CSCI-331 **Spring 2023**

Project 1: Vector Drawing Program

Total Points: 100 pts Due: Wednesday, February 22

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 and color are done HTML input controls (provided). The user clicks 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. You may assume that at most 100,000 vertices will ever be drawn in the program.

The HTML document along with the `init()` and `initProgram()` functions are completed for you. Three other functions called from `init()` are declared but empty. You will likely need to add additional functions and global variables. You will likely need to look up additional JavaScript functions such as `substr()` or `parseInt()`.

GRADING

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

- 20 pts for proper pre-allocation of buffers and adding vertices and currently color as the user clicks
- 20 pts for drawing the correct shapes depending on the mode and not keeping previous shapes
- 20 pts for keeping all previously drawn shapes and colors as-is when the drawing mode or color changes
- 20 pts for not keeping any unnecessary data within JavaScript but only in the GPU buffers
- 20 pts for good coding style (i.e. reduce code duplication) and documentation/comments
- EC: 10 pts for supporting downloading the drawing as a JSON file and later uploading an appropriate JSON file to reload the image, you will want to look at gl.getBufferSubData(), creating JSON files for download on the client-side, and reading files client-side.

APPROACH

1. Assume the mode and color are always the same (just pick them) and having it grow the drawing as you click each new point. Get it working properly by "appending" to the buffer properly as each point is added.
2. Make every point added use the current color. You may need a change event listener.
3. Listen for the drop-down to change and advance to the next series of shapes. You will have to set up variable(s) that allow 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, once for each time the mode was changed.
4. Clean up your code with comments, documentation, etc.