Kadir GÜVEN 210315001

Zeynep Naz CEYHAN 210315003

Report of Homework #2: WebGL Basics and Modelling

HTML Code:

```
uniform float scale;

void main()
{
gl_Position =vec4(scale*vPosition.x, scale*vPosition.y, 0.0, 1.0) ;
}
</script>
```

JAVASCRIPT Code:

Attribute

```
var angle=radians(10);
var r=0.4;
var w=0.8;
var scalingFactor = 0.5;
var points = [vec2(r,0),vec2(w,0)];
var color = vec4(Math.random(),Math.random(),Math.random(),1.0);
```

Color Location

```
var colorLoc = gl.getUniformLocation(program,"color");
gl.uniform4fv(colorLoc,color);
```

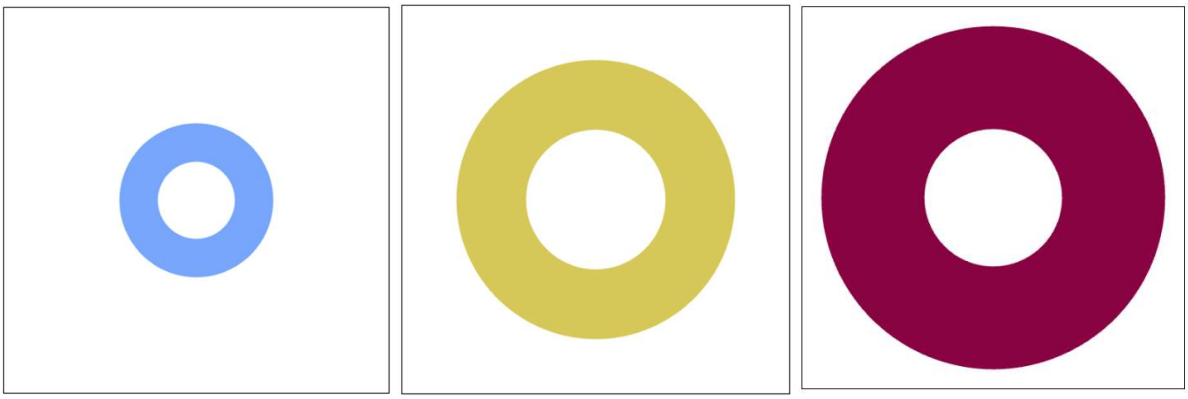
Scale Location

```
var scaleLoc = gl.getUniformLocation(program, "scale");
gl.uniform1f(scaleLoc, scalingFactor);
```

Function

```
function circle(){
    for(var i=0;i<360/angle;i++){
        points.push(vec2(points[0][0]*Math.cos(angle*i),points[0][0]*Math.sin(angle*i)))
        points.push(vec2(points[1][0]*Math.cos(angle*i),points[1][0]*Math.sin(angle*i)))
    }
}
```

The Outputs:



Conclusion

We learned a lot in this project. In the HTML code, we have sized the *gl_position* value with our *Scale* variable that we assigned as static. We have assigned our *Scale* variable to Uniform type. We used the uniform keyword and applied what we learned. We have defined our *Angle* variable in the javascript code. Since there was an error in our code when we started the project, as we changed the *angle* variable, the output was not as we wanted. We had no problems assigning our *colorLoc* and *scaleLoc* variables. We created our for loop inside the *circle function*. We had a hard time writing this for loop at first, but then we figured it out.