CSE 3219

COMPUTER GRAPHICS

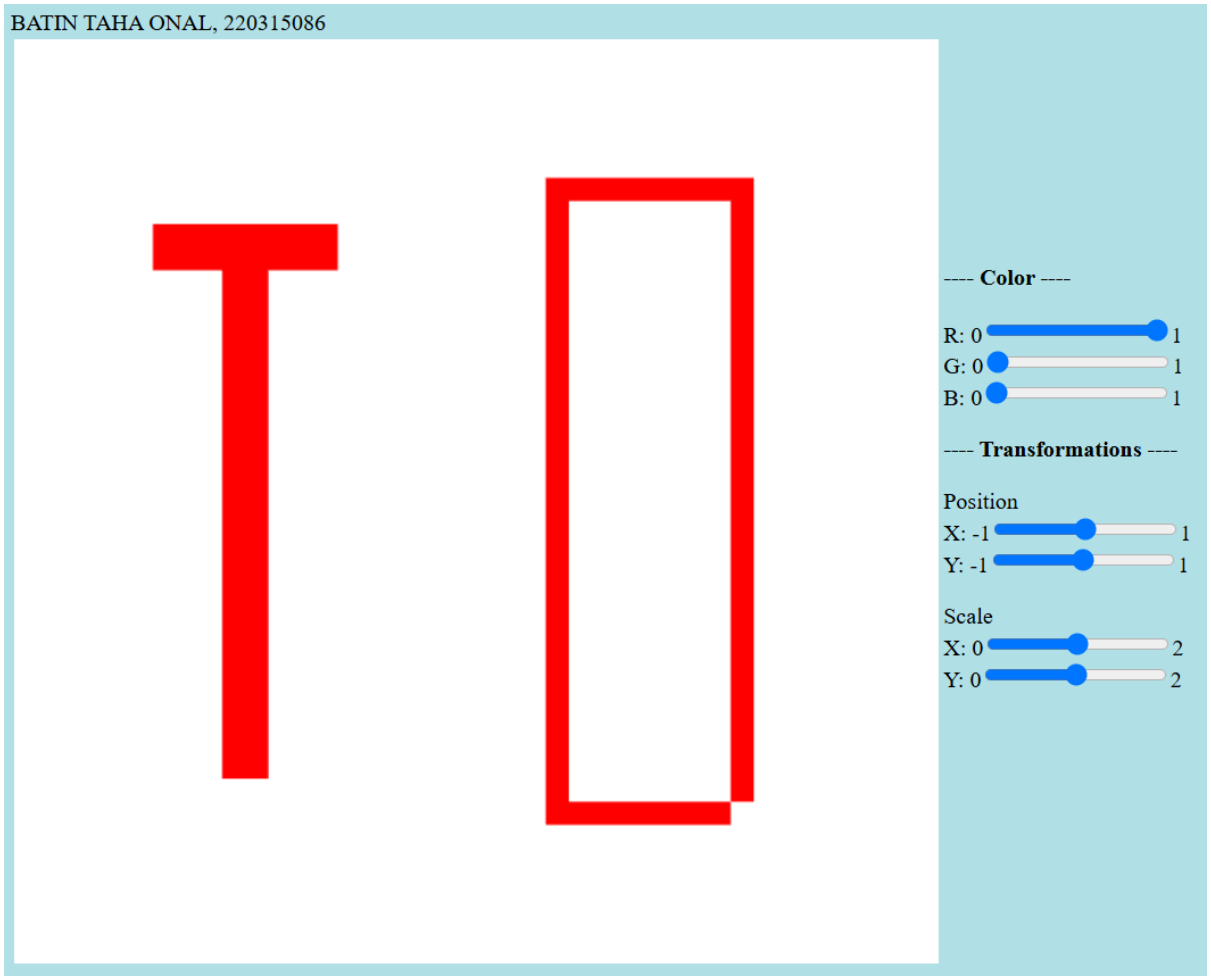
SPRING 2025

Midterm Assignment Report

BATIN TAHA ONAL - 220315086

Submission Date: 8 April 2025

Program Output



Reflections

I learned basic graphic coloring and shaping. Also, I learned teoric topics like shaders(vertex,fragment). I had a hard time about memorization topics like trigonometry formulas (rotation angle- sin-cos-tetha formulas).

Source Code

(html)

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8" >
<title>Midterm Assignment</title>

<script id="vertex-shader" type="x-shader/x-vertex">
attribute vec4 vPosition;
uniform vec2 uTranslation;
```

```
uniform vec2 uScale;

void main()
{
    vec4 scaledPosition = vec4(vPosition.x * uScale.x, vPosition.y * uScale.y,
0.0, 1.0);
    vec4 translatedPosition = scaledPosition + vec4(uTranslation, 0.0, 0.0);
    gl_Position = translatedPosition;

}
</script>

<script id="fragment-shader" type="x-shader/x-fragment">
precision mediump float;
uniform vec4 uColor;

void main()
{
    gl_FragColor = uColor;

}
</script>

<script type="text/javascript" src="../Common/webgl-utils.js"></script>
<script type="text/javascript" src="../Common/initShaders.js"></script>
<script type="text/javascript" src="../Common/MV.js"></script>
<script type="text/javascript" src="midterm.js"></script>
</head>

<body style="background-color:powderblue;">

<div>
BATIN TAHA ONAL, 220315086
</div>
<table>
    <td>
    <canvas id="gl-canvas" width="650" height="650">
        Oops ... your browser doesn't support the HTML5 canvas element
    </canvas>
    </td>
    <td>
        <div> <strong>---- Color ----</strong> </div><br>
        <div>
        R: 0<input id="redSlider" type="range"
         min="0" max="1" step="0.05" value="1" />1
        </div>
        <div>
        G: 0<input id="greenSlider" type="range"
```

```
         min="0" max="1" step="0.05" value="0" />1
        </div>
        <div>
        B: 0<input id="blueSlider" type="range"
         min="0" max="1" step="0.05" value="0" />1
        </div>
        <br>

        <div> <strong>---- Transformations ----</strong> </div><br>

        <div>Position</div>
        <div>X: -1<input id="posX" type="range"
         min="-1" max="1" step="0.05" value="0" />1</div>
        <div>Y: -1<input id="posY" type="range"
         min="-1" max="1" step="0.05" value="0" />1</div><br>

        <div>Scale</div>
        <div>X: 0<input id="scaleX" type="range"
         min="0" max="2" step="0.05" value="1" />2</div>
        <div>Y: 0<input id="scaleY" type="range"
         min="0" max="2" step="0.05" value="1" />2</div><br>

        <br>
    </td>

</table>
<div>
</body>
</html>
```

(js)

```
var canvas;
var gl;
var program;
var vPosition;

var letter1vertices, letter2vertices;
var buffer1, buffer2;

// Renk değerlerini tutacak değişkenler
var red = 1.0;
var green = 0.0;
var blue = 0.0;

// Pozisyon değerlerini tutacak değişkenler
var posX = 0.0;
```

```
var posY = 0.0;

// Ölçek değerlerini tutacak değişkenler
var scaleX = 1.0;
var scaleY = 1.0;

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

    gl = WebGLUtils.setupWebGL(canvas);
    if (!gl) {
        alert("WebGL isn't available");
    }

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

    //  Load shaders and initialize attribute buffers
    program = initShaders(gl, "vertex-shader", "fragment-shader");
    gl.useProgram(program);

    // Get attribute location
    vPosition = gl.getAttribLocation(program, "vPosition");

    // Create geometry data for the letter T (using triangles)
    letter1vertices = [
        // T'nin yatay çubuğu (üst kısım)
        vec2(-0.7, 0.6), vec2(-0.3, 0.6), vec2(-0.3, 0.5),
        vec2(-0.7, 0.6), vec2(-0.3, 0.5), vec2(-0.7, 0.5),

        // T'nin dikey çubuğu (gövde) - daha geniş hale getirdik
        vec2(-0.55, 0.5), vec2(-0.45, 0.5), vec2(-0.45, -0.6),
        vec2(-0.55, 0.5), vec2(-0.45, -0.6), vec2(-0.55, -0.6)
    ];

    // Create geometry data for the letter O (using triangles)
    letter2vertices = [
        // Üst İnce Kenar
        vec2(0.2, 0.7), vec2(0.6, 0.7), vec2(0.2, 0.65),
        vec2(0.6, 0.7), vec2(0.2, 0.65), vec2(0.6, 0.65),

        // Sağ İnce Kenar
        vec2(0.6, 0.65), vec2(0.6, -0.65), vec2(0.55, 0.65),
        vec2(0.6, -0.65), vec2(0.55, 0.65), vec2(0.55, -0.65),

        // Alt İnce Kenar
        vec2(0.55, -0.65), vec2(0.2, -0.65), vec2(0.55, -0.7),
```

```
        vec2(0.2, -0.65), vec2(0.55, -0.7), vec2(0.2, -0.7),

        // Sol Kalın Kenar
        vec2(0.2, -0.7), vec2(0.2, 0.7), vec2(0.15, -0.7),
        vec2(0.2, 0.7), vec2(0.15, -0.7), vec2(0.15, 0.7)
    ];

    // Load the data into the GPU
    buffer1 = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, buffer1);
    gl.bufferData(gl.ARRAY_BUFFER, flatten(letter1vertices), gl.STATIC_DRAW);

    buffer2 = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, buffer2);
    gl.bufferData(gl.ARRAY_BUFFER, flatten(letter2vertices), gl.STATIC_DRAW);

    document.getElementById("posX").oninput = function(event) {
        posX = parseFloat(event.target.value); // X değerini güncelle
    };
    document.getElementById("posY").oninput = function(event) {
        posY = parseFloat(event.target.value); // Y değerini güncelle
    };
    document.getElementById("scaleX").oninput = function(event) {
        scaleX = parseFloat(event.target.value); // X ölçek değerini güncelle
    };
    document.getElementById("scaleY").oninput = function(event) {
        scaleY = parseFloat(event.target.value); // Y ölçek değerini güncelle
    };
    document.getElementById("redSlider").oninput = function(event) {
        red = event.target.value; // Kırmızı değerini güncelle
    };
    document.getElementById("greenSlider").oninput = function(event) {
        green = event.target.value; // Yeşil değerini güncelle
    };
    document.getElementById("blueSlider").oninput = function(event) {
        blue = event.target.value; // Mavi değerini güncelle
    };

    render();
};

function render() {
    gl.clear(gl.COLOR_BUFFER_BIT);

    // Get uniform location for color
    var colorLocation = gl.getUniformLocation(program, "uColor");
    gl.uniform4f(colorLocation, red, green, blue, 1.0); // Rengi shader'a
gönder
```

```
    // Get uniform location for translation
    var translationLocation = gl.getUniformLocation(program, "uTranslation");
    gl.uniform2f(translationLocation, posX, posY); // Pozisyonu shader'a
gönder

    // Get uniform location for scale
    var scaleLocation = gl.getUniformLocation(program, "uScale");
    gl.uniform2f(scaleLocation, scaleX, scaleY); // Ölçeği shader'a gönder

    // bind vertex buffer and associate position data with shader variables
    gl.bindBuffer(gl.ARRAY_BUFFER, buffer1);
    gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(vPosition);

    // draw letter T (önceki harf B idi, ama şimdi T olarak çiziliyor)
    gl.drawArrays(gl.TRIANGLES, 0, letter1vertices.length);

    //
    // Şimdi ikinci harfi (O) çiziyoruz:
    //
    gl.bindBuffer(gl.ARRAY_BUFFER, buffer2);
    gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(vPosition);

    // draw letter O
    gl.drawArrays(gl.TRIANGLES, 0, letter2vertices.length);

    window.requestAnimFrame(render);
}
```