Lab 05

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

a. Enter the x and y coordinates for several points between (-1 , -1) and (1 , 1) in `Lab05ColorArray.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 `gl_PointSize` in the vertex shader in `Lab05ColorArray.html`, making it much larger, 20 or so.
c. Change the `vColor` assignment in the vertex shader so the color depends on the position of the vertex. The x and y values are between -1 and 1, while the colors are specified between 0 and 1. A simple mapping is to add 1 to the location, and then divide by 2. Rather than doing that multiple times, for each x, y (and possibly z) location, GLSL has a shortcut. If `aPosition` is the position value from the application, `aPosition.xyz` is a `vec3` comprised of the first three components of `aPosition`. The `vec4` constructor is then used to add the fourth color component, the Alpha, to the output color.

```
vColor = vec4((1.0+aPosition.xyz)/2.0,1.0);
```