Name: Dima Bochkarev

ID(UIN): 01211656

Class: CS460 – Assignment 2

Computer: Windows PC, Visual Studio Code

The goal of this program is to draw a picture of smiling face using WebGL API.

The program doesn’t take any input.

The program doesn’t give any output value.

First, the program gets a reference to the canvas element and its WebGL rendering context. Next, a buffer is created and bound to the ARRAY\_BUFFER. The generateArc function is defined to generate points for an arc. This function takes the center coordinates, radius, start and end angles, number of points, and scale factor for the x-coordinate as parameters. Once the points for the eyebrows, nose, and lips are generated using the generateArc, they are concatenated into a single array and stored in the buffer using the bufferData method. The code then creates a simple shader program. Lastly, the code draws the eyebrows, nose, and lips using the drawArrays method. The color of each part is set using the uniform4f method before drawing.

Source Code:  
index.html

<!DOCTYPE html>

<html>

<body>

<canvas id="canvas" width="400" height="400"></canvas>

<script src="script.js"></script>

</body>

</html>

script.js

*// Get the canvas and WebGL context*

var canvas = document.getElementById('canvas');

var gl = canvas.getContext('webgl');

*// Set the clear color to a darker green*

gl.clearColor(0.0, 0.39, 0.0, 1.0);

gl.clear(gl.COLOR\_BUFFER\_BIT);

*// Create a buffer for the points*

var buffer = gl.createBuffer();

gl.bindBuffer(gl.ARRAY\_BUFFER, buffer);

*// Function to generate points for an arc*

function generateArc(*centerX*, *centerY*, *radius*, *startAngle*, *endAngle*, *numPoints*, *scaleX*) {

    var points = [];

    var angleStep = (*endAngle* - *startAngle*) / (*numPoints* - 1);

*for* (var i = 0; i < *numPoints*; i++) {

      var angle = *startAngle* + i \* angleStep;

      points.push(*centerX* + *radius* \* Math.cos(angle) \* *scaleX*); *// Scale the x-coordinate*

      points.push(*centerY* + *radius* \* Math.sin(angle));

    }

*return* points;

  }

*// Generate points for the eyebrows, nose, and lips*

var leftEyebrow = generateArc(-0.20, 0.05, 0.1, 0, Math.PI, 50, 1.25);

var rightEyebrow = generateArc(0.20, 0.05, 0.1, 0, Math.PI, 50, 1.25);

var nose = [0.0, 0.0, 0.0, -0.1];

var lips = generateArc(0.0, -0.15, 0.1, Math.PI, 2 \* Math.PI, 50, 1.25);

var vertices = [].concat(leftEyebrow, rightEyebrow, nose, lips);

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

*// Create a simple shader program*

var vertexShader = gl.createShader(gl.VERTEX\_SHADER);

gl.shaderSource(vertexShader, `

  attribute vec2 a\_position;

  void main() {

    gl\_Position = vec4(a\_position, 0.0, 1.0);

    gl\_PointSize = 15.0; // Set the point size here

  }

`);

gl.compileShader(vertexShader);

var fragmentShader = gl.createShader(gl.FRAGMENT\_SHADER);

gl.shaderSource(fragmentShader, `

  precision mediump float;

  uniform vec4 u\_color;

  void main() {

    gl\_FragColor = u\_color;

  }

`);

gl.compileShader(fragmentShader);

var program = gl.createProgram();

gl.attachShader(program, vertexShader);

gl.attachShader(program, fragmentShader);

gl.linkProgram(program);

*// Use the shader program and bind the buffer*

gl.useProgram(program);

gl.bindBuffer(gl.ARRAY\_BUFFER, buffer);

*// Get the position location and enable it*

var positionLocation = gl.getAttribLocation(program, 'a\_position');

gl.enableVertexAttribArray(positionLocation);

*// Point the position attribute to the buffer*

gl.vertexAttribPointer(positionLocation, 2, gl.FLOAT, false, 0, 0);

*// Get the color uniform location*

var colorLocation = gl.getUniformLocation(program, 'u\_color');

*// Draw the left eyebrow*

gl.uniform4f(colorLocation, 0.0, 0.0, 1.0, 1.0); *// Blue*

gl.drawArrays(gl.LINE\_STRIP, 0, leftEyebrow.length / 2);

*// Draw the right eyebrow*

gl.uniform4f(colorLocation, 0.0, 0.0, 1.0, 1.0); *// Blue*

gl.drawArrays(gl.LINE\_STRIP, leftEyebrow.length / 2, rightEyebrow.length / 2);

*// Draw the nose*

gl.uniform4f(colorLocation, 0.0, 0.0, 0.0, 1.0); *// Black*

gl.drawArrays(gl.LINES, (leftEyebrow.length + rightEyebrow.length) / 2, nose.length / 2);

*// Draw the lips*

gl.uniform4f(colorLocation, 1.0, 0.0, 0.0, 1.0); *// Red*

gl.drawArrays(gl.LINE\_STRIP, (leftEyebrow.length + rightEyebrow.length + nose.length) / 2, lips.length / 2);

Drawing/Output screenshot:

