Lab 6 Group 1

Yashiya Minor
Noah Parks
Adriel Campos
Henry Fiantaca

```
//*************************global variables*****************//
// Canvas: reference to the canvas in the DOM
// draw_points_bttn: reference to the Draw Points button in the DOM
// updt_world_bttn: reference to the Update World button in the DOM
// world_xmin_text: reference to the World XMin text input in the DOM
// world_ymin_text: reference to the World YMin text input in the DOM
// world_xmax_text: reference to the World XMax text input in the DOM
// world_ymax_text: reference to the World YMax text input in the DOM
// msg_area: reference to the Message Area element in the DOM
// draw_mode: toggle state of the draw mode
// vertices: list of vertices in world coordinates
// colors: list of colors assigned to vertices
// world_xform_pos: position of world2NDC_xform in shader program
// counter: number of points stored
// world_xmin = 0, world_xmax = 100
// world_ymin = 0, world_ymax = 100



// This function will load when the page loads
window.onload = function init(){
      // get references to relevant HTML elements in global variables
```

```
    // Initialize WebGL on the canvas element
    // set up the viewport

    // set the clear color

    // compile shader program using shaders from the html

    // create GPU buffer for vertices
    // bind to new vertex buffer
    // associate vertex buffer with vPosition in shader program
    // format vertex buffer: vec4, floats, non-normalized, stride 0,
offset 0
    // enable vertex buffer

    // create GPU buffer for colors
    // bind to new color buffer
    // associate color buffer with vColor in shader program
    // format color buffer: vec4, floats, non-normalized, stride 0,
offset 0
    // enable color buffer

    // get position of world2NDC_xform in the shader program

    // call setEventHandlers() to enable interaction
    // call render() to begin render loop
}
```

```
//This function will draw the points based on selected world coords
function render() {
	// clear the view
	// draw array of vertices, offset 0, counter points
	// schedule next render call
}

// This function will handle all of the event handlers such as when the user clicks on a button and it should change the mode
function setEventHandlers(){
	// Update World Coords when button is clicked
	updt_world_bttn.onclicked = function (evt) {
		// if not all values in world_*_text are all numbers
			// alert() and return

		// get value for world_ymax_text on change
		// get value for world_ymin_text on change
		// get value for world_xmax_text on change
		// get value for world_xmin_text on change

		// update transform with xform_world()
	}

	// Add new vertex when user clicks on the canvas
	canvas.onclicked = function(evt) {
		// get world coords from getMousePosInWorld()
		// add coords using generatePoint()
```

```
            // add coords (devices and world) to the msg_area
      }
}


function xform_world()
{
      // generate a xform matrix to convert world coords to NDC
      // get translation from (world_xmin, world_ymin) to (-1, -1)
using Translate() from Lab05
      // get scale of (1/(xmax-xmin), 1/(ymax-ymin)) using Scale() from
Lab05
      // get complete xform by using matMult() from mat_vec.js


      // use gl uniform function to update the xform on the GPU
}


function generatePoint(xCoordinate, yCoordinate)
{
      //bind array buffer
      //Set point position, push vertices
      //bind array buffer
      //Set point colors, push color values
}


function getMousePosInWorld(canvas, evt)
{
    // get the bounding rectangle of the canvas
    var rect = canvas.getBoundingClientRect(),
```

```
      // get the scale of the canvas to the rectangle (browser
scaling)

      xScale = canvas.width / rect.width,

      yScale = canvas.height / rect.height,

      // get the pixel position of the click in the canvas (eventpos -
rectangle)

      xPix = (evt.x - rect.left) / xScale,

      yPix = (evt.y - rect.top) / yScale,

      // map the pixel position to World coords (0, width/height) =>
(x/ymin, x/ymax)

     //
https://stackoverflow.com/questions/53897723/mapping-float-values-into
-a-new-value-within-a-specified-range-in-c-or-python#53897806

      x = world_xmin + xPix * (world_xmax-world_xmin) / width,

      y = world_ymin + yPix * (world_ymax-world_ymin) / height;



    // return pos

    return [x, y];

}
```