Assignment #1: Simple Graphics Program with Keyboard Inputs

Shane Steiner
T00622768

1/27/2021

Problem descriptions:
In this problem we need to create different 2d shapes and change their color.

```
<html>
  <style>
    div {
      width: 200px;
      height: 100px;
      border: 1px solid black;
    }
    </style>

<script id="vertex-shader" type="x-shader/x-vertex">
#version 300 es

in vec4 aPosition;//attribute
in vec4 aColor;//attribute

out vec4 vColor;//vertex

void main()
{
  gl_Position = aPosition;
  vColor = aColor;
}
</script>

<script id="fragment-shader" type="x-shader/x-fragment">
#version 300 es

precision mediump float;

in vec4 vColor;
out vec4  fColor; //fragment
//u is uniform

void main()
{
  // fColor = vec4( 1.0, 0.0, 0.0, 1.0 );
```

```
   fColor = vColor;
}
</script>

<script type="text/javascript" src="../Common/initShaders.js"></script>
<script type="text/javascript" src="../Common/MV.js"></script>
<script type="text/javascript" src="triangle.js"></script>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></s
cript>
<body>
  <canvas id="gl-canvas" width="512" height="512"> </canvas>
  <button id="initalButton">Click for first initial</button>
  <div onmousemove="myFunction(event)" ></div>
  <p id="demo"></p>

</body>

</html>
```

```
"use strict";

var gl;
var points, pointsSquare, circlePoints;

var colors = [1, 0, 0, 0, 1, 0, 0, 0, 1];
var program;
var bufferId;
var cBuffer;
var squareTryangleCircle;
var s = [];

window.onload = function init() {
    var canvas = document.getElementById("gl-canvas");

    gl = canvas.getContext('webgl2');
    if (!gl) { alert("WebGL 2.0 isn't available"); }
```

```
    //
    //  Configure WebGL
    //
    //clip cordinants
    gl.viewport(0, 0, canvas.width, canvas.height);
    gl.clearColor(0.7, 0.7, 1.0, 1.0);

    //  Load shaders and initialize attribute buffers

    program = initShaders(gl, "vertex-shader", "fragment-shader");
    gl.useProgram(program);

    //
    //  Initialize our data for a single triangle
    //

    // First, initialize the  three points.

    points = new Float32Array([-1, -1,
        0, 1,
        1, -1,
    ]);

    pointsSquare = new Float32Array([-0.75, 0.75,
        0.75, 0.75,
        0.75, -0.75, -0.75, -0.75
    ]);

    const numVerts = 100;
    var radius = 0.8
    circlePoints = [];
    for (var i = 0; i < numVerts; i++) {
        var u = i / numVerts;
        var angle = u * 3.14159 * 2.0;
        var pos = vec2(Math.cos(angle) * radius, Math.sin(angle) *
radius);
        circlePoints.push(pos);
    }
```

```
    squareTryangleCircle = "circle";
    var colors = [1, 0, 1, 1, 0, 1];
    var numPoints = numVerts;
    for (var i = 0; i < numPoints; i++)colors.push(1, 0, 1);


    var bufferId = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
    gl.bufferData(gl.ARRAY_BUFFER, flatten(circlePoints), gl.STATIC_DRAW);

    var aPosition = gl.getAttribLocation(program, "aPosition");
    gl.vertexAttribPointer(aPosition, 2, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(aPosition);

    cBuffer = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, cBuffer);
    gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(colors),
gl.STATIC_DRAW); //new Float32Array(colors)

    var aColor = gl.getAttribLocation(program, "aColor");
    //3 points colors?
    gl.vertexAttribPointer(aColor, 3, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(aColor);

    render("red");

    window.addEventListener('keydown', this.checkKey);
    $("#initalButton").click(displayLetter)

};

//function to set rendering peramiters
function render(colorName) {
    var numPoints;
    var colors = [];
    var colorArray = [];

    if (colorName == "red") colorArray = [1, 0, 0];
```

```
    else if (colorName == "green") colorArray = [0, 1, 0];
    else if (colorName == "blue") colorArray = [0, 0, 1];
    else if (colorName == "favourite") colorArray = [.7, .1, 1];
    else if (colorName == "random") colorArray = [Math.random(),
Math.random(), Math.random()];

    if (squareTryangleCircle == "square") {
        numPoints = 4;
        for (var i = 0; i < numPoints; i++)colors.push(...colorArray);
    }
    else if (squareTryangleCircle == "trinagle") {
        numPoints = 3;
        for (var i = 0; i < numPoints; i++)colors.push(...colorArray);
    }
    else if (squareTryangleCircle == "circle") {
        numPoints = 100;
        for (var i = 0; i < numPoints; i++)colors.push(...colorArray);
    }
    else if (squareTryangleCircle == "letter") {
        numPoints = 240;
        for (var i = 0; i < numPoints; i++)colors.push(...colorArray);

        gl.bufferSubData(gl.ARRAY_BUFFER, points, new
Float32Array(colors));
        gl.clear(gl.COLOR_BUFFER_BIT);
        gl.drawArrays(gl.LINE_STRIP, 0, numPoints);
        return;
    }


    gl.bufferSubData(gl.ARRAY_BUFFER, points, new Float32Array(colors));
    gl.clear(gl.COLOR_BUFFER_BIT);
    gl.drawArrays(gl.TRIANGLE_FAN, 0, numPoints);
}

//displays the letter s
function displayLetter() {

    var xypoints =
[125,526,141,526,140,526,140,527,139,527,138,528,137,528,136,529,135,529,1
```

```
34,529,133,529,133,530,132,530,131,530,130,531,129,531,128,531,127,531,126
,531,126,532,125,532,124,533,123,533,122,533,121,533,120,533,120,534,119,5
34,118,534,117,534,117,535,116,535,115,535,115,536,114,536,114,537,113,537
,112,537,111,537,110,537,109,537,108,537,108,538,107,538,106,538,106,539,1
05,539,104,539,103,539,102,540,101,540,100,541,99,541,98,542,97,542,96,542
,96,543,95,543,94,543,94,544,93,544,92,544,92,545,92,546,91,546,91,547,90,
547,89,547,89,548,88,548,88,549,87,549,86,550,86,551,85,551,84,552,84,553,
83,554,83,555,83,556,82,556,82,557,82,558,82,559,82,560,81,560,81,561,81,5
62,81,563,81,564,81,565,81,566,81,567,81,568,82,568,82,569,83,569,83,570,8
4,570,85,570,86,570,86,571,87,571,88,571,89,572,90,573,91,573,91,574,92,57
4,93,574,94,574,94,575,95,575,96,575,97,575,97,576,98,576,98,577,99,577,10
0,577,101,577,102,577,102,578,103,578,104,578,105,578,106,578,107,578,108,
578,109,578,110,578,111,579,112,579,113,579,114,579,115,579,116,579,117,57
9,118,579,118,580,119,580,120,581,121,581,122,581,122,582,123,582,124,582,
125,583,126,583,126,584,127,584,127,585,128,585,129,585,129,586,130,586,13
0,587,131,587,131,588,132,588,132,589,133,589,133,590,134,590,134,591,134,
592,134,593,135,593,135,594,135,595,135,596,135,597,135,598,135,599,135,60
0,135,601,135,602,134,603,133,603,133,604,132,604,132,605,132,606,131,606,
129,606,129,607,128,607,127,608,127,609,126,609,125,609,125,610,124,610,12
4,611,123,611,122,611,121,612,120,612,120,613,119,613,118,614,117,614,116,
614,115,614,114,614,114,615,113,615,112,615,111,615,110,615,109,615,109,61
6,108,616,107,616,106,616,105,616,104,616,103,616,102,616,102,617,101,617,
100,617,99,617,98,617,97,617,96,617,95,617,94,617,93,617,92,617,91,617,90,
617,89,617,88,617,87,617,86,617,85,617,84,617,84,618,84,619,84,620,84,621,
84,622,84,623,84,624,84,625,84,626,84,627];

    var x;
    var even = true;
    xypoints = $.map(xypoints, function (v) {
        x= v/627.0;
        x = (x * 1.6) - 1
        if (even) {
            x = (x * -1) - .8
            even = false;
        }
        else {
            x = x - .5
            even = true;
        }
```

```
        return ((x * 4) + .3);
        return(x);
    });
    //alert(xypoints);
    var numbers = new Float32Array(xypoints);
    squareTryangleCircle = "letter";
    var colors =
[1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1
,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0
,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0
,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1
,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0
,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0
,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1
,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0
,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0
,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1
,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0
,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0
,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1
,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0
,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0
,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1
,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0
,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0
,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1
,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0];

    var bufferId = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
    gl.bufferData(gl.ARRAY_BUFFER, numbers, gl.STATIC_DRAW);

    var aPosition = gl.getAttribLocation(program, "aPosition");
    gl.vertexAttribPointer(aPosition, 2, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(aPosition);

    cBuffer = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, cBuffer);
    gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(colors),
gl.STATIC_DRAW); //new Float32Array(colors)
```

```
    var aColor = gl.getAttribLocation(program, "aColor");
    //3 points colors?
    gl.vertexAttribPointer(aColor, 3, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(aColor);

    render("red");
    // gl.clear(gl.COLOR_BUFFER_BIT);
    // gl.drawArrays(gl.LINE_STRIP, 0, 240);

}

//helper to get points for letter s
function myFunction(e) {
    var x = e.clientX;
    var y = e.clientY;
    var coor = "Coordinates: (" + x + "," + y + ")";
    document.getElementById("demo").innerHTML = coor;
    s.push(x);
    s.push(y);
}

//function that handles key presses
function checkKey(e) {
    //alert(e.keyCode);
    switch (e.keyCode) {
        //red
        case 49:
            render("red");
            break;
        //green
        case 50:
            render("green");
            break;
        //blue
        case 51:
            render("blue");
            break;
        //random color
        case 52:
```

```
            render("random");
            break;
        //favourite color
        case 53:
            render("favourite");
            break;
        //triangle
        case 84:
            squareTryangleCircle = "trinagle";
            var colors = [1, 0, 0, 1, 0, 0, 1, 0, 0];

            var bufferId = gl.createBuffer();
            gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
            gl.bufferData(gl.ARRAY_BUFFER, points, gl.STATIC_DRAW);

            var aPosition = gl.getAttribLocation(program, "aPosition");
            gl.vertexAttribPointer(aPosition, 2, gl.FLOAT, false, 0, 0);
            gl.enableVertexAttribArray(aPosition);

            cBuffer = gl.createBuffer();
            gl.bindBuffer(gl.ARRAY_BUFFER, cBuffer);
            gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(colors),
gl.STATIC_DRAW); //new Float32Array(colors)

            var aColor = gl.getAttribLocation(program, "aColor");
            //3 points colors?
            gl.vertexAttribPointer(aColor, 3, gl.FLOAT, false, 0, 0);
            gl.enableVertexAttribArray(aColor);

            render();
            break;
        // square
        case 83:
            squareTryangleCircle = "square";
            var colors = [0, 1, 1, 0, 1, 1, 0.5, 0, 1, .6, .1, 1];

            var bufferId = gl.createBuffer();
            gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
            gl.bufferData(gl.ARRAY_BUFFER, pointsSquare, gl.STATIC_DRAW);
```

```
            var aPosition = gl.getAttribLocation(program, "aPosition");
            gl.vertexAttribPointer(aPosition, 2, gl.FLOAT, false, 0, 0);
            gl.enableVertexAttribArray(aPosition);

            cBuffer = gl.createBuffer();
            gl.bindBuffer(gl.ARRAY_BUFFER, cBuffer);
            gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(colors),
gl.STATIC_DRAW); //new Float32Array(colors)

            var aColor = gl.getAttribLocation(program, "aColor");
            gl.vertexAttribPointer(aColor, 3, gl.FLOAT, false, 0, 0);
            gl.enableVertexAttribArray(aColor);

            render();
            break;
        // circle
        case 67:
            squareTryangleCircle = "circle";
            var colors = [1, 0, 1, 1, 0, 1];
            var numPoints = 100;
            for (var i = 0; i < numPoints; i++)colors.push(1, 0, 1);


            var bufferId = gl.createBuffer();
            gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
            gl.bufferData(gl.ARRAY_BUFFER, flatten(circlePoints),
gl.STATIC_DRAW);

            var aPosition = gl.getAttribLocation(program, "aPosition");
            gl.vertexAttribPointer(aPosition, 2, gl.FLOAT, false, 0, 0);
            gl.enableVertexAttribArray(aPosition);

            cBuffer = gl.createBuffer();
            gl.bindBuffer(gl.ARRAY_BUFFER, cBuffer);
            gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(colors),
gl.STATIC_DRAW); //new Float32Array(colors)

            var aColor = gl.getAttribLocation(program, "aColor");
            //3 points colors?
            gl.vertexAttribPointer(aColor, 3, gl.FLOAT, false, 0, 0);
```

```
            gl.enableVertexAttribArray(aColor);

            render("favourite");
            break;
    }
}
```