HTML JAVASCRIPT CODE DESCRIPTION

General view of code ''html file''

```
<html>
<head>
    <style>
        body{
            margin:0 auto;
            }
        .first{
            float:left;
            height:721px;
            width:800px;
         }
        .div1{
            float:left;
            position:relative;
            height:30px;
            margin-top:20px;
            margin-left:30px;
            }
        .div2{
            float:left;
            position:relative;
            margin-top:20px;
            margin-left:30px;
            height:30px;
            width:200px;
            }
        .div3{
            float:right;
            position:relative;
            margin-top:37px;
            margin-right:80px;
            height:20px;
            }
        .div4{
            float:left;
            position:relative;
            margin-top:30px;
            margin-left:30px;}
        .div5{
            margin-top:10px;
            margin-left:10px;
            position:relative;
            width:60px;
            }
        .div6{
            margin-top:25px;
            margin-left:10px;
            position:relative;
            width:60px;
            }
        .div7{
            width:75px;}

    </style>
</head>
```

```
<body>

   <canvas id="glcanvas" width="800" height="721" class = "first"></canvas>

   <button id = "DirectionButton" class="div2">
    <font color="green" >
      Change Rotation Direction </font>
   </button>

   <button id = "StartRotation" class="div2">
   <font color="green" >
      StartRotation </font>
   </button>

   <button id = "StopRotation" class="div2">
   <font color="green" >
      StopRotation </font>
   </button>

   <div class = "div1">
        <br>
        <b><font color="red" size="4">CHANGE THE COLOR :</font> </b>
        <select name="colors" id="color" >
            <option value="white"> WHITE </option>
            <option value="red">RED </option>
            <option value="blue"> BLUE </option>
            <option value="green"> GREEN </option>
        </select>

        <button id="ColorButton" class="div7"><font color="purple">
        Select </font>
       </button>
    </div>

   <div  class = "div3">
   speed 0 <input id="slide" type="range"
   min="0" max="1000" step="50" value="500"/>
   100
   </div>

    <div class="div4">
        <br> <b> <font color="red" size="4"> REPOSITION :</font> </b>
        <br> (on the x-axis)
        <button id="xleft" class="div5"> <font color="blue" >LEFT</font></button>
        <button id="xright" class="div5"><font color="blue" >RIGHT</font> </button>
    </div>

    <div class="div4">
        <br>(on the y-axis)
        <button id="yup" class="div6"><font color="blue" >UP</font></button>
        <button id="ydown" class="div6"><font color="blue" >DOWN</font></button>
    </div>

</body>
```

```
<script id="vertex-shader" type="x-shader/x-vertex">
    attribute vec4 vPosition;
    uniform float theta;
    void main(){

            gl_Position.x = cos(theta) * vPosition.x - sin(theta) * vPosition.y;
            gl_Position.y = sin(theta) * vPosition.x + cos(theta) * vPosition.y;
            gl_Position.z = 0.0;
            gl_Position.w = 1.0;
    }
</script>

<script id="fragment-shader" type="x-shader/x-fragment">
    precision mediump float;
    uniform vec4 fColor;
    void main() {
       gl_FragColor= fColor;
    }
</script>

<script src="../common/MV.js"></script>
<script src="../common/initShaders.js"></script>
<script src="20290310.js"></script>
<script src="../common/webgl-utils.js"></script>

</html>
```

General view of code ''javascript file''

```
var gl;
var numPoints = 5000;
var thetaLoc;
var theta;
var isDirClockwise = false;
var delay = 500;
var stoprotating = false;
var changecoordinates_x = 0;
var changecoordinates_y = 0;
window.onload = function main() {

    const canvas = document.querySelector("#glcanvas");
    // Initialize the GL context
    gl = WebGLUtils.setupWebGL(canvas);
    // Only continue if WebGL is available and working
    if (!gl) {
      alert("Unable to initialize WebGL. Your browser or machine may not support it.");
      return;
    }

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

```
// adding button
var myButton = document.getElementById("DirectionButton");
myButton.addEventListener("click", function() {(isDirClockwise = !isDirClockwise);});
//changing speed
document.getElementById("slide").onchange = function() {delay = this.value;};

// stop button
var stopbutton = document.getElementById("StopRotation");
stopbutton.addEventListener("click", function(){stoprotating = false;});

// start button
var startbutton = document.getElementById("StartRotation");
startbutton.addEventListener("click", function(){stoprotating = true;});

// color buttons
var buttonColor = document.getElementById("ColorButton");
var colorLocation = gl.getUniformLocation(program,"fColor");

// renklendirme kısmı
document.getElementById("color").onchange = function(){delay2 = this.value;};

var blue =[0.0,0.0,0.5,0.7];
var red =[1.0, 0.0, 0.0, 1.0];
var white =[1.0,1.0,1.0,1.0];
var green =[0.0, 0.5, 0.0,1.0];
buttonColor.addEventListener("click",
                    function(){
                      if(delay2=="white"){color = white;gl.uniform4fv(colorLocation,color);}
                      if(delay2=="blue"){color = blue;gl.uniform4fv(colorLocation,color);}
                      if(delay2=="red"){color = red;gl.uniform4fv(colorLocation,color);}
                      if(delay2=="green"){color = green;gl.uniform4fv(colorLocation,color);}
                        ;}
                      );
//x-y düzlemlerinde kaydırma islemi

var changecoordinatesy1 = document.getElementById("yup");
    changecoordinatesy1.addEventListener("click",
                        function(){  changecoordinates_y += 10;
                        });
var changecoordinatesy2 = document.getElementById("ydown");
    changecoordinatesy2.addEventListener("click",
                        function(){ changecoordinates_y -= 10;
                        });

var changecoordinatesx1 = document.getElementById("xright");
    changecoordinatesx1.addEventListener("click",
                        function(){ changecoordinates_x += 10;
                        });
var changecoordinatesx2 = document.getElementById("xleft");
    changecoordinatesx2.addEventListener("click",
                        function(){  changecoordinates_x -= 10;
                        });
```

```
// coordinates of letters
var vertices = [ // M harfi (30 satır)
                 vec2(-.65, -.5), //sol alt
                 vec2(-.65, 0.5), // sol üst
                 vec2(-.60, 0.5), // sağ üst
                 vec2(-.60, -.5), // sağ alt
                 vec2(-.65, -.5),
                 vec2(-.60, 0.5),
                 vec2(-0.35, -0.5),
                 vec2(-0.40, -0.5),
                 vec2(-.60, 0.32),
                 vec2(-0.35, -0.5), // 4.nokta
                 vec2(-.60, 0.5),
                 vec2(-.60, 0.32),
                 vec2(-0.39, -0.48),
                 vec2(-0.35, -0.5),
                 vec2(-.60, 0.32),
                 vec2(-0.16, 0.5),
                 vec2(-0.16, 0.32),
                 vec2(-0.39, -0.48),
                 vec2(-0.16, 0.5),
                 vec2(-0.16, 0.32),
                 vec2(-0.40, -0.5),
                 vec2(-0.16, 0.32),
                 vec2(-0.35, -0.5),
                 vec2(-0.40, -0.5),
                 vec2(-0.16, 0.5),
                 vec2(-0.11, 0.5),
                 vec2(-0.11, -.5),
                 vec2(-0.11, -.5),
                 vec2(-0.16, -.5),
                 vec2(-0.16, 0.5),
                 // A harfi
                 vec2(0.15, -.5),
                 vec2(0.2, -.5),
                 vec2(0.38, 0.5),
                 vec2(0.15, -.5),
                 vec2(0.2, -.5),
                 vec2(0.38, 0.5),
                 vec2(0.15, -.5),
                 vec2(0.38, 0.5),
                 vec2(0.32, 0.5),
                 vec2(0.55, -.5),
                 vec2(0.32, 0.5),
                 vec2(0.38, 0.5),
                 vec2(0.60, -.5),
                 vec2(0.55, -.5),
                 vec2(0.38, 0.5),
                 vec2(0.26, -0.09),
                 vec2(0.26, -0.04),
                 vec2(0.47, -.04),
                 vec2(0.47, -.04),
                 vec2(0.47, -.09),
                 vec2(0.26, -0.09),
                  ];
```

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

        // Associate out shader variables with our data buffer
        var vPosition = gl.getAttribLocation( program, "vPosition" );
        gl.vertexAttribPointer( vPosition, 2, gl.FLOAT, false, 0, 0 );
        gl.enableVertexAttribArray( vPosition );

        thetaLoc = gl.getUniformLocation(program, "theta");
        theta=0;
        gl.uniform1f(thetaLoc, theta);


        render();

}

function render(){

      // default color
      gl.clearColor(0.90, 0.85, 0.75, 1.0);
      gl.clear(gl.COLOR_BUFFER_BIT);
      gl.drawArrays(gl.TRIANGLES, 0, 60);
      gl.viewport(changecoordinates_x,changecoordinates_y,glcanvas.width,glcanvas.height);

      if(stoprotating==true){
          theta += (isDirClockwise ? -0.1 : 0.1);
          gl.uniform1f(thetaLoc, theta);
      }
      setTimeout(function(){
        requestAnimFrame(render);
      },delay);


}
```

CODE DESCRIPTION

First of all, I removed the spaces in the margins of the page.

```
<html>
<head>
    <style>
        body{
            margin:0 auto;
            }
```

I created canvas width equal to 800px and height equal to 721px.

```
<body>

    <canvas id="glcanvas" width="800" height="721" class = "first"></canvas>
```

then I choose class of canvas called first that I created before.

```
.first{
    float:left;
    height:721px;
    width:800px;
}
```

For designing to rotation feature. I need stop, start, and change rotation direction button.

```
<button id = "DirectionButton" class="div2">
 <font color="green" >
   Change Rotation Direction </font>
</button>

<button id = "StartRotation" class="div2">
<font color="green" >
   StartRotation </font>
</button>

<button id = "StopRotation" class="div2">
<font color="green" >
   StopRotation </font>
</button>
```

I created class of rotation buttons called div2.

```
.div2{
    float:left;
    position:relative;
    margin-top:20px;
    margin-left:30px;
    height:30px;
    width:200px;
    }
```

Other feature of the project is the slider that speeds up the rotation. Slider has class named div3 and also slider entries range from 0-1000, also 50 speed changes with each step and default value is 500.

```
<div  class = "div3">
speed 0 <input id="slide" type="range"
min="0" max="1000" step="50" value="500"/>
100
</div>
```

The attributes of the div3 class.

```
.div3{
    float:right;
    position:relative;
    margin-top:37px;
    margin-right:80px;
    height:20px;
    }
```

After I complete all rotation buttons, and slider in html file. I turn to the js file, and created buttons.

Identities are inside of document.getElementById("..."),

addEventListener assigns a property.(when click button apply function what to do.)

```
// adding button
var myButton = document.getElementById("DirectionButton");
myButton.addEventListener("click", function() {(isDirClockwise = !isDirClockwise);});
//changing speed
document.getElementById("slide").onchange = function() {delay = this.value;};

// stop button
var stopbutton = document.getElementById("StopRotation");
stopbutton.addEventListener("click", function(){stoprotating = false;});

// start button
var startbutton = document.getElementById("StartRotation");
startbutton.addEventListener("click", function(){stoprotating = true;});
```

I have a color button to change color of text.

I define 4 option as white, red, blue, green that can be selected. Also, a select button to see all options.

```
<div class = "div1">
    <br>
    <b><font color="red" size="4">CHANGE THE COLOR :</font> </b>
    <select name="colors" id="color" >
        <option value="white"> WHITE </option>
        <option value="red">RED </option>
        <option value="blue"> BLUE </option>
        <option value="green"> GREEN </option>
    </select>

    <button id="ColorButton" class="div7"><font color="purple">
    Select </font>
   </button>
</div>
```

div1 provides height, margin, and position for web design.

```
        .div1{
            float:left;
            position:relative;
            height:30px;
            margin-top:20px;
            margin-left:30px;
            }
```

div7 has 75px width that enlarge select button.

```
        .div7{
            width:75px;}

    </style>
</head>
<body>
```

Html part to reach location of fcolor.

```
<script id="fragment-shader" type="x-shader/x-fragment">
   precision mediump float;
   uniform vec4 fColor;
   void main() {
      gl_FragColor= fColor;
   }
</script>
```

document.getElementById part is same like in rotation buttons. I used the getUniformLocation function because we have to get to the location of the color. The first parameter of the function is the program we created in the laboratory, and the second parameter is fcolor, which is equal to gl_fragcolor(above photo).

Onchange event in JavaScript is an important event which is used for the event changes occurring at the time of performing the event, so I used it. We have 4 variables blue, red, white, green and their rgb color values.

Finally, addEventListener will add the changed color when we click the button as a delay2 value, and onchange function will show us.

```
// color buttons
var buttonColor = document.getElementById("ColorButton");
var colorLocation = gl.getUniformLocation(program,"fColor");


// renklendirme kısmı
document.getElementById("color").onchange = function(){delay2 = this.value;};

var blue =[0.0,0.0,0.5,0.7];
var red =[1.0, 0.0, 0.0, 1.0];
var white =[1.0,1.0,1.0,1.0];
var green =[0.0, 0.5, 0.0,1.0];

buttonColor.addEventListener("click",
                    function(){
                      if(delay2=="white"){color = white;gl.uniform4fv(colorLocation,color);}
                      if(delay2=="blue"){color = blue;gl.uniform4fv(colorLocation,color);}
                      if(delay2=="red"){color = red;gl.uniform4fv(colorLocation,color);}
                      if(delay2=="green"){color = green;gl.uniform4fv(colorLocation,color);}
                        ;}
                      );
```

I guess, it is not included in the minimum project specifications. But I stumbled upon it while researching how to change the letters color. Thus, I thought that I should add it to my projects. Below code are going to relocate our letters in x-y coordinate sytem. First I created buttons in html file.

```
<div class="div4">
    <br> <b> <font color="red" size="4"> REPOSITION :</font> </b>
    <br> (on the x-axis)
    <button id="xleft" class="div5"> <font color="blue" >LEFT</font></button>
    <button id="xright" class="div5"><font color="blue" >RIGHT</font> </button>
</div>

<div class="div4">
    <br>(on the y-axis)
    <button id="yup" class="div6"><font color="blue" >UP</font></button>
    <button id="ydown" class="div6"><font color="blue" >DOWN</font></button>
</div>


</body>
```

It is head part of code that has class features(div4-div5-div6).

```
.div4{
    float:left;
    position:relative;
    margin-top:30px;
    margin-left:30px;}
.div5{
    margin-top:10px;
    margin-left:10px;
    position:relative;
    width:60px;
    }
.div6{
    margin-top:25px;
    margin-left:10px;
    position:relative;
    width:60px;
    }
```

After classic assignment of getElementById and addEventListener, I declare a function that change the x-y coordinates(btw changecoordinates_y, changecoordinates_x are global variable). I can change location of text by increasing, or decreasing x-y coordinates. Our text will be at 0,0 coordinates as default.

```
//x-y düzlemlerinde kaydırma islemi

var changecoordinatesy1 = document.getElementById("yup");
    changecoordinatesy1.addEventListener("click",
                        function(){  changecoordinates_y += 10;
                        });
var changecoordinatesy2 = document.getElementById("ydown");
    changecoordinatesy2.addEventListener("click",
                        function(){ changecoordinates_y -= 10;
                        });

var changecoordinatesx1 = document.getElementById("xright");
    changecoordinatesx1.addEventListener("click",
                        function(){ changecoordinates_x += 10;
                        });
var changecoordinatesx2 = document.getElementById("xleft");
    changecoordinatesx2.addEventListener("click",
                        function(){  changecoordinates_x -= 10;
                        });
```

The following are the coordinates of the characters. I used gl.TRIANGLES so that every 3 coordinates corresponds to 1 triangle and every 6 coordinates will form part of the character.

```
// coordinates of letters
var vertices = [ // M harfi (30 satır)
                    vec2(-.65, -.5), //sol alt
                    vec2(-.65, 0.5), // sol üst
                    vec2(-.60, 0.5), // sağ üst
                    vec2(-.60, -.5), // sağ alt
                    vec2(-.65, -.5),
                    vec2(-.60, 0.5),
                    vec2(-0.35, -0.5),
                    vec2(-0.40, -0.5),
                    vec2(-.60, 0.32),
                    vec2(-0.35, -0.5), // 4.nokta
                    vec2(-.60, 0.5),
                    vec2(-.60, 0.32),
                    vec2(-0.39, -0.48),
                    vec2(-0.35, -0.5),
                    vec2(-.60, 0.32),
                    vec2(-0.16, 0.5),
                    vec2(-0.16, 0.32),
                    vec2(-0.39, -0.48),
                    vec2(-0.16, 0.5),
                    vec2(-0.16, 0.32),
                    vec2(-0.40, -0.5),
                    vec2(-0.16, 0.32),
                    vec2(-0.35, -0.5),
                    vec2(-0.40, -0.5),
                    vec2(-0.16, 0.5),
                    vec2(-0.11, 0.5),
                    vec2(-0.11, -.5),
                    vec2(-0.11, -.5),
                    vec2(-0.16, -.5),
                    vec2(-0.16, 0.5),
```

```
                // A harfi
                vec2(0.15, -.5),
                vec2(0.2, -.5),
                vec2(0.38, 0.5),
                vec2(0.15, -.5),
                vec2(0.2, -.5),
                vec2(0.38, 0.5),
                vec2(0.15, -.5),
                vec2(0.38, 0.5),
                vec2(0.32, 0.5),
                vec2(0.55, -.5),
                vec2(0.32, 0.5),
                vec2(0.38, 0.5),
                vec2(0.60, -.5),
                vec2(0.55, -.5),
                vec2(0.38, 0.5),

                vec2(0.26, -0.09),
                vec2(0.26, -0.04),
                vec2(0.47, -.04),
                vec2(0.47, -.04),
                vec2(0.47, -.09),
                vec2(0.26, -0.09),
                 ];
```

We used buffer to get vertices into the buffer, and vPosition for vertex attributes, also thetaloc and theta value to rotate our letters. It is same as laboratuary videos.

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

        // Associate out shader variables with our data buffer
        var vPosition = gl.getAttribLocation( program, "vPosition" );
        gl.vertexAttribPointer( vPosition, 2, gl.FLOAT, false, 0, 0 );
        gl.enableVertexAttribArray( vPosition );

        thetaLoc = gl.getUniformLocation(program, "theta");
        theta=0;
        gl.uniform1f(thetaLoc, theta);


        render();

}
```

I defined stoprotating as global variable defaul value is false. When you open the html file you will see that letters are not moving this is the reason. But you click start button stoprotating will be true, and rotation are going to start. Others are same.

```
function render(){

        // default color
        gl.clearColor(0.90, 0.85, 0.75, 1.0);
        gl.clear(gl.COLOR_BUFFER_BIT);
        gl.drawArrays(gl.TRIANGLES, 0, 60);
        gl.viewport(changecoordinates_x,changecoordinates_y,glcanvas.width,glcanvas.height);

        if(stoprotating==true){
            theta += (isDirClockwise ? -0.1 : 0.1);
            gl.uniform1f(thetaLoc, theta);
        }
        setTimeout(function(){
          requestAnimFrame(render);
        },delay);


}
```