Lab 03

In this lab we will familiarize ourselves with the structure of a WebGL application.

a. Enter the x and y coordinates for three points between (-1 , -1) and (1 , 1) in `Lab03Points.js`. Note that these are two-dimensional, there are no z coordinates needed, so the `vec2` definition is used, and `points` is then an array of `vec2` objects.
b. Change the value of `gl_PointSize` in the vertex shader in `Lab03Points.html`.
c. Note that data sent to the vertex shader from the application uses the in variable `aPosition`. The `a` is for "application." When we send data from the vertex shader to the fragment shader, the prefix would be `v`.
d. `aPosition` was defined in the application as a `vec2`, but it's incoming to the vertex shader as a `vec4`. GLSL can handle conversions like these automatically.
e. Change the color defined in the fragment shader. Initially the RGBA values are (1.0, 0.0, 0.0, 1.0), or all red, no green, and no blue. The fourth value, the Alpha, will be used when we discuss lighting. For right now, we will use 1.0.
f. Add more (x,y) coordinates to the `points` array.