Creating and Classifying Polygons

Using a Block-Coding Program

**Patterning and Algebra**

**Unit 3 Line Master 5a**

Follow the link to access the file: What type of polygon?   
<https://scratch.mit.edu/projects/546910232/editor>

**Let’s explore.**

Enter 3 for the number of sides. What polygon did you make?

A picture containing graphical user interface

Description automatically generated

Try again. This time enter 5.   
It seems as like the program isn’t quite right.  
We need to alter the code so that the words   
and drawings are accurate.

A picture containing graphical user interface

Description automatically generated

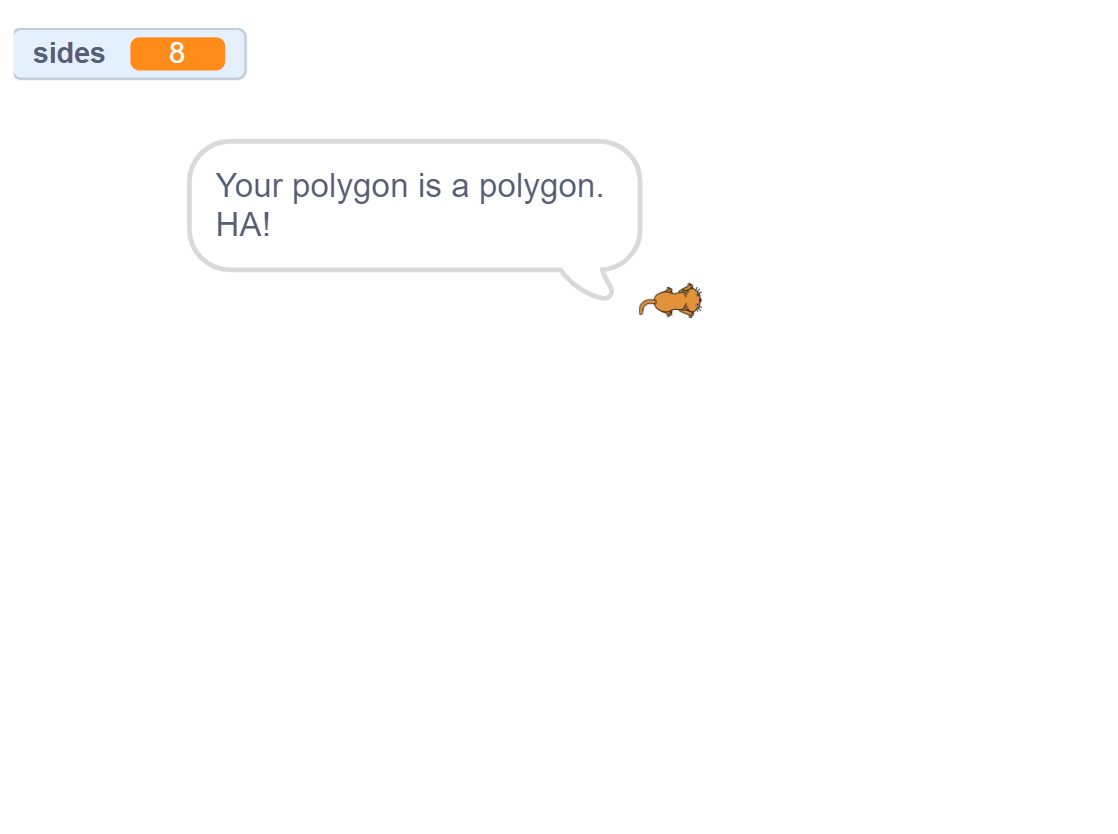
Creating and Classifying Polygons (cont’d)

Using a Block-Coding Program

**Patterning and Algebra**

**Unit 3 Line Master 5b**

Try one more time. Enter 8.



Ha! I guess it would take forever to code this for   
ALL possible polygons.   
But maybe we can code a few more polygons.

**Let’s look inside the code.**

This program uses a conditional statement where something is either   
true or false. We are going to use these to classify different polygons.

It either has 3 sides or it doesn’t.

● If it does, a triangle is named and drawn.

● If it doesn’t, nothing happens.

Notice that the code for 3 sides is working for our program.

You might use this code as you determine how to make the other   
polygons work properly.

You will see that some Blocks have already been created and   
labelled by their polygon name. These are part of the conditional   
statements that this program runs on, but there are mistakes.

Creating and Classifying Polygons (cont’d)

Using a Block-Coding Program

**Patterning and Algebra**

**Unit 3 Line Master 5c**

Text, timeline

Description automatically generated

What are some of the blocks you recognize? Any new ones?

Notice for the triangle code, we used a “repeat 3” block. If we wrote this without a repeat block, it would require more blocks. We’d have to use the “move” and “turn” blocks three times! Repeats help our code to be more efficient.

When we are coding, we try to make our code as efficient as possible. The more experience we have with coding, the more efficient we are to make our applications!

Creating and Classifying Polygons (cont’d)

Using a Block-Coding Program

**Patterning and Algebra**

**Unit 3 Line Master 5d**

**Tips**

This code allows the “answer” to be used in the rest of the code,   
since the classification is based on the answer that is given.   
Graphical user interface, text, application, chat or text message

Description automatically generated

We’ve stored the user’s answer in a variable called “sides.”

You might notice that you can use this variable block (found under   
Variables) to make your code even more *efficient* by incorporating it   
in your repeat values. There’s that word again – “efficient”. Variables are another great way to help us make our code more efficient.

|  |  |
| --- | --- |
| This code incorporates a conditional  statement that means:  “If the answer entered by the user is 3,  then the triangle code will be executed.” | Graphical user interface  Description automatically generated |

This is the resulting code.

|  |  |
| --- | --- |
| This defines the code to create  a triangle.  When you click the green flag,  the main program begins.  The main program “calls” the triangle  code (or subprogram). The code for the triangle is executed  if the user typed 3 for number  of sides. | Graphical user interface, application  Description automatically generated |

Creating and Classifying Polygons (cont’d)

Using a Block-Coding Program

**Patterning and Algebra**

**Unit 3 Line Master 5e**

|  |  |  |
| --- | --- | --- |
| For this conditional statement,  the user is told to enter a new value  if they enter a number less than 3.  This is because we cannot create  a polygon with fewer than 3 sides!  Why is that? | | Graphical user interface  Description automatically generated |
| For the other conditional statements,  you will need to ensure the proper  polygon is named and drawn according  to the number of sides.  For example, if the user enters 4 for  number of sides, a quadrilateral should  be named and drawn: | | Graphical user interface, application  Description automatically generated |
| But what happens when the user  types in 4 now?  Look at the quadrilateral code as  it is currently written: | Graphical user interface  Description automatically generated with low confidence | |

What a mess! We can see several mistakes:

● The polygon is named incorrectly.

● The repeat value is incorrect.

● The angle turn is incorrect.

Make the required changes for the quadrilateral.

Creating and Classifying Polygons (cont’d)

Using a Block-Coding Program

**Patterning and Algebra**

**Unit 3 Line Master 5f**

**What to Do**

Alter all the code so it is accurate.

Remember to change only one value at a time,   
so that you can reflect on the impact of your change.

After you fix the broken parts of the code, try to add more   
blocks to make this program define and draw more polygons.

You will find these in the My Blocks Tab 

|  |  |
| --- | --- |
| You can see that the block for triangle,  quadrilateral and pentagon have already  been created. | Diagram, text  Description automatically generated |

Alter the code for each of the blocks that are there so they are   
accurate with naming, classifying, and drawing.

Try to add more blocks (My Blocks, Make a Block) for other polygons,   
so that more polygons are included in the classification.

Creating and Classifying Polygons (cont’d)

Using a Block-Coding Program

**Patterning and Algebra**

**Unit 3 Line Master 5g**

**Self-check in**

What have you learned about block coding so far?

What is one way to make your code more efficient?

Did you get stuck? If so, what did you do?

Did you turn to your classmates for help? If so, how did they help?

What are you doing to help the learning of others?

This is “hard fun.” What do you think we mean by “hard fun”? What other activities do you do that are “hard fun”?

Go on “spy walks” to see what your classmates have done.