



OoR Tech – Session Plan, SPRK Makes Shapes - Part 2

rev 20171206

Grade Range: Readers+

Prerequisite: SPRK Makes Shapes Part 1

(15-20mins)

Refresh from Part 1.

Go over square program again and ensure students are understanding the concept of loops and variables. Quickly discuss regular polygons and robot angle direction changes to create the square and polygons in general.

If EXTRA was explored in Part 1 ($>$ vs \geq) explore this in detail and try out both in the robot. Modify all square programs so that loop is conditional upon ≥ 360 based upon this exploration. Finally copy this program to a new program named "polygon"

(15mins)

Create a new variable called "corner_angle".

Add "corner_angle" into area that originally had the robot heading increment statically by 90 degrees. Add a "set" command to the start of the program that sets the new variable to 90.

Explore with kids how this doesn't change the functionality of the program and robot still creates a square.

(10mins)

Have students change the set value command of "increment_value" to 120 and run the program to see what happens (should create a triangle).

(10mins)

Challenge students to create different shapes. Ask, what value must be placed in "increment_value" to create an octagon (45)? How about a hexagon (60)?

What happens if we place 180 into the set value command?

(5mins)

Wrap up

Program Example:

The image shows a code editor interface with a blue header bar containing a back arrow, a green 'Start' button, and icons for AIM, a menu, and a settings icon. The main workspace contains the following code blocks:

- On Start Program** (black block)
- Set Number** (blue block): `corner_angle = 45`
- Set Number** (blue block): `heading = 0`
- Loop Until** (purple block): `heading >= 360`
- Roll** (cyan block): `heading 64 1s`
- Delay** (purple block): `1s`
- Set Number** (blue block): `heading = heading + corner_angle`

At the bottom, there is a horizontal menu with the following categories: Movement, Lights & Sounds, Controls, Operators, Comparators, Sensors, Events, Variables, and Functions.