



**CSE 3114 / CSE 3219**

**COMPUTER GRAPHICS**

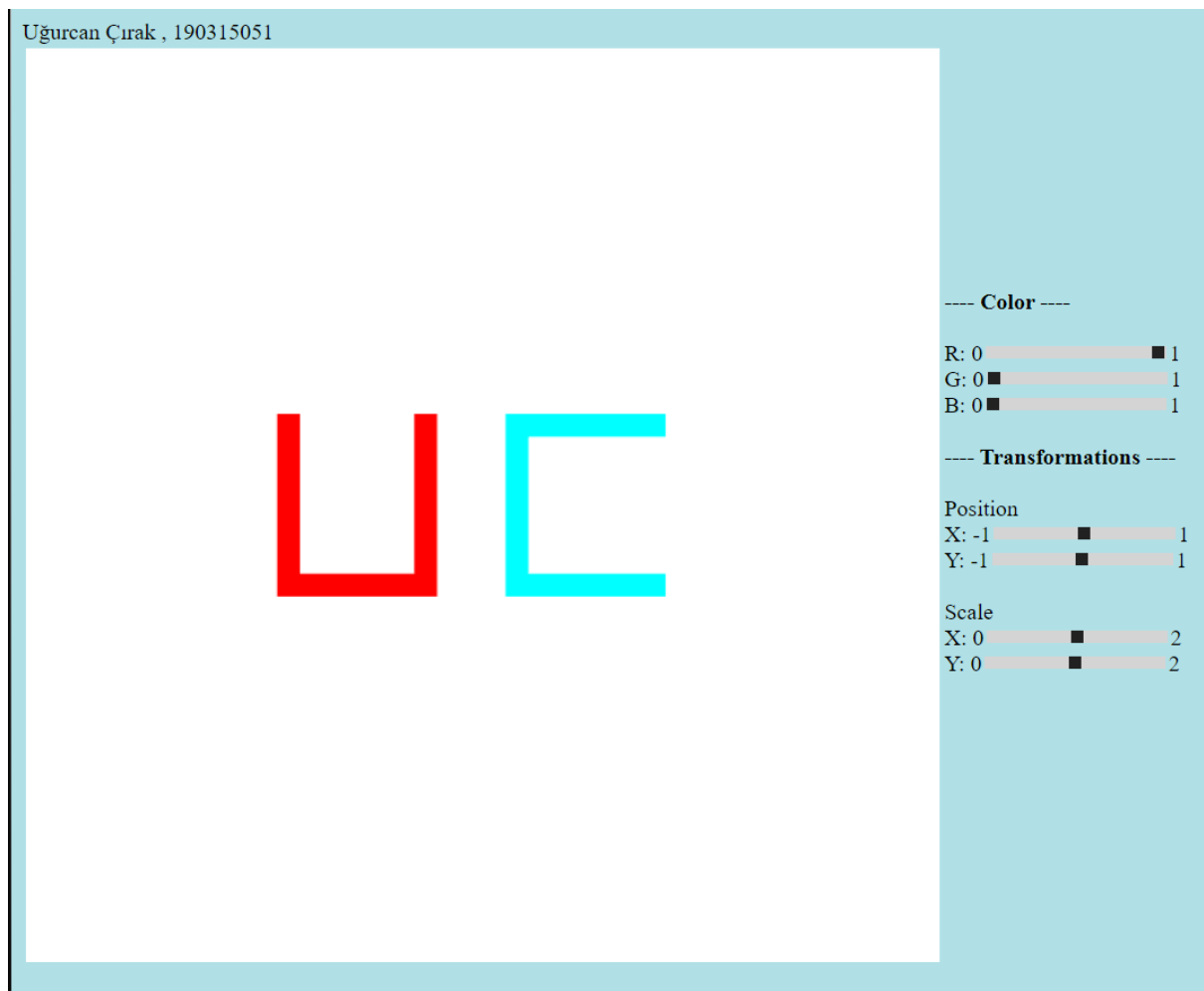
**SPRING 2023**

***Midterm Assignment Report***

***UĞURCAN ÇIRAK– 190315051***

***Submission Date: 9 April 2023***

## Program Output



## Reflections

*I learned how to do Transformations in general. I had no trouble with the color change. I didn't have any difficulties with the scale operation, but because I entered the posX/posY eventargets incorrectly while performing the scale operation, I did not have the Position operation, the image disappeared every time I moved it. Then I found this error while trying. It was a fun assignment. If we enter the wrong coordinate order of the name initials, the figure does not appear. I also learned that ordering is important.*

## Source Code

**JS:**

```
var canvas;
```

```
var gl;
```

```
var vPosition;

var program;


var letter1vertices, letter2vertices;

var buffer1, buffer2;


// TODO: define any global variables you need


var vColor;

var posX = 0.0;

var posY = 0.0;

var scaleX = 1.0;

var scaleY = 1.0;

var redSlider = 1.0;

var greenSlider = 0.0;

var blueSlider = 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 );
```

```
// Create geometry data
```

```
letter1vertices = [

    vec2(-0.9 , -0.3 ),

    vec2(-0.8 , -0.3),

    vec2(-0.9 , 0.4 ),

    vec2(-0.8 , 0.4 ),


    vec2(-0.9 , -0.4 ),

    vec2(-0.9 , -0.3 ),

    vec2(-0.2 , -0.4 ),

    vec2(-0.2 , -0.3),


    vec2(-0.3 , -0.3),

    vec2(-0.2 , -0.3 ),

    vec2(-0.3 , 0.4 ),

    vec2(-0.2 , 0.4 )

];
```

```
letter2vertices = [

    vec2(0.1 , 0.4 ),

    vec2(0.2 , 0.4 ),
```

```
vec2(0.1 , -0.4),
```

```
vec2(0.2 , -0.4 ),
```

```
vec2(0.2 , -0.4 ),
```

```
vec2(0.2 , -0.3 ),
```

```
vec2(0.8 , -0.4 ),
```

```
vec2(0.8 , -0.3 ),
```

```
vec2(0.8 , 0.4 ),
```

```
vec2(0.8 , 0.3 ),
```

```
vec2(0.2 , 0.4 ),
```

```
vec2(0.2 , 0.3 )
```

```
];
```

```
// TODO: create vertex coordinates for your initial letters instead of these vertices
```

```
// 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 );
```

```
vColor = gl.getUniformLocation(program, "vColor");
```

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

```
gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
```

```
gl.enableVertexAttribArray(vPosition);
```

```
    document.getElementById("posX").oninput = function(event) {  
        //TODO: fill here to adjust translation according to slider value  
        posX = (event.target.value );  
    };
```

```
document.getElementById("posY").oninput = function(event) {  
    //TODO: fill here to adjust translation according to slider value  
    posY = (event.target.value );  
};
```

```
document.getElementById("scaleX").oninput = function(event) {  
    //TODO: fill here to adjust scale according to slider value  
    scaleX = event.target.value;  
};
```

```
document.getElementById("scaleY").oninput = function(event) {  
    //TODO: fill here to adjust scale according to slider value  
    scaleY = event.target.value;  
};
```

```
document.getElementById("redSlider").oninput = function(event) {
```

```

//TODO: fill here to adjust color according to slider value

redSlider = event.target.value;

};

document.getElementById("greenSlider").oninput = function(event) {

//TODO: fill here to adjust color according to slider value

greenSlider = event.target.value;

};

document.getElementById("blueSlider").oninput = function(event) {

//TODO: fill here to adjust color according to slider value

blueSlider = event.target.value;

};


render();

};


function render() {

gl.clear( gl.COLOR_BUFFER_BIT );


// TODO: Send necessary uniform variables to shader and


gl.uniform4fv(vColor, vec4(redSlider, greenSlider, blueSlider, 1));

// perform draw calls for drawing letters


// 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 triangle

    gl.uniform1f(gl.getUniformLocation(program, "posX"), posX);

    gl.uniform1f(gl.getUniformLocation(program, "posY"), posY);

    gl.uniform1f(gl.getUniformLocation(program, "scaleX"), scaleX);

    gl.uniform1f(gl.getUniformLocation(program, "scaleY"), scaleY);


    gl.drawArrays(gl.TRIANGLE_STRIP, 0, letter1vertices.length);


gl.uniform4fv(vColor, vec4(1 - redSlider, 1 - greenSlider, 1 - blueSlider, 1));


    // bind vertex buffer and associate position data with shader variables
gl.bindBuffer( gl.ARRAY_BUFFER, buffer2 );

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

gl.enableVertexAttribArray( vPosition );

// draw rectangle


    gl.uniform1f(gl.getUniformLocation(program, "posX"), posX);

    gl.uniform1f(gl.getUniformLocation(program, "posY"), posY);

    gl.uniform1f(gl.getUniformLocation(program, "scaleX"), scaleX);

    gl.uniform1f(gl.getUniformLocation(program, "scaleY"), scaleY);


    gl.drawArrays(gl.TRIANGLE_STRIP, 0, letter2vertices.length);


window.requestAnimationFrame(render);

}

```



## 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 float scaleX;
```

```
uniform float scaleY;
```

```
uniform float posX;
```

```
uniform float posY;
```

```
void main() {
```

```
    gl_Position = vec4(posX + (vPosition.x * scaleX), posY + (vPosition.y * scaleY), 0.2, 2.0);
```

```
// TODO: get required variables
```

```
}
```

```
</script>
```

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

```
precision mediump float;
```

```
uniform vec4 vColor;

void main()
{
    gl_FragColor = vColor;
}
```

```
</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>
```

Uğurcan Çırak , 190315051

```
</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>