CS161: Introduction to Computer Science
Homework Assignment 11
Due: Wednesday 5/15 by 11:59pm (our finals time)
Note: This is an individual assignment

If Java is like an industrial manufacturing plant, then Processing is like a maker space. Processing uses a subset of Java to enable quick prototyping of ideas. Processing is both an *interactive development environment* or "IDE" for short – just like BlueJ – as well as a collection of libraries that help you write visual programs called "sketches".

In this assignment, you will write a sketch that uses recursion to draw a Sierpinski Triangle.

▬▬ The Sierpinski Triangle ▬▬

The Sierpinski Triangle is a fractal that can be drawn using the following algorithm:

1. Draw an equilateral triangle (i.e all sides have the same length)

2. Compute the midpoint of each side of the triangle

3. Remove the triangle defined by the midpoints. This leaves behind 3 smaller triangles: one on the bottom left, one on the bottom right, and one on the top.

4. Repeat steps (2)-(3) recursively for each of the 3 smaller triangles.

For this assignment, you'll create a sketch that draws an interactive Sierpinski Triangle. Your finished assignment will look something like the image below, where the user can slide the value in the slider at the top of the screen to change the depth of the recursion (i.e. the slider controls the base case):
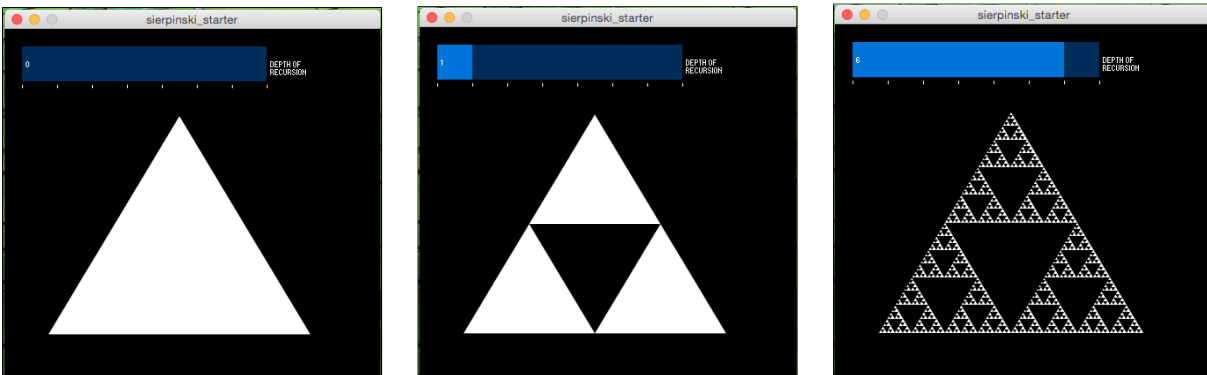
Download the starter code for the assignment. The starter code uses a library called `controlP5` to create the slider. This library may already be installed for you. You can check by going to "Sketch > Import Library". If you see the "ControlP5" library listed at the bottom then it is already installed. Otherwise, go to "Sketch > Import Library> Add Library". This will bring up a list of all the libraries you can install. Search for and select "controlP5" and then click "Install" in the lower-right corner. This will download and install the library for you. You may need to close and reopen the sketch after installing the library.

In the `setup()` method, the slider is constructed and initialized. In the `draw()` method, the initial white triangle is drawn and the recursive `sierpinski()` method is called.

The `sierpinski()` method is a recursive method that you will implement. This method should use recursion to draw the Sierpinski Triangle. Since there is no way to "remove" a triangle (as the above algorithm requires), you can instead draw a triangle (whose vertices are the midpoints) that matches the background color. This will give the illusion that you have removed a piece of the triangle.

The midpoint between two points is simply the average of their x and y coordinates. For example, if I have a point at $(20, 100)$ and another point at $(50, 10)$, then the midpoint is:

$$x_{\text{mid}} = \frac{20 + 50}{2} = 35$$
$$y_{\text{mid}} = \frac{100 + 10}{2} = 55$$

So, the midpoint is $(35, 55)$. It is sufficient to use integer division for this application.

Feel free to play around with your code: change the colors, re-order the lines of code inside of the `sierpinski()` method, draw different shapes, etc. With recursion, it's always interesting to see what sorts of weird patterns you can produce.

━━━━━ Submitting your homework assignment ━━━━━

You should submit your `sierpinski` folder, renamed, and zipped. Upload to Moodle.