

Scratch Geometry Lesson Plan

Purpose: This lesson is primarily reinforcement for geometric shape knowledge. It also encourages the development of mathematical thinking and modeling skills, and provides an opportunity to apply shape knowledge in a hands-on manner. It embodies all of the Common Core Standards for Mathematical Practice. And, it introduces a number of Scratch blocks and programming practices that can be reused in future student endeavors.

This lesson was specifically targeted for 3rd Grade, but can easily be adapted for other levels.

Materials: For this lesson students should have access to the Scratch programming environment. Three solid options for using Scratch exist:

- 1) The current version (Scratch 2.0) is run from a web browser using Adobe Flash (works with PC, Macintosh, Linux, and Android platforms, but not iPads). It can be reached at: <http://scratch.mit.edu/>
- 2) A downloadable, stand-alone executable exists for the previous version (Scratch 1.4) with installers for Mac OS X, Windows (XP, Vista, 7, & 8), and Linux. It can be downloaded here: http://scratch.mit.edu/scratch_1.4/. This version lacks some functionality (Sprite cloning and custom blocks)
- 3) For iPads, you can use an HTML5 variant of Scratch developed by Berkley: Snap! It is based on Scratch 1.4, but adds its own implementation of custom blocks. It can be found here: <http://byob.berkeley.edu/>

In addition, you should print out enough copies of “Drawing a Triangle with Scratch” and “Drawing Any Regular Polygon with Scratch” for each student to have their own copy.

The folder also contains multiple Scratch projects. These are solutions to drawing different regular shapes, for you to familiarize yourself with the process. They should not be given to students, as they will be writing the projects themselves.

Background: In this lesson, students will draw regular polygons using the Scratch programming language. In doing so, they will practice problem solving and algorithmic thinking, by abstracting the final shape into a sequence of instructions of how to draw that shape. Additionally, by attempting to develop a general algorithm for drawing an arbitrary polygon, they will learn about the relationship of sides and angles in a regular polygon (i.e. that the sum of the interior angles must be 360 degrees and that sides must be of equal length).

Further, students may observe that circles can be approximated by a regular polygon with a high number of sides – a discovery that is an important foundational experience for later developing an understanding of calculus.

Finally, this exercise exposes graphics programming techniques which are the basis of all modern graphical user interfaces.



Objectives: This is a bulleted list of what they will do during the lesson and know or be able to do when they complete the lesson. Keep in mind, the procedure or directions will be in another section. If you are tying your lesson to K-12 Standards, add that here.

- Mathematics objectives
 - Draw regular polygons (CCSS.Math.Content.2.G.A.1) using the Scratch programming environment (CCSS.Math.Practice.MP4, CCSS.Math.Practice.MP5)
 - Work with other students to develop an algorithm for drawing a regular polygon with an arbitrary number of sides (CCSS.Math.Practice.MP1, CCSS.Math.Practice.MP2, CCSS.Math.Practice.MP3, CCSS.Math.Practice.MP7, CCSS.Math.Practice.MP8)
 - Recognize that a circle can be approximated by a regular polygon with a large number of sides (CCSS.Math.Practice.MP4, CCSS.Math.Practice.MP8)
- Scratch objectives
 - Use the Pen blocks
 - Use the Question and Answer blocks
 - Use the Repeat block
 - Use the algebra operation blocks (division & multiplication)

Discussion questions:

What makes a shape regular?

It must have sides of equal length, and the sum of interior angles equaling 360 degrees. These requirements are a key reason to why we can write a general algorithm to draw any regular shape.

Can we draw a non-regular shape?

Indeed you can – any shape can be drawn, but probably won't be generalizable (extended for any number of sides) the way a regular shape can be. There is a five-pointed star example in the folder.

How can you draw a circle?

The same algorithm used to draw a regular polygon can be used to approximate a circle – adding more and smaller sides gets you closer and closer to a true circle.

Is such an “approximate” circle really a circle? This is a good question for sparking discussion – but don't expect a consensus (even mathematicians and logicians argue this). But such approximate circles are used every day by scientists, engineers, surveyors, and other professionals. Also, every circle you see drawn on your computer screen or printed on a printer is approximated in this fashion.

Procedure:

Part I: Creating a triangle-drawing algorithm

Distribute the handout “Drawing a Triangle with Scratch” for each of your students, and have your students work through the procedure. You may either lead the class (demonstrating with a



projector), or have the students work through it at their own speed, addressing questions as they arise.

This portion of the lesson is intended to be heavily scaffolded; the most important goal here is that the students are comfortable programming using Scratch and using the blocks involved, in preparation for Part II. Ideally, students should each have their own machine and every student should complete the step before moving on.

Part II: Creating additional regular polygon drawing algorithms

Once the students have created the triangle-drawing algorithm, challenge them to create scripts to draw other regular shapes – rectangles, pentagons, octagons, etc. You may either have all students do the same shapes, or assign different shapes to each. Students who finish quickly can be given additional shapes while the rest of the class works. For especially gifted individuals, you might consider non-regular shapes as well, such as a five-pointed star.

Again, this portion is ideally an individual activity. It is important that each of the students has the opportunity to write an algorithm script on their own; simply observing does not result in the deep-seated learning experience necessary.

Part III: Discussing the Algorithms

As a class, discuss the algorithms the students have created. If possible, have each student demonstrate one of their algorithms, and describe the process by which they wrote it. It is likely you will see variations in how the algorithms are written – this is perfectly fine, provided the desired result was achieved. Have the students identify the similarities and differences in the algorithms for drawing different shapes.

Part IV: A General Regular Shape Drawing Algorithm

Divide the class into small groups – mixed learners work best, but be sure that the groups are not being dominated by one student. Distribute the second handout “Drawing Any Regular Polygon in Scratch.” Challenge the groups to write a single script that can draw *any* regular polygon – you or the handout can provide instruction on using Ask and Answer blocks to determine the number of sides to draw, as well as assist with using algebra operator blocks, but let the students develop the algorithm to draw the shapes themselves.

A good hint is that the sum of the interior angles of any closed polygon has to equal 360 degrees.

Once the students finish the general solution, ask them to draw a shape with 50 sides. With 100. Ask them how they would draw a circle (see discussion questions above).

Extension or Exploration: Beyond drawing the shapes mentioned here, there are many other ways you can utilize Scratch in teaching Geometry:

- Divide shapes into parts (CCSS.Math.Content.2.GA.2)
- Add non-regular polygons (CCSS.Math.Content.3.GA.1)

