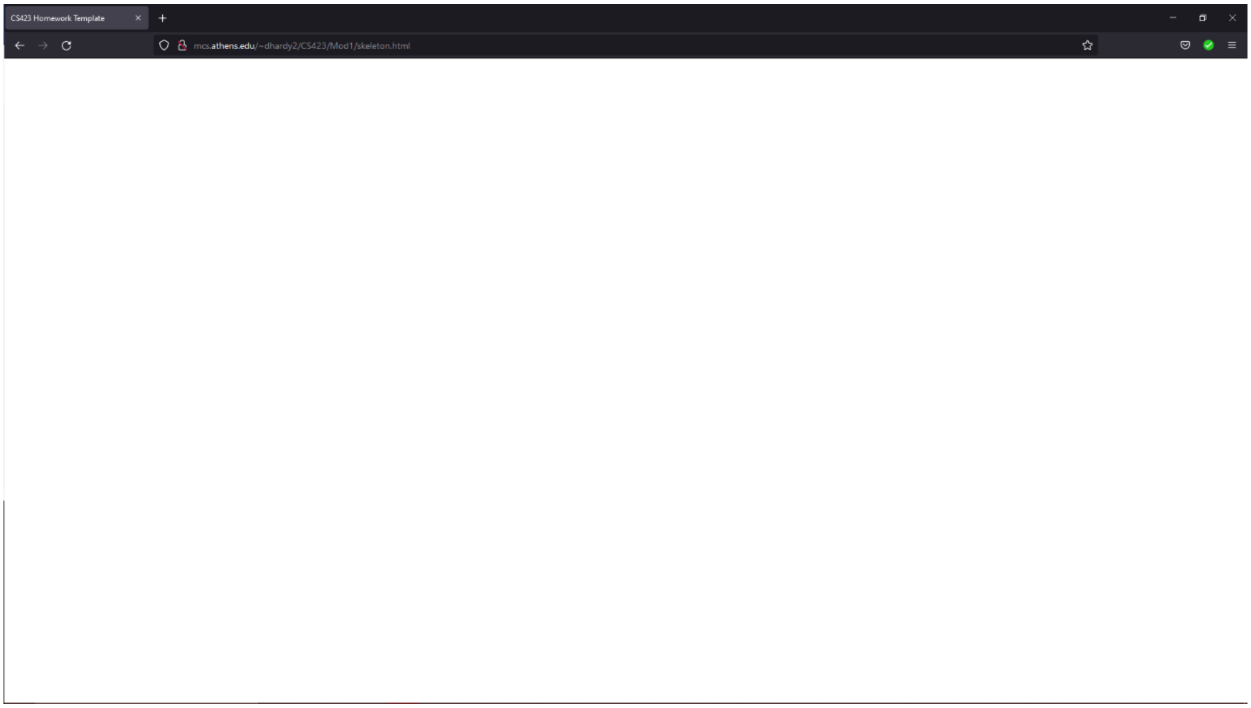
Devin Hardy

00076619

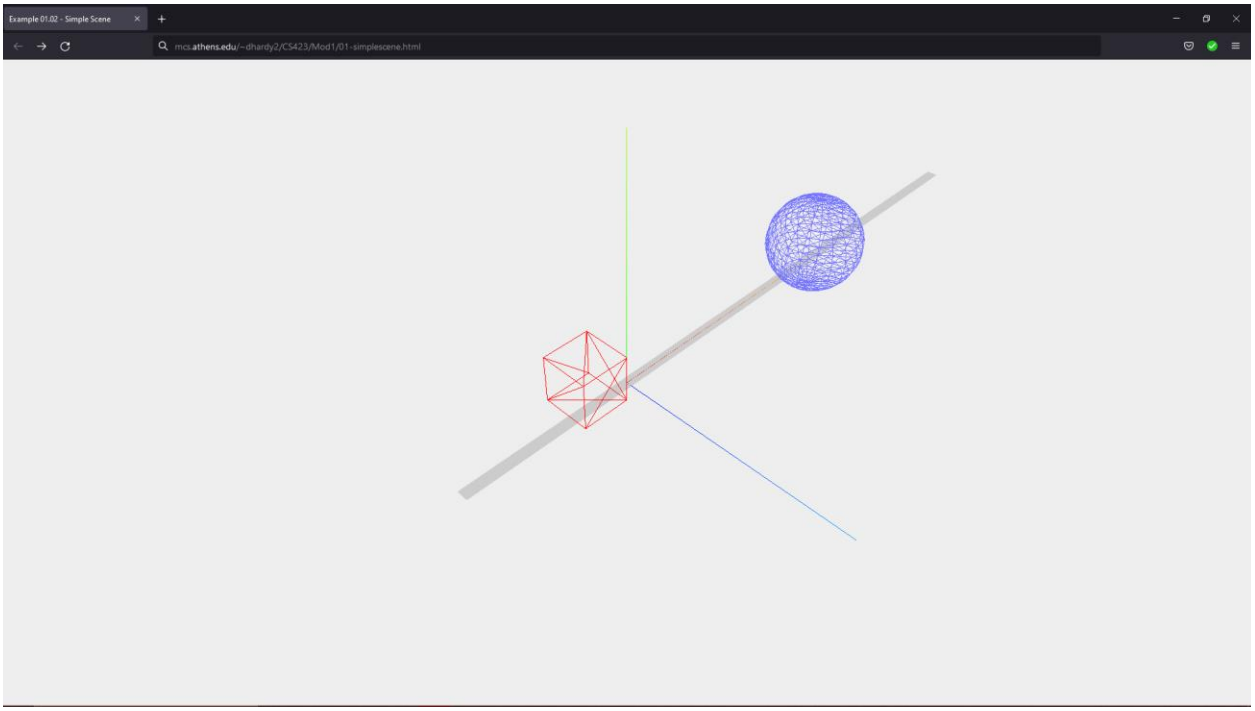
CS423

Skeleton



```
<!DOCTYPE html>

<HTML>

        <HEAD>

         <TITLE>CS423 Homework Template</TITLE>

         <SCRIPT

          TYPE="text/javascript"

          SRC="../libs/three.js">

         </SCRIPT>

         <STYLE>

          body {

          margin: 0;

          overflow: hidden;

          }
        </STYLE>
       </HEAD>
       <BODY>
        <DIV ID="WebGL-output">
        </DIV>
        <!-- Scripts that we use for running things -->
        <SCRIPT TYPE="text/javascript">
         // Put the bulk of what we do in the onlad handler
         // for the window.
         funcion init() {
        // Put Three.js stuff here.
         }
         window.onload = init
        </SCRIPT>
       </BODY>
</HTML>
```

Simple Scene


Example 01.02 - Simple Scene
mcs.athens.edu/~dhardy2/CS423/Mod1/01-simplescene.html

```
<!DOCTYPE html>

<!--File: 01-simplescene.html -->

<html>

 <head>

  <title> Example 01.02 - Simple Scene </title>

  <script

   TYPE ="text/javascript"

   SRC="../libs/three.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="./01-simplescene.js"></script>
  </body>
</html>
```

```
//
// File: 01-simplescene.js
// Purpose:
// Demo some of the basics of working wit the scenegraph.
// Thisis an extension of code from the Learning Three.js textbook
// once everything is loadded, we run our Three.js stuff
function init() {
  var scene = new THREE.Scene();
  var extent = window.innerWidth / window.innerHeight;
  var camera = new THREE.PerspectiveCamera(45, extent, 0.1, 1000);
  var renderer = new THREE.WebGLRenderer();
  renderer.setClearColor(0xEEEEEE,1.0);
  renderer.setSize(window.innerWidth,window.innerHeight);

  // Drop a axis set into the scene
  var axes = new THREE.AxisHelper(20);
  scene.add(axes);
```

```
// Let's add a base plane upon which we place objects.
var planeGeometry = new THREE.PlaneGeometry(60.20);
var planeMaterial = new THREE.MeshBasicMaterial({color:0xCCCCCC});
var plane = new THREE.Mesh(planeGeometry, planeMaterial);
plane.rotation.x = -0.5 * Math.PI;
plane.position.x = 15;
plane.position.y = 0;
plane.position.z = 0;
scene.add(plane);

// Add a cube
var cubeGeometry = new THREE.BoxGeometry(4,4,4);
var cubeMaterial = new THREE.MeshBasicMaterial({color: 0xFF0000, wireframe: true});
var cube = new THREE.Mesh(cubeGeometry, cubeMaterial);
cube.position.x = -4;
cube.position.y = 3;
cube.position.z = 0;
scene.add(cube);
// Add a sphere
var sphereGeometry = new THREE.SphereGeometry(4, 20, 20);
var sphereMaterial = new THREE.MeshBasicMaterial({color: 0x7777ff, wireframe: true});
var sphere = new THREE.Mesh(sphereGeometry, sphereMaterial);
sphere.position.x = 20;
sphere.position.y = 4;
sphere.position.z = 2;
scene.add(sphere);

//Need to tell Three.js the point from where we're viewing the scene
camera.position.x = -30;
```

```
    camera.position.y = 40;
    camera.position.z = 30;
    camera.lookAt(scene.position);
    //Now update the page by attaching the renderer to appropriate place in the
    // HTML DOM for a page and then tell the renderer to render the scene
    document.getElementById("WebGL-output").appendChild(renderer.domElement);
    renderer.render(scene,camera);

}
window.onload = init;
```

Files not included three.js