Lesson 2.3: Polygons: Any Number of Sides

Objectives

In this lesson, students will:

- ❖ Calculate the external angle between 2 sides of a polygon, based on the number of sides.
- ❖ Gain familiarity with user input
- ❖ Review and practice using variables

Agenda

1. Introduction	10 mins
2. Draw Any Polygon	20 mins
3. Getting User Input	10 mins
4. Wrap Up and Reflections	10 mins

Preparation

- ❏ Remix the solution project and become familiar with it.

Resources & Links

- ❏ Solution project: https://scratch.mit.edu/projects/283388336

1. Introduction

Engage students in a teacher led discussion:

Last time we drew a square that has 4 sides and a triangle that has 3 sides.
What relationship do you see between the number of sides in the polygon and the number of repeats in the loop to draw this polygon? Is it the same?

How many degrees does the pen need to turn in each repeat loop of a square? (90 degrees). We have a loop that repeats 4 times, and each time we turn 90 degrees. What is the total number of degrees once we complete the square? (360 degrees)

How about for a triangle? We repeat the loop 3 times and turn 120 degrees each time. What is the total number of degrees once we complete the triangle? (again 360 degrees)

What can we conclude from this: We know the total number of degrees is 360, and we know the number of sides, can we calculate the number of degrees the pen turns each time? (yes, the number of degrees to turn each time is 360/ number of sides).

So for example, if we want to draw a polygon with 5 sides, how many degrees do we need to turn each time we draw a side? (360/5 = 72 degrees)

2. Draw Any Polygon

Explain to students that in this next activity they will write a program that can draw any polygon by giving it the number of sides. We want to be able to store the number of sides somewhere so that every time we change the number of sides, the code can just pick up the number. In order to do that, we need to store the number of sides.

Prompt students to call out how we store a value in our program. The answer is with a variable. A variable is a place in memory where we can store values. Each variable has a name so we can refer to it.

So lets create a variable to hold the number of sides. Create a variable called **sides**.

Student Activity:

Instructions to give to students:

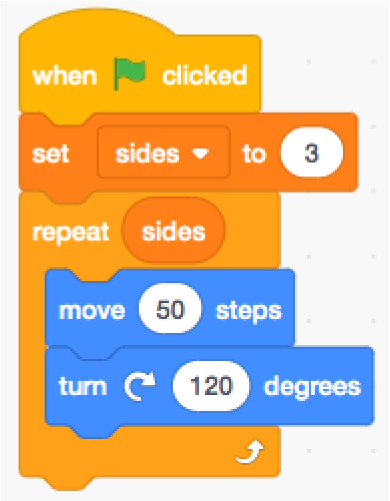
1. Create a new project and give it a name.
2. Create a variable called ***sides.*** The name of a variable should make sense for what it stores.
3. Set the variable to 3 when the green flag is clicked. All commands for creating and setting variables are in the *Variables* category.

Direct students attention to the ***Variables*** category. The value of “**sides”** can be accessed by dragging the round block into any command with an oval shape in it. For example the repeat loop. When you use the variable in your program you can use its value.

Student Activity:

Instruct students to add the instructions to draw a triangle using the variable. Instead of repeating 3 times the instructions to move forward and turn, let’s repeat “sides” times.

Our program now looks like this:



```
when ⚑ clicked
set sides ▾ to 3
repeat sides
    move 50 steps
    turn ↻ 120 degrees
```

Continue the class instruction:

If we set sides to 4, would the code now work to draw a polygon with 4 sides? In order to test our code, we need a pen to draw.

At the beginning of our program we want to make sure the stage and the pen are ready before we start drawing, this means we want to make sure the stage is clear, we have a starting position on the stage and we draw in the direction we want.

Let's add the code to do that so we can test our code:

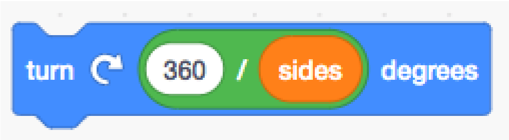
1. Clear our screen with the erase all
2. Pen up so we don't accidentally draw when moving to our starting position
3. Let's start drawing in the upper left corner so we have enough room for our polygon. Go to x: y:-100 100
4. Point in direction 90 (to the right)
5. Put pen down so we can draw
6. Add the wait so we can see the polygon being drawn.

Student Activity:

Instruct students to add the same code to their program and test it and report their findings.

Prompt students: Did we draw a polygon with 4 sides ? No, we did not draw a square, and the cat drew one side it had already drawn.

What did we forget to change? How many degrees do we need to turn in the repeat loop to draw a polygon that has 4 sides? (90). How did we get that answer? By dividing 360 by 4. Remember, our variable "**sides**" stores the number of sides for our polygon. So instead of the number 120 in our turn command, we need to divide 360 by the number of "**sides**". In the operator category we can find a block to do division. In computer science the division character is '**/**'. So let's replace 120 with:



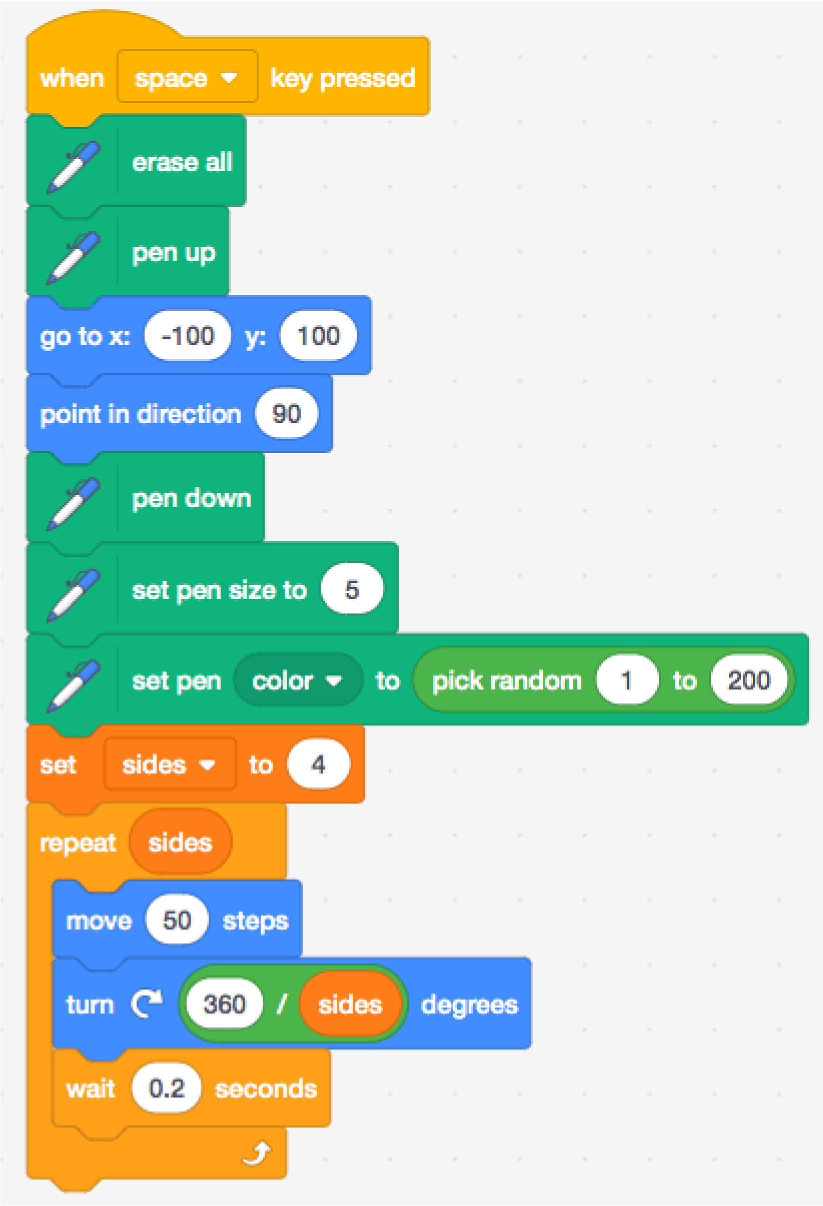
```
turn ↻ 360 / sides degrees
```

Now run the code again to show that it works.

Student Activity:

Instruct students to fix the code by adding the division. Tell them to also add code to draw in different random colors and line sizes as they did with the triangle and square. Remind students to always test their code every time they make a change to their program.

The code at this point should now look like this:

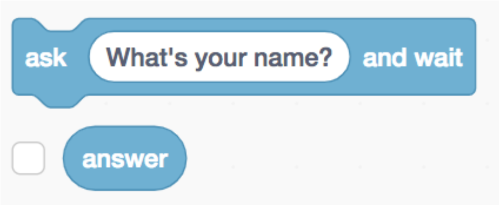


```
when space key pressed
erase all
pen up
go to x: -100 y: 100
point in direction 90
pen down
set pen size to 5
set pen color to pick random 1 to 200
set sides to 4
repeat sides
    move 50 steps
    turn ↻ 360 / sides degrees
    wait 0.2 seconds
```

3. Getting User Input

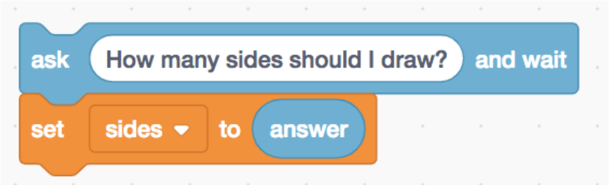
Engage students in a teacher led discussion:
The keyboard is a way for the program to communicate with the user if the program wants information from them. The Keyboard is an input device. What if we ask the user how many sides they want in the polygon?

Look in the Sensing category. The "ask" block is to get values from the program user.
The value is placed in the special block "answer," which is like a variable in that it stores a value that we can retrieve, but it cannot be set other than through asking the user to type a value.



```
ask (What's your name?) and wait
□ (answer)
```

Replace the "What's your name?" message with something more relevant; for example, "Enter the number of sides to draw."
Use the "set <sides> to" block from the Data category to assign a value to the variable.
Drag the "answer" block to the value area of the "set <sides> to" block.



```
ask (How many sides should I draw?) and wait
set [sides ▾] to (answer)
```

We want the user to be able to draw a polygon of any side repeatedly instead of once when the green flag is clicked. How can we do that? We replace the **when green flag clicked** with a keyboard event like we did by clicking "**t**" for triangle. Since the user can draw a polygon of any number of sides, let's use the space bar key.

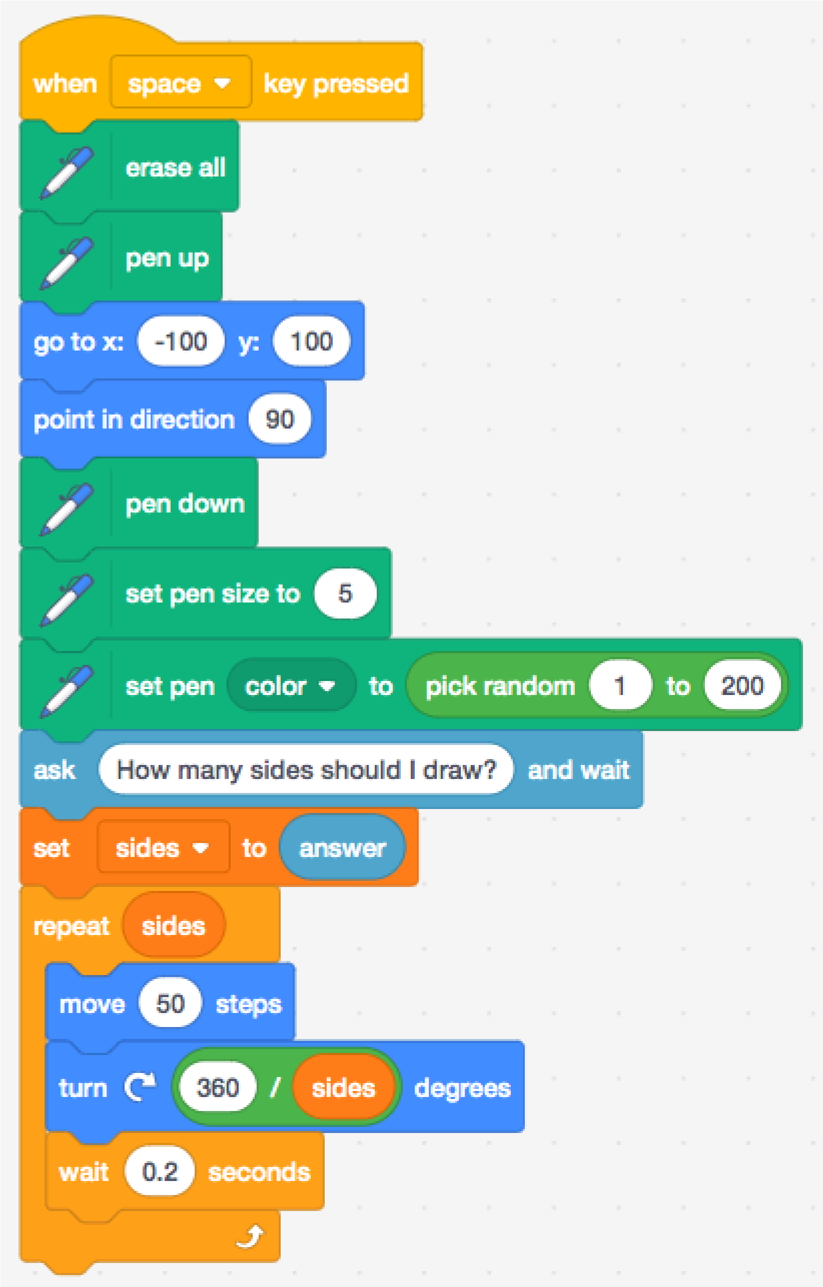
Student Activity:

Instruct students to update their program with the 2 user input changes you just demonstrated. The first user input asking for the user's name and the second asking for how many sides the user wants.

Remind students to save their program. If time allows, give them free coding time to run their code with different values and explore pen sizes and locations.

Solution:

The final code looks like this:



```
when [space ▾] key pressed
erase all
pen up
go to x: (-100) y: (100)
point in direction (90)
pen down
set pen size to (5)
set pen [color ▾] to (pick random (1) to (200))
ask [How many sides should I draw?] and wait
set [sides ▾] to (answer)
repeat (sides)
    move (50) steps
    turn ↻ ((360) / (sides)) degrees
    wait (0.2) seconds
```

4. Wrap Up and Reflections

Reflection Points:

- What did you learn today?
- What did you like about today's activity?
- How do you get input from the user?
- What are the 2 user inputs we used in our program?