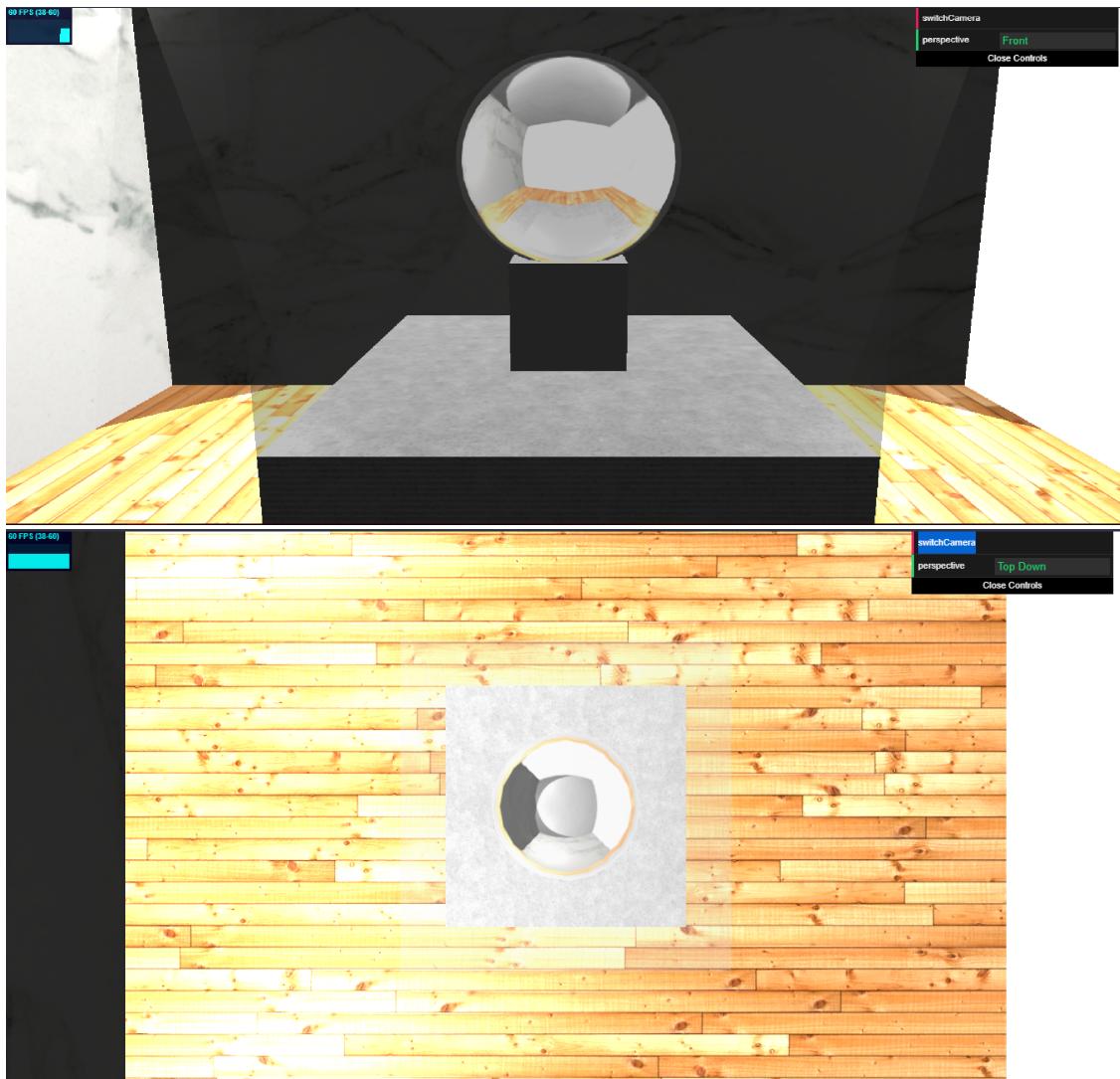


Devin Hardy

00076619

CS423



```
<!DOCTYPE html>
```

```
<HTML>
```

```
<HEAD>
```

```
 <TITLE>Assignment 4</TITLE>
```

```
 <SCRIPT TYPE="text/javascript" SRC="../libs/three.min.js"></SCRIPT>
```

```
 <SCRIPT TYPE="text/javascript" SRC="../libs/stats.min.js"></SCRIPT>
```

```
 <SCRIPT TYPE="text/javascript" SRC="../libs/gui.min.js"></SCRIPT>
```

```
<STYLE>
    body {
        /* set margin to 0 and overflow to hidden, to go fullscreen */
        margin: 0;
        overflow: hidden;
    }
</STYLE>

</HEAD>

<BODY>

<DIV id="Stats-output">
</DIV>

<!-- DIV which will hold the Output --&gt;

&lt;DIV id="WebGL-output"&gt;
&lt;/DIV&gt;

<!-- Javascript code that runs our Three.js examples --&gt;

&lt;SCRIPT TYPE="text/javascript" SRC="04-ASG-Setup.js"&gt;
&lt;/SCRIPT&gt;

&lt;/BODY&gt;

&lt;/HTML&gt;</pre>
```

```
//
// File:
// Author:
// Purpose:
//
function init() {

    var stats = initStats();
```

```
// create a scene, that will hold all our elements such as objects, cameras and lights.  
var scene = new THREE.Scene();  
  
// create a camera, which defines where we're looking at.  
var camera = new THREE.PerspectiveCamera(45, window.innerWidth / window.innerHeight, 0.1,  
1000);  
camera.position.x = 30;  
camera.position.y = 30;  
camera.position.z = 0;  
  
// create a render and set the size  
var renderer = new THREE.WebGLRenderer();  
  
renderer.setClearColor(new THREE.Color(0xEEEEEE, 1.0));  
renderer.setSize(window.innerWidth, window.innerHeight);  
  
// create the ground plane  
var cube = createMesh(new THREE.BoxGeometry(100, 1, 100), "floor-wood.jpg");  
cube.position.y = 0;  
scene.add(cube);  
  
//Ceiling  
var cube = createMesh(new THREE.BoxGeometry(100, 1, 100), "ceiling.jpg");  
cube.position.y = 75;  
scene.add(cube);  
  
// Room  
var cubeGeometry = new THREE.BoxGeometry(1,100,100);
```

```
var cube = createMesh(cubeGeometry, "wall1.jpg");
cube.position.x = 50;
cube.position.y = 25;
cube.position.z = 0;
scene.add(cube);
```

```
var cubeGeometry = new THREE.BoxGeometry(100,100,1);
var cube = createMesh(cubeGeometry, "wall2.jpg");
cube.position.x = 0;
cube.position.y = 25;
cube.position.z = 50;
scene.add(cube);
```

```
var cubeGeometry = new THREE.BoxGeometry(1,100,100);
var cube = createMesh(cubeGeometry, "wall3.jpg");
cube.position.x = -50;
cube.position.y = 25;
cube.position.z = 0;
scene.add(cube);
```

```
var cubeGeometry = new THREE.BoxGeometry(100,100,1);
var cube = createMesh(cubeGeometry, "wall4.jpg");
cube.position.x = 0;
cube.position.y = 25;
cube.position.z = -50;
scene.add(cube);
```

```
// Pedestal cube
```

```

var cubeGeometry = new THREE.BoxGeometry(20,20,20);

var cube = createMesh(cubeGeometry, "stone.jpg");

cube.position.x = 0;

cube.position.y = 10;

cube.position.z = 0;

scene.add(cube);

//Box

var cubeGeometry = new THREE.BoxGeometry(5, 5, 5);

var cube = new THREE.Mesh(cubeGeometry, new THREE.MeshPhongMaterial({color: 0xC4C4C4,
specular: 0x606060, reflectivity: 1.0}));

cube.position.x = 0;

cube.position.y = 23;

cube.position.z = 0;

scene.add(cube);

cubeRenderTarget1 = new THREE.WebGLCubeRenderTarget(128, {format: THREE.RGBFormat,
generateMipmaps: true,minFilter: THREE.LinearMipmapLinearFilter});

cubeCamera1 = new THREE.CubeCamera(0.1, 1000, cubeRenderTarget1);

// Item Sphere

var sphereGeometry = new THREE.SphereGeometry(5, 20, 20);

var sphereMaterial = new THREE.MeshBasicMaterial({reflectivity: 0.9, envMap:
cubeRenderTarget1.texture});

var sphere = new THREE.Mesh(sphereGeometry, sphereMaterial);

```

```
sphere.position.x = 0;  
sphere.position.y = 30;  
sphere.position.z = 0;  
sphere.castShadow = true;  
sphere.add(cubeCamera1);  
scene.add(sphere);
```

```
var directionalLight = new THREE.DirectionalLight(0xffffff, 0.7);  
directionalLight.position.set(-20, 40, 60);  
scene.add(directionalLight);
```

```
// Attempt at glass box  
  
var cubeGeometry = new THREE.BoxGeometry(20, 20, 20);  
  
var cubeMaterial = new THREE.MeshPhongMaterial({color: 0xF7F7F7, opacity: 0.3, transparent:  
true});  
  
var cube = new THREE.Mesh(cubeGeometry, cubeMaterial);  
cube.position.x = 0;  
cube.position.y = 25;
```

```
cube.position.z = 0;
scene.add(cube);

// SpotLights
var spotLight1 = new THREE.SpotLight(0xffffff);
spotLight1.position.set(-49, 50, -49);
spotLight1.castShadow = true;
scene.add(spotLight1);
spotLight1.target.position.set(0, 25, -1);
scene.add(spotLight1.target)

var spotLight2 = new THREE.SpotLight(0xffffff);
spotLight2.position.set(-49, 50, 49);
spotLight2.castShadow = true;
scene.add(spotLight2);
spotLight2.target.position.set(0, 25, 1);
scene.add(spotLight2.target)

// add subtle ambient lighting
var ambientLight = new THREE.AmbientLight(0x292929);
scene.add(ambientLight);

// add the output of the renderer to the html element
document.getElementById("WebGL-output").appendChild(renderer.domElement);
```

```
// call the render function  
  
var step = 0;  
  
  
// Insert Lab03 code here.  
  
// New Controls  
  
var camSpot = 1;  
  
var controls = new function () {  
  
    this.perspective = "Front";  
  
    this.switchCamera = function () {  
  
        if (camSpot == 0) {  
  
            camSpot = 1;  
  
            camera = new THREE.PerspectiveCamera(45, window.innerWidth /  
window.innerHeight, 0.1, 1000);  
  
            camera.position.x = 30;  
  
            camera.position.y = 30;  
  
            camera.position.z = 0;  
  
            camera.lookAt(0, 25, 0);  
  
            this.perspective = "Front";  
  
        } else if (camSpot == 1) {  
  
            camSpot = 2;  
  
            camera = new THREE.PerspectiveCamera(45, window.innerWidth /  
window.innerHeight, 0.1, 1000);  
  
            camera.position.x = 0;  
  
            camera.position.y = 30;  
  
            camera.position.z = 30;  
  
            camera.lookAt(0, 25, 0);  
  
            this.perspective = "Left";  
  
        } else if (camSpot == 2) {  
  
        }  
    }  
};
```

```
        camSpot = 3;

        camera = new THREE.PerspectiveCamera(45, window.innerWidth /
window.innerHeight, 0.1, 1000);

        camera.position.x = -30;
        camera.position.y = 30;
        camera.position.z = 0;
        camera.lookAt(0, 25, 0);
        this.perspective = "Back";

    } else if (camSpot == 3) {

        camSpot = 4;

        camera = new THREE.PerspectiveCamera(45, window.innerWidth /
window.innerHeight, 0.1, 1000);

        camera.position.x = 0;
        camera.position.y = 30;
        camera.position.z = -30;
        camera.lookAt(0, 25, 0);
        this.perspective = "Right";

    } else {

        camSpot = 0;

        camera = new THREE.PerspectiveCamera(45, window.innerWidth /
window.innerHeight, 0.1, 1000);

        camera.position.x = 0;
        camera.position.y = 75;
        camera.position.z = 0;
        camera.lookAt(0, 25, 0);
        this.perspective = "Top Down";

    }

};

};
```

```

var gui = new dat.GUI();
gui.add(controls, 'switchCamera');
gui.add(controls, 'perspective').listen();

// make sure that for the first time, the
// camera is looking at the scene
camera.lookAt(0, 25, 0);
render();

function createMesh(geom, imageFile) {
    var texLoader = new THREE.TextureLoader();
    var texture = texLoader.load("../assets/textures/" + imageFile);
    var mat = new THREE.MeshPhongMaterial();
    mat.map = texture;

    var mesh = new THREE.Mesh(geom, mat);
    return mesh;
}

function render() {

    stats.update();

    cubeCamera1.update(renderer, scene);
    // render using requestAnimationFrame
    requestAnimationFrame(render);
    renderer.render(scene, camera);
}

```

```
function initStats() {  
  
    var stats = new Stats();  
  
    stats.setMode(0); // 0: fps, 1: ms  
  
    // Align top-left  
    stats.domElement.style.position = 'absolute';  
    stats.domElement.style.left = '0px';  
    stats.domElement.style.top = '0px';  
  
    document.getElementById("Stats-output").appendChild(stats.domElement);  
  
    return stats;  
}  
}  
  
window.onload = init
```

60 FPS (38-60)

