

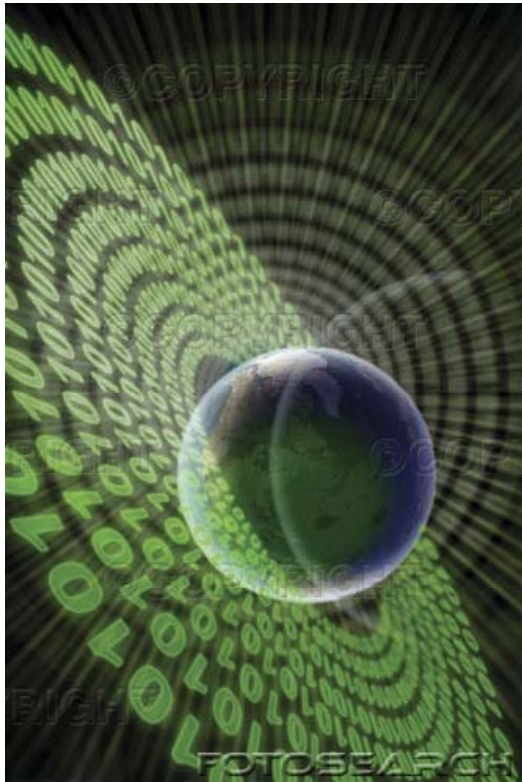


*Drexel-SDP GK-12 ACTIVITY*

## **Science**

### **Scratch Programming and Problem Solving**

#### **Introduction to Scratch - Tessellations**



**Grade Level**      6   (  5   -   8  )

## **Lesson # 2**

### **Lesson Dependency Lessons in the Scratch group**

#### **Time Required: 2 Hours**

#### **Summary**

In this lesson, we build on the first lesson by introducing variables, and then using the motion and shapes idea to draw tessellations on the screen. Students will experiment with angles and see that not all shapes can form tessellations. Students will experimentally determine what dictates that a shape can form a tessellation, and then create a generic program to draw tessellations given the desired number of sides (if possible).

#### **Engineering Connection**

A knowledge of software engineering facilitates in the general approach to problem solving. The goal is to provide students with the skills needed to identify processes appropriate to solving a particular problem. Often students have the requisite technical skills to solve a problem like division, but they do not have the skills required to identify when division is appropriate to solve a problem. Rather, they will only use division when it is explicitly asked of them. Computers are exceptional problem solving tools, but they require the user to direct them in their action. Scratch, as a kid-friendly programming environment, allows students to explore problem solving and obtain immediate feedback.

#### **Keywords**

- Scratch
- Programming
- Software Engineering
- Problem Solving
- Tessellations
- Angles
- Geometry

#### **PA Science Educational Standards**

- 3.1.7A Explain the parts of a simple system and their relationship to each other.
- 3.1.7B Describe the use of models as an application of scientific or technological concepts.
- 3.2.7A Explain and apply scientific and technological knowledge.
- 3.2.7C Identify and use the elements of scientific inquiry to solve problems.

#### **Learning Objectives**

After this lesson, students should be able to:

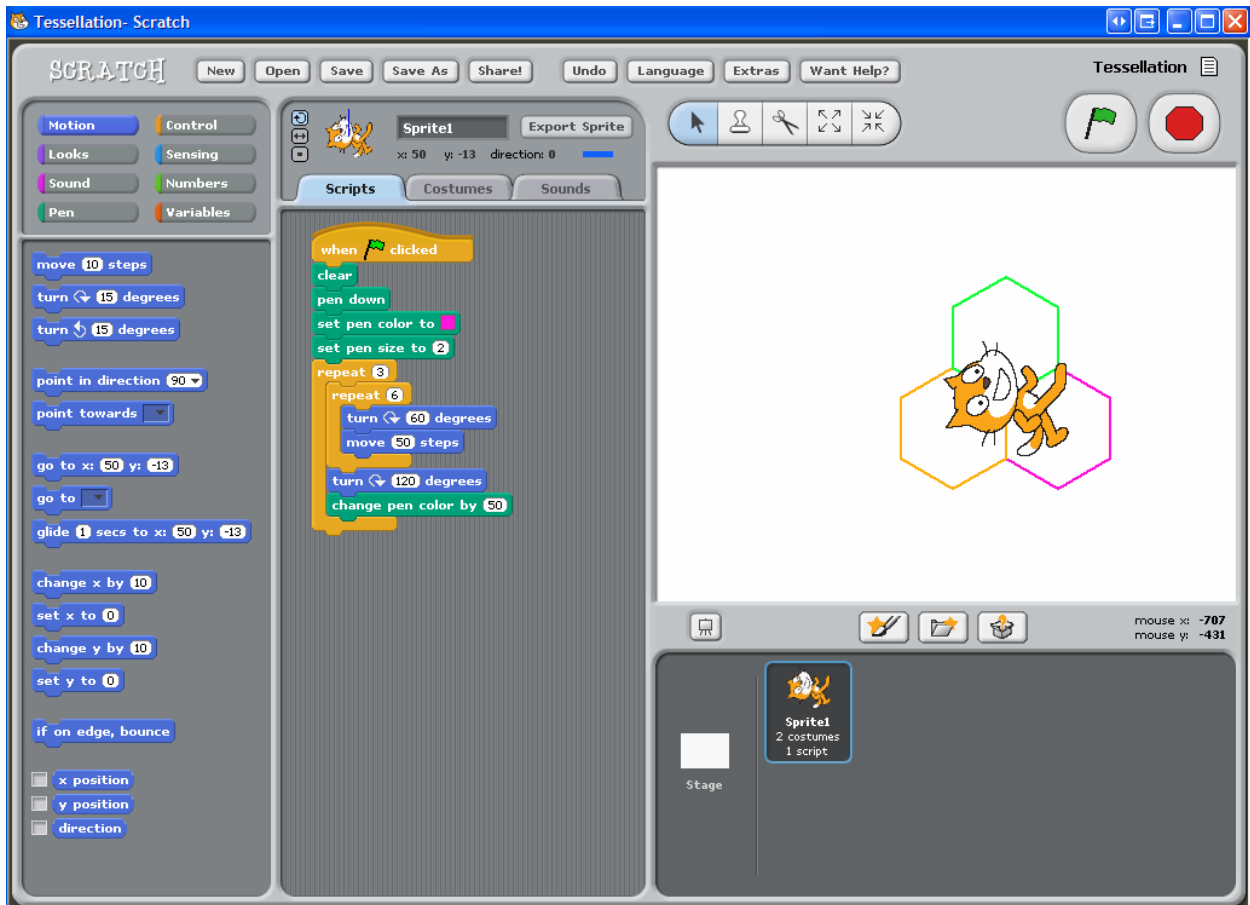
- Create variables in Scratch, and manipulate them.
- Use the pen tool in Scratch.
- Modify a given (or created) Scratch program to draw tessellations using 3 hexagons.
- Determining the interior angle formed by the sides of the hexagon as well as the adjacent hexagons, students should find what determines whether a shape can form a tessellation, and how many shapes are required to do so.
- Modify the specific Tessellation Scratch program to a generic one that uses a variable for the desired number of sides.

## Introduction / Motivation

Last week, we explored how to move the Scratch sprite around the screen. We observed that we made a perfect square by moving the sprite four times in 90 degree angles (and then moving the same distance, of course). We even made a loop out of those steps. This week, we will use that loop to draw a hexagon. How many sides does a hexagon have? Six. That's easy. But how many degrees should he turn this time? Not 90, we needed 90 degree turns when there were four sides to draw. Let students experiment with this to determine that 60 degree turns are needed. Show students the "Pen" commands to clear, put the pen down, and set pen size and color to draw as the Sprite moves. These steps should go outside of the loop and before the program begins, just like the "Face right" step we had in last week's program.

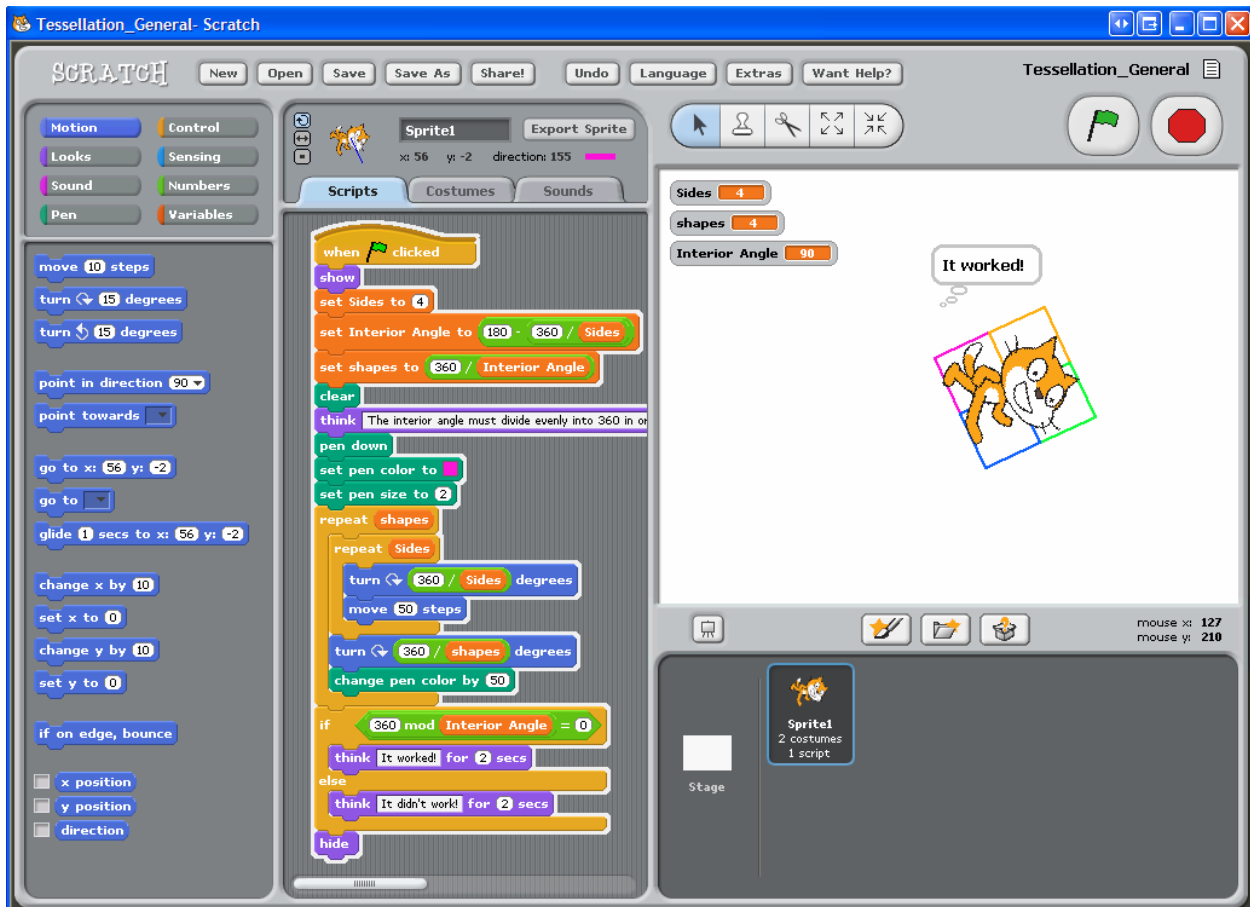
Once one hexagon is drawn, we would like to draw three adjacent hexagons. This would be a tessellation on the hexagon! We can do this by turning and drawing another hexagon. But again, how many degrees would we turn? Again allow students to experiment and guess to determine that, if three turns are made and we wish to make 360 degrees of turn, 120 degrees must be turned each time. The resulting program looks like the one below. We change the pen color between hexagons just to make things interesting.

While we're thinking about it, notice the two loops in the program. There is one loop inside the other. What does each loop represent? The outer loop represents the number of shapes to draw, and the inner loop is the number of sides in each figure. How many total lines will the program draw? It will draw the product of the outer and inner loop numbers! Does this make sense? Well, if you wanted to draw three hexagons, how many lines would you draw? Three hexagons, each containing six sides, so  $3 * 6 = 18$  lines.



Now, how would you modify this program to make a tessellation on a triangle or a square? What would the angle of turn be between each side, and what would be the angle of turn be between each shape? Here in both cases it is 90 for a square, and for a triangle it is a 120 degree turn on the inside, and a 60 degree turn between triangles (or vice versa). See if students can derive a generic formula that for each side in the shape (the inner loop), one turns 360 divided by the number of sides in the shape, and for each shape in the tessellation (the outer loop), one turns 360 divided by the number of shapes in the tessellation. If these divisions can't be done evenly (say, for a pentagon), the figure will not tessellate properly.

Use the "Numbers" tab to manipulate variables to take in the number of sides, compute the number of shapes, and draw them. As a class, add the IF Control statement to check whether or not the shape could be tessellated as per the Generic Tessellation example.



## Materials

- Enough laptops for the students, groups up to 3 are appropriate.
- The Scratch software environment (see References section).
- A laptop and projector combo with Scratch for presenting to the students.

## References

<http://scratch.mit.edu>

### Author

William M. Mongan

### Date

12/2/2007

## Copyright

Copyright 2007 Drexel University GK12 Program. Reproduction permission is granted for non-profit educational use