CS-396 **Fall 2018**

Project 00: Vector Drawing Program

Total Points: 100 pts Due: September 21, 2018

DIRECTIONS

WebGL is based on vector graphics (defining the shapes and their properties instead of labelling individual pixels with a certain color). In this homework we will be creating a vector drawing program where the user can draw vector shapes using points, lines, and triangles of selectable (or random) colors.

The program will allow the user to select one of the WebGL drawing modes (points, lines, line strip, line loop, triangles, triangle strip, and triangle fan) and current vertex color which will be used to add shapes to the canvas display. The selection of the drawing mode should be a drop-down menu while the vertex color should be done using 3 slider controls plus a button that causes a random color to be chosen. Then the user can click inside the WebGL canvas to add vertices. In modes that are not points nothing may show up during the first click or two which is fine since some shapes require a minimum number of points before they draw anything. Whenever the drawing mode or color is changed all previously drawn shapes must remain as-is and future clicks will draw with the new mode and color.

The currently selected color must be visually shown to the user somewhere in the WebGL canvas before they click. Shapes should use smooth shading, i.e. colors should be interpolated between vertices. You may assume that at most 100,000 vertices will ever be drawn in the program.

GRADING

For full credit, be sure to follow the directions above. There are 100 pts and they are broken down as follows:

- 10 pts for having a dropdown menu, 3 sliders, and a button properly labelled with the correct text
- 10 pts for having the button randomly select a color from all possible colors and updating the sliders
- 10 pts for showing the currently selected color visually before clicking (*hint:* add an extra shape to the set of shapes drawn that is in one of the corners that you then update the color of)
- 15 pts for proper pre-allocation of the buffers to store the positions and colors and adding vertices and the currently selected color as the user clicks
- 15 pts for drawing shapes as soon as there are enough vertices and drawing the correct shapes depending on the mode (including having points be larger than size 0)
- 20 pts for keeping all previously drawn shapes and colors as-is when the drawing mode or color changes
- 20 pts for good coding style and documentation/comments

APPROACH

1. Get just one mode (such as line strip) of a solid color working first, growing as you click each new point. Get it working properly by growing the buffer properly as each point is added.
2. Next make every vertex a random color. This requires each vertex to also have a color attribute and for the color buffer to grow as the vertex buffer grows.
3. Make the color user-selectable/user-randomizable.
4. Add the mode drop-down. You will have to set up variable(s) that allows you to keep track of the number of vertices drawn with each mode in the order drawn so that the appropriate number can be draw with each mode. During the rendering you will need to call `gl.drawArrays` multiple times.
5. Complete finishing touches like color indicator, documentation, nice interface.