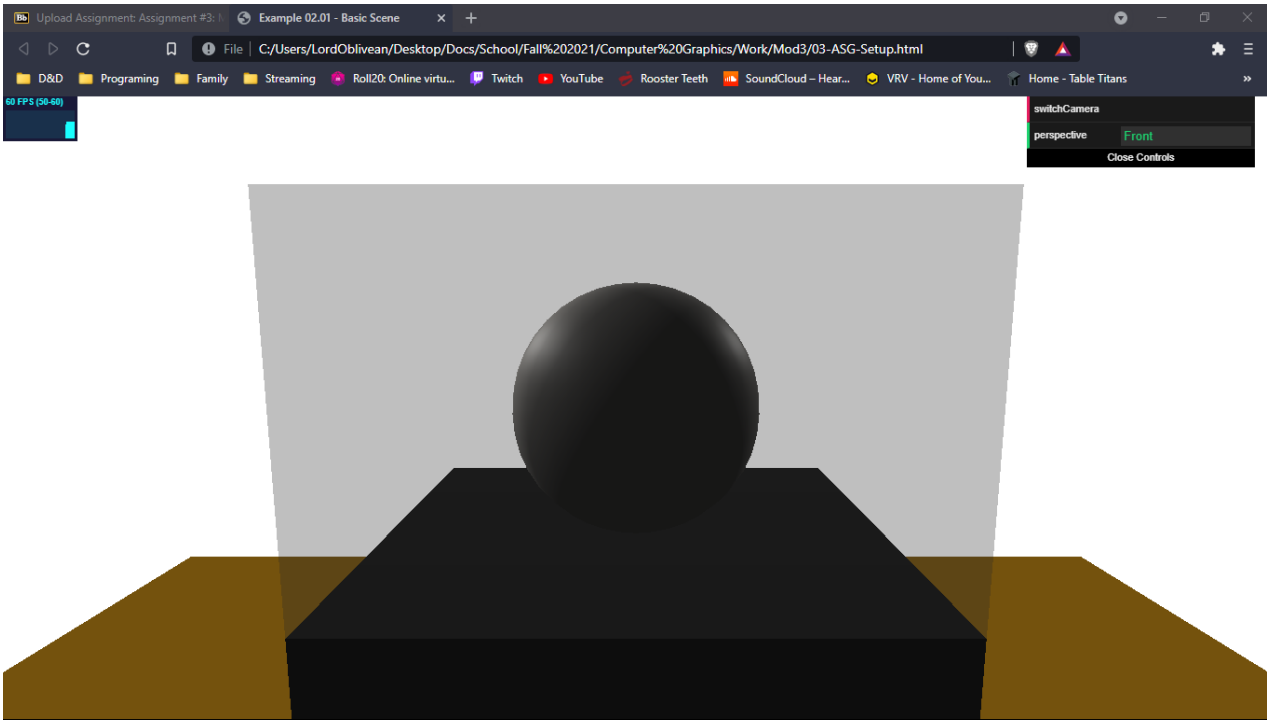
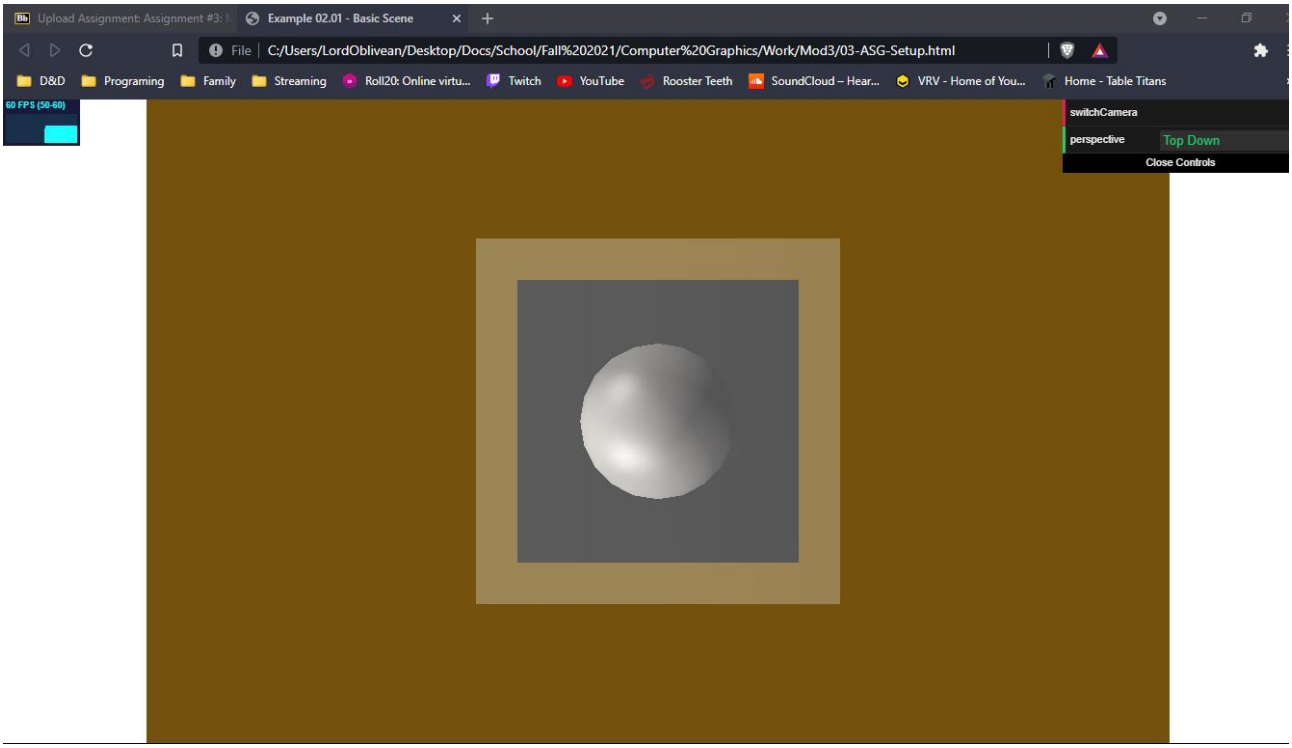
Devin Hardy

00076619

CS423





```
<!DOCYTPE html>
<HTML>
<HEAD>
	<TITLE>Example 02.01 - Basic Scene</TITLE>
	<SCRIPT TYPE="text/javascript" SRC="../libs/three.js"></SCRIPT>
	<SCRIPT TYPE="text/javascript" SRC="../libs/stats.min.js"></SCRIPT>
	<SCRIPT TYPE="text/javascript" SRC="../libs/dat.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 -->
<DIV id="WebGL-output">
</DIV>
<!-- Javascript code that runs our Three.js examples -->
<SCRIPT TYPE="text/javascript" SRC="03-ASG-Setup.js">
</SCRIPT>
</BODY>
</HTML>

//
// 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 planeGeometry = new THREE.PlaneGeometry(180, 180);
    var planeMaterial = new THREE.MeshBasicMaterial({color: 0x74520D});
    var plane = new THREE.Mesh(planeGeometry, planeMaterial);
```

```
// rotate and position the plane
plane.rotation.x = -0.5 * Math.PI;
plane.position.x = 0;
plane.position.y = 0;
plane.position.z = 0;

// add the plane to the scene
scene.add(plane);

// Room
var cubeGeometry = new THREE.BoxGeometry(1,100,100);
var cubeMaterial = new THREE.MeshBasicMaterial({color: 0xFFFFFF});
var cube = new THREE.Mesh(cubeGeometry, cubeMaterial);
cube.position.x = 50;
cube.position.y = 25;
cube.position.z = 0;
scene.add(cube);

var cubeGeometry = new THREE.BoxGeometry(100,100,1);
var cubeMaterial = new THREE.MeshBasicMaterial({color: 0xFFFFFF});
var cube = new THREE.Mesh(cubeGeometry, cubeMaterial);
cube.position.x = 0;
cube.position.y = 25;
cube.position.z = 50;
scene.add(cube);

var cubeGeometry = new THREE.BoxGeometry(1,100,100);
var cubeMaterial = new THREE.MeshBasicMaterial({color: 0xFFFFFF});
var cube = new THREE.Mesh(cubeGeometry, cubeMaterial);
```

```
cube.position.x = -50;
cube.position.y = 25;
cube.position.z = 0;
scene.add(cube);

var cubeGeometry = new THREE.BoxGeometry(100,100,1);
var cubeMaterial = new THREE.MeshBasicMaterial({color: 0xFFFFFF});
var cube = new THREE.Mesh(cubeGeometry, cubeMaterial);
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 cubeMaterial = new THREE.MeshLambertMaterial({color: 0x0F0F0F});
var cube = new THREE.Mesh(cubeGeometry, cubeMaterial);
cube.position.x = 0;
cube.position.y = 10;
cube.position.z = 0;
scene.add(cube);

        // Item Sphere
var sphereGeometry = new THREE.SphereGeometry(5, 20, 20);
var sphereMaterial = new THREE.MeshPhongMaterial({color: 0x64625F});
var sphere = new THREE.Mesh(sphereGeometry, sphereMaterial);
sphere.position.x = 0;
```

```
    sphere.position.y = 25;
    sphere.position.z = 0;
    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(20,15,20);
        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 {
				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 render() {

        stats.update();
        // 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
```