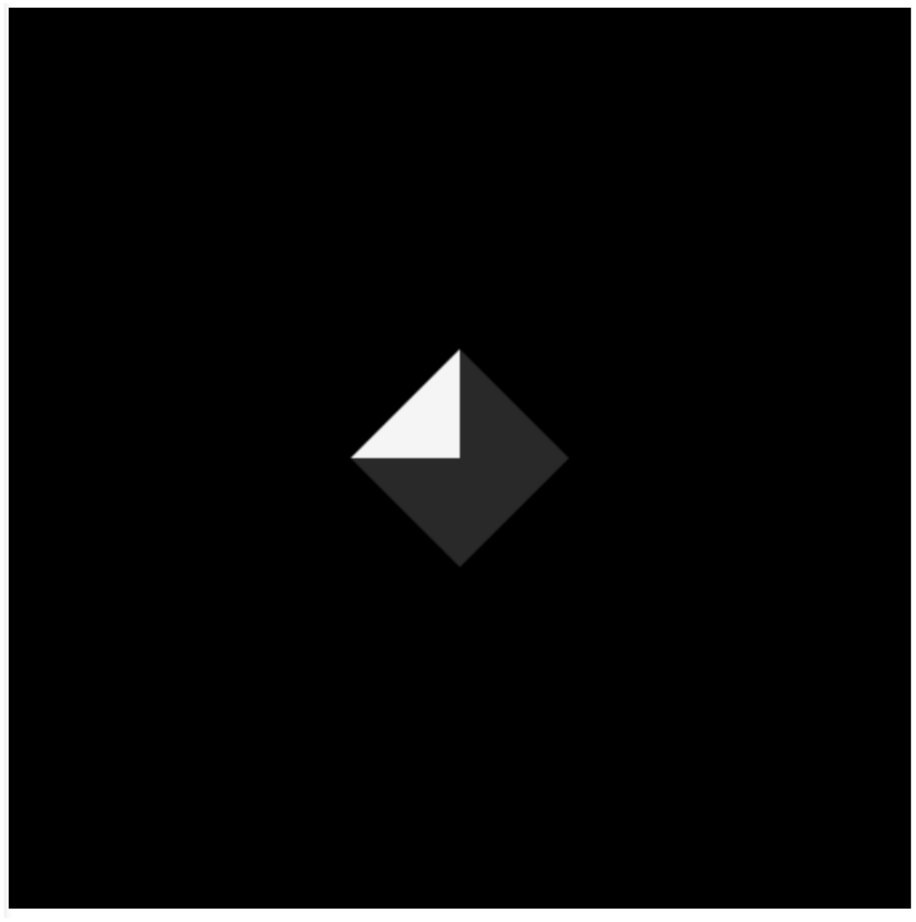
Deren Bozer
COSC-556
MW 11:00 Lab 4

Part 1 - Normals:

- Exercises completed in “bozer-LE1.html” and “bozer-LE1.js”

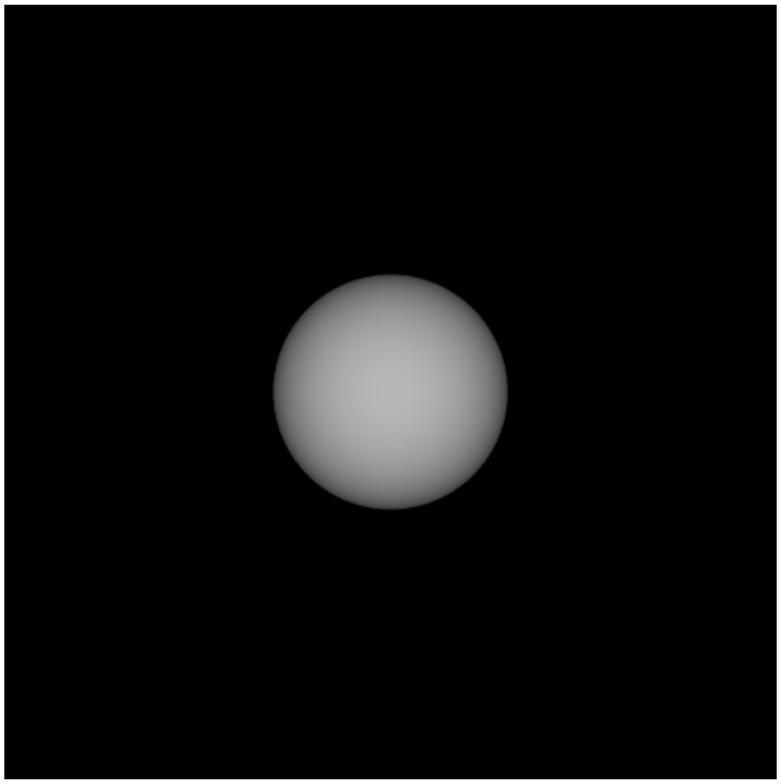


Part 2 - Lights and Materials

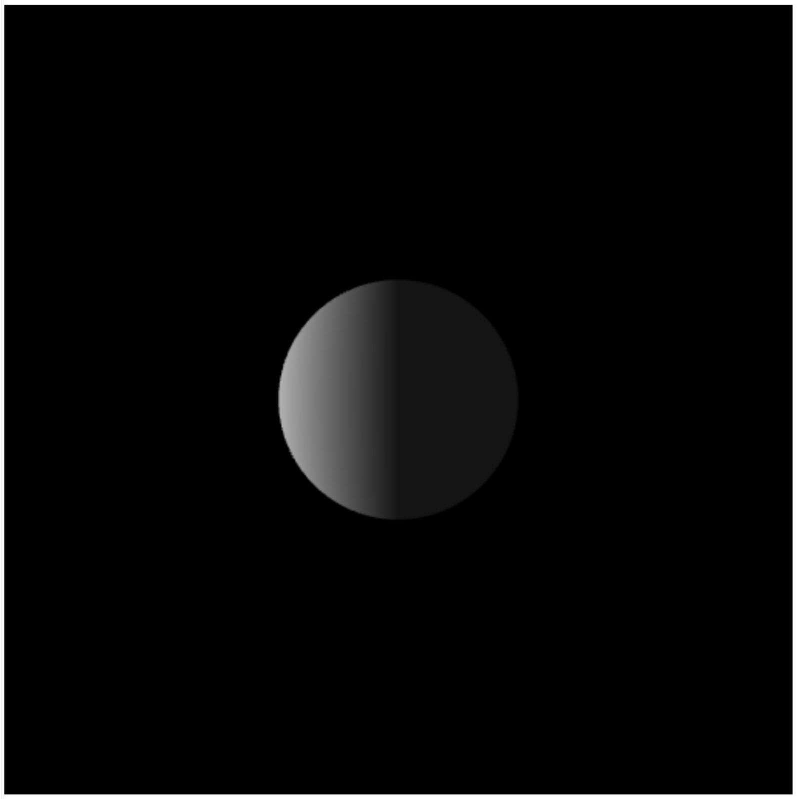
- Exercises completed in “bozer-LE2.html” and “bozer-LE2.js”

2) The default light setting is positional light because it’s not being multiplied against the light’s position.

3.a)

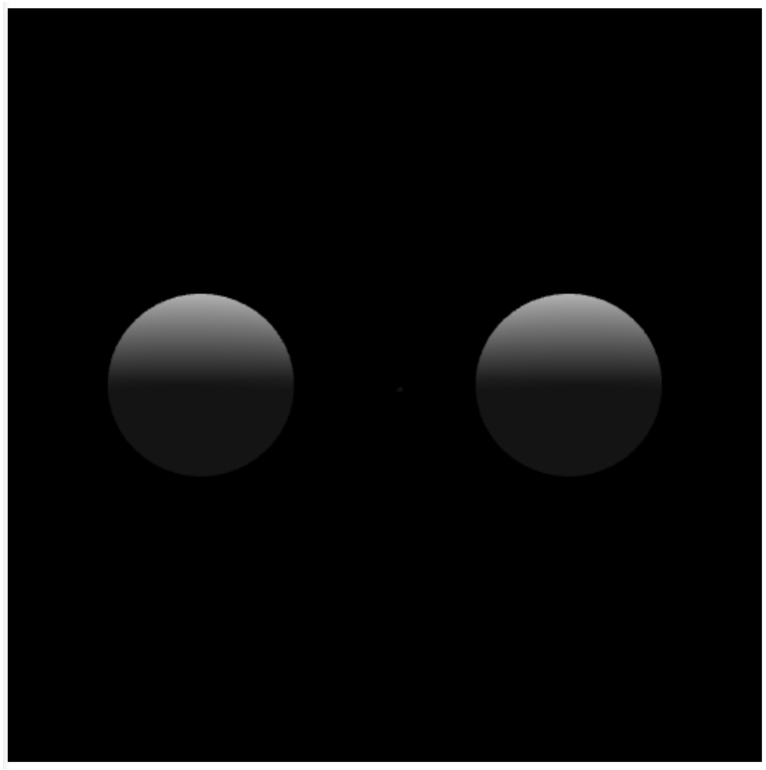


3.b)

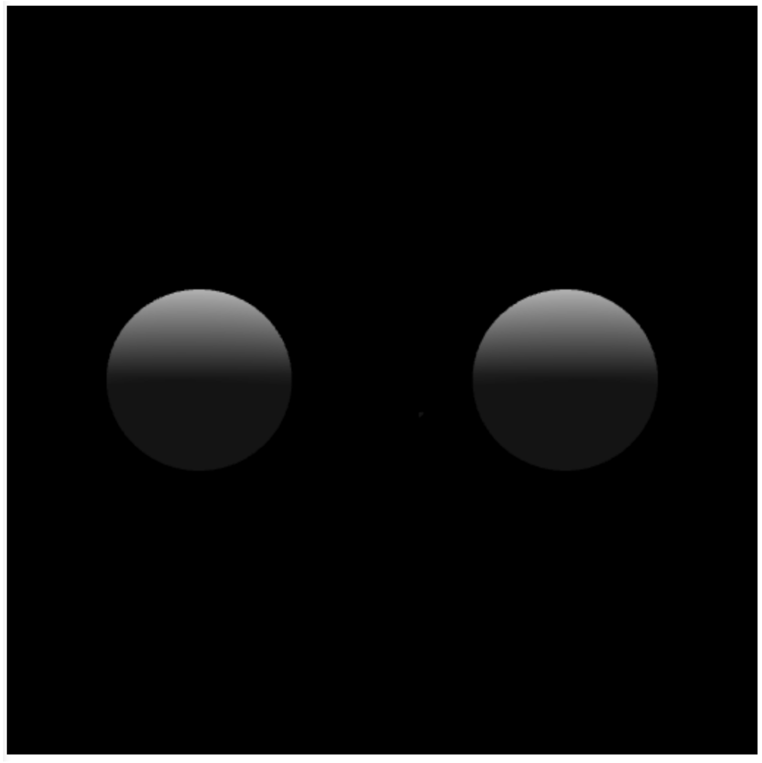


3.c) restored previous values

3.d)

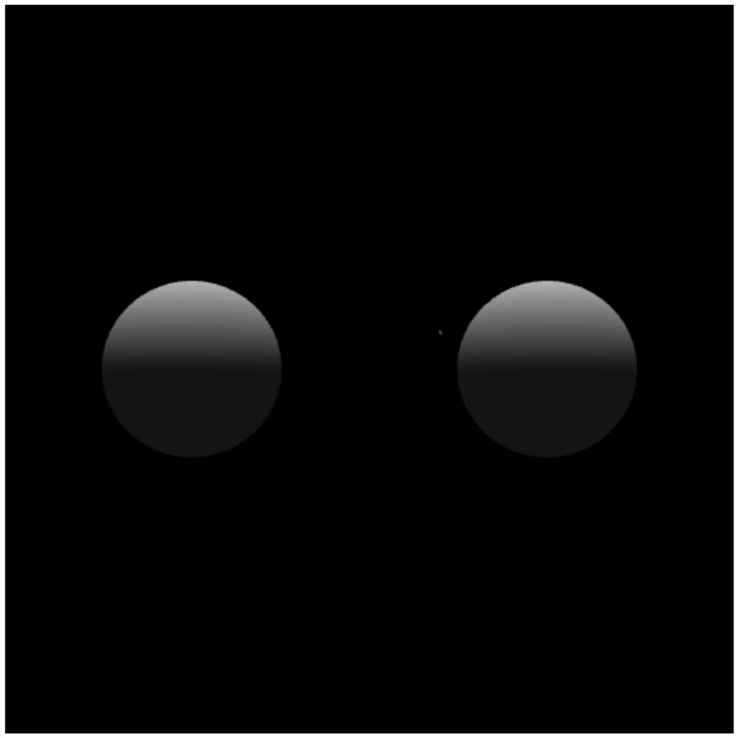


3.e)

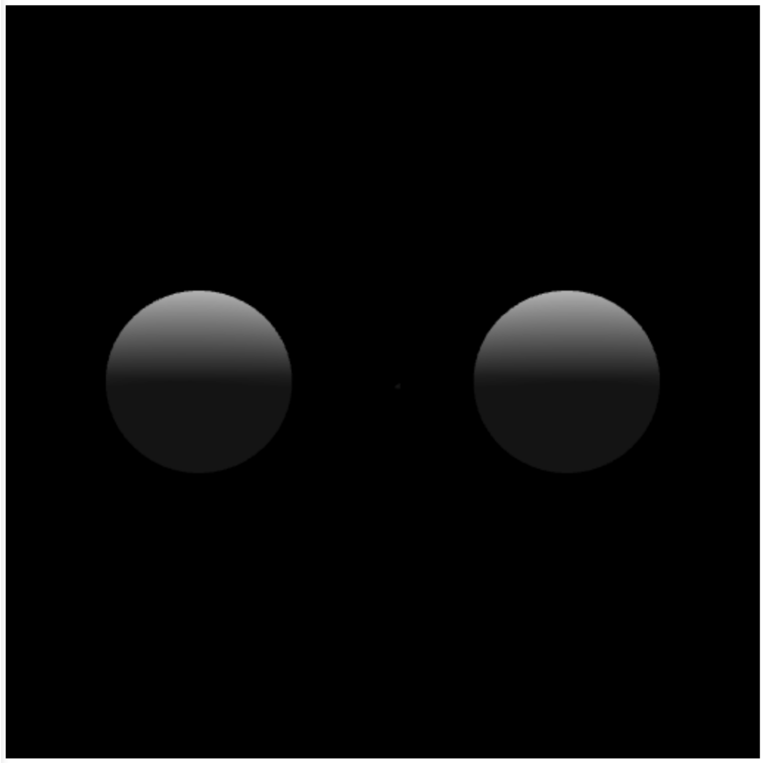


3.f) The MV matrix did not have an effect, it stayed the same.

4b)

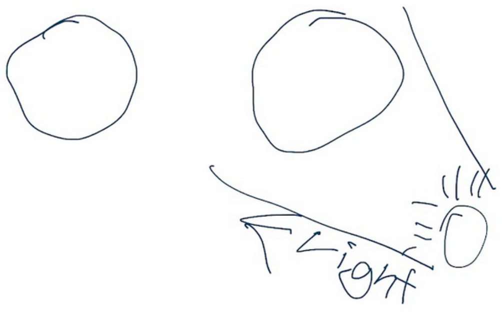


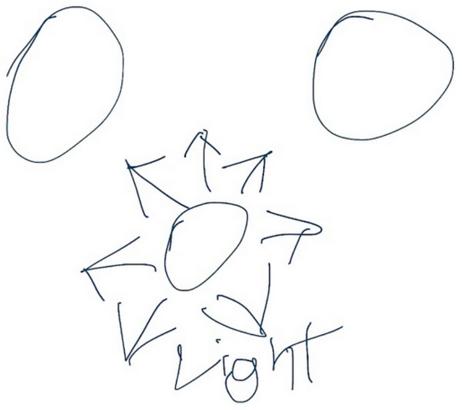
4c)



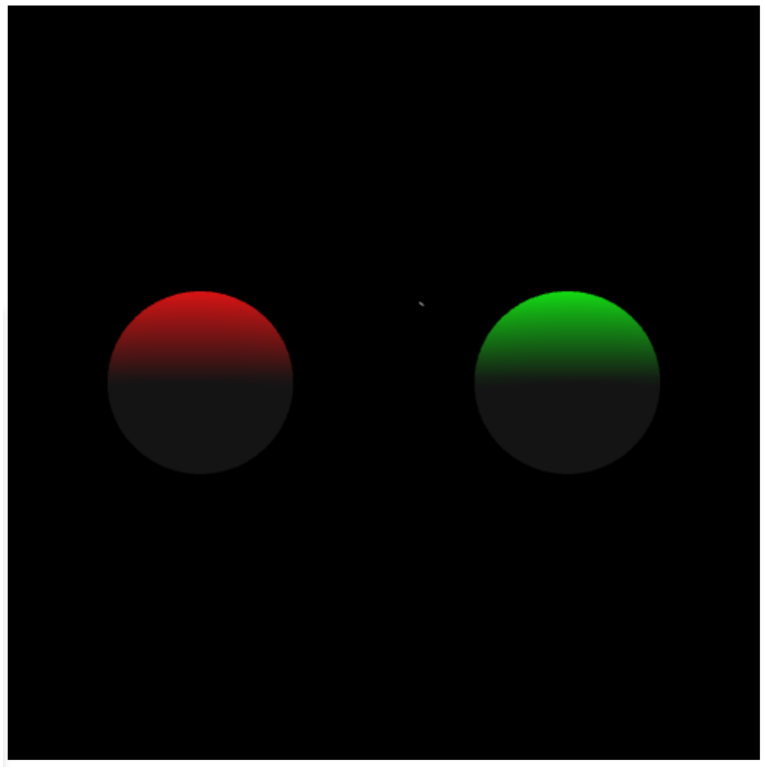
5) The effect is so different because the world coordinate lighting affects the points while the other affects the vectors.

6)

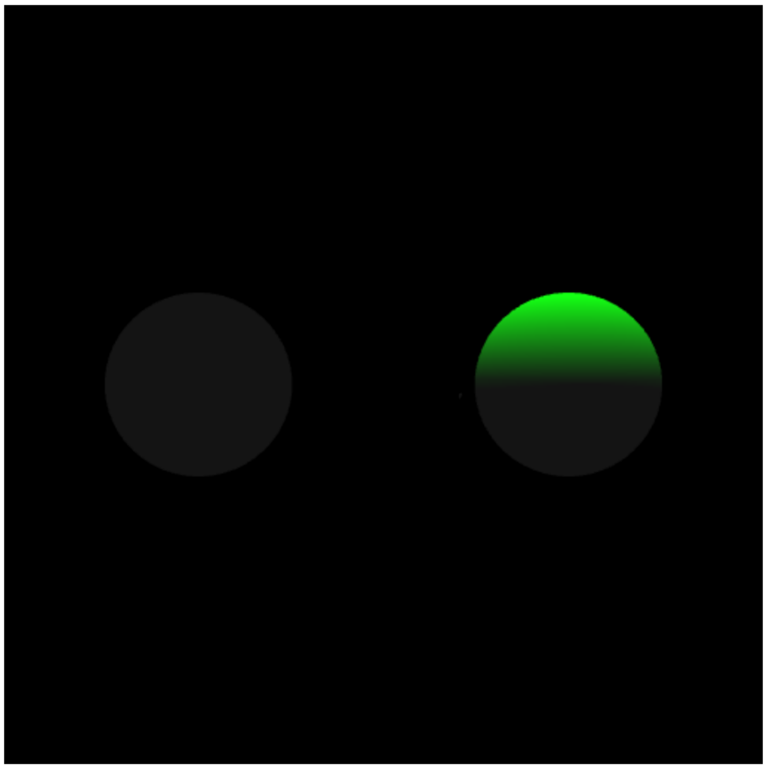




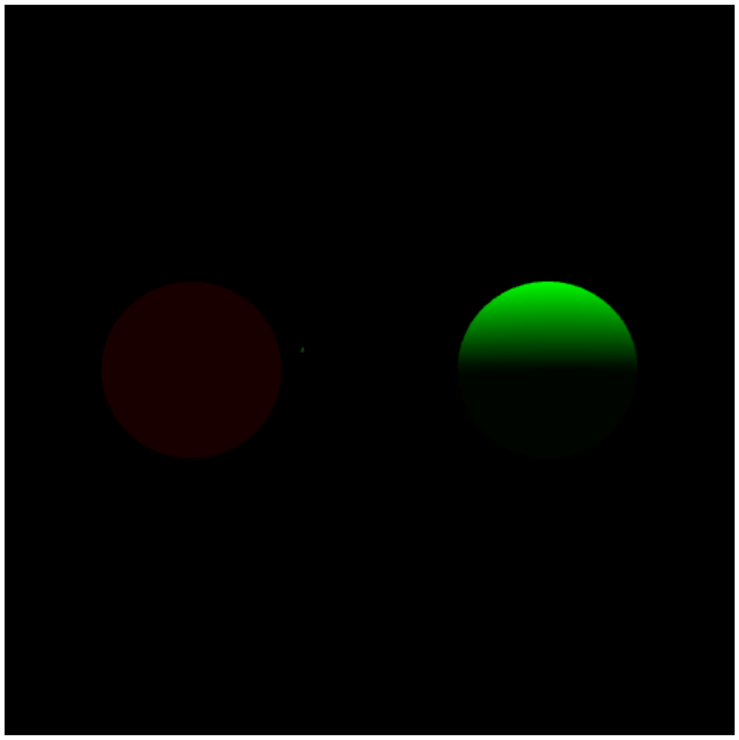
7)



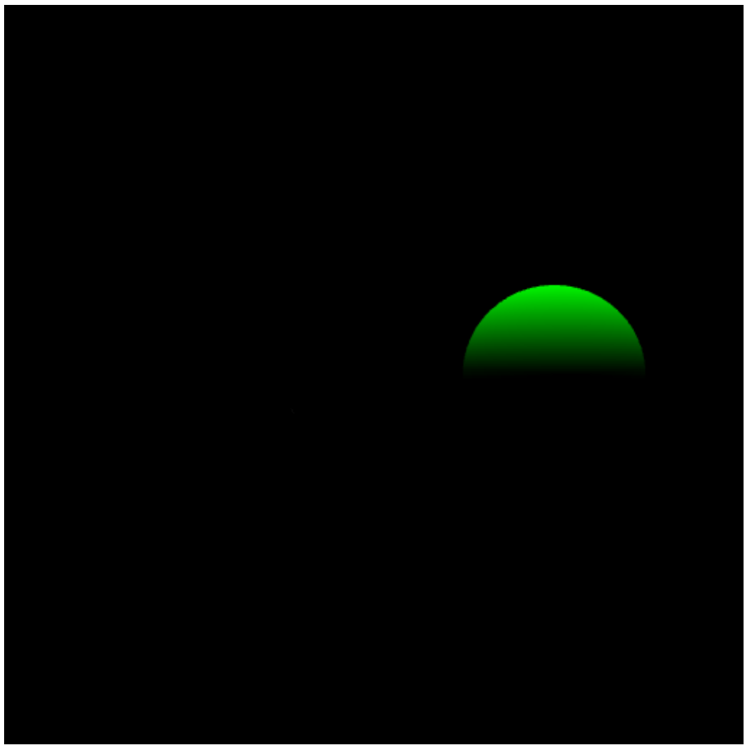
8)



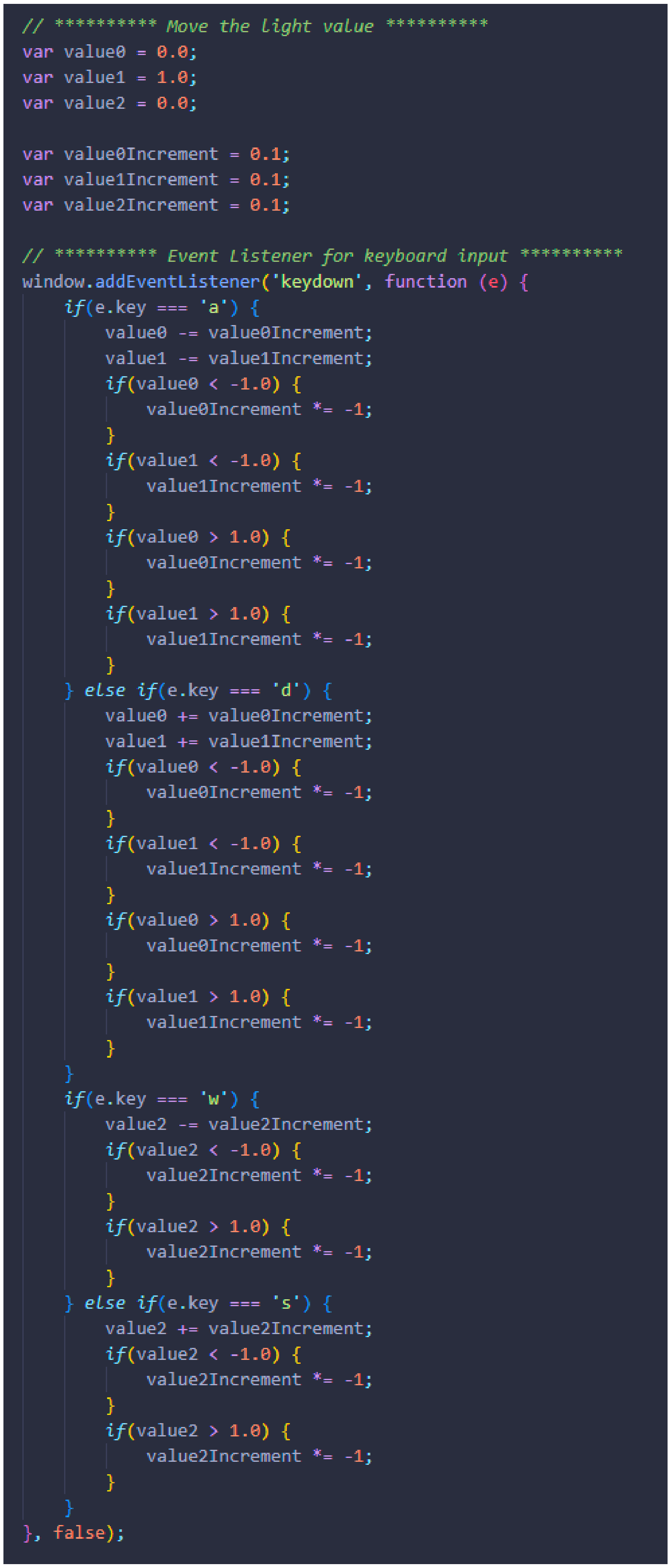
10)



11)



Move that Light



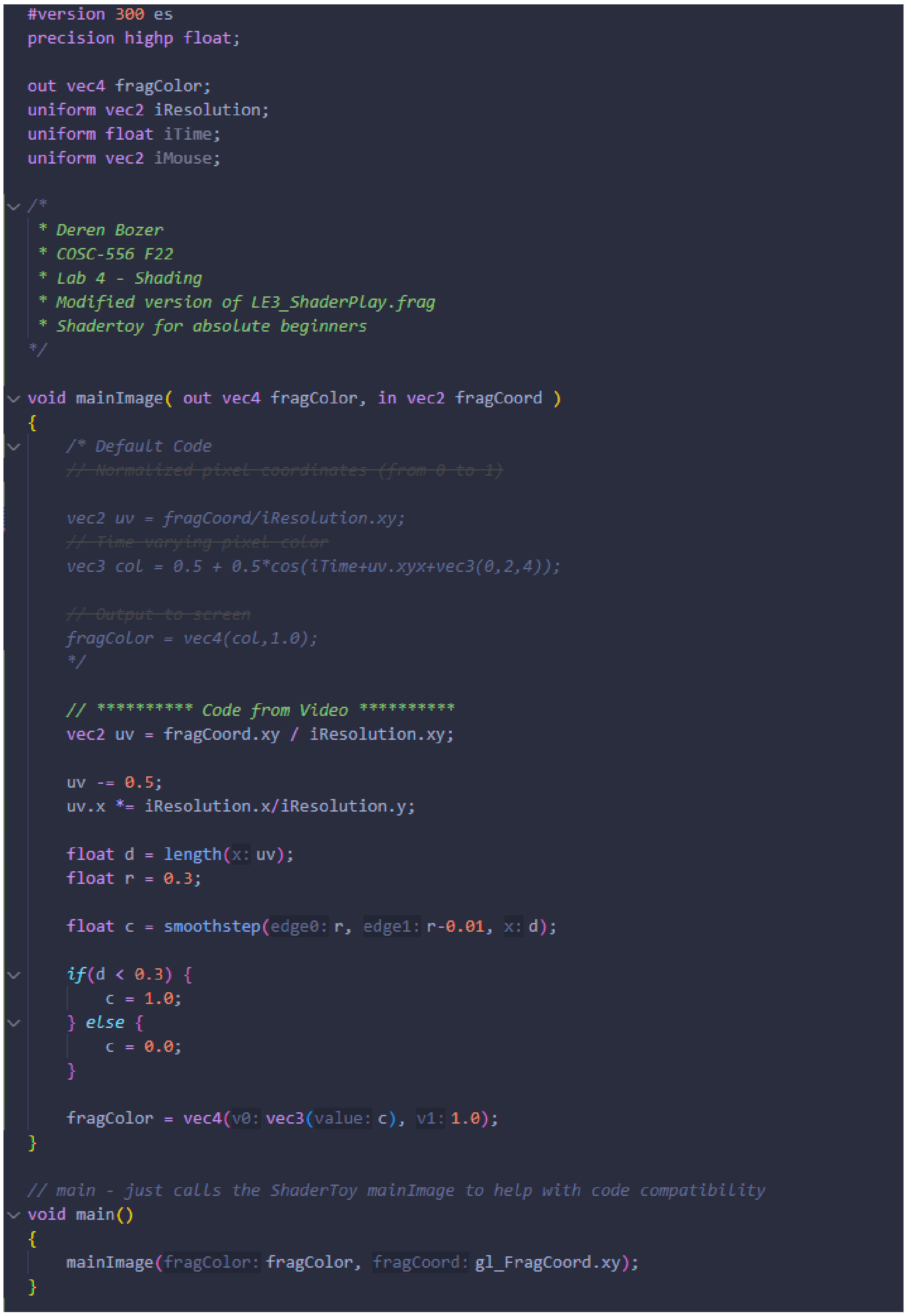
```
// ********** Move the Light value **********
var value0 = 0.0;
var value1 = 1.0;
var value2 = 0.0;

var value0Increment = 0.1;
var value1Increment = 0.1;
var value2Increment = 0.1;

// ********** Event Listener for keyboard input **********
window.addEventListener('keydown', function (e) {
    if(e.key === 'a') {
        value0 -= value0Increment;
        value1 -= value1Increment;
        if(value0 < -1.0) {
            value0Increment *= -1;
        }
        if(value1 < -1.0) {
            value1Increment *= -1;
        }
        if(value0 > 1.0) {
            value0Increment *= -1;
        }
        if(value1 > 1.0) {
            value1Increment *= -1;
        }
    } else if(e.key === 'd') {
        value0 += value0Increment;
        value1 += value1Increment;
        if(value0 < -1.0) {
            value0Increment *= -1;
        }
        if(value1 < -1.0) {
            value1Increment *= -1;
        }
        if(value0 > 1.0) {
            value0Increment *= -1;
        }
        if(value1 > 1.0) {
            value1Increment *= -1;
        }
    }
    if(e.key === 'w') {
        value2 -= value2Increment;
        if(value2 < -1.0) {
            value2Increment *= -1;
        }
        if(value2 > 1.0) {
            value2Increment *= -1;
        }
    } else if(e.key === 's') {
        value2 += value2Increment;
        if(value2 < -1.0) {
            value2Increment *= -1;
        }
        if(value2 > 1.0) {
            value2Increment *= -1;
        }
    }
}, false);
```

Bonus: Shader Play

- Code from following the tutorial



```
#version 300 es
precision highp float;

out vec4 fragColor;
uniform vec2 iResolution;
uniform float iTime;
uniform vec2 iMouse;

/*
 * Deren Bozer
 * COSC-556 F22
 * Lab 4 - Shading
 * Modified version of LE3_ShaderPlay.frag
 * Shadertoy for absolute beginners
 */

void mainImage( out vec4 fragColor, in vec2 fragCoord )
{
    /* Default Code
    ~~// Normalized pixel coordinates (from 0 to 1)~~

    vec2 uv = fragCoord/iResolution.xy;
    ~~// Time varying pixel color~~
    vec3 col = 0.5 + 0.5*cos(iTime+uv.xyx+vec3(0,2,4));

    ~~// Output to screen~~
    fragColor = vec4(col,1.0);
    */

    // ********** Code from Video **********
    vec2 uv = fragCoord.xy / iResolution.xy;

    uv -= 0.5;
    uv.x *= iResolution.x/iResolution.y;

    float d = length(x: uv);
    float r = 0.3;

    float c = smoothstep(edge0: r, edge1: r-0.01, x: d);

    if(d < 0.3) {
        c = 1.0;
    } else {
        c = 0.0;
    }

    fragColor = vec4(v0: vec3(value: c), v1: 1.0);
}

// main - just calls the ShaderToy mainImage to help with code compatibility
void main()
{
    mainImage(fragColor: fragColor, fragCoord: gl_FragCoord.xy);
}
```