MANİSA CELAL BAYAR UNIVERSITY

CSE 3219

COMPUTER GRAPHICS

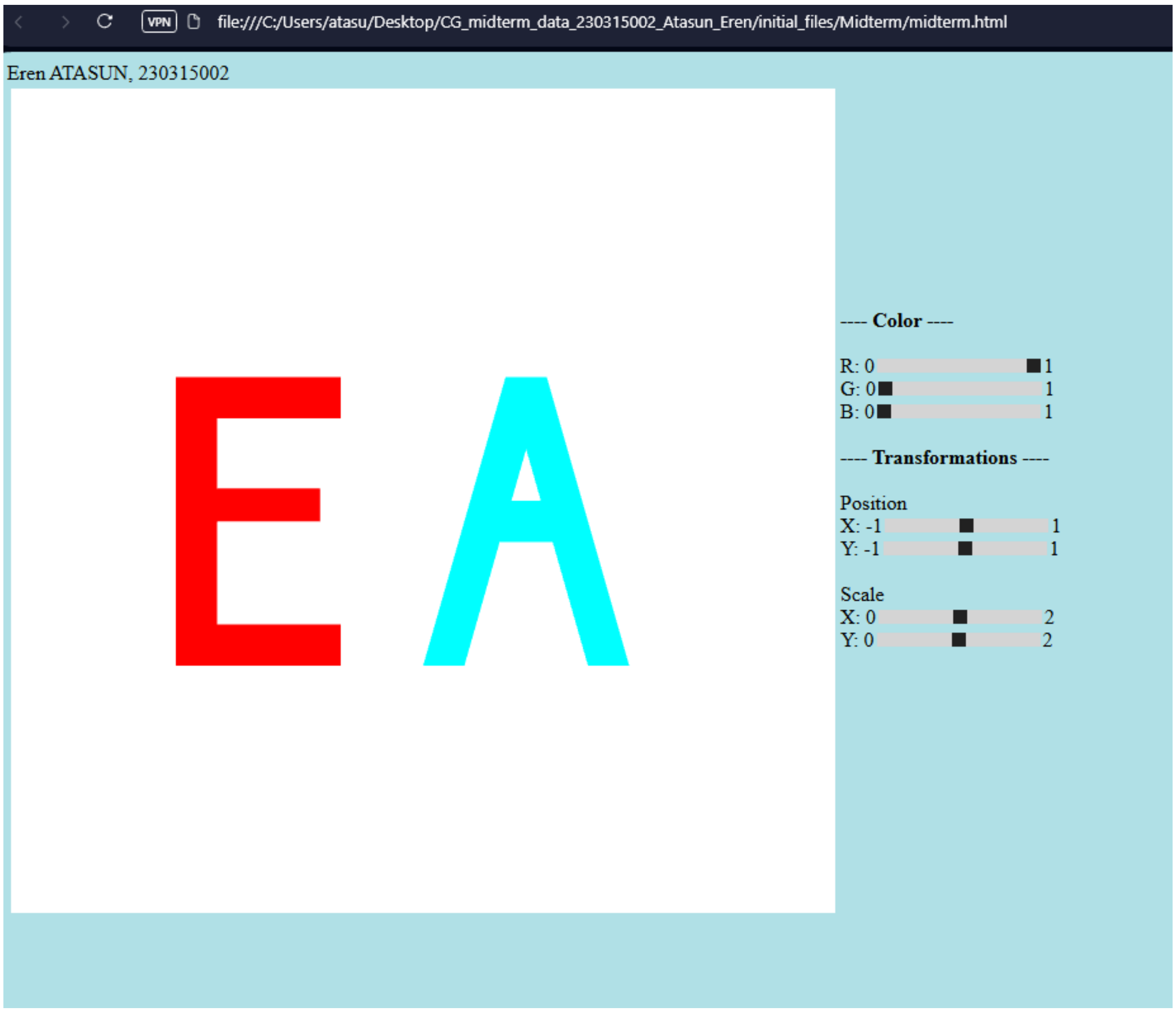
SPRING 2025

Midterm Assignment Report

Eren Atasun – 230315002

Submission Date: *8 April 2025*

Program Output


file:///C:/Users/atasu/Desktop/CG_midterm_data_230315002_Atasun_Eren/initial_files/Midterm/midterm.html

Eren ATASUN, 230315002

---- **Color** ----

R: 0 1
G: 0 1
B: 0 1

---- **Transformations** ----

Position
X: -1 1
Y: -1 1

Scale
X: 0 2
Y: 0 2

Reflections

During this assignment, I learned how to use WebGL to model 2D geometric shapes using triangle-based vertex definitions. I gained a deeper understanding of the coordinate system in WebGL, how shaders work, and how to implement interactive transformations such as position, scale, and color changes via sliders. One of the most valuable experiences was learning to manage multiple objects (letters) and apply transformations correctly using uniform variables.

Source Code

midterm.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()

{

   // TODO: calculate gl_Position appropriately

   vec2 scaledPosition = vPosition.xy * uScale;

   vec2 translatedPosition = scaledPosition + uTranslation;

   gl_Position = vec4(translatedPosition, 0.0, 1.0);


}

</script>


<script id="fragment-shader" type="x-shader/x-fragment">

precision mediump float;
```

```
uniform vec3 uColor;


void main()
{
   gl_FragColor = vec4(uColor, 1.0);
}
</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>
Eren ATASUN, 230315002
</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>
```

and midterm.js

```
var canvas;
var gl;
var program;
var vPosition;
var letter1vertices, letter2vertices;
var buffer1, buffer2;

// Global uniform locations
var uTranslationLoc, uScaleLoc, uColorLoc;

// Transformation and color state
var translation = [0.0, 0.0];
var scale = [1.0, 1.0];
var color = [1.0, 0.0, 0.0];

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 );


vPosition = gl.getAttribLocation(program, "vPosition");

uTranslationLoc = gl.getUniformLocation(program, "uTranslation");

uScaleLoc = gl.getUniformLocation(program, "uScale");

uColorLoc = gl.getUniformLocation(program, "uColor");


// Define vertices for 'E' (left)

letter1vertices = [

   // Sol dikey çubuk (daha uzun hale getirildi)

   vec2(-0.6,  0.3), vec2(-0.5,  0.3), vec2(-0.6, -0.4),

   vec2(-0.5,  0.3), vec2(-0.5, -0.4), vec2(-0.6, -0.4),


   // Üst yatay çubuk

   vec2(-0.5,  0.3), vec2(-0.2,  0.3), vec2(-0.5,  0.2),

   vec2(-0.2,  0.3), vec2(-0.2,  0.2), vec2(-0.5,  0.2),


   // Orta yatay çubuk (biraz daha yukarı alındı)

   vec2(-0.5,  0.03), vec2(-0.25, 0.03), vec2(-0.5, -0.05),

   vec2(-0.25, 0.03), vec2(-0.25, -0.05), vec2(-0.5, -0.05),


   // Alt yatay çubuk (aşağıya uzatıldı)
```

```
    vec2(-0.5, -0.3), vec2(-0.2, -0.3), vec2(-0.5, -0.4),
    vec2(-0.2, -0.3), vec2(-0.2, -0.4), vec2(-0.5, -0.4)
];



// Define vertices for 'A' (right)
letter2vertices = [
    vec2(0.2, 0.3), vec2(0.0, -0.4), vec2(0.1, -0.4),
    vec2(0.3, 0.3), vec2(0.2, 0.3), vec2(0.1, -0.4),

    vec2(0.3, 0.3), vec2(0.4, -0.4), vec2(0.5, -0.4),
    vec2(0.3, 0.3), vec2(0.2, 0.3), vec2(0.4, -0.4),

    vec2(0.2, 0.0), vec2(0.3, 0.0), vec2(0.4, -0.1),
    vec2(0.1, -0.1), vec2(0.2, 0.0), vec2(0.4, -0.1)

];



// 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 );



// Slider event handlers
document.getElementById("posX").oninput = function(event) {
   translation[0] = parseFloat(event.target.value);
};
document.getElementById("posY").oninput = function(event) {
   translation[1] = parseFloat(event.target.value);
};
document.getElementById("scaleX").oninput = function(event) {
   scale[0] = parseFloat(event.target.value);
};
document.getElementById("scaleY").oninput = function(event) {
   scale[1] = parseFloat(event.target.value);
};
document.getElementById("redSlider").oninput = function(event) {
   color[0] = parseFloat(event.target.value);
};
```

```
    document.getElementById("greenSlider").oninput = function(event) {
        color[1] = parseFloat(event.target.value);
    };
    document.getElementById("blueSlider").oninput = function(event) {
        color[2] = parseFloat(event.target.value);
    };

    render();
};

function render() {
    gl.clear( gl.COLOR_BUFFER_BIT );
// Letter 1
gl.bindBuffer(gl.ARRAY_BUFFER, buffer1);
gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
gl.enableVertexAttribArray(vPosition);

gl.uniform2fv(uTranslationLoc, translation);
gl.uniform2fv(uScaleLoc, scale);
gl.uniform3fv(uColorLoc, color);
gl.drawArrays(gl.TRIANGLES, 0, letter1vertices.length);

// Letter 2 (opposite color)
```

```
gl.bindBuffer(gl.ARRAY_BUFFER, buffer2);
gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
gl.enableVertexAttribArray(vPosition);


gl.uniform2fv(uTranslationLoc, [translation[0], translation[1]]);
gl.uniform2fv(uScaleLoc, scale);
gl.uniform3fv(uColorLoc, [1 - color[0], 1 - color[1], 1 - color[2]]);
gl.drawArrays(gl.TRIANGLES, 0, letter2vertices.length);


   window.requestAnimFrame(render);
}
```