is a very open-ended question and allows students to explore for any remaining time left in the period.

Activity created by Noor Mahmood and Allan Hickey for MATH 3P41 at Brock University in collaboration with three teachers from the Niagara Catholic DSB under the Math Knowledge Network. Coordinated by Laura Cronshaw and Jeffery Martin (Niagara Catholic DSB) and Dr. Chantal Buteau and Dr. Laura Broley (Brock University). Adapted by Chantal Lof. June


Brock
University
Mathematics Knowledge Network
Réseau de connaissances en mathématiques
PROGRAMMING IN
SCRATCH


Scratch is a “Block-Based” coding language where pieces of code “snap together”
2


STARTING FROM SCRATCH!
https://scratch.mit.edu/
3


USING THE BLOCKS ON THE SIDE, RECREATE THIS CODE
when clicked
erase all
pen down
repeat 4
wait 1 seconds
move 100 steps
turn 90 degrees
ONCE DONE, CLICK THE GREEN FLAG ON THE TOP OF YOUR SCREEN AND SEE WHAT HAPPENS!
Regular Polygons and Int...
Shared
See Project Page
4

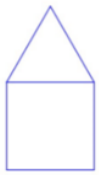
ARE YOU DONE ALREADY?

HERE ARE SOME EXTRA QUESTIONS TO TRY!

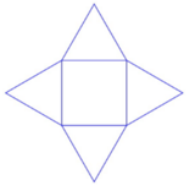
Extended Question (BONUS):

If you have extra time, try these problems out!

a) **Code a program for the drawing below:**



b) **Based on your code from a), try to draw this shape using repeats:**



c) **Using your code, try to create a shape that is NOT a regular polygon by adjusting the *repeat* and the *turn* blocks? If you find any, draw it below and write the numbers in the *repeat* and the *turn* blocks.**

5