

Building a Basic Three.js Web Application with .gltf and .obj 3D Models

Content

- [Foreword](#)
- [.GLTF MODEL](#)
 - Obtain model files and appropriate loader
 - [Import the Appropriate files](#)
 - [Create a scene](#)
 - Scene
 - [Camera](#)
 - Renderer
 - [Loading The Model](#)
 - [Run a Local Server](#)
 - [Resources Used](#)
- [.OBJ Model](#)
 - Obtain model files and appropriate loader
 - [Import appropriate files](#)
 - [Create a scene](#)
 - Scene
 - [Camera](#)
 - Renderer
 - [Loading The Model](#)
 - [Run a local server](#)
 - [Resources Used](#)

Foreword

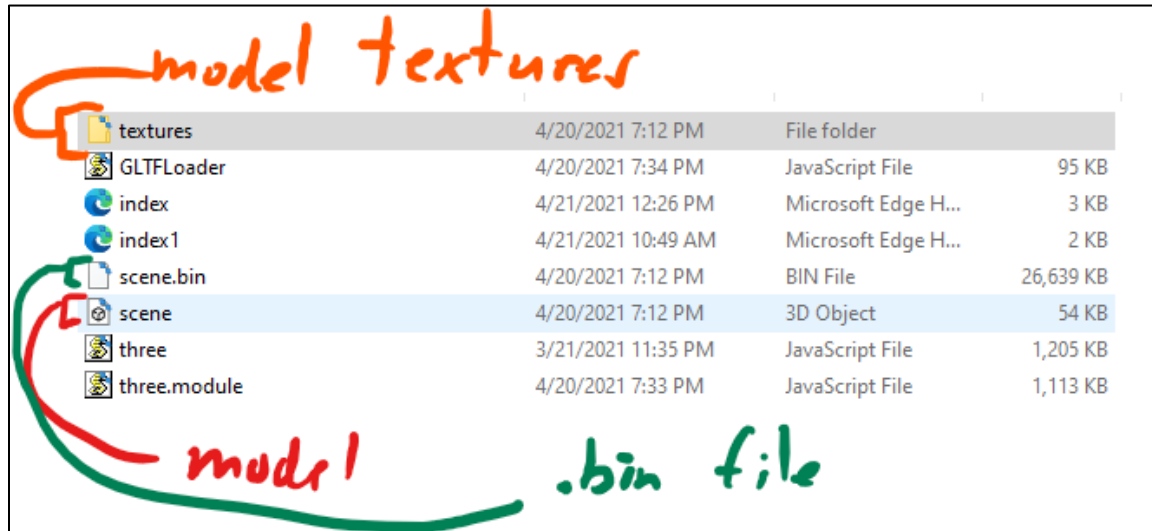
This guide assumes that you have access to code editing software such as Visual Studio Code and have the ability to access a server to run the three.js application on. The script shown in this guide is all contained in Visual Studio Code and is available in the available Github repository:

[lifesaver3/loading_OBJ_and_GLTF_in_three.js: A quick reference guide on loading .obj and .gltf 3d models in the web browser using three.js \(github.com\)](https://github.com/lifesaver3/loading_OBJ_and_GLTF_in_three.js)

.GLTF MODEL

Obtain a 3D Model in a .gltf file format. The model in this guide consists of:

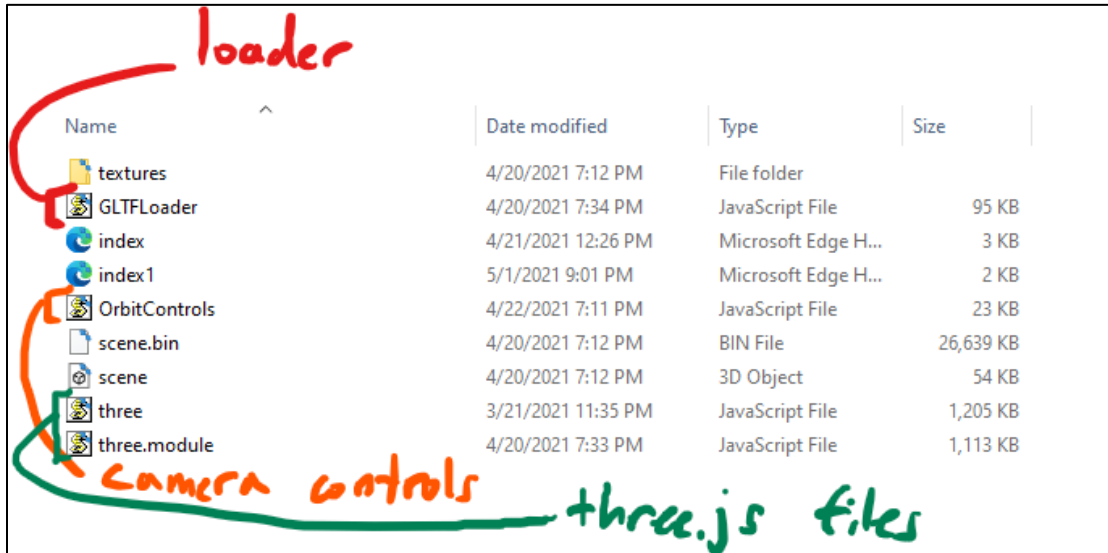
- A 3D Object (a car),
- Textures for the model,
- And a .bin file.



textures	4/20/2021 7:12 PM	File folder	
GLTFLoader	4/20/2021 7:34 PM	JavaScript File	95 KB
index	4/21/2021 12:26 PM	Microsoft Edge H...	3 KB
index1	4/21/2021 10:49 AM	Microsoft Edge H...	2 KB
scene.bin	4/20/2021 7:12 PM	BIN File	26,639 KB
scene	4/20/2021 7:12 PM	3D Object	54 KB
three	3/21/2021 11:35 PM	JavaScript File	1,205 KB
three.module	4/20/2021 7:33 PM	JavaScript File	1,113 KB

Import the Appropriate Files

For all three.js applications, a copy of the “three.js” file is required. For this example, “GLTFLoader.js” and “three.module.js” are also needed. For simple pathing, all of the necessary files are located in the same directory.



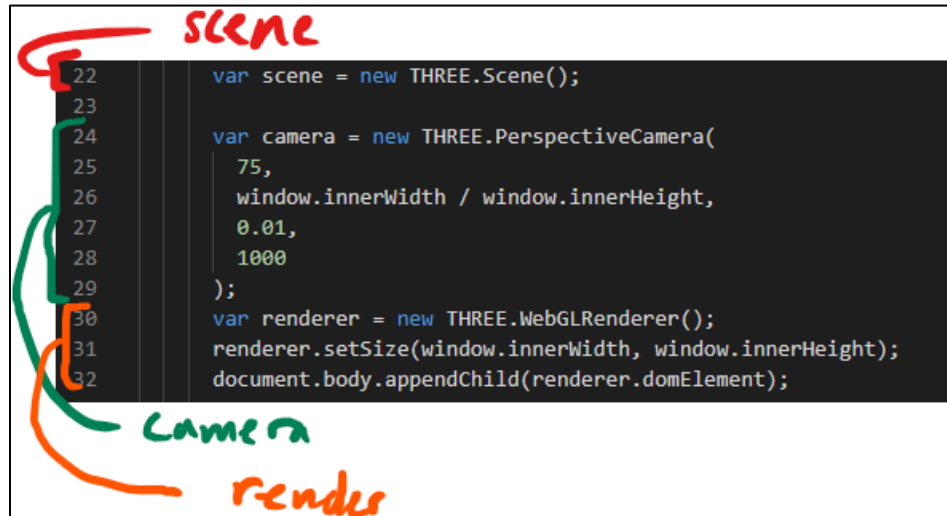
Name	Date modified	Type	Size
textures	4/20/2021 7:12 PM	File folder	
GLTFLoader	4/20/2021 7:34 PM	JavaScript File	95 KB
index	4/21/2021 12:26 PM	Microsoft Edge H...	3 KB
index1	5/1/2021 9:01 PM	Microsoft Edge H...	2 KB
OrbitControls	4/22/2021 7:11 PM	JavaScript File	23 KB
scene.bin	4/20/2021 7:12 PM	BIN File	26,639 KB
scene	4/20/2021 7:12 PM	3D Object	54 KB
three	3/21/2021 11:35 PM	JavaScript File	1,205 KB
three.module	4/20/2021 7:33 PM	JavaScript File	1,113 KB

In order to use the necessary files, they must be called upon and imported into the application. GLTFLoader.js is a module and must be called in as a module-type in order to load the .GLTF model.

```
17 <script src="three.js"></script>
18 <script src="OrbitControls.js"></script>
19 <script type="module" src="GLTFLoader.js"></script>
20 <script type="module">
21   import { GLTFLoader } from "./GLTFLoader.js";
```

Create a Scene

To display the 3D model using three.js, three things are needed first: **a scene**, **a camera**, and **a renderer**.



Scene

In order to see the model, lighting is created as variables and then added to the scene. A background is also set in order to provide color into the picture.

```
49 var light = new THREE.HemisphereLight(0xffffff, 0x000000, 10);
50 scene.add(light);
51 camera.position.set(0, 10, 40);

42 scene.background = new THREE.Color(0x20B2AA);
```

Camera

In order to be able to use a mouse to rotate the camera view, three.js' `OrbitalControl` application must be imported. Then, a variable for controls of Three.js; `OrbitalControls` camera can be created.

```
18 <script src="OrbitControls.js"></script>
```

```
25 var camera = new THREE.PerspectiveCamera(  
26     75,  
27     window.innerWidth / window.innerHeight,  
28     0.01,  
29     1000  
30 );
```

```
44 var controls = new THREE.OrbitControls (camera, renderer.domElement);  
45 controls.enableDamping = true;  
46 controls.dampingFactor = 0.25;  
47 controls.enableZoom = true;
```

Renderer

With these arguments, the application will always render to the size of the web browser window.

```
31 var renderer = new THREE.WebGLRenderer();  
32 renderer.setSize(window.innerWidth, window.innerHeight);  
33 document.body.appendChild(renderer.domElement);
```

Loading The Model

GLTFLoader is added in as a new variable and can then load in the 3D model file and add it to the scene.

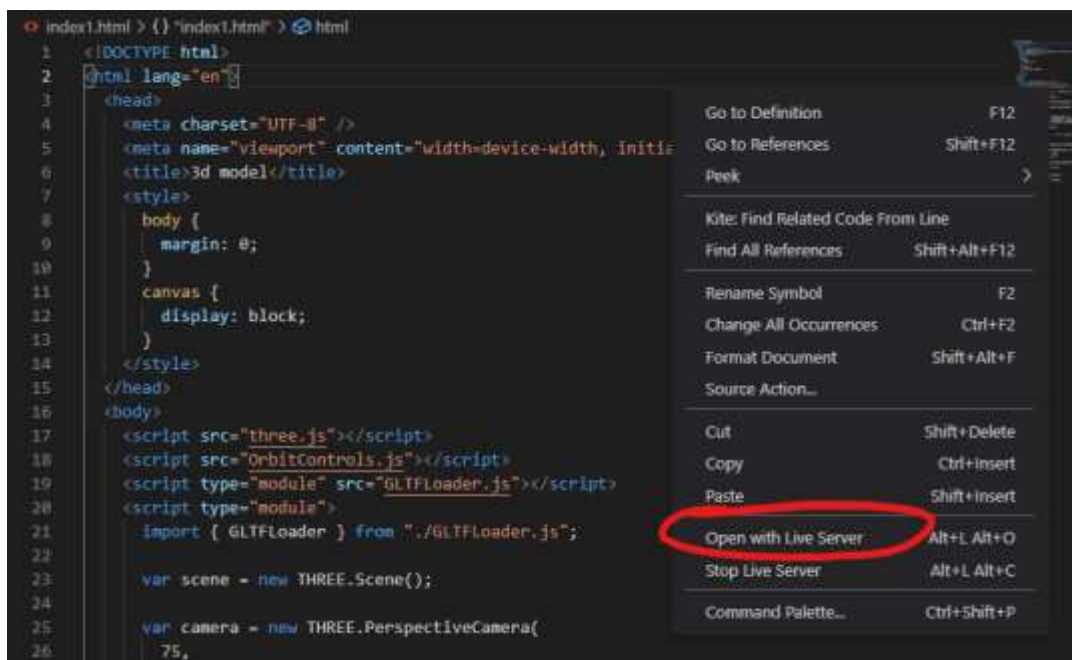
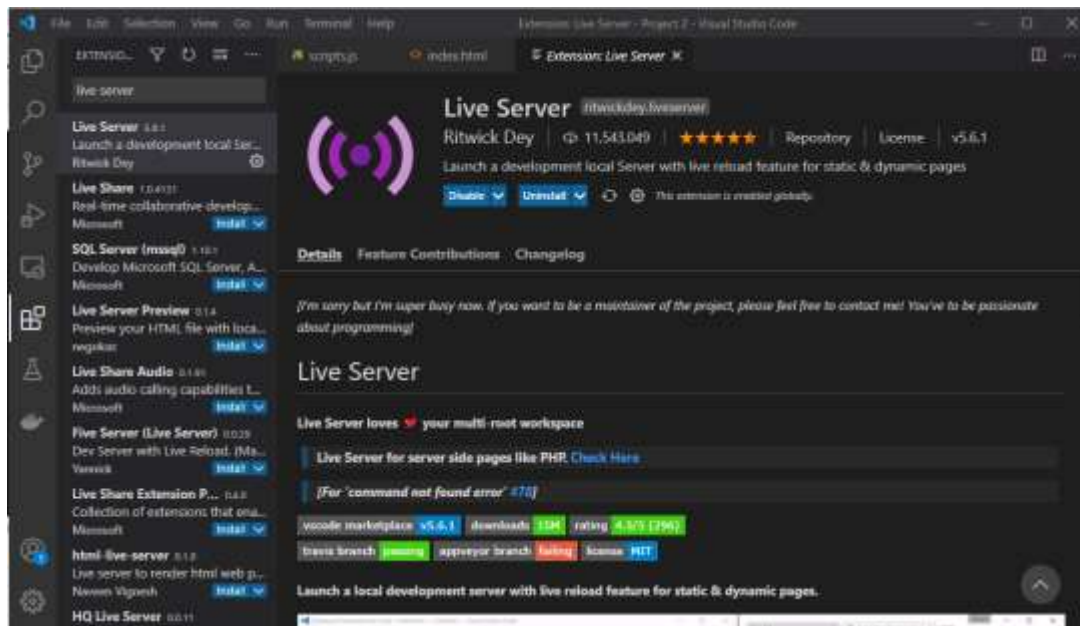
```
35     var loader = new GLTFLoader();
36
37     var obj;
38     loader.load("scene.gltf", function (gltf) {
39         obj = gltf.scene;
40         scene.add(gltf.scene);
41     });
```

This code will rotate the model along it's y-axis as well as animate the entire scene.

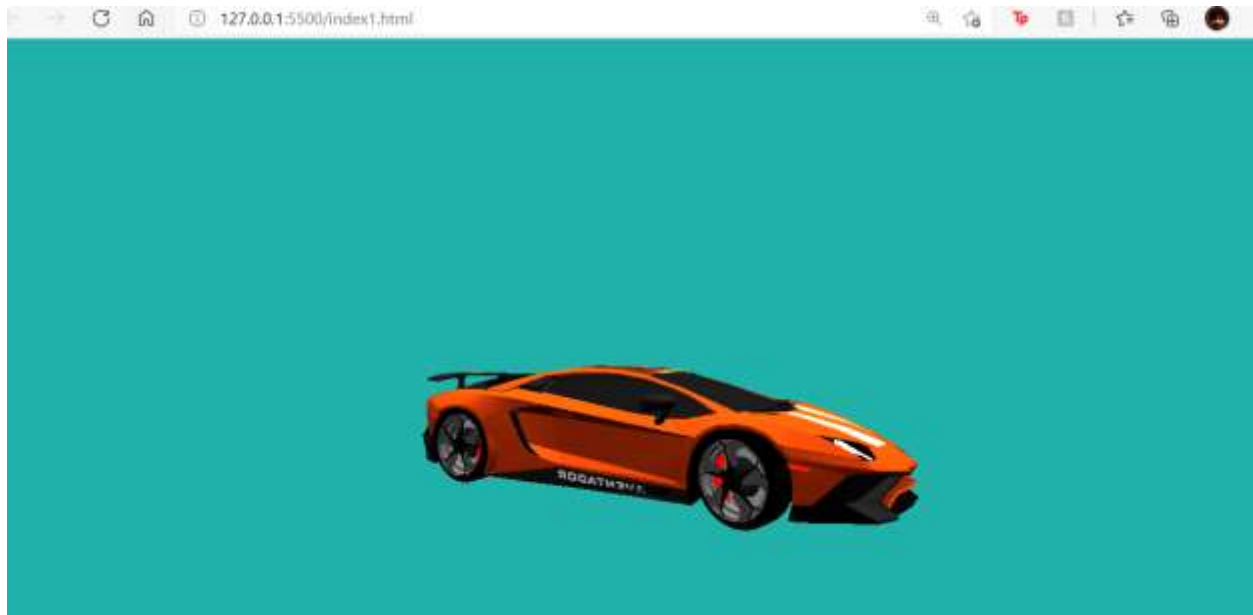
```
53     function animate() {
54         requestAnimationFrame(animate);
55         obj.rotation.y += 0.01;
56         renderer.render(scene, camera);
57     }
58     animate();
```

Run a Local Server

With VSCode, an extension called “Live Server” can be added in order to run a local live server for the application.



If a camera, renderer and scene with sufficient lighting was created, opening the .html file with Live Server will result in being able to view the model in a web browser similar to the image below. With OrbitControls successfully incorporated, the camera can be panned, rotated and zoomed all with the use of a mouse.



Resources Used:

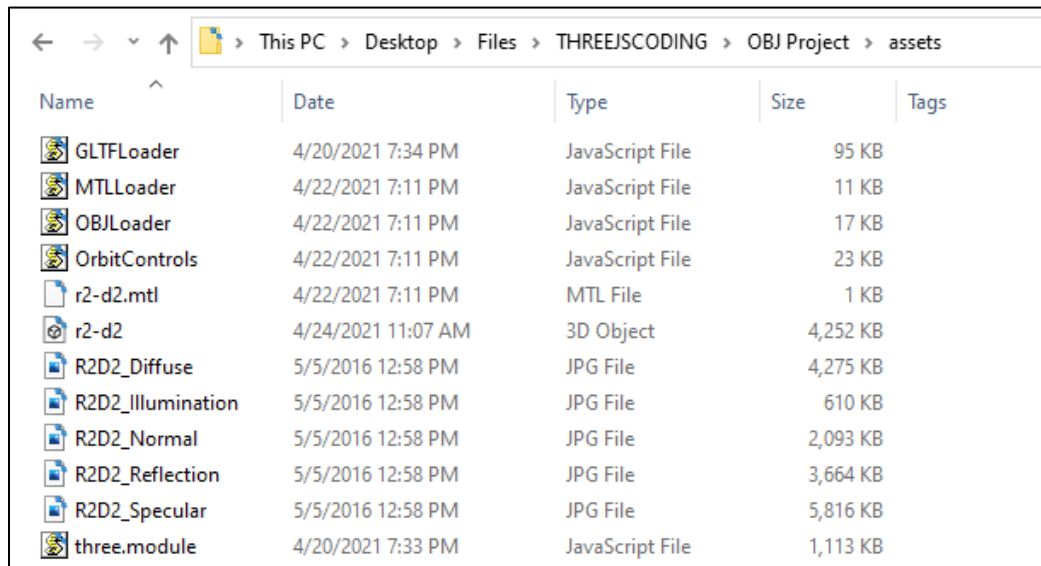
[Display your own 3D Model with Javascript | three.js - YouTube](#)

[\(FREE \) Lamborghini Aventador SV tunig by SDC - Download Free 3D model by SDC \(@3Duae\) \[4350c8d\] \(sketchfab.com\)](#)

.OBJ MODEL

Obtain a 3D Model in a .obj file format. The model in this guide consists of:

- A 3D Object (R2-D2)
- Textures for the model,
- And a .mtl file.



Name	Date	Type	Size	Tags
GLTFLoader	4/20/2021 7:34 PM	JavaScript File	95 KB	
MTLLoader	4/22/2021 7:11 PM	JavaScript File	11 KB	
OBJLoader	4/22/2021 7:11 PM	JavaScript File	17 KB	
OrbitControls	4/22/2021 7:11 PM	JavaScript File	23 KB	
r2-d2.mtl	4/22/2021 7:11 PM	MTL File	1 KB	
r2-d2	4/24/2021 11:07 AM	3D Object	4,252 KB	
R2D2_Diffuse	5/5/2016 12:58 PM	JPG File	4,275 KB	
R2D2_Illumination	5/5/2016 12:58 PM	JPG File	610 KB	
R2D2_Normal	5/5/2016 12:58 PM	JPG File	2,093 KB	
R2D2_Reflection	5/5/2016 12:58 PM	JPG File	3,664 KB	
R2D2_Specular	5/5/2016 12:58 PM	JPG File	5,816 KB	
three.module	4/20/2021 7:33 PM	JavaScript File	1,113 KB	

Import the Appropriate Files

In order to use the necessary files, they must be called upon and imported into the application. In this example, the script for the application is not written in the same index.html file, but in it's own "script.js" file, so that file is also imported.



Create a Scene

To display the 3D model using three.js, three things are needed first: **a scene, a camera, and a renderer.**



Scene

In order to see the model, lighting is created as variables and then added to the scene. A background is also set in order to provide color into the picture.

```
15 var keyLight = new THREE.DirectionalLight (new THREE.Color ( 'hsl(30, 100%, 75%)'), 1.0);
16 keyLight.position.set(-100,0,100);
17
18 var fillLight = new THREE.DirectionalLight (new THREE.Color ( 'hsl(240, 100%, 75%)'), 0.75);
19 fillLight.position.set(100,0,-100);
20
21 var backLight = new THREE.DirectionalLight (0xffffff, 1.0);
22 backLight.position.set(100,0,-100).normalize();
23
24 scene.add(keyLight);
25 scene.add(fillLight);
26 scene.add(backLight);
27 scene.background = new THREE.Color(0x000000);
```

In order to see the model, lighting is added as variables and added to the scene. The background can also added to bring color to the application.

Camera

In order to be able to use a mouse to rotate the camera view, three.js' `OrbitalControl` application must be imported. Then, a variable for controls of Three.js; `OrbitalControls` camera can be created.

```
12 <script src="assets\OrbitControls.js"></script>
```

```
2 var camera = new THREE.PerspectiveCamera( 75, window.innerWidth/window.innerHeight, 0.1, 1000 );
```

```
10 var controls = new THREE.OrbitControls (camera, renderer.domElement);  
11 controls.enableDamping = true;  
12 controls.dampingFactor = 0.25;  
13 controls.enableZoom = true;
```

Renderer

With these arguments, the application will always render to the size of the web browser window.

```
4 var renderer = new THREE.WebGLRenderer();  
5 renderer.setSize( window.innerWidth, window.innerHeight );  
6 document.body.appendChild( renderer.domElement );
```

Loading The Model

The 3D model we have is a .OBJ file and has texture files that can be applied to the model using three.js' MTLLoader. The model itself can be loaded in using three.js' OBJLoader.

```
11 <script src="assets\OBJLoader.js"></script>
12 <script src="assets\OrbitControls.js"></script>
13 <script src="assets\MTLLoader.js"></script>
14 <script src="scripts.js"></script>
```

OBJ Loader MTLLoader

In our script, the materials MUST be loaded in before the object itself.

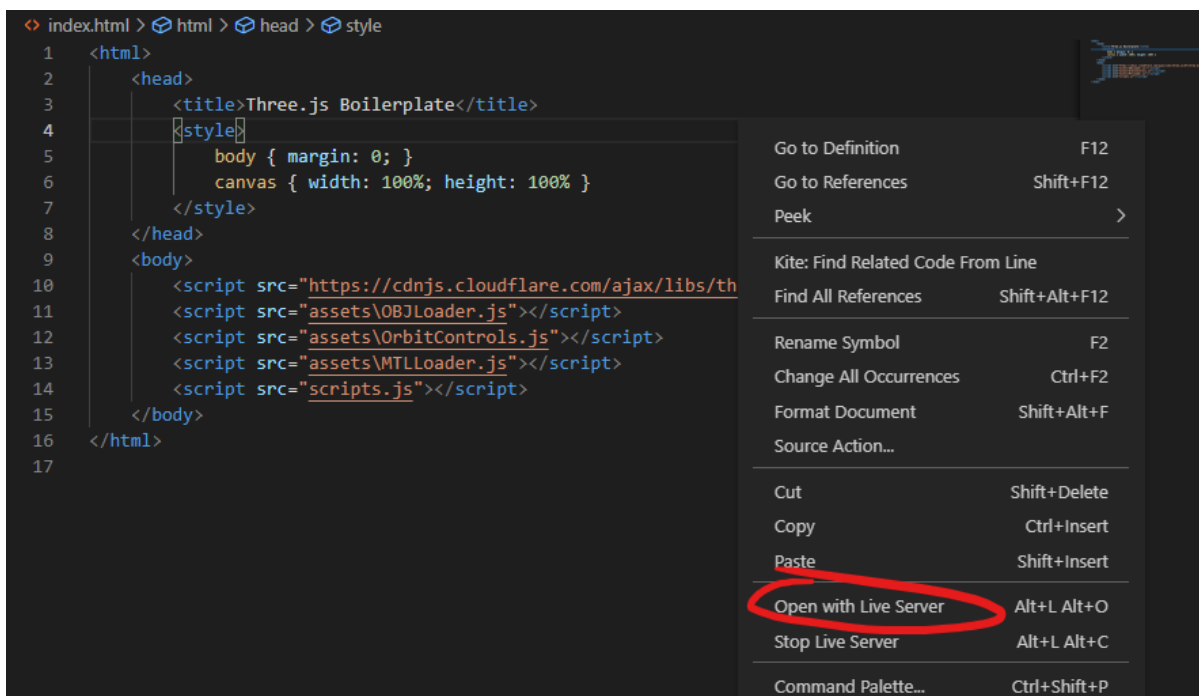
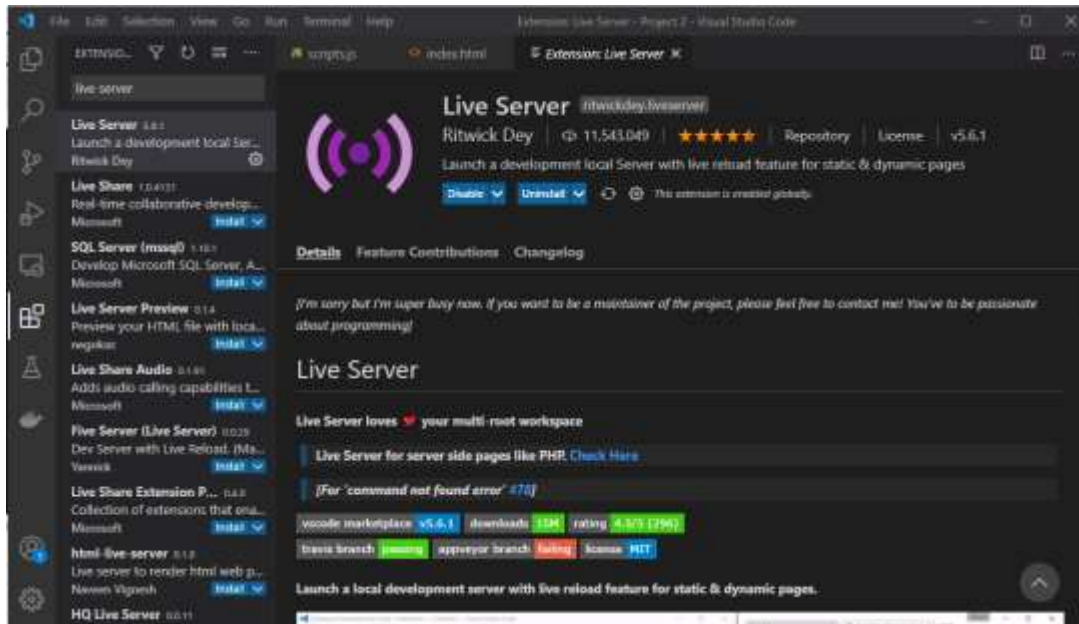
```
29
30 var mtlLoader = new THREE.MTLLoader();
31 mtlLoader.setTexturePath('assets/');
32 mtlLoader.setPath('assets/');
33 mtlLoader.load('r2-d2.mtl',function(materials) {
34     materials.preload();
35
36     var objLoader = new THREE.OBJLoader();
37     objLoader.setMaterials(materials);
38     objLoader.setPath('assets/');
39     objLoader.load('r2-d2.obj',function(object){
40         object.position.y -=60;
41         scene.add(object);
42     });
43
44 });
45
```

MTLLoader

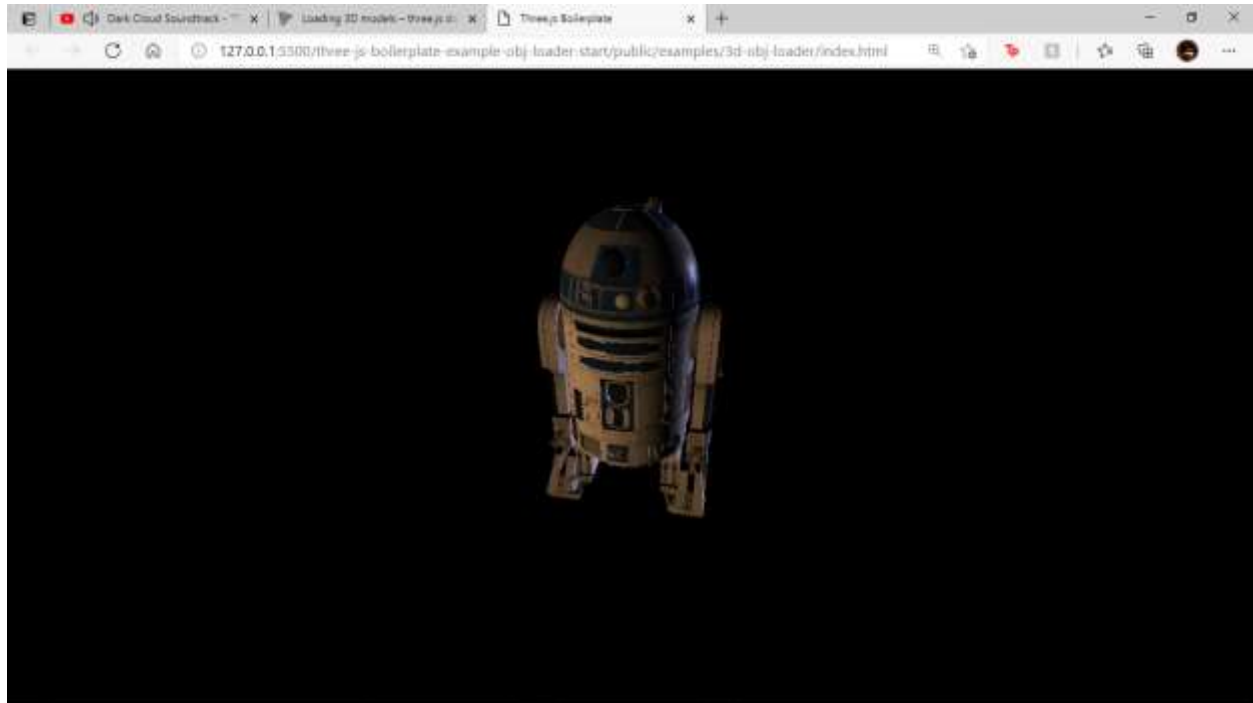
OBJ Loader

Run a Local Server

With VSCode, an extension called “Live Server” can be added in order to run a local live server for the application.



If a camera, renderer and scene with sufficient lighting was created, opening the .html file with Live Server will result in being able to view the model in a web browser similar to the image below. With OrbitControls successfully incorporated, the camera can be panned, rotated and zoomed all with the use of a mouse.



Resources Used:

[Load a 3D Model Using the Three.js OBJ Loader - YouTube](#)

[learnthreejs/three.js-boilerplate at example-obj-loader-finish \(github.com\)](#)