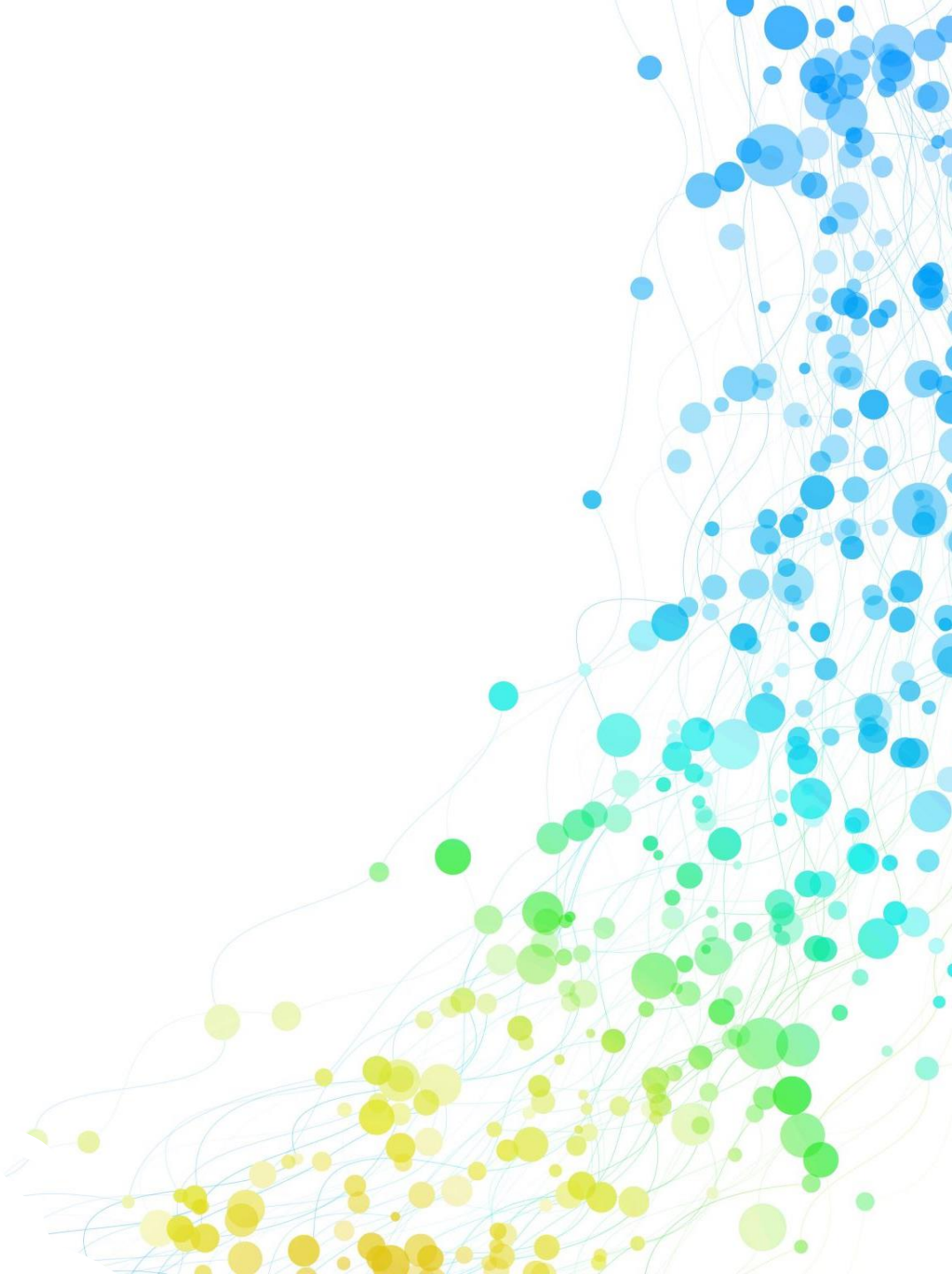
Three.js를 이용한 CG 웹 프로그래밍

2024. 10. 29



three.js r169

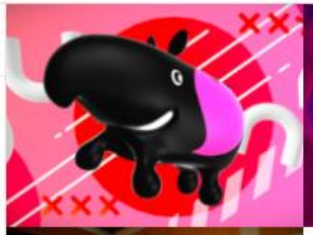
Learn
documentation
examples
editor
nodes
gpt

Community
questions
discord
forum
twitter

Code
github
download

Resources
Three.js Fundamentals
Three.js Journey
Learn Three.js
初めてのThree.js

Merch
T-Shirts



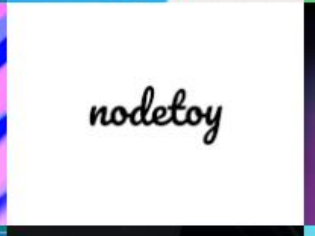




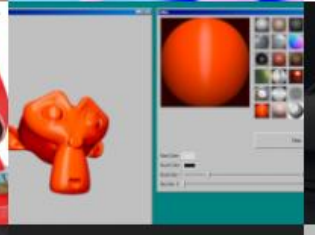
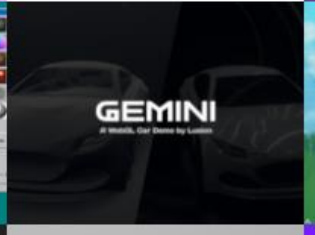













OpenGL

- 산업계에서 가장 널리 사용되고 있는 API 표준 (실질적인 표준)
- 1992년 플랫폼 독립적으로 개발됨
  - Silicon Graphics사 GL(Graphics Library)로부터 발전
- 1999년 Java3D
  - OpenGL 또는 Direct3D 위에서 실행되는 상위 수준의 그래픽스 API를 제공
  - 인터넷 환경의 가상현실 모델링 언어 VRML과 같이 장면 그래프 개념 사용
- OpenGL ES – 모바일 환경에서 3차원 컴퓨터 그래픽스 표준
  - 2003년 OpenGL ES 1.0 → 2012년 OpenGL ES 3.0
- Vulkan API – OpenGL의 차기 버전

WebGL

- 웹 환경에서 플러그인을 사용하지 않고 웹 브라우저에서 3차원 및 2차원 그래픽을 지원하는 JavaScript API를 제공
- WebGL 2.0
  - OpenGL ES 3.0에 기반
  - WWW 표준 HTML5의 Canvas 사용(2차원 그래픽스를 지원하는 웹 스크립트 언어)
- 웹 브라우저 – Chrome, Edge, Firefox, Safari 등 WebGL을 지원함
  - 추가 설치 프로그램 X (소스코드 편집기만 필요)
- HTML과 JavaScript에 대한 이해가 필요
  - GLSL – C/C++ 프로그래밍에 대한 기초적인 지식 필요
  - 웹페이지에 Canvas 생성 → WebGL 초기화 → GLSL 컴파일 → 렌더링

WebGL™

LOW-LEVEL 3D GRAPHICS API BASED ON OPENGL ES

WebGL™ is a cross-platform, royalty-free open web standard for a low-level 3D graphics API based on OpenGL ES, exposed to ECMAScript via the HTML5 Canvas element. Developers familiar with OpenGL ES 2.0 will recognize WebGL as a Shader-based API using GLSL, with constructs that are semantically similar to those of the underlying OpenGL ES API. It stays very close to the OpenGL ES specification, with some concessions made for what developers expect out of memory-managed languages such as JavaScript. WebGL 1.0 exposes the OpenGL ES 2.0 feature set; WebGL 2.0 exposes the OpenGL ES 3.0 API.

WebGL brings plugin-free 3D to the web, implemented right into the browser. Major browser vendors Apple (Safari), Google (Chrome), Microsoft (Edge), and Mozilla (Firefox) are members of the WebGL Working Group.

Essential Resources for WebGL Development

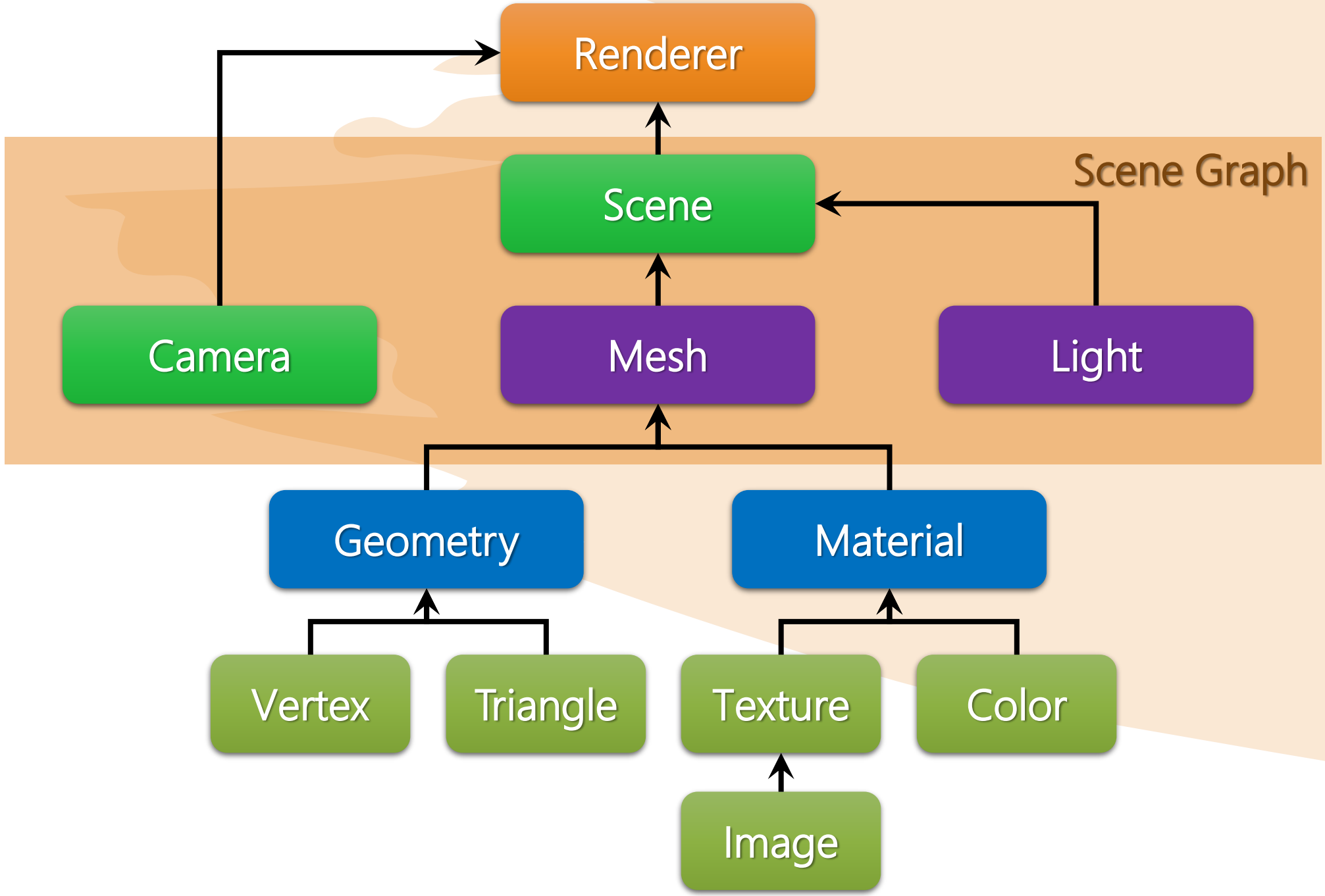
There is a full set of well-supported developer information and educational resources to help quickly get you up and running with your WebGL application development.

The Standard

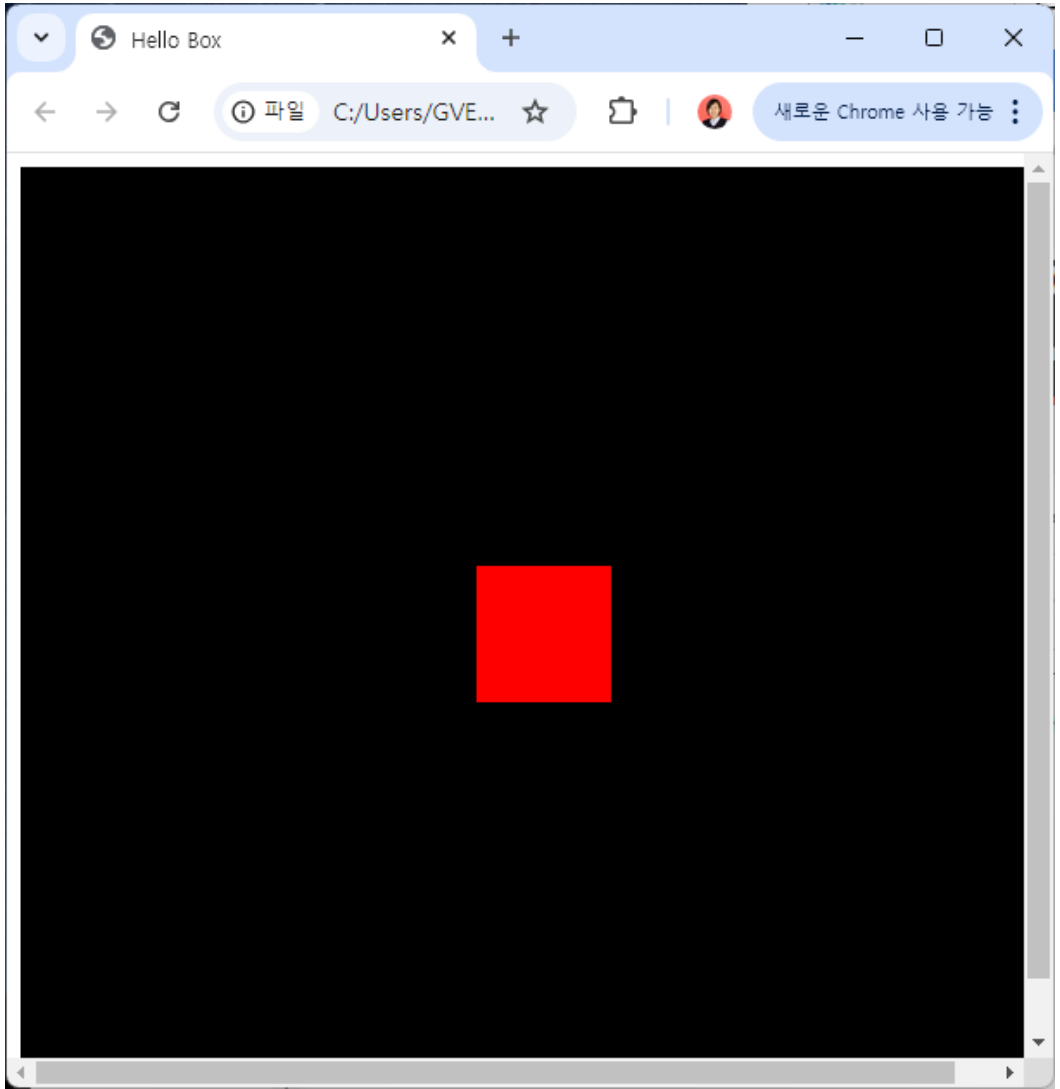
- WebGL 1.0 Specification
- WebGL 2.0 Specification
- WebGL Reference Card
- Filing bugs about the WebGL spec or conformance tests

Three.js를 이용한 WebGL 프로그래밍

- WebGL은 GLSL을 통해 대부분의 그래픽스 기능을 구현
  - 그래픽스 파이프라인의 구조에 대한 이해 필요 / 배우는데 많은 시간 소요
- Three.js
  - WebGL 보다는 상위 계층의 유틸리티 라이브러리
  - GLSL 언어를 이용하여 프로그래밍 해야 하는 물체의 모양, 변환, 셰이딩, 재질, 조명, 카메라 등을 추상화된 객체로 쉽게 생성하여 사용할 수 있게 되어 있음


Renderer
Scene Graph
Scene
Camera
Mesh
Light
Geometry
Material
Vertex
Triangle
Texture
Color
Image

Three.js로 Box 그리기



File Edit Selection View Go Run ··· Search

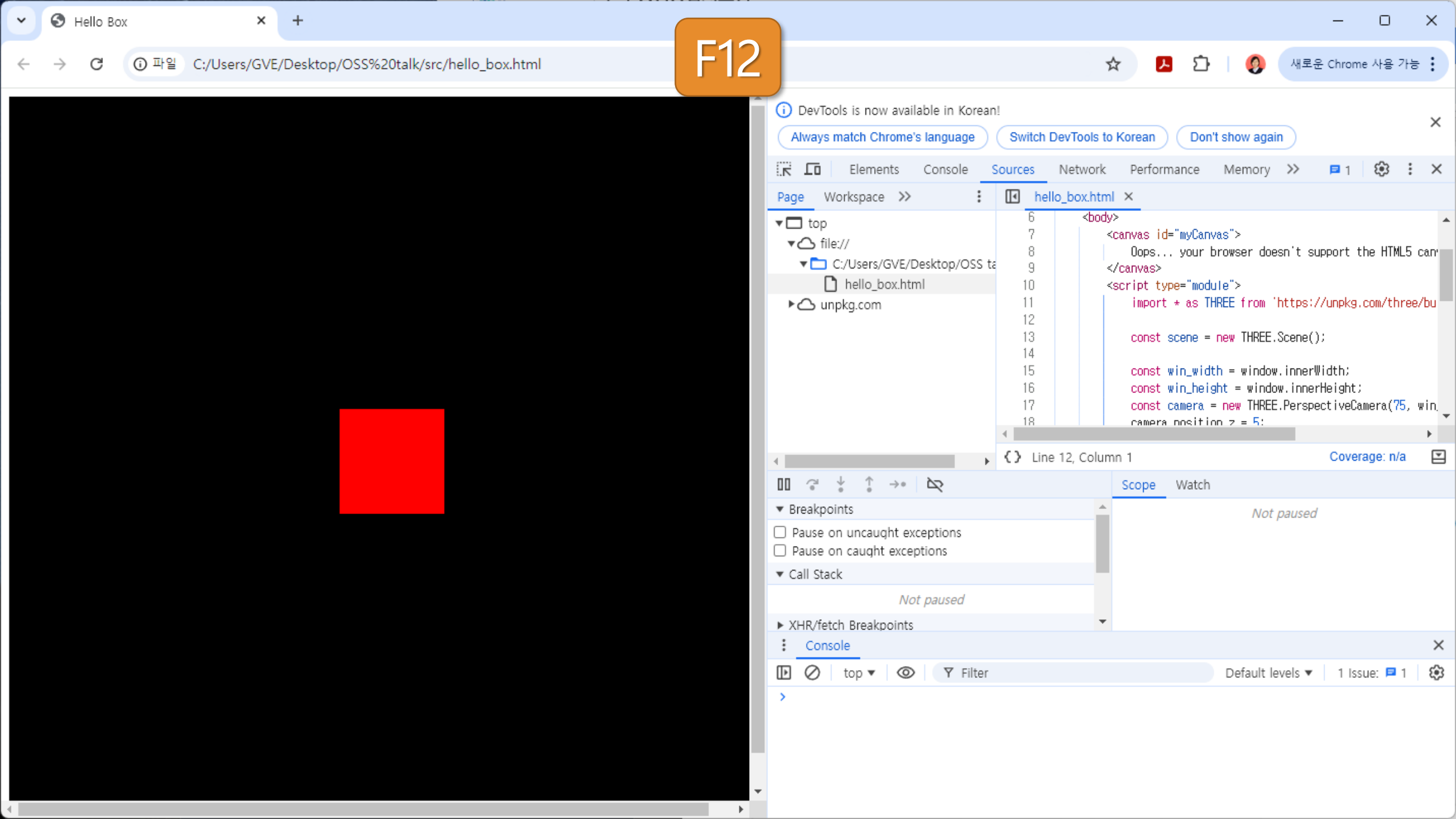
hello_box.html

C: > Users > GVE > Desktop > OSS talk > src > hello_box.html > {} "hello_box.html"

https://github.com/ProfSunKim/threejs

```html
<html>
    <head>
        <meta charset="utf-8">
        <title>Hello Box</title>
    </head>
    <body>
        <canvas id="myCanvas">
            Oops... your browser doesn't support the HTML5 canvas element!
        </canvas>
        <script type="module">
            import * as THREE from 'https://unpkg.com/three/build/three.module.js';

            const scene = new THREE.Scene();

            const win_width = window.innerWidth;
            const win_height = window.innerHeight;
            const camera = new THREE.PerspectiveCamera(75, win_width/win_height, 0.001, 1000);
            camera.position.z = 5;

            const renderer = new THREE.WebGLRenderer({canvas:myCanvas});
            renderer.setSize(win_width, win_height);

            const geometry = new THREE.BoxGeometry();
            const material = new THREE.MeshBasicMaterial({color: 0xff0000});
            const cube = new THREE.Mesh(geometry, material);
            scene.add(cube);

            renderer.render(scene, camera);
        </script>
    </body>
</html>

```

Restricted Mode ⊗ 0 ⚠ 0 0 Ln 32, Col 1 Spaces: 4 UTF-8 CRLF HTML

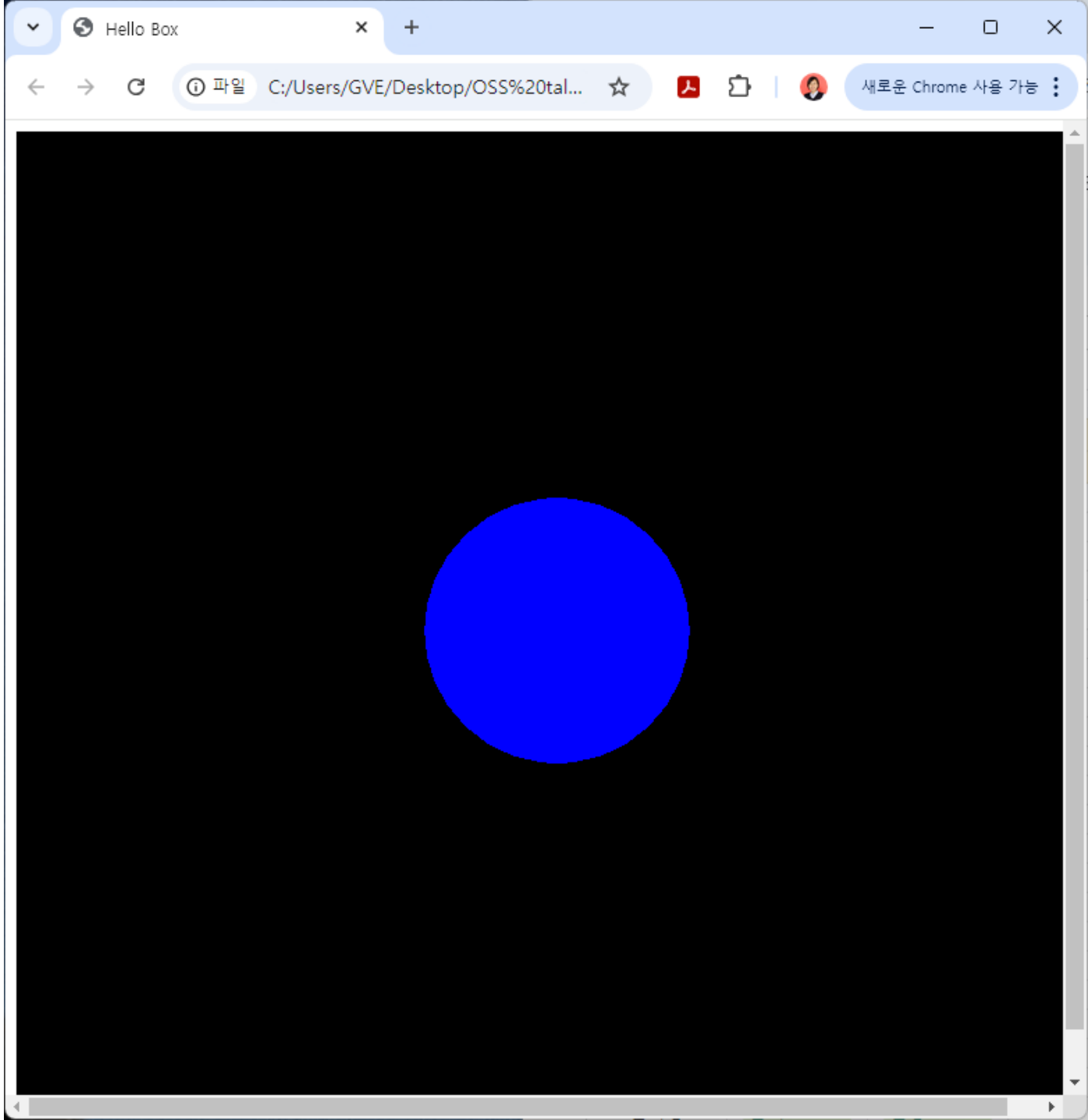

Hello Box
F12
C:/Users/GVE/Desktop/OSS%20talk/src/hello_box.html
DevTools is now available in Korean!
Always match Chrome's language
Switch DevTools to Korean
Don't show again
Elements
Console
Sources
Network
Performance
Memory
Page
Workspace
hello_box.html
top
file://
C:/Users/GVE/Desktop/OSS ta
hello_box.html
unpkg.com
<body>
<canvas id="myCanvas">
Oops... your browser doesn't support the HTML5 can
</canvas>
<script type="module">
import * as THREE from 'https://unpkg.com/three/bu
const scene = new THREE.Scene();
const win_width = window.innerWidth;
const win_height = window.innerHeight;
const camera = new THREE.PerspectiveCamera(75, win
camera.position.z = 5;
Line 12, Column 1
Coverage: n/a
Scope
Watch
Not paused
Breakpoints
Pause on uncaught exceptions
Pause on caught exceptions
Call Stack
Not paused
XHR/fetch Breakpoints
Console
top
Filter
Default levels
1 Issue: 1

Three.js 주요 기능

- 기하 형상 모델링 (Geometric Shape Modeling)
  - Plane, Box, Circle, Sphere, Cone, Cylinder, Text 등
- 기하 변환과 투영 (Geometric Transformation and Projection)
  - Viewing, Projection, Viewport 등
- 셰이딩 (Shading)
  - Material, Lighting 등
- 텍스처 매핑 (Texture Mapping), 블렌딩 (Blending), 애니메이션 (Animation) 등

연습 문제 (1)

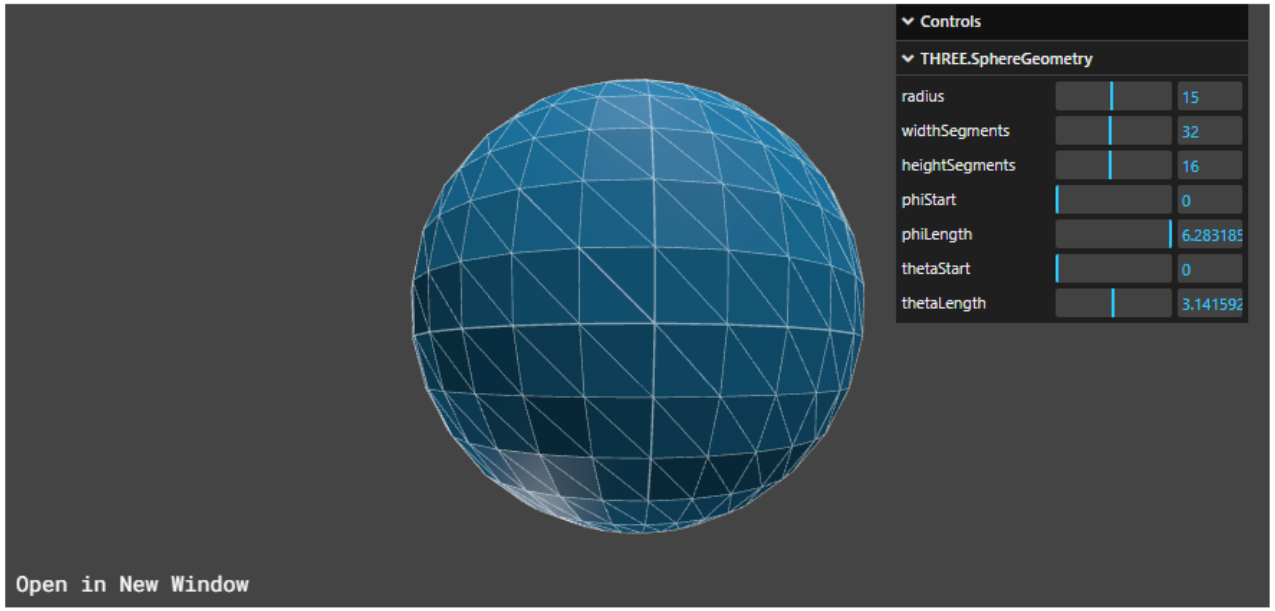
- Box 대신 Sphere를 파란색으로 그려보세요.



BufferGeometry →

SphereGeometry

A class for generating sphere geometries.



Code Example

```
const geometry = new THREE.SphereGeometry( 15, 32, 16 );
const material = new THREE.MeshBasicMaterial( { color: 0xffff00 } );
const sphere = new THREE.Mesh( geometry, material ); scene.add( sphere );
```

Constructor

SphereGeometry(radius : Float, widthSegments : Integer, heightSegments :

File Edit Selection View Go Run ··· Search

hello_box.html

C: > Users > GVE > Desktop > OSS talk > src > hello_box.html > {} "hello_box.html"

```
<html>
    <head>
        <meta charset="utf-8">
        <title>Hello Box</title>
    </head>
    <body>
        <canvas id="myCanvas">
            Oops... your browser doesn't support the HTML5 canvas element!
        </canvas>
        <script type="module">
            import * as THREE from 'https://unpkg.com/three/build/three.module.js';

            const scene = new THREE.Scene();

            const win_width = window.innerWidth;
            const win_height = window.innerHeight;
            const camera = new THREE.PerspectiveCamera(75, win_width/win_height, 0.001, 1000);
            camera.position.z = 5;

            const renderer = new THREE.WebGLRenderer({canvas:myCanvas});
            renderer.setSize(win_width, win_height);

            const geometry = new THREE.SphereGeometry(1, 32, 16);
            const material = new THREE.MeshBasicMaterial({color: 0x0000ff});
            const sphere = new THREE.Mesh(geometry, material);
            scene.add(sphere);

            renderer.render(scene, camera);
        </script>
    </body>
</html>
```

Restricted Mode ⊗ 0 ⚠ 0 0 Ln 32, Col 1 Spaces: 4 UTF-8 CRLF HTML

File Edit Selection View Go Run ··· Search

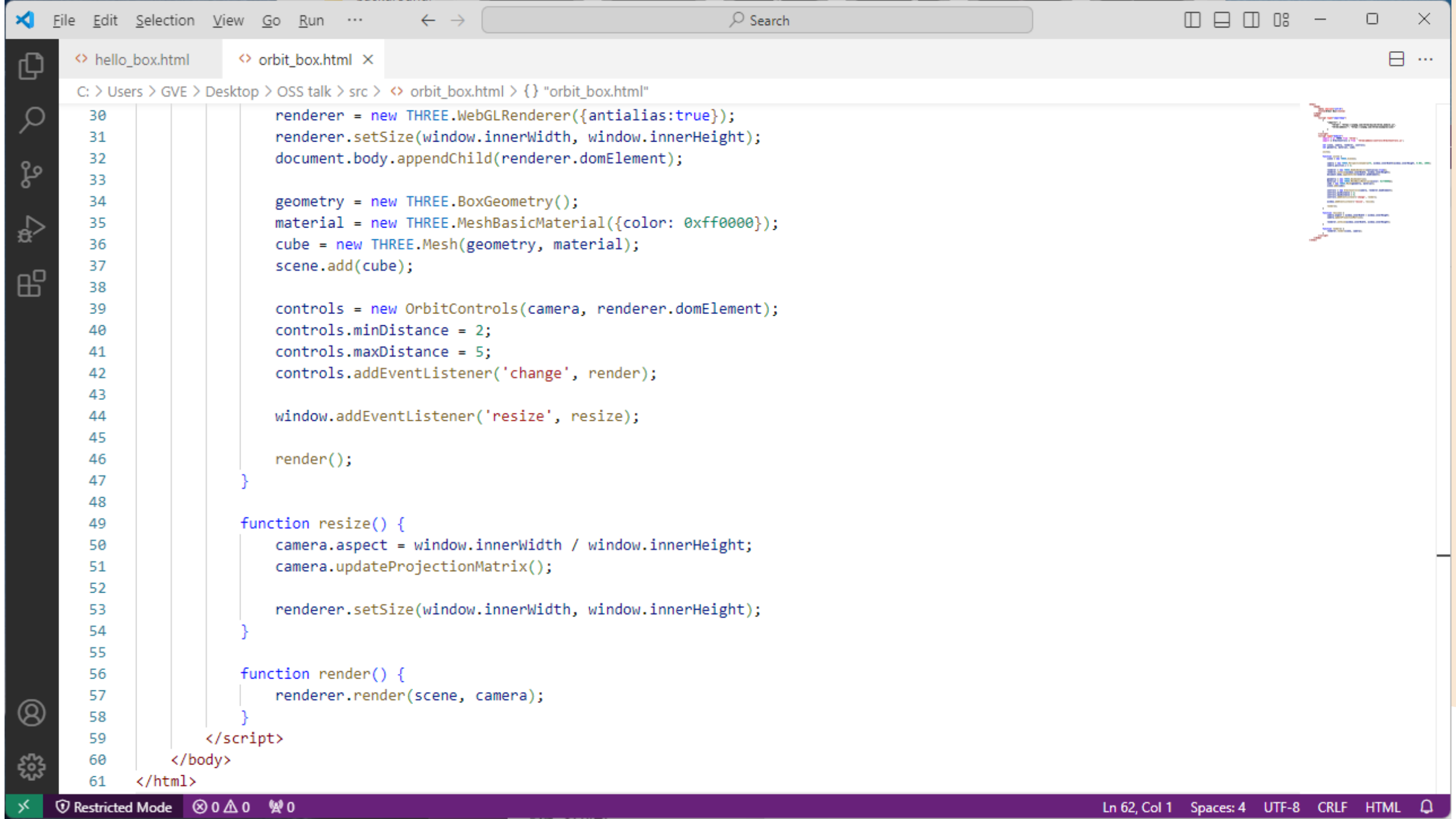
hello_box.html | orbit_box.html

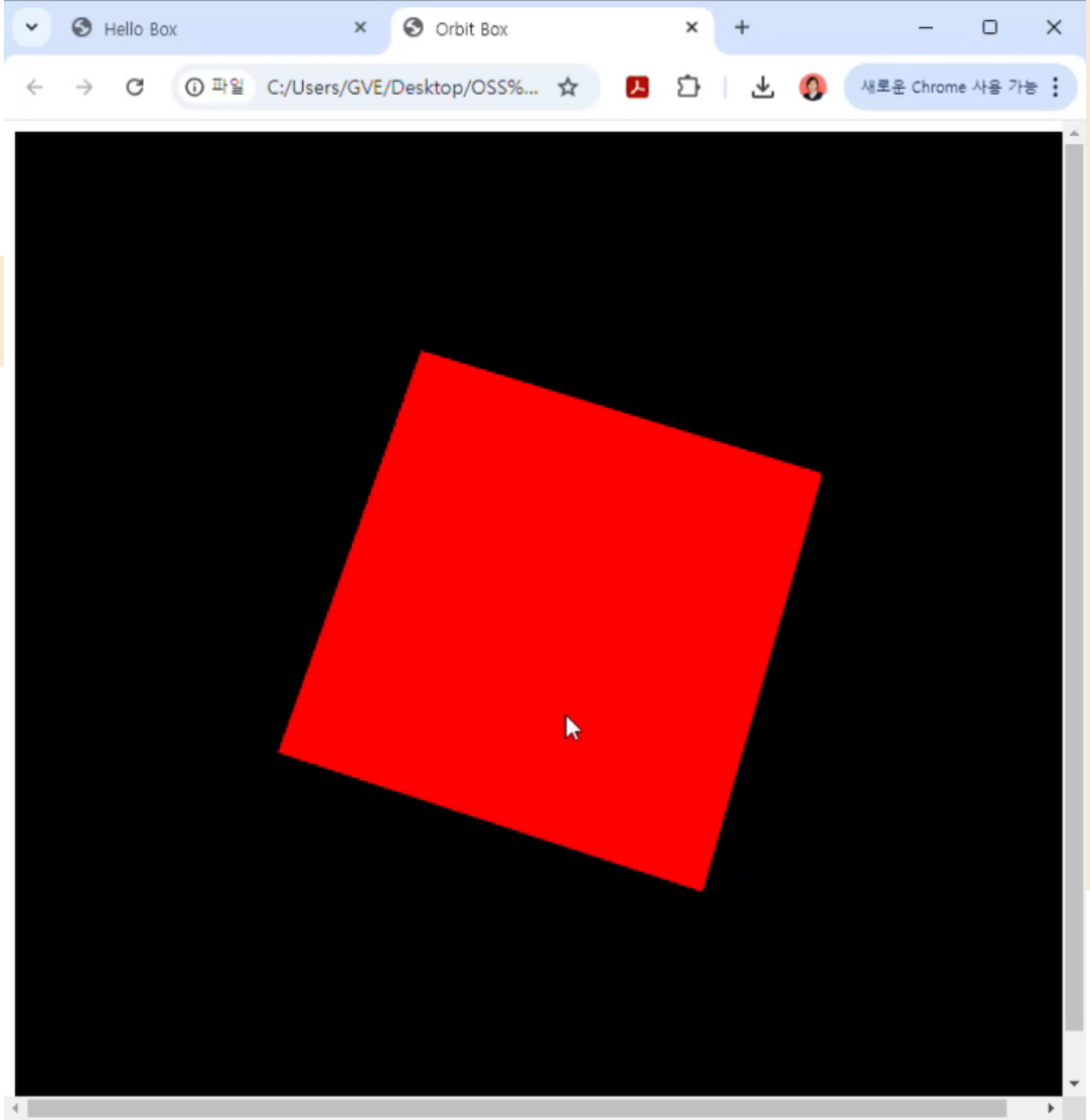
C: > Users > GVE > Desktop > OSS talk > src > orbit_box.html > {} "orbit_box.html"

https://github.com/ProfSunKim/threejs

```
<html>
    <head>
        <meta charset="utf-8">
        <title>Orbit Box</title>
    </head>
    <body>
        <script type="importmap">
            {
                "imports": {
                    "three": "https://unpkg.com/three/build/three.module.js",
                    "three/addons/": "https://unpkg.com/three/examples/jsm/"
                }
            }
        </script>
        <script type="module">
            import * as THREE from 'three';
            import { OrbitControls } from  'three/addons/controls/OrbitControls.js';

            let scene, camera, renderer, controls;
            let geometry, material, cube;

            init();

            function init() {
                scene = new THREE.Scene();

                camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight, 0.001, 1000);
                camera.position.z = 2;

                renderer = new THREE.WebGLRenderer({antialias:true});
                renderer.setSize(window.innerWidth, window.innerHeight);
                document.body.appendChild(renderer.domElement);
```

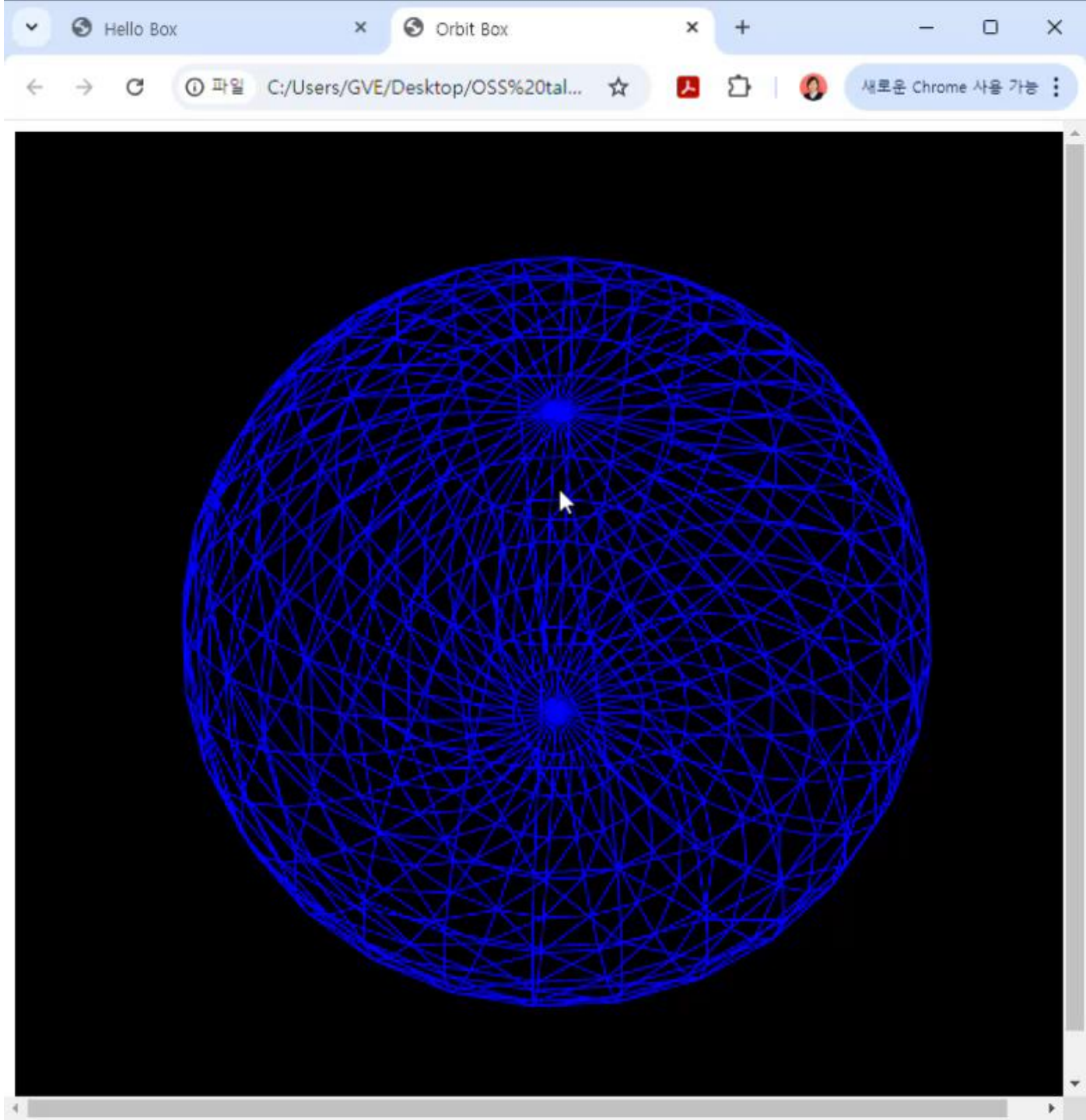
Restricted Mode ⊗ 0 ⚠ 0 0 Ln 62, Col 1 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run ···
Search
hello_box.html
orbit_box.html
C: > Users > GVE > Desktop > OSS talk > src > orbit_box.html > {} "orbit_box.html"
renderer = new THREE.WebGLRenderer({antialias:true});
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);
geometry = new THREE.BoxGeometry();
material = new THREE.MeshBasicMaterial({color: 0xff0000});
cube = new THREE.Mesh(geometry, material);
scene.add(cube);
controls = new OrbitControls(camera, renderer.domElement);
controls.minDistance = 2;
controls.maxDistance = 5;
controls.addEventListener('change', render);
window.addEventListener('resize', resize);
render();
}
function resize() {
camera.aspect = window.innerWidth / window.innerHeight;
camera.updateProjectionMatrix();
renderer.setSize(window.innerWidth, window.innerHeight);
}
function render() {
renderer.render(scene, camera);
}
</script>
</body>
</html>
Restricted Mode
Ln 62, Col 1
Spaces: 4
UTF-8
CRLF
HTML


Hello Box
Orbit Box
파일
C:/Users/GVE/Desktop/OSS%...
새로운 Chrome 사용 가능

연습 문제 (2)

- Sphere를 파란색 wireframe으로 그린 후 Orbit Control으로 조작해 보시오.

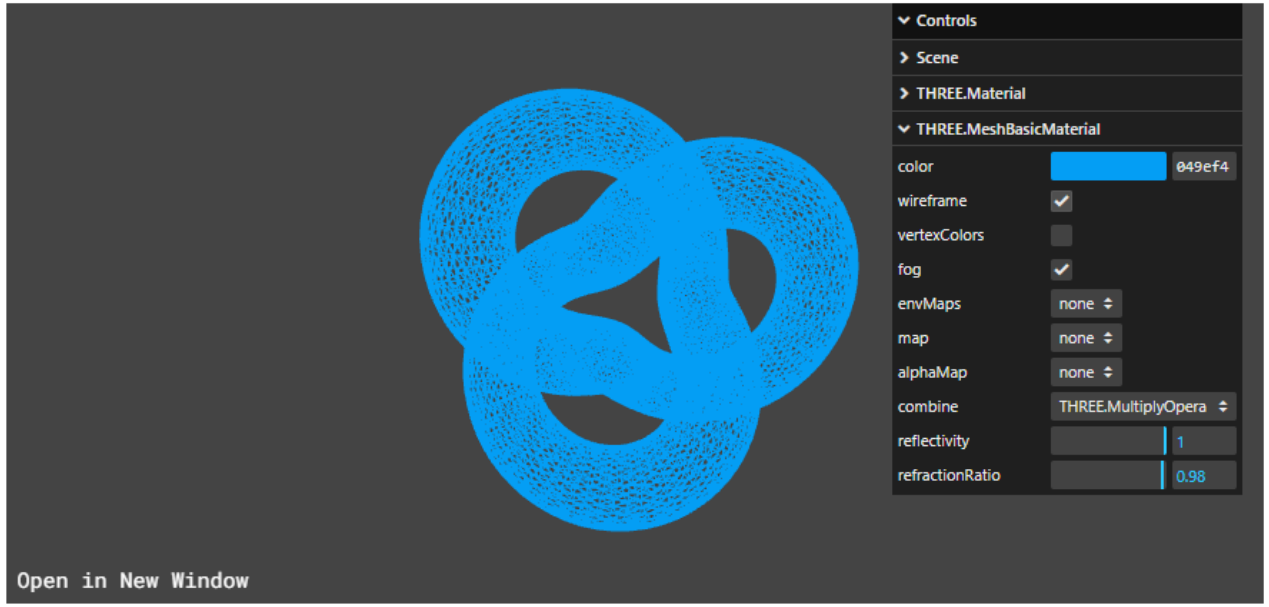


Material →

MeshBasicMaterial

A material for drawing geometries in a simple shaded (flat or wireframe) way.

This material is not affected by lights.

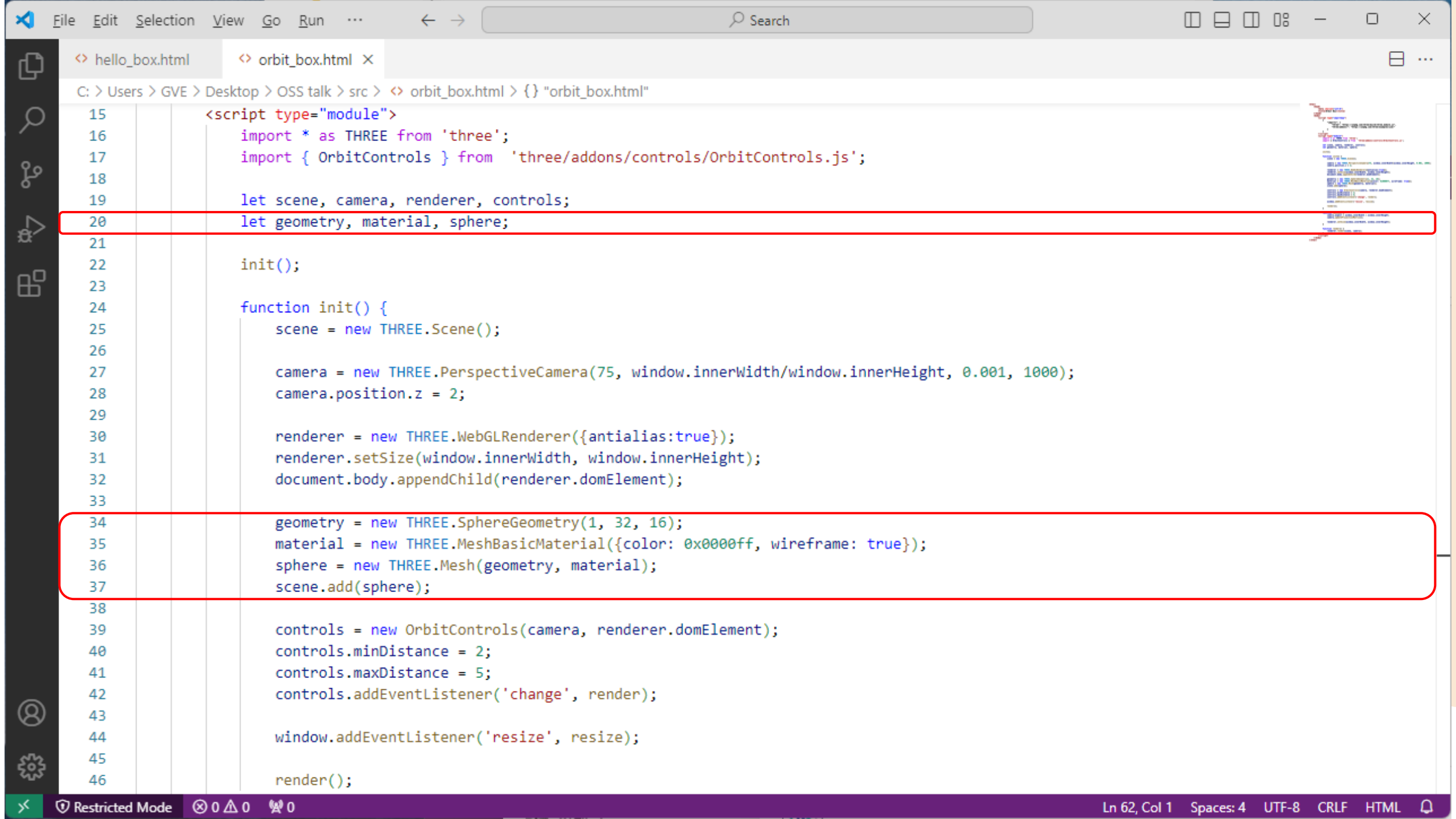


Constructor

MeshBasicMaterial(parameters : Object)

parameters - (optional) an object with one or more properties defining the material's appearance. Any property of the material (including any property inherited from Material) can be passed in here.

The exception is the property color, which can be passed in as a hexadecimal string and is `0xffffff` (white) by default. Color.set(color) is called internally.


File Edit Selection View Go Run
Search
hello_box.html
orbit_box.html
C: > Users > GVE > Desktop > OSS talk > src > orbit_box.html > {} "orbit_box.html"
<script type="module">
import * as THREE from 'three';
import { OrbitControls } from 'three/addons/controls/OrbitControls.js';
let scene, camera, renderer, controls;
let geometry, material, sphere;
init();
function init() {
scene = new THREE.Scene();
camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight, 0.001, 1000);
camera.position.z = 2;
renderer = new THREE.WebGLRenderer({antialias:true});
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);
geometry = new THREE.SphereGeometry(1, 32, 16);
material = new THREE.MeshBasicMaterial({color: 0x0000ff, wireframe: true});
sphere = new THREE.Mesh(geometry, material);
scene.add(sphere);
controls = new OrbitControls(camera, renderer.domElement);
controls.minDistance = 2;
controls.maxDistance = 5;
controls.addEventListener('change', render);
window.addEventListener('resize', resize);
render();
Restricted Mode
Ln 62, Col 1
Spaces: 4
UTF-8
CRLF
HTML

File Edit Selection View Go Run ··· Search

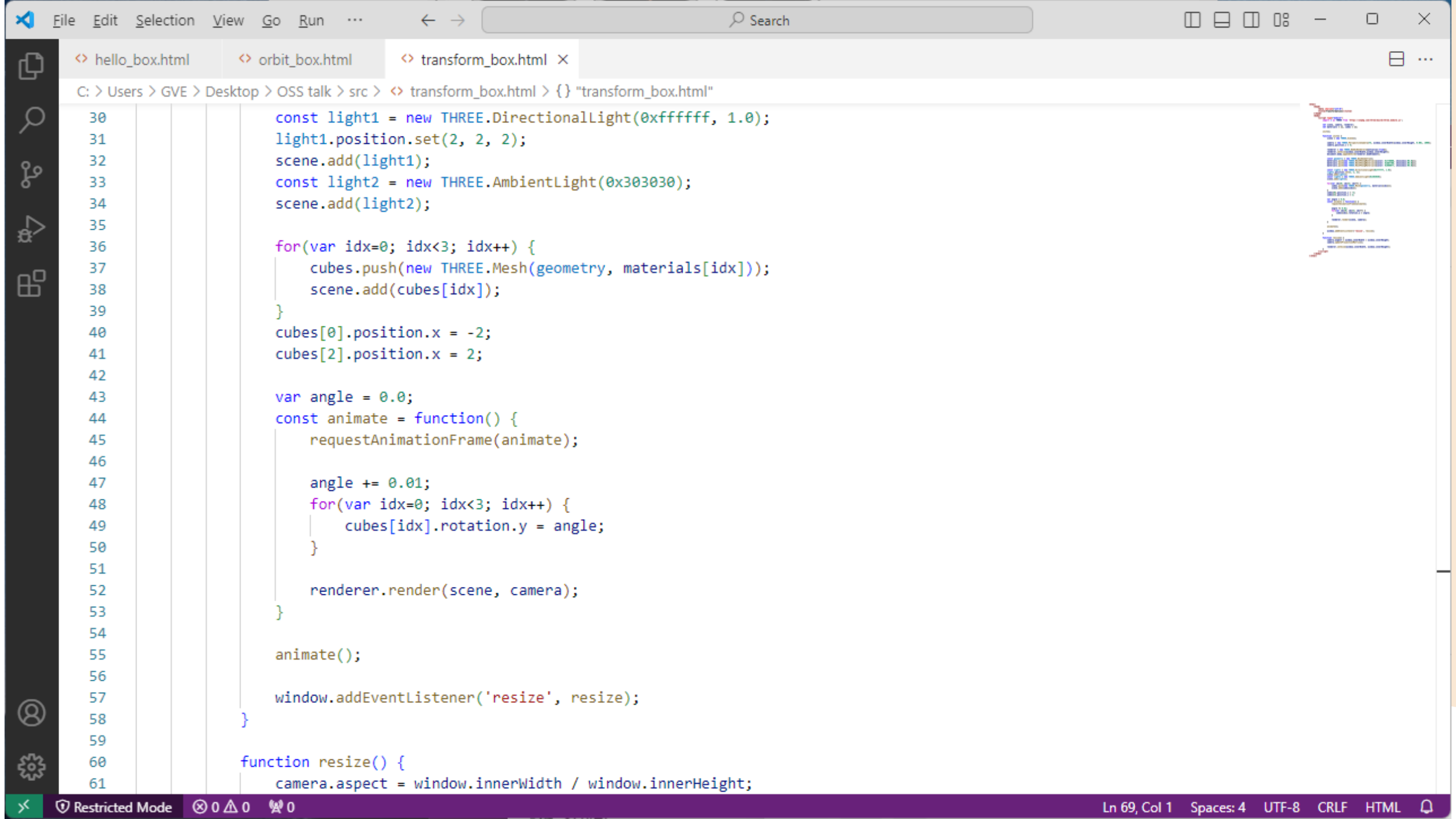
hello_box.html orbit_box.html transform_box.html

C: > Users > GVE > Desktop > OSS talk > src > transform_box.html > {} "transform_box.html"

https://github.com/ProfSunKim/threejs

```
<html>
    <head>
        <meta charset="utf-8">
        <title>Transformations</title>
    </head>
    <body>
        <script type="module">
            import * as THREE from 'https://unpkg.com/three/build/three.module.js';

            let scene, camera, renderer;
            let materials = [], cubes = [];

            init();

            function init() {
                scene = new THREE.Scene();

                camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight, 0.001, 1000);
                camera.position.z = 5;

                renderer = new THREE.WebGLRenderer({antialias:true});
                renderer.setSize(window.innerWidth,window.innerHeight);
                document.body.appendChild(renderer.domElement);

                const geometry = new THREE.BoxGeometry();
                materials.push(new THREE.MeshPhongMaterial({color: 0xff0000, shininess:90.0}));
                materials.push(new THREE.MeshPhongMaterial({color: 0x00ff00, shininess:90.0}));
                materials.push(new THREE.MeshPhongMaterial({color: 0x0000ff, shininess:90.0}));

                const light1 = new THREE.DirectionalLight(0xffffff, 1.0);
                light1.position.set(2, 2, 2);
                scene.add(light1);
```

Restricted Mode ⊗ 0 ⚠ 0 0 Ln 69, Col 1 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run ···
Search
hello_box.html
orbit_box.html
transform_box.html
C: > Users > GVE > Desktop > OSS talk > src > transform_box.html > {} "transform_box.html"
const light1 = new THREE.DirectionalLight(0xffffff, 1.0);
light1.position.set(2, 2, 2);
scene.add(light1);
const light2 = new THREE.AmbientLight(0x303030);
scene.add(light2);
for(var idx=0; idx<3; idx++) {
cubes.push(new THREE.Mesh(geometry, materials[idx]));
scene.add(cubes[idx]);
}
cubes[0].position.x = -2;
cubes[2].position.x = 2;
var angle = 0.0;
const animate = function() {
requestAnimationFrame(animate);
angle += 0.01;
for(var idx=0; idx<3; idx++) {
cubes[idx].rotation.y = angle;
}
renderer.render(scene, camera);
}
animate();
window.addEventListener('resize', resize);
}
function resize() {
camera.aspect = window.innerWidth / window.innerHeight;
Restricted Mode
Ln 69, Col 1
Spaces: 4
UTF-8
CRLF
HTML

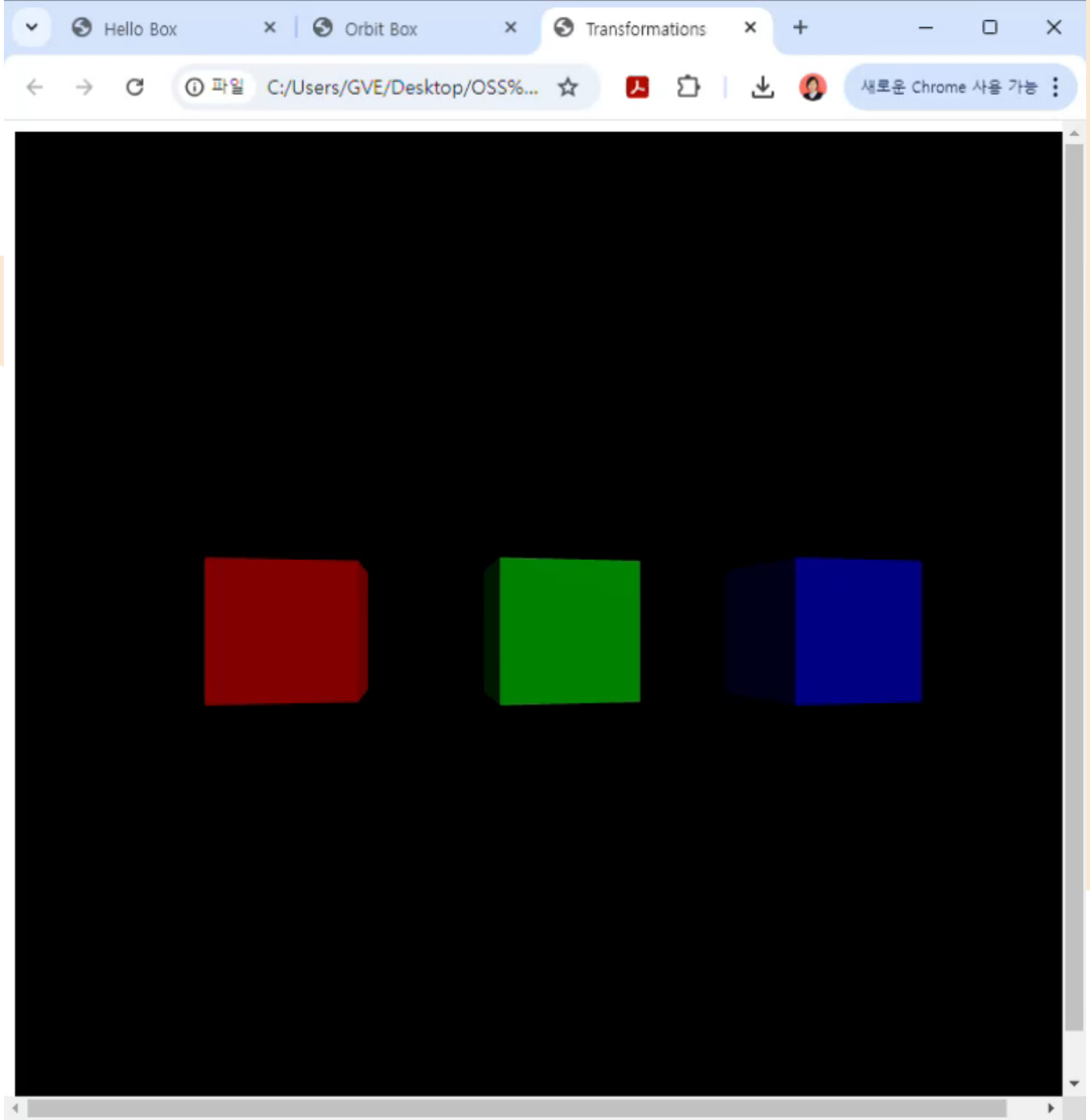
File Edit Selection View Go Run ··· Search

hello_box.html | orbit_box.html | transform_box.html

C: > Users > GVE > Desktop > OSS talk > src > transform_box.html > {} "transform_box.html"

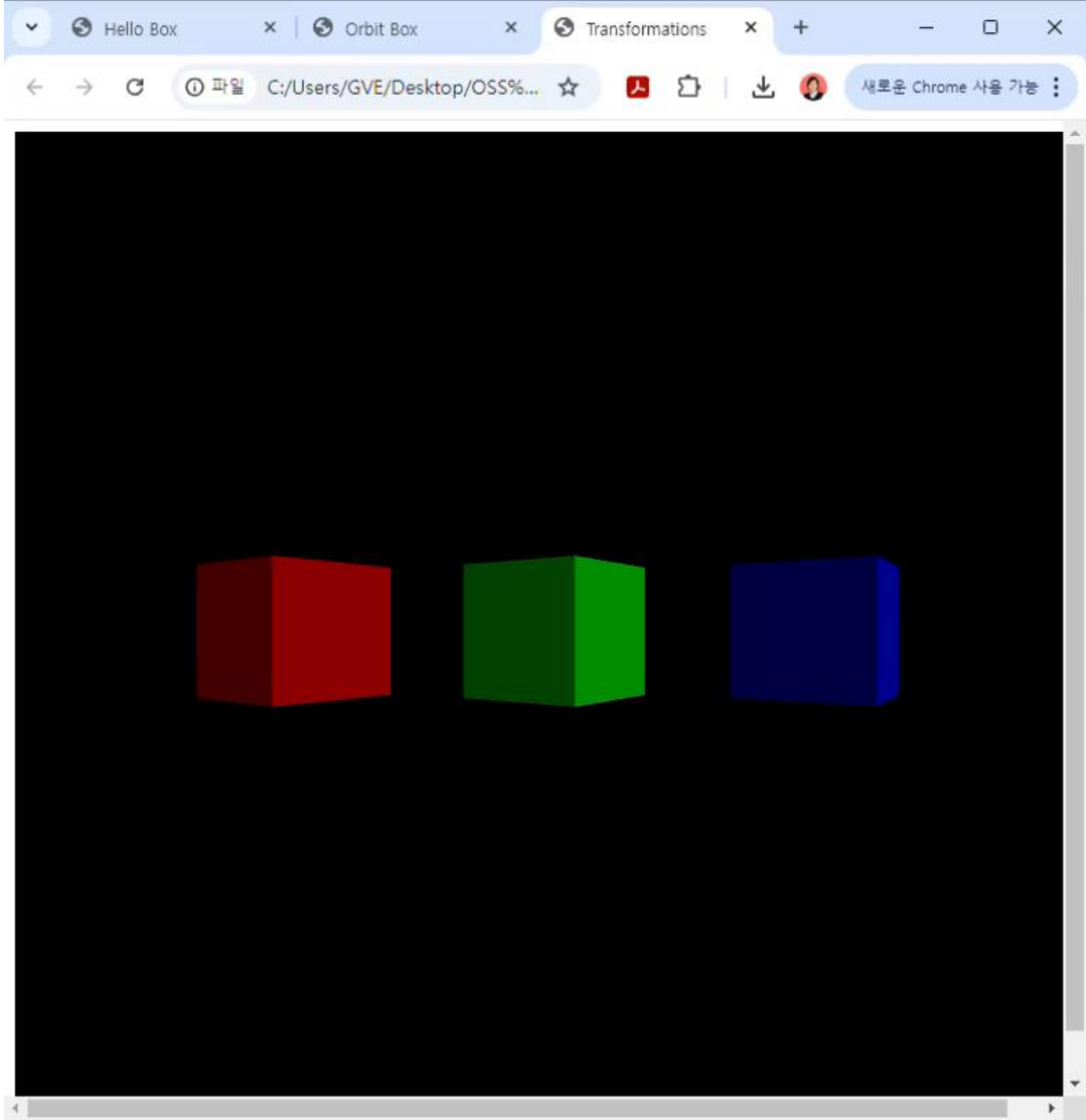
```
                cubes.push(new THREE.Mesh(geometry, materials[idx]));
                scene.add(cubes[idx]);
            }
            cubes[0].position.x = -2;
            cubes[2].position.x = 2;

            var angle = 0.0;
            const animate = function() {
                requestAnimationFrame(animate);

                angle += 0.01;
                for(var idx=0; idx<3; idx++) {
                    cubes[idx].rotation.y = angle;
                }

                renderer.render(scene, camera);
            }

            animate();

            window.addEventListener('resize', resize);
        }

        function resize() {
            camera.aspect = window.innerWidth / window.innerHeight;
            camera.updateProjectionMatrix();

            renderer.setSize(window.innerWidth, window.innerHeight);
        }
    </script>
</body>
</html>
```

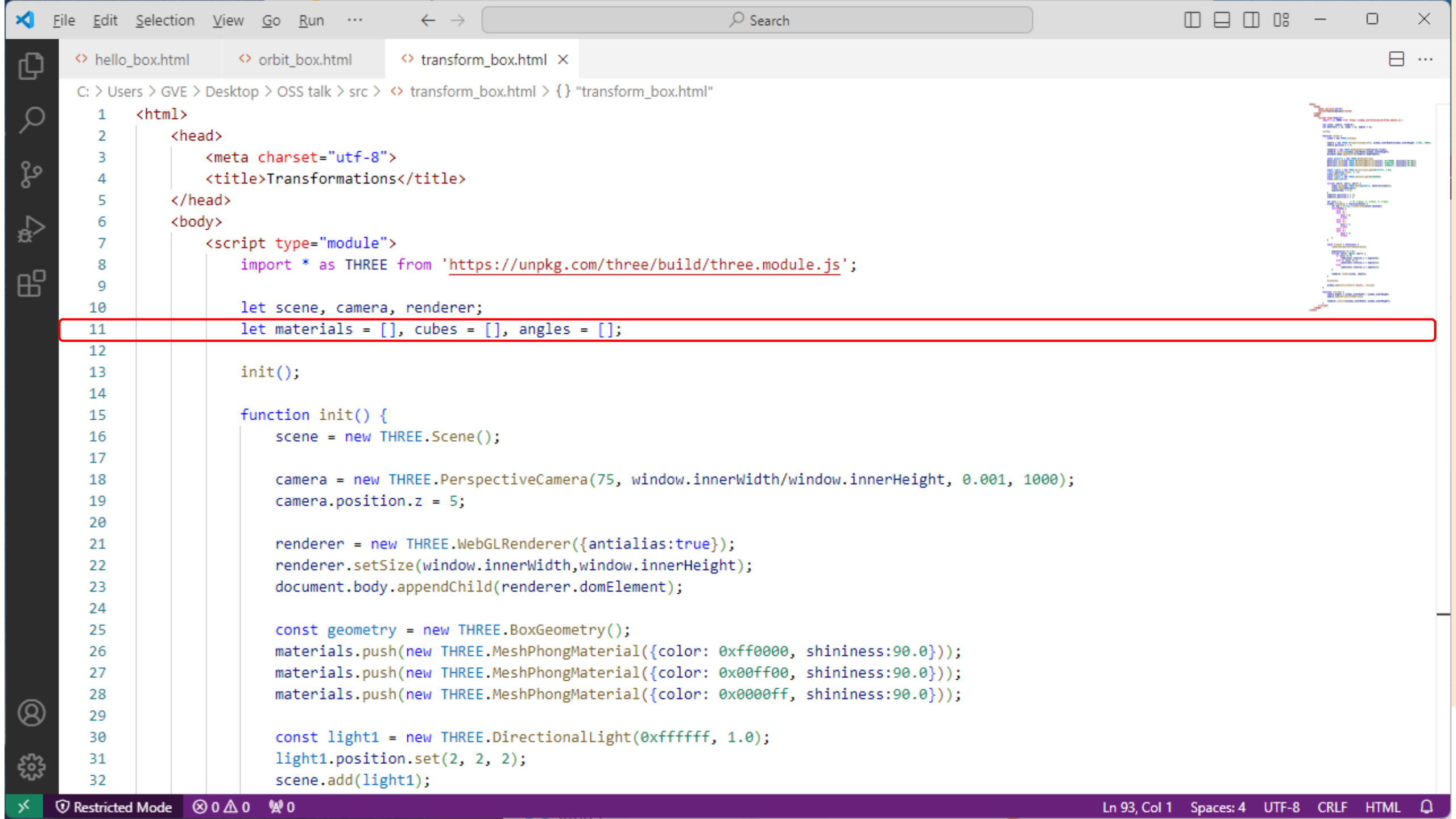
Restricted Mode ⊗ 0 ⚠ 0 0 Ln 69, Col 1 Spaces: 4 UTF-8 CRLF HTML

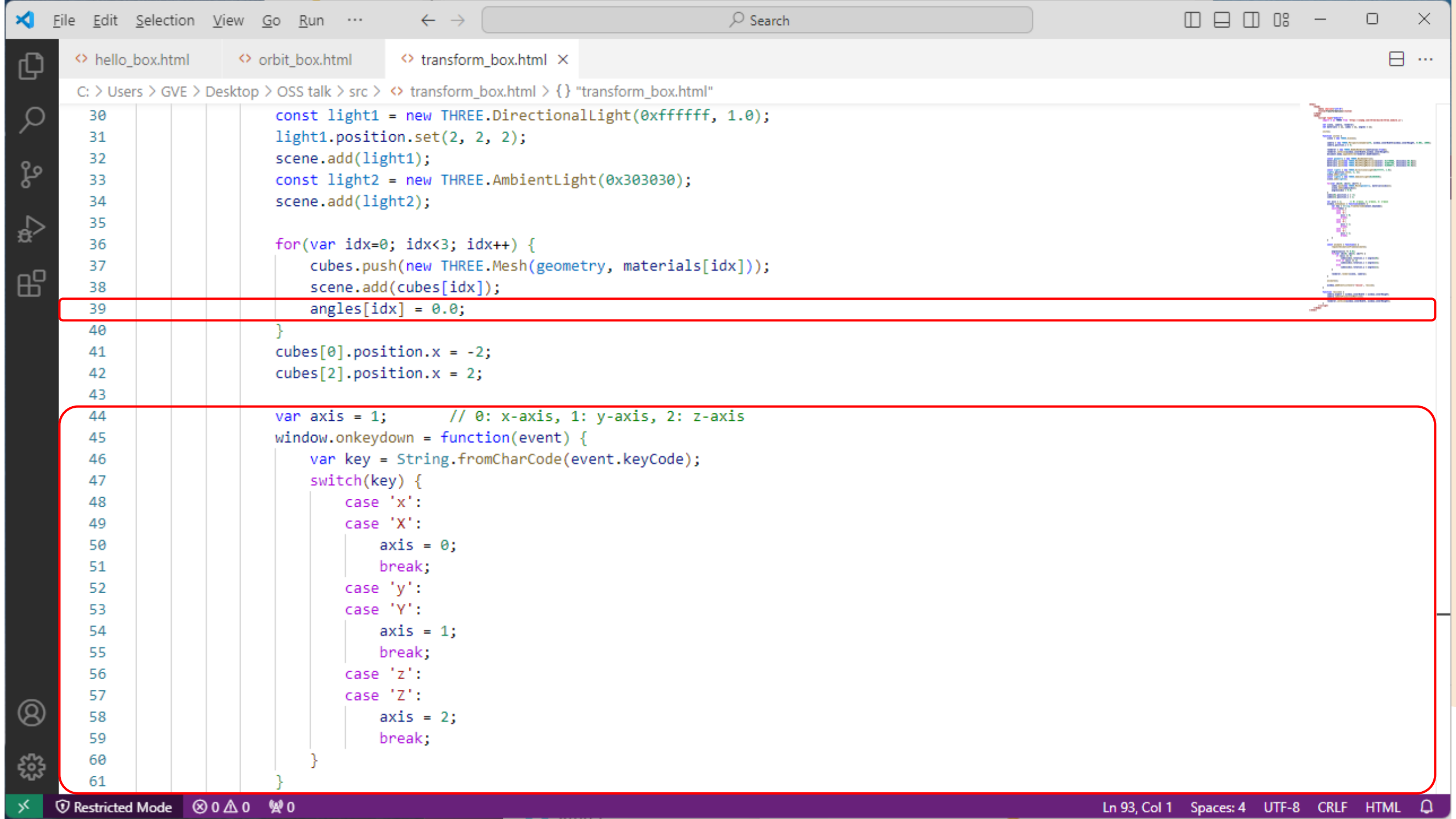

Hello Box
Orbit Box
Transformations
파일
C:/Users/GVE/Desktop/OSS%...
새로운 Chrome 사용 가능

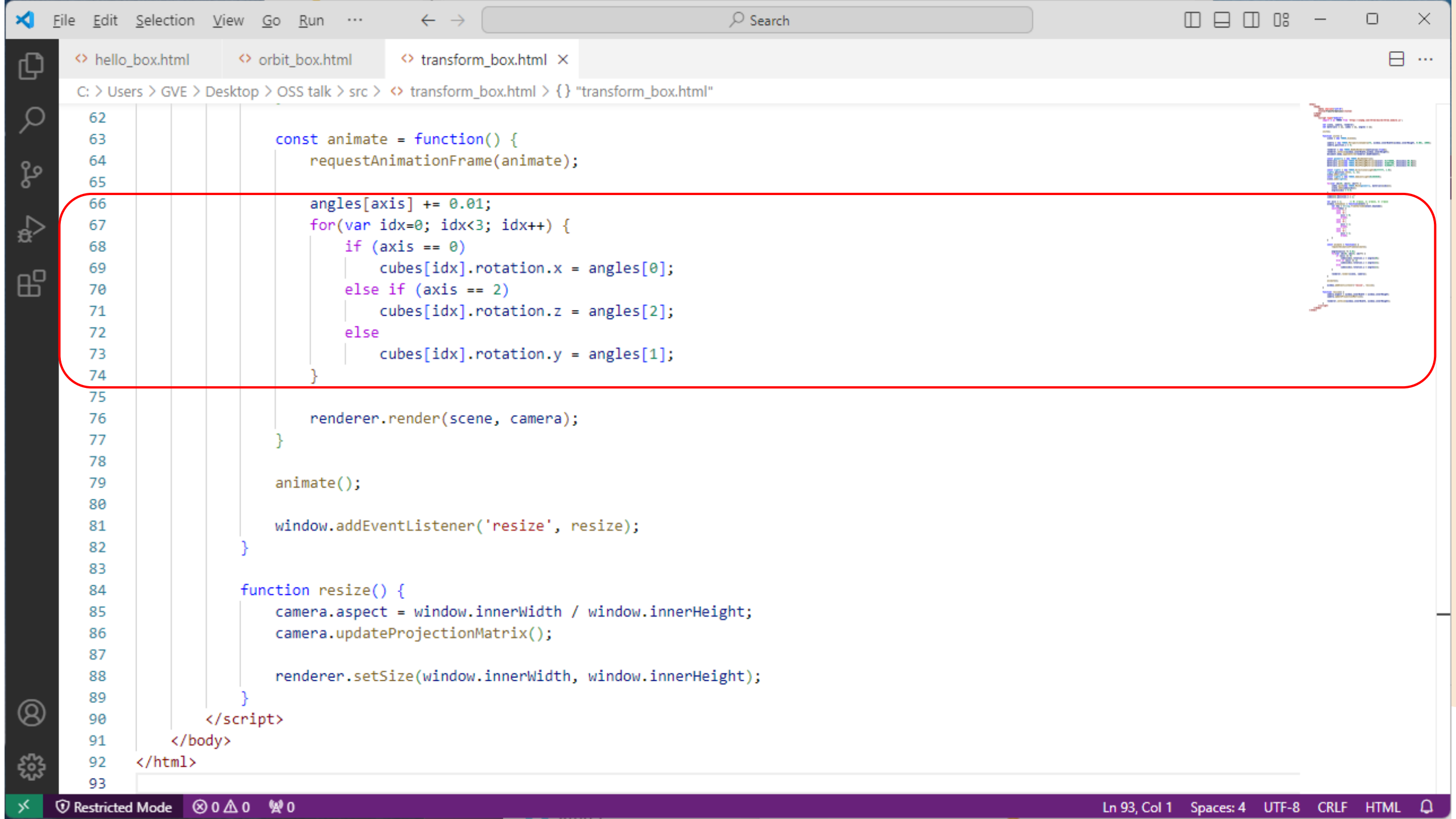
연습 문제 (3)

- X, Y, Z 키 입력으로 회전 축을 변경해 보시오.




File Edit Selection View Go Run ···
Search
hello_box.html
orbit_box.html
transform_box.html
C: > Users > GVE > Desktop > OSS talk > src > transform_box.html > {} "transform_box.html"
<html>
<head>
<meta charset="utf-8">
<title>Transformations</title>
</head>
<body>
<script type="module">
import * as THREE from 'https://unpkg.com/three/build/three.module.js';
let scene, camera, renderer;
let materials = [], cubes = [], angles = [];
init();
function init() {
scene = new THREE.Scene();
camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight, 0.001, 1000);
camera.position.z = 5;
renderer = new THREE.WebGLRenderer({antialias:true});
renderer.setSize(window.innerWidth,window.innerHeight);
document.body.appendChild(renderer.domElement);
const geometry = new THREE.BoxGeometry();
materials.push(new THREE.MeshPhongMaterial({color: 0xff0000, shininess:90.0}));
materials.push(new THREE.MeshPhongMaterial({color: 0x00ff00, shininess:90.0}));
materials.push(new THREE.MeshPhongMaterial({color: 0x0000ff, shininess:90.0}));
const light1 = new THREE.DirectionalLight(0xffffff, 1.0);
light1.position.set(2, 2, 2);
scene.add(light1);
Restricted Mode
Ln 93, Col 1
Spaces: 4
UTF-8
CRLF
HTML


File Edit Selection View Go Run
Search
hello_box.html
orbit_box.html
transform_box.html
C: > Users > GVE > Desktop > OSS talk > src > transform_box.html > {} "transform_box.html"
30 const light1 = new THREE.DirectionalLight(0xffffff, 1.0);
31 light1.position.set(2, 2, 2);
32 scene.add(light1);
33 const light2 = new THREE.AmbientLight(0x303030);
34 scene.add(light2);
35
36 for(var idx=0; idx<3; idx++) {
37 cubes.push(new THREE.Mesh(geometry, materials[idx]));
38 scene.add(cubes[idx]);
39 angles[idx] = 0.0;
40 }
41 cubes[0].position.x = -2;
42 cubes[2].position.x = 2;
43
44 var axis = 1; // 0: x-axis, 1: y-axis, 2: z-axis
45 window.onkeydown = function(event) {
46 var key = String.fromCharCode(event.keyCode);
47 switch(key) {
48 case 'x':
49 case 'X':
50 axis = 0;
51 break;
52 case 'y':
53 case 'Y':
54 axis = 1;
55 break;
56 case 'z':
57 case 'Z':
58 axis = 2;
59 break;
60 }
61 }
Restricted Mode
Ln 93, Col 1
Spaces: 4
UTF-8
CRLF
HTML


File Edit Selection View Go Run ···
Search
hello_box.html
orbit_box.html
transform_box.html
C: > Users > GVE > Desktop > OSS talk > src > transform_box.html > {} "transform_box.html"
62
63 const animate = function() {
64 requestAnimationFrame(animate);
65
66 angles[axis] += 0.01;
67 for(var idx=0; idx<3; idx++) {
68 if (axis == 0)
69 cubes[idx].rotation.x = angles[0];
70 else if (axis == 2)
71 cubes[idx].rotation.z = angles[2];
72 else
73 cubes[idx].rotation.y = angles[1];
74 }
75
76 renderer.render(scene, camera);
77 }
78
79 animate();
80
81 window.addEventListener('resize', resize);
82 }
83
84 function resize() {
85 camera.aspect = window.innerWidth / window.innerHeight;
86 camera.updateProjectionMatrix();
87
88 renderer.setSize(window.innerWidth, window.innerHeight);
89 }
90 </script>
91 </body>
92 </html>
93
Restricted Mode
0 0
0
Ln 93, Col 1
Spaces: 4
UTF-8
CRLF
HTML

File Edit Selection View Go Run ··· Search

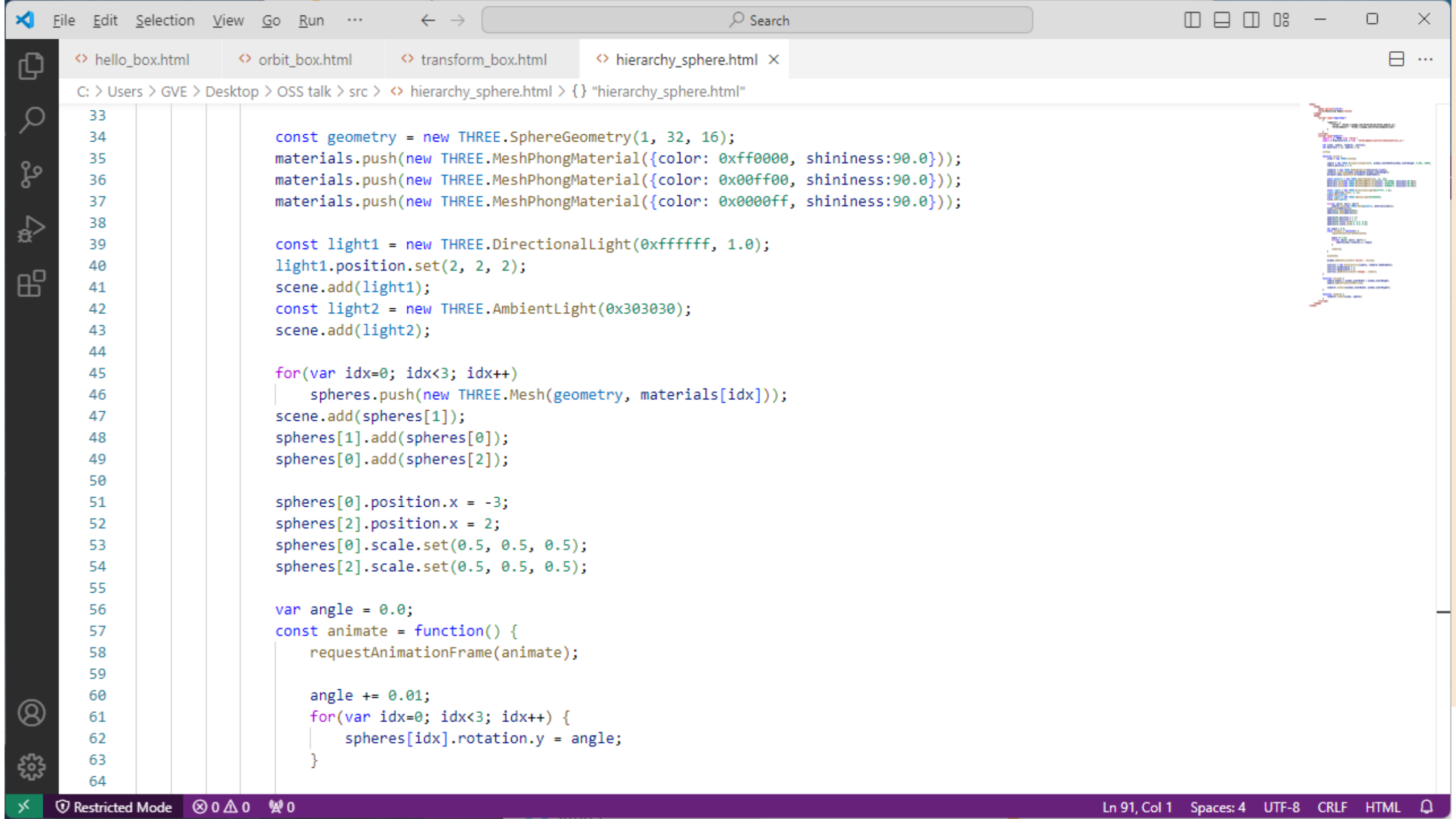
hello_box.html | orbit_box.html | transform_box.html | hierarchy_sphere.html

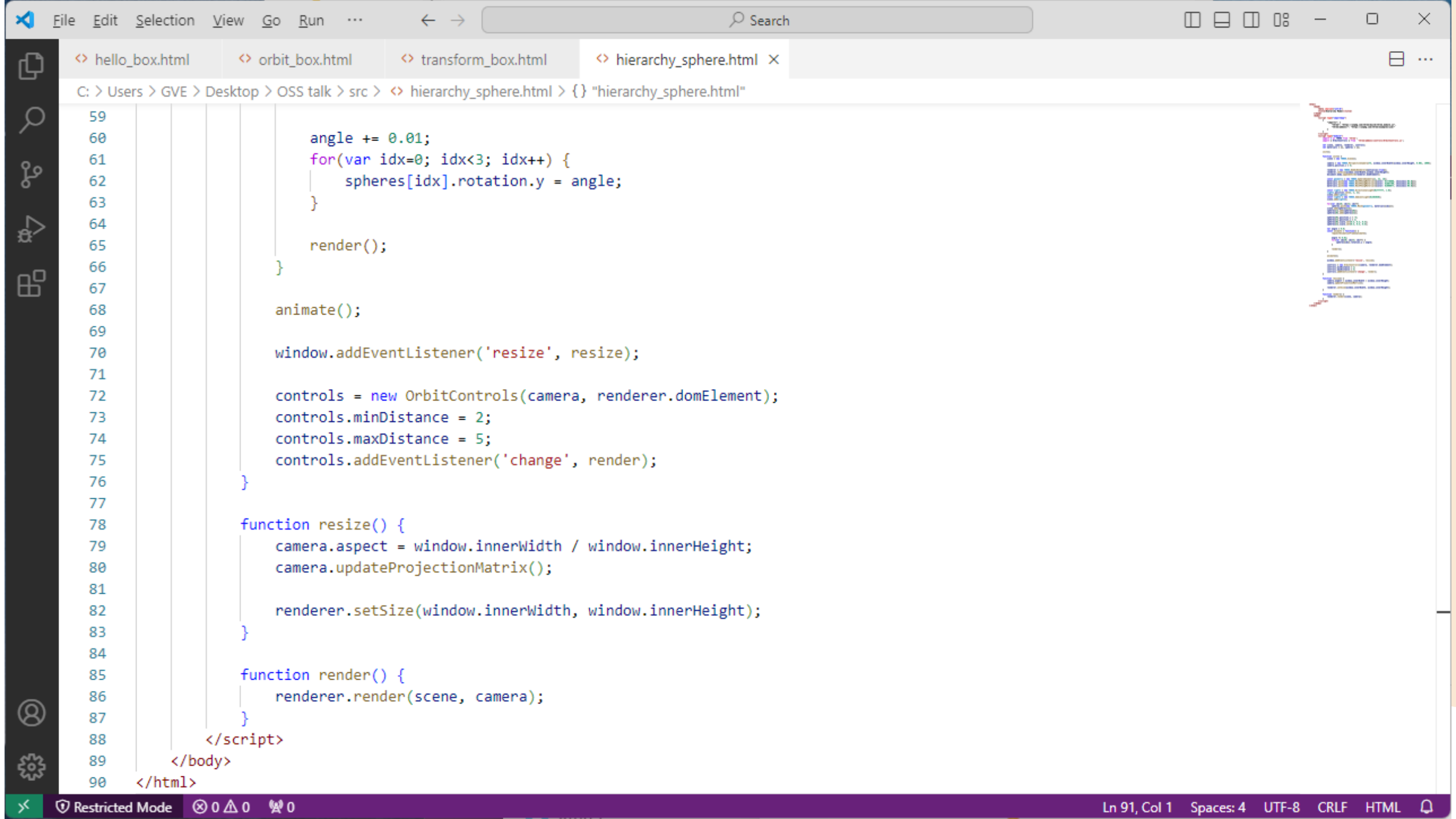
C: > Users > GVE > Desktop > OSS talk > src > hierarchy_sphere.html > {} "hierarchy_sphere.html"

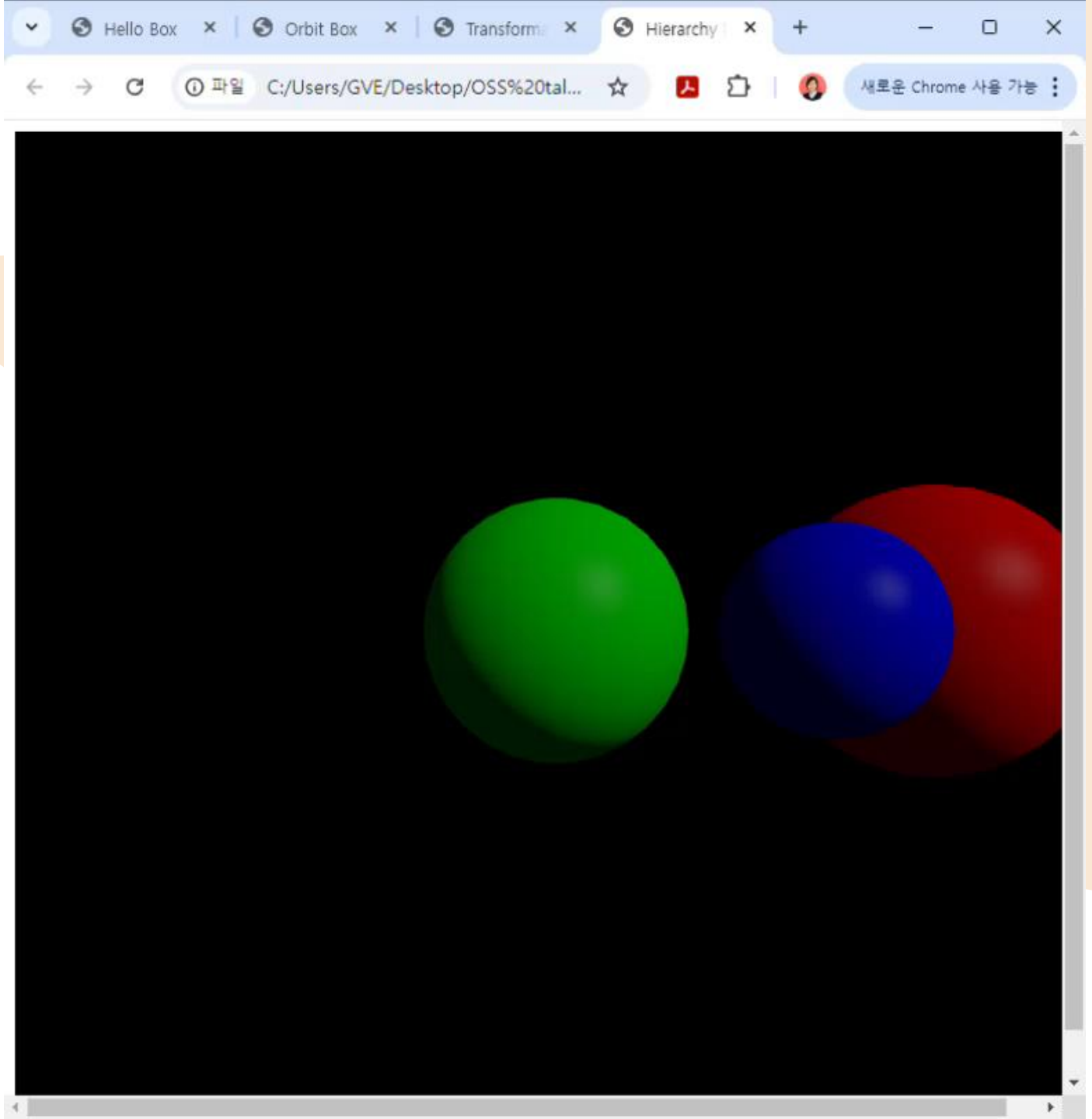
https://github.com/ProfSunKim/threejs

```
<html>
    <head>
        <meta charset="utf-8">
        <title>Hierarchy Model</title>
    </head>
    <body>
        <script type="importmap">
            {
                "imports": {
                    "three": "https://unpkg.com/three/build/three.module.js",
                    "three/addons/": "https://unpkg.com/three/examples/jsm/"
                }
            }
        </script>
        <script type="module">
            import * as THREE from 'three';
            import { OrbitControls } from  'three/addons/controls/OrbitControls.js';

            let scene, camera, renderer, controls;
            let materials = [], spheres = [];

            init();

            function init() {
                scene = new THREE.Scene();

                camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight, 0.001, 1000);
                camera.position.z = 5;

                renderer = new THREE.WebGLRenderer({antialias:true});
                renderer.setSize(window.innerWidth,window.innerHeight);
                document.body.appendChild(renderer.domElement);
```

Restricted Mode ⊗ 0 ⚠ 0 0 Ln 91, Col 1 Spaces: 4 UTF-8 CRLF HTML

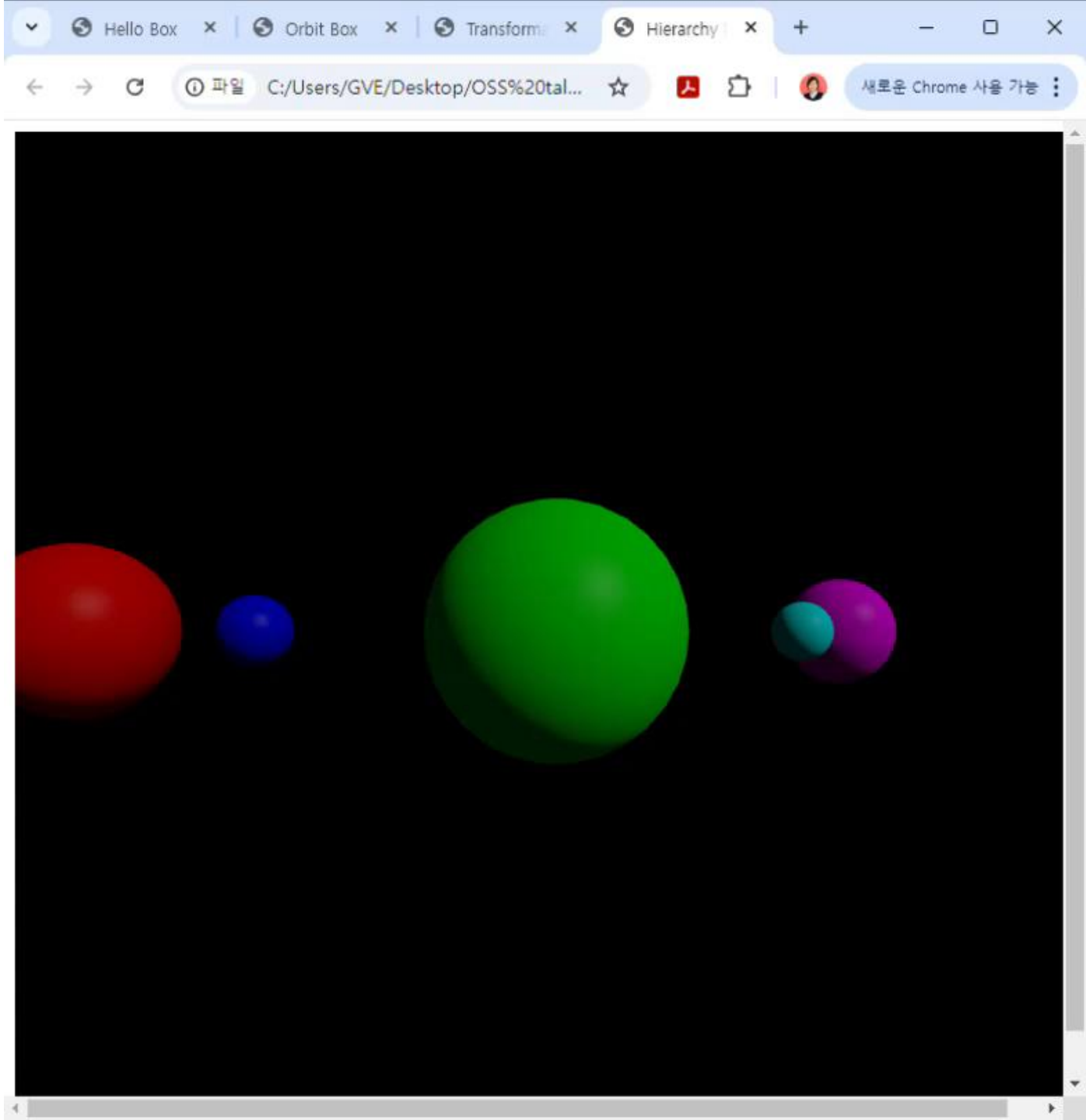

File Edit Selection View Go Run ···
Search
hello_box.html
orbit_box.html
transform_box.html
hierarchy_sphere.html
C: > Users > GVE > Desktop > OSS talk > src > hierarchy_sphere.html > {} "hierarchy_sphere.html"
const geometry = new THREE.SphereGeometry(1, 32, 16);
materials.push(new THREE.MeshPhongMaterial({color: 0xff0000, shininess:90.0}));
materials.push(new THREE.MeshPhongMaterial({color: 0x00ff00, shininess:90.0}));
materials.push(new THREE.MeshPhongMaterial({color: 0x0000ff, shininess:90.0}));
const light1 = new THREE.DirectionalLight(0xffffff, 1.0);
light1.position.set(2, 2, 2);
scene.add(light1);
const light2 = new THREE.AmbientLight(0x303030);
scene.add(light2);
for(var idx=0; idx<3; idx++)
spheres.push(new THREE.Mesh(geometry, materials[idx]));
scene.add(spheres[1]);
spheres[1].add(spheres[0]);
spheres[0].add(spheres[2]);
spheres[0].position.x = -3;
spheres[2].position.x = 2;
spheres[0].scale.set(0.5, 0.5, 0.5);
spheres[2].scale.set(0.5, 0.5, 0.5);
var angle = 0.0;
const animate = function() {
requestAnimationFrame(animate);
angle += 0.01;
for(var idx=0; idx<3; idx++) {
spheres[idx].rotation.y = angle;
}
Restricted Mode
Ln 91, Col 1
Spaces: 4
UTF-8
CRLF
HTML

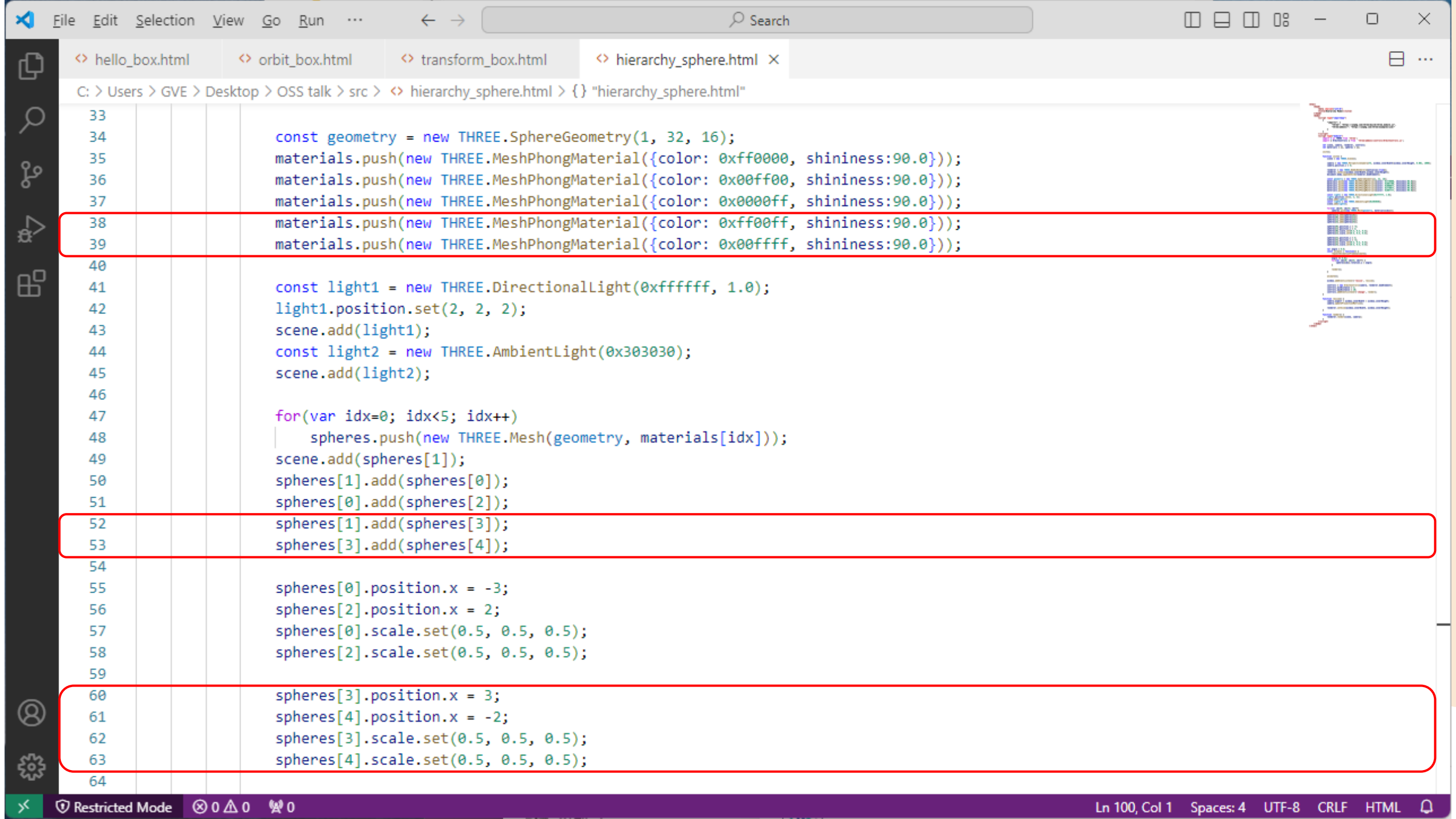

File Edit Selection View Go Run
Search
hello_box.html
orbit_box.html
transform_box.html
hierarchy_sphere.html
C: > Users > GVE > Desktop > OSS talk > src > hierarchy_sphere.html > {} "hierarchy_sphere.html"
59
60 angle += 0.01;
61 for(var idx=0; idx<3; idx++) {
62 spheres[idx].rotation.y = angle;
63 }
64
65 render();
66 }
67
68 animate();
69
70 window.addEventListener('resize', resize);
71
72 controls = new OrbitControls(camera, renderer.domElement);
73 controls.minDistance = 2;
74 controls.maxDistance = 5;
75 controls.addEventListener('change', render);
76 }
77
78 function resize() {
79 camera.aspect = window.innerWidth / window.innerHeight;
80 camera.updateProjectionMatrix();
81
82 renderer.setSize(window.innerWidth, window.innerHeight);
83 }
84
85 function render() {
86 renderer.render(scene, camera);
87 }
88 </script>
89 </body>
90 </html>
Restricted Mode
0 0
0
Ln 91, Col 1
Spaces: 4
UTF-8
CRLF
HTML

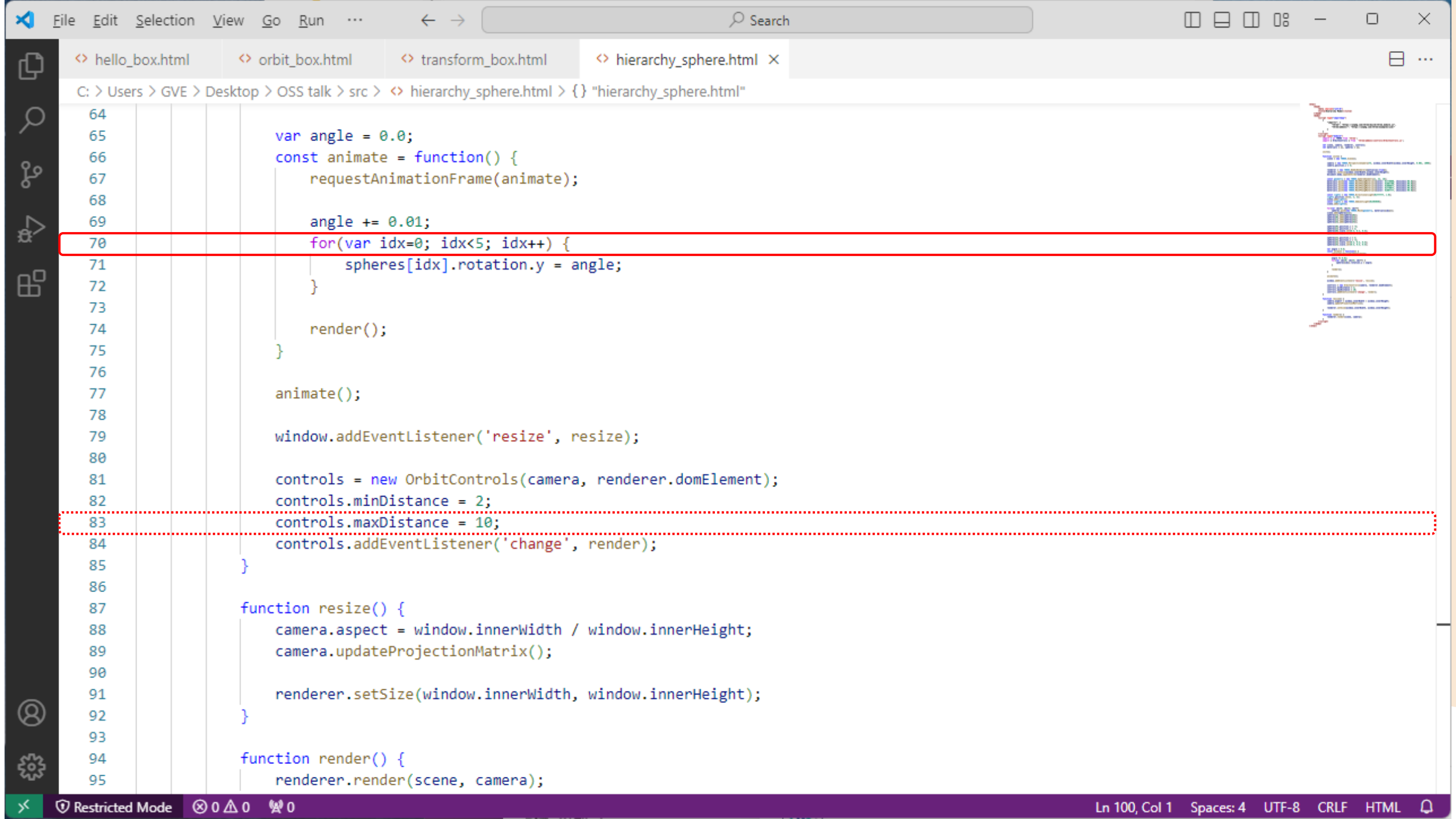

Hello Box
Orbit Box
Hierarchy
파일
C:/Users/GVE/Desktop/OSS%20tal...
새로운 Chrome 사용 가능

연습 문제 (4)

- 오른쪽 그림과 같이 Sphere 2개를 더 추가해 보시오.




File Edit Selection View Go Run
Search
hello_box.html
orbit_box.html
transform_box.html
hierarchy_sphere.html
C: > Users > GVE > Desktop > OSS talk > src > hierarchy_sphere.html > {} "hierarchy_sphere.html"
33
34 const geometry = new THREE.SphereGeometry(1, 32, 16);
35 materials.push(new THREE.MeshPhongMaterial({color: 0xff0000, shininess:90.0}));
36 materials.push(new THREE.MeshPhongMaterial({color: 0x00ff00, shininess:90.0}));
37 materials.push(new THREE.MeshPhongMaterial({color: 0x0000ff, shininess:90.0}));
38 materials.push(new THREE.MeshPhongMaterial({color: 0xff00ff, shininess:90.0}));
39 materials.push(new THREE.MeshPhongMaterial({color: 0x00ffff, shininess:90.0}));
40
41 const light1 = new THREE.DirectionalLight(0xffffff, 1.0);
42 light1.position.set(2, 2, 2);
43 scene.add(light1);
44 const light2 = new THREE.AmbientLight(0x303030);
45 scene.add(light2);
46
47 for(var idx=0; idx<5; idx++)
48 spheres.push(new THREE.Mesh(geometry, materials[idx]));
49 scene.add(spheres[1]);
50 spheres[1].add(spheres[0]);
51 spheres[0].add(spheres[2]);
52 spheres[1].add(spheres[3]);
53 spheres[3].add(spheres[4]);
54
55 spheres[0].position.x = -3;
56 spheres[2].position.x = 2;
57 spheres[0].scale.set(0.5, 0.5, 0.5);
58 spheres[2].scale.set(0.5, 0.5, 0.5);
59
60 spheres[3].position.x = 3;
61 spheres[4].position.x = -2;
62 spheres[3].scale.set(0.5, 0.5, 0.5);
63 spheres[4].scale.set(0.5, 0.5, 0.5);
64
Restricted Mode
Ln 100, Col 1 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run ···
Search
hello_box.html
orbit_box.html
transform_box.html
hierarchy_sphere.html
C: > Users > GVE > Desktop > OSS talk > src > hierarchy_sphere.html > {} "hierarchy_sphere.html"
64
65 var angle = 0.0;
66 const animate = function() {
67 requestAnimationFrame(animate);
68
69 angle += 0.01;
70 for(var idx=0; idx<5; idx++) {
71 spheres[idx].rotation.y = angle;
72 }
73
74 render();
75 }
76
77 animate();
78
79 window.addEventListener('resize', resize);
80
81 controls = new OrbitControls(camera, renderer.domElement);
82 controls.minDistance = 2;
83 controls.maxDistance = 10;
84 controls.addEventListener('change', render);
85 }
86
87 function resize() {
88 camera.aspect = window.innerWidth / window.innerHeight;
89 camera.updateProjectionMatrix();
90
91 renderer.setSize(window.innerWidth, window.innerHeight);
92 }
93
94 function render() {
95 renderer.render(scene, camera);
Restricted Mode
0 0
0
Ln 100, Col 1
Spaces: 4
UTF-8
CRLF
HTML

File Edit Selection View Go Run ··· Search

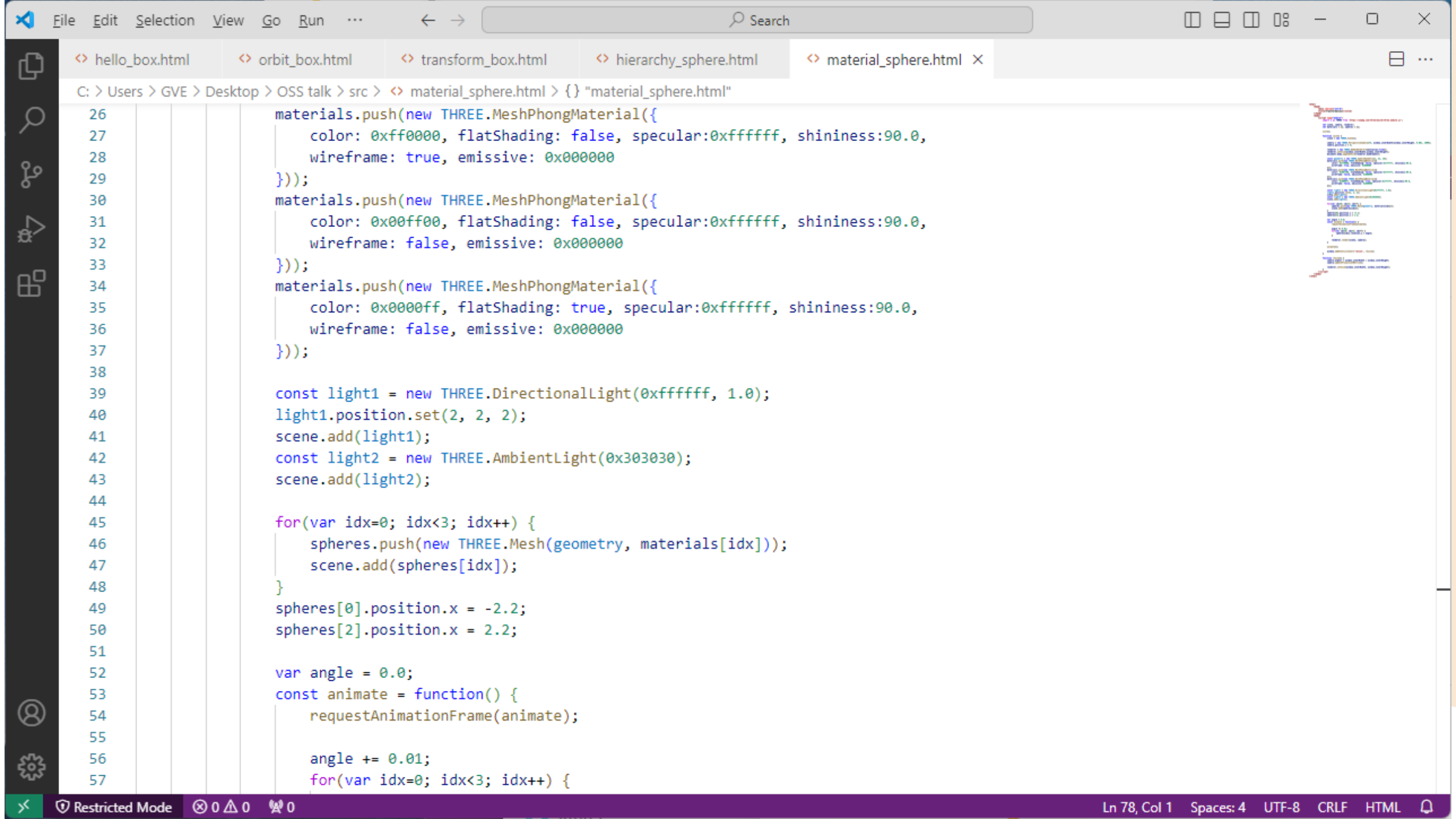
hello_box.html | orbit_box.html | transform_box.html | hierarchy_sphere.html | material_sphere.html

C: > Users > GVE > Desktop > OSS talk > src > material_sphere.html > {} "material_sphere.html"

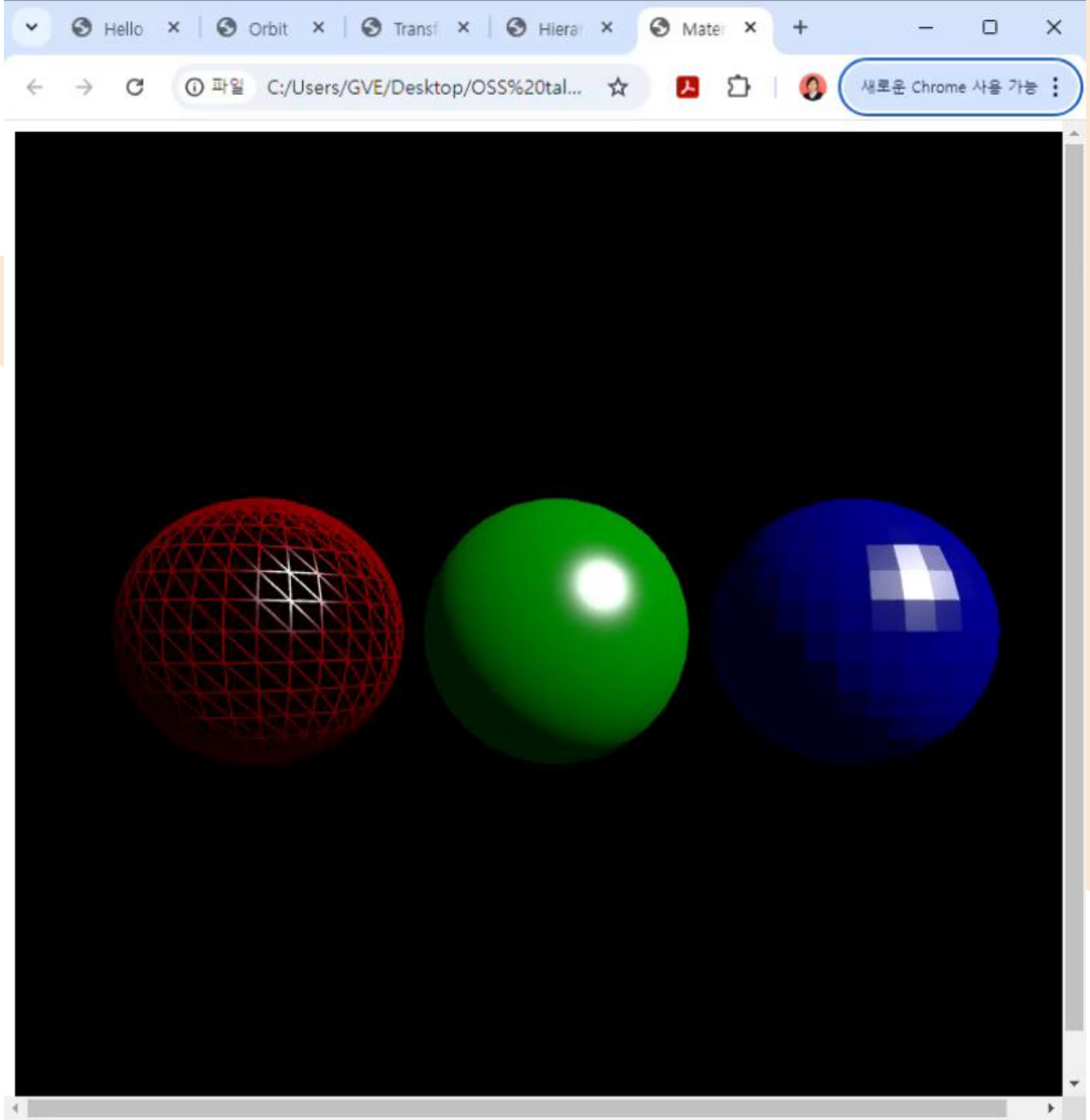
https://github.com/ProfSunKim/threejs

```
<html>
    <head>
        <meta charset="utf-8">
        <title>Materials</title>
    </head>
    <body>
        <script type="module">
            import * as THREE from 'https://unpkg.com/three/build/three.module.js';

            let scene, camera, renderer;
            let materials = [], spheres = [];

            init();

            function init() {
                scene = new THREE.Scene();

                camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight, 0.001, 1000);
                camera.position.z = 5;

                renderer = new THREE.WebGLRenderer({antialias:true});
                renderer.setSize(window.innerWidth,window.innerHeight);
                document.body.appendChild(renderer.domElement);

                const geometry = new THREE.SphereGeometry(1, 32, 16);
                materials.push(new THREE.MeshPhongMaterial({
                    color: 0xff0000, flatShading: false, specular:0xffffff, shininess:90.0,
                    wireframe: true, emissive: 0x000000
                }));
                materials.push(new THREE.MeshPhongMaterial({
                    color: 0x00ff00, flatShading: false, specular:0xffffff, shininess:90.0,
                    wireframe: false, emissive: 0x000000
```

Restricted Mode ⊗ 0 ⚠ 0 0 Ln 78, Col 1 Spaces: 4 UTF-8 CRLF HTML

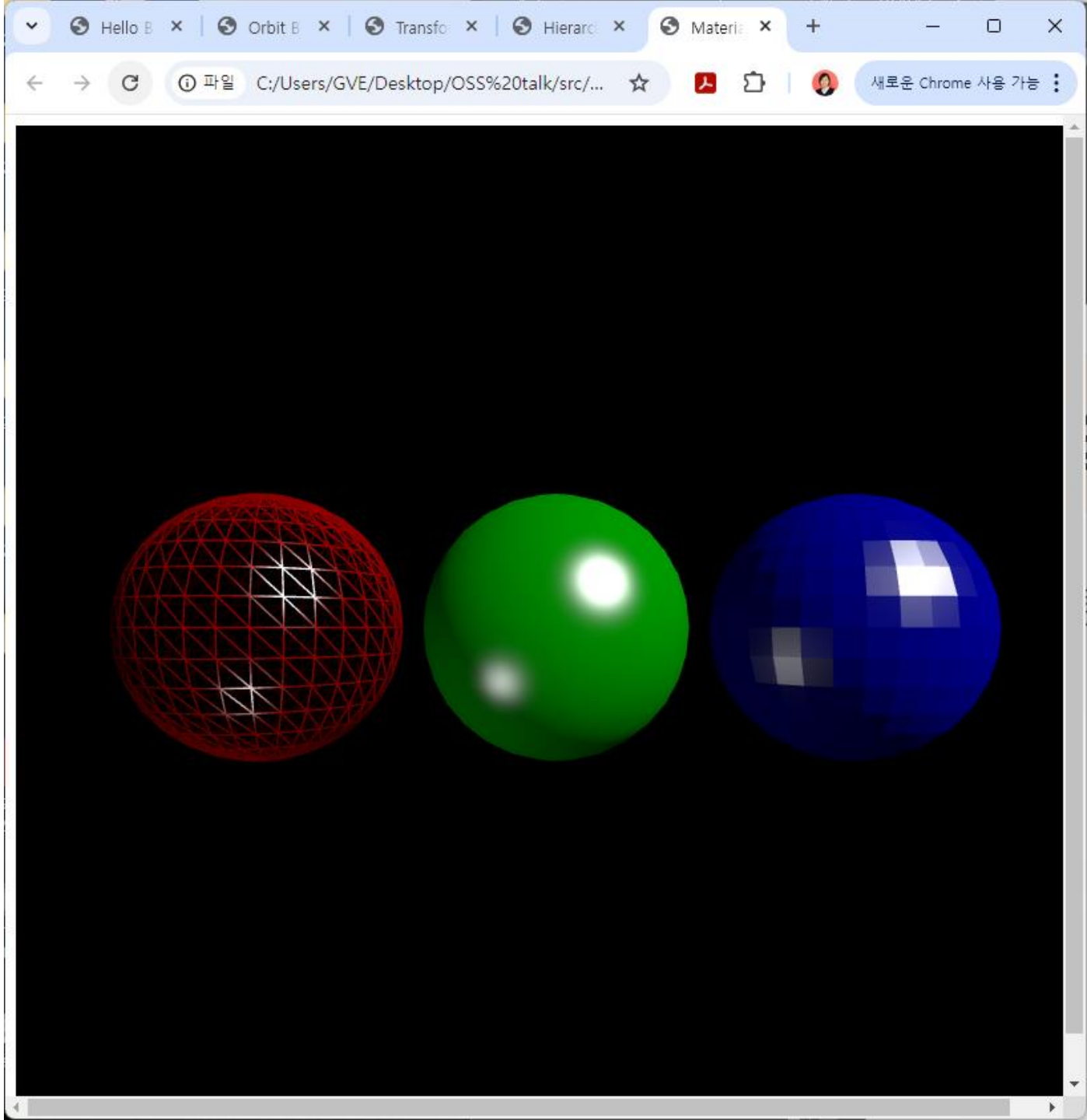

File Edit Selection View Go Run ···
Search
hello_box.html
orbit_box.html
transform_box.html
hierarchy_sphere.html
material_sphere.html
C: > Users > GVE > Desktop > OSS talk > src > material_sphere.html > {} "material_sphere.html"
materials.push(new THREE.MeshPhongMaterial({
color: 0xff0000, flatShading: false, specular:0xffffff, shininess:90.0,
wireframe: true, emissive: 0x000000
}));
materials.push(new THREE.MeshPhongMaterial({
color: 0x00ff00, flatShading: false, specular:0xffffff, shininess:90.0,
wireframe: false, emissive: 0x000000
}));
materials.push(new THREE.MeshPhongMaterial({
color: 0x0000ff, flatShading: true, specular:0xffffff, shininess:90.0,
wireframe: false, emissive: 0x000000
}));
const light1 = new THREE.DirectionalLight(0xffffff, 1.0);
light1.position.set(2, 2, 2);
scene.add(light1);
const light2 = new THREE.AmbientLight(0x303030);
scene.add(light2);
for(var idx=0; idx<3; idx++) {
spheres.push(new THREE.Mesh(geometry, materials[idx]));
scene.add(spheres[idx]);
}
spheres[0].position.x = -2.2;
spheres[2].position.x = 2.2;
var angle = 0.0;
const animate = function() {
requestAnimationFrame(animate);
angle += 0.01;
for(var idx=0; idx<3; idx++) {
Restricted Mode
Ln 78, Col 1
Spaces: 4
UTF-8
CRLF
HTML

```
51
52                  var angle = 0.0;
53                  const animate = function() {
54                      requestAnimationFrame(animate);
55
56                      angle += 0.01;
57                      for(var idx=0; idx<3; idx++) {
58                          spheres[idx].rotation.y = angle;
59                      }
60
61                      renderer.render(scene, camera);
62                  }
63
64                  animate();
65
66                  window.addEventListener('resize', resize);
67              }
68
69              function resize() {
70                  camera.aspect = window.innerWidth / window.innerHeight;
71                  camera.updateProjectionMatrix();
72
73                  renderer.setSize(window.innerWidth, window.innerHeight);
74              }
75          </script>
76      </body>
77  </html>
78
```


Hello
Orbit
Transf
Hiera
Mate
파일
C:/Users/GVE/Desktop/OSS%20tal...
새로운 Chrome 사용 가능

연습 문제 (5)

- 오른쪽 그림과 같이 점광원을 추가해 보시오.



PointLight

A light that gets emitted from a single point in all directions. A common use case for this is to replicate the light emitted from a bare lightbulb.

This light can cast shadows - see PointLightShadow page for details.

Code Example

```
const light = new THREE.PointLight( 0xff0000, 1, 100 );
light.position.set( 50, 50, 50 );
scene.add( light );
```

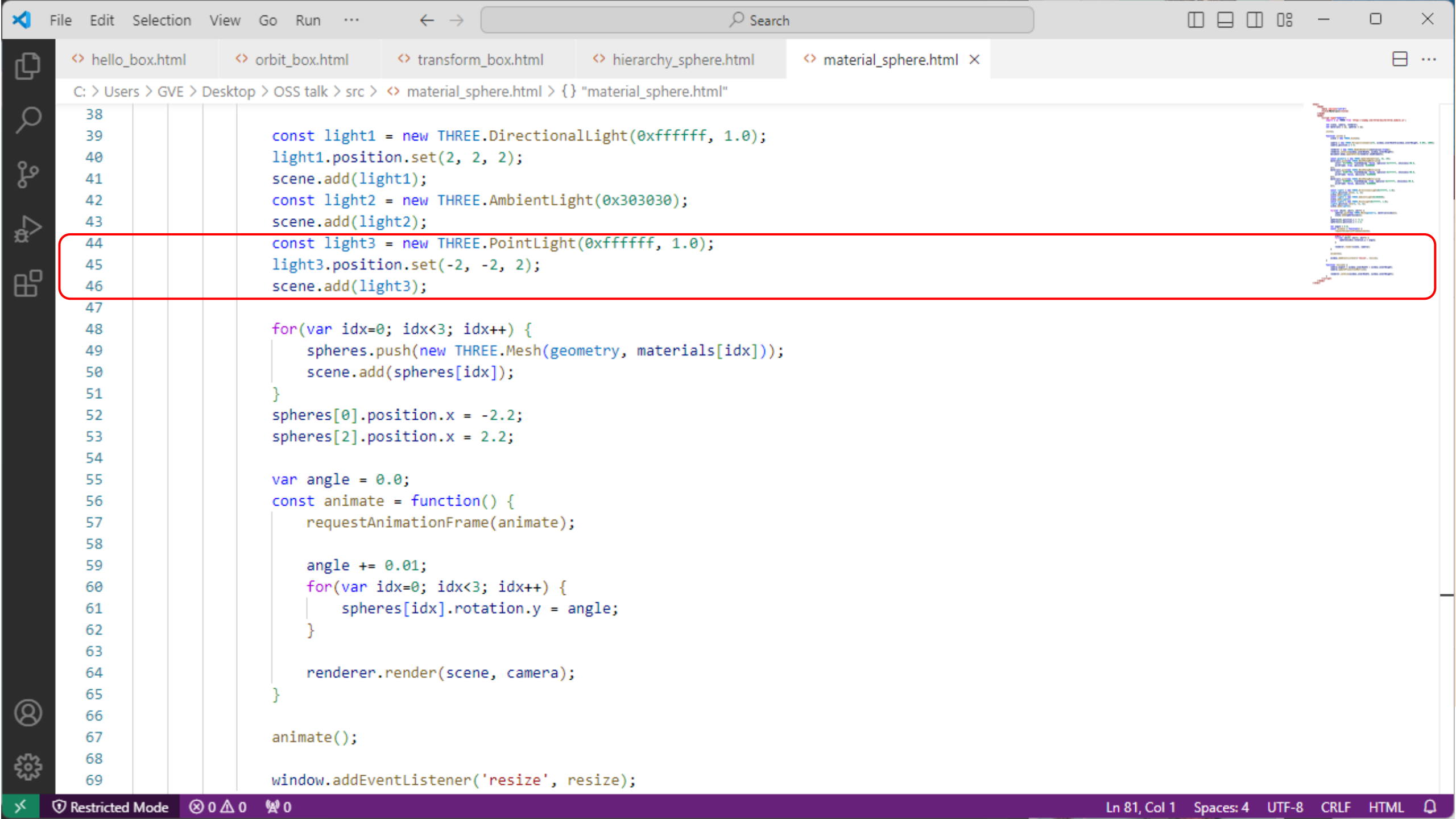
Examples

lights / pointlights
effects / anaglyph
geometry / text
lensflares

Constructor

PointLight(color : Integer, intensity : Float, distance : Number, decay : Float)

color - (optional) hexadecimal color of the light. Default is 0xffffff (white).
intensity - (optional) numeric value of the light's strength/intensity. Default is `1`.
distance - Maximum range of the light. Default is `0` (no limit).
decay - The amount the light dims along the distance of the light. Default is `2`.


File Edit Selection View Go Run ···
Search
hello_box.html
orbit_box.html
transform_box.html
hierarchy_sphere.html
material_sphere.html
C: > Users > GVE > Desktop > OSS talk > src > material_sphere.html > {} "material_sphere.html"
38
39 const light1 = new THREE.DirectionalLight(0xffffff, 1.0);
40 light1.position.set(2, 2, 2);
41 scene.add(light1);
42 const light2 = new THREE.AmbientLight(0x303030);
43 scene.add(light2);
44 const light3 = new THREE.PointLight(0xffffff, 1.0);
45 light3.position.set(-2, -2, 2);
46 scene.add(light3);
47
48 for(var idx=0; idx<3; idx++) {
49 spheres.push(new THREE.Mesh(geometry, materials[idx]));
50 scene.add(spheres[idx]);
51 }
52 spheres[0].position.x = -2.2;
53 spheres[2].position.x = 2.2;
54
55 var angle = 0.0;
56 const animate = function() {
57 requestAnimationFrame(animate);
58
59 angle += 0.01;
60 for(var idx=0; idx<3; idx++) {
61 spheres[idx].rotation.y = angle;
62 }
63
64 renderer.render(scene, camera);
65 }
66
67 animate();
68
69 window.addEventListener('resize', resize);
Restricted Mode
0 0
0
Ln 81, Col 1
Spaces: 4
UTF-8
CRLF
HTML

File Edit Selection View Go Run ··· Search

hello_box.html | orbit_box.html | transform_box.html | hierarchy_sphere.html | material_sphere.html | texture_box.html

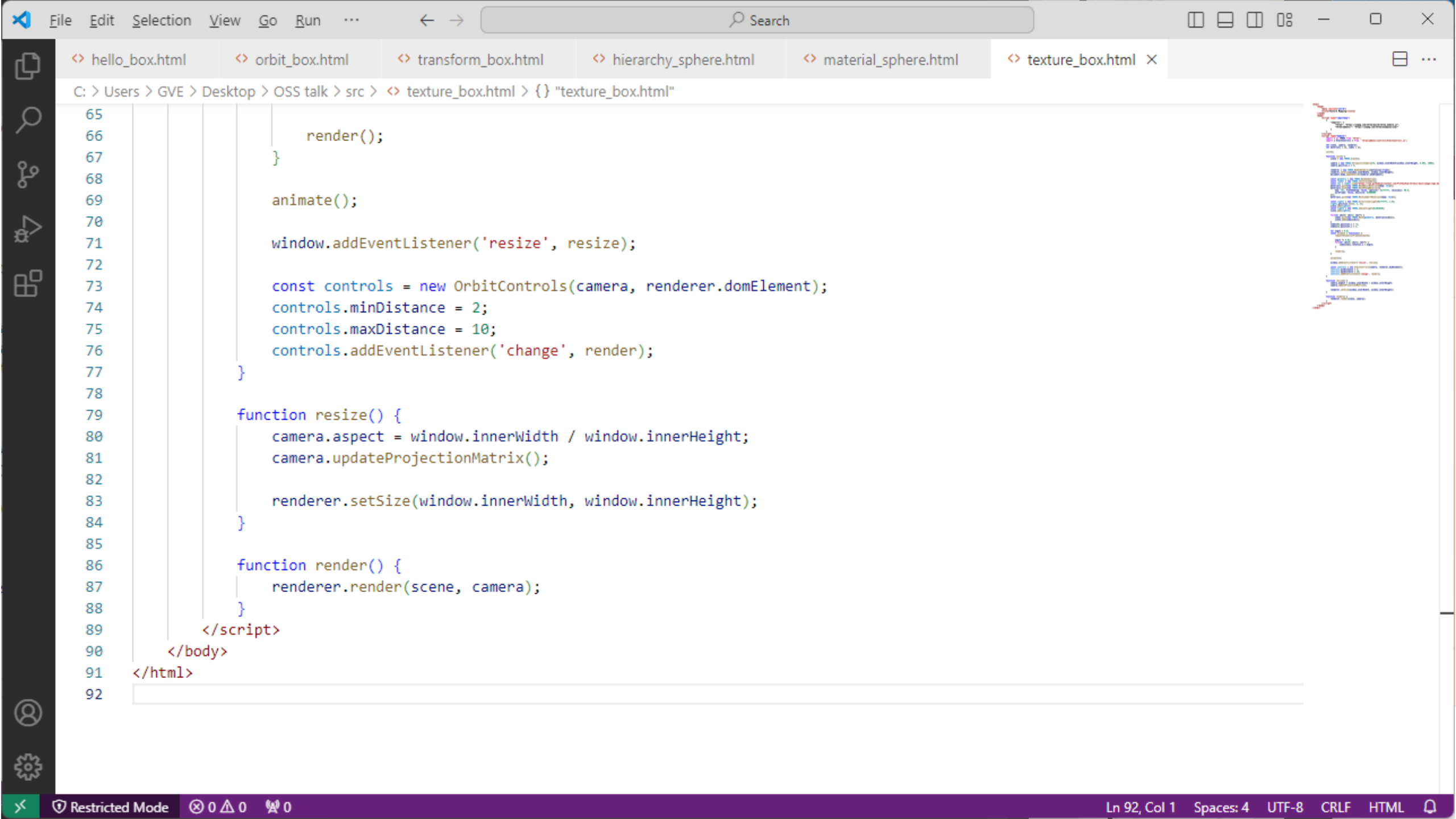
C: > Users > GVE > Desktop > OSS talk > src > texture_box.html > {} "texture_box.html"

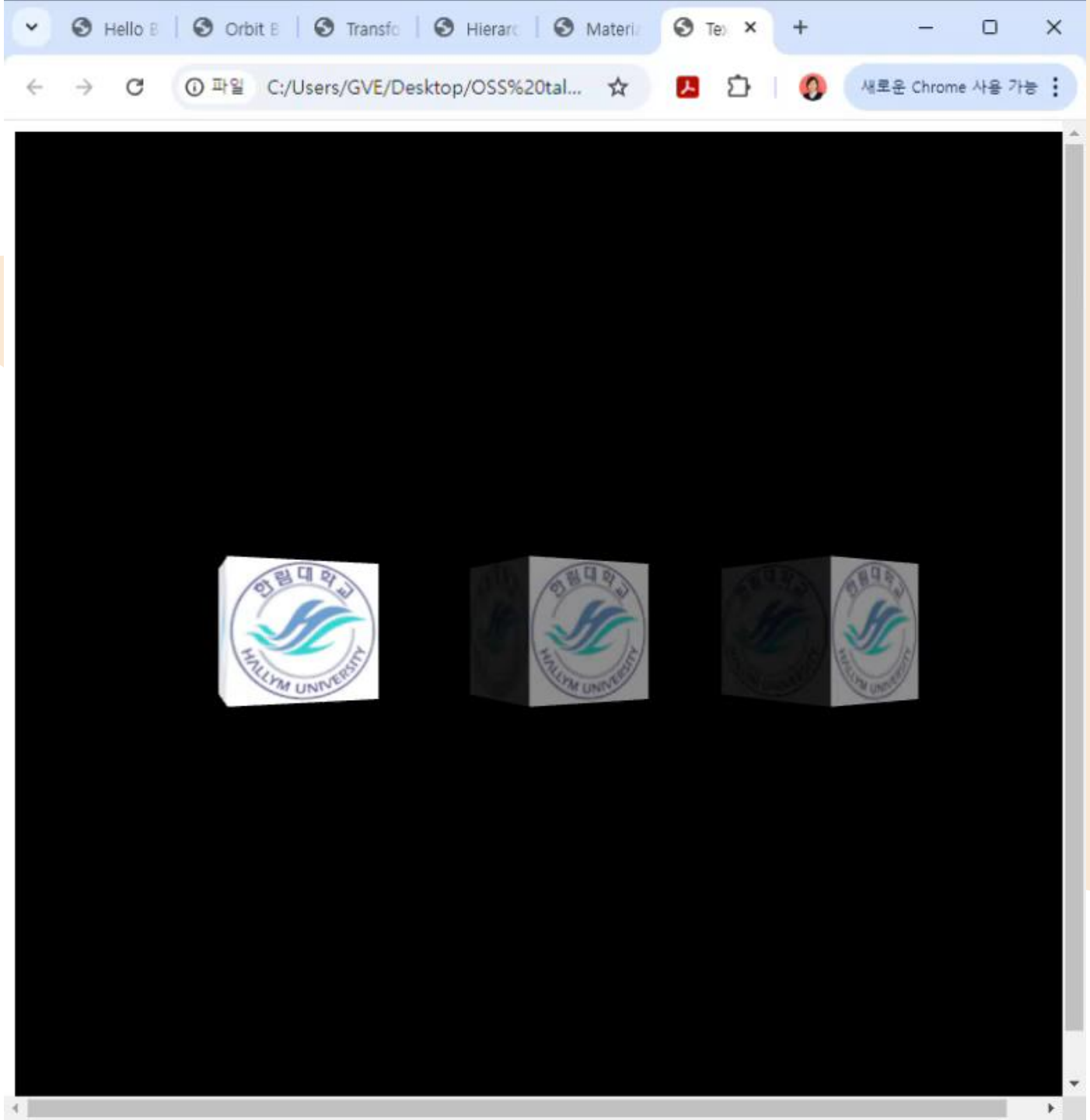
https://github.com/ProfSunKim/threejs

```
<html>
    <head>
        <meta charset="utf-8">
        <title>Texture Mapping</title>
    </head>
    <body>
        <script type="importmap">
            {
                "imports": {
                    "three": "https://unpkg.com/three/build/three.module.js",
                    "three/addons/": "https://unpkg.com/three/examples/jsm/"
                }
            }
        </script>
        <script type="module">
            import * as THREE from 'three';
            import { OrbitControls } from  'three/addons/controls/OrbitControls.js';

            let scene, camera, renderer;
            let materials = [], cubes = [];

            init();

            function init() {
                scene = new THREE.Scene();

                camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight, 0.001, 1000);
                camera.position.z = 5;

                renderer = new THREE.WebGLRenderer({antialias:true});
                renderer.setSize(window.innerWidth, window.innerHeight);
                document.body.appendChild(renderer.domElement);
```

Restricted Mode ⊗ 0 ⚠ 0 0 Ln 92, Col 1 Spaces: 4 UTF-8 CRLF HTML

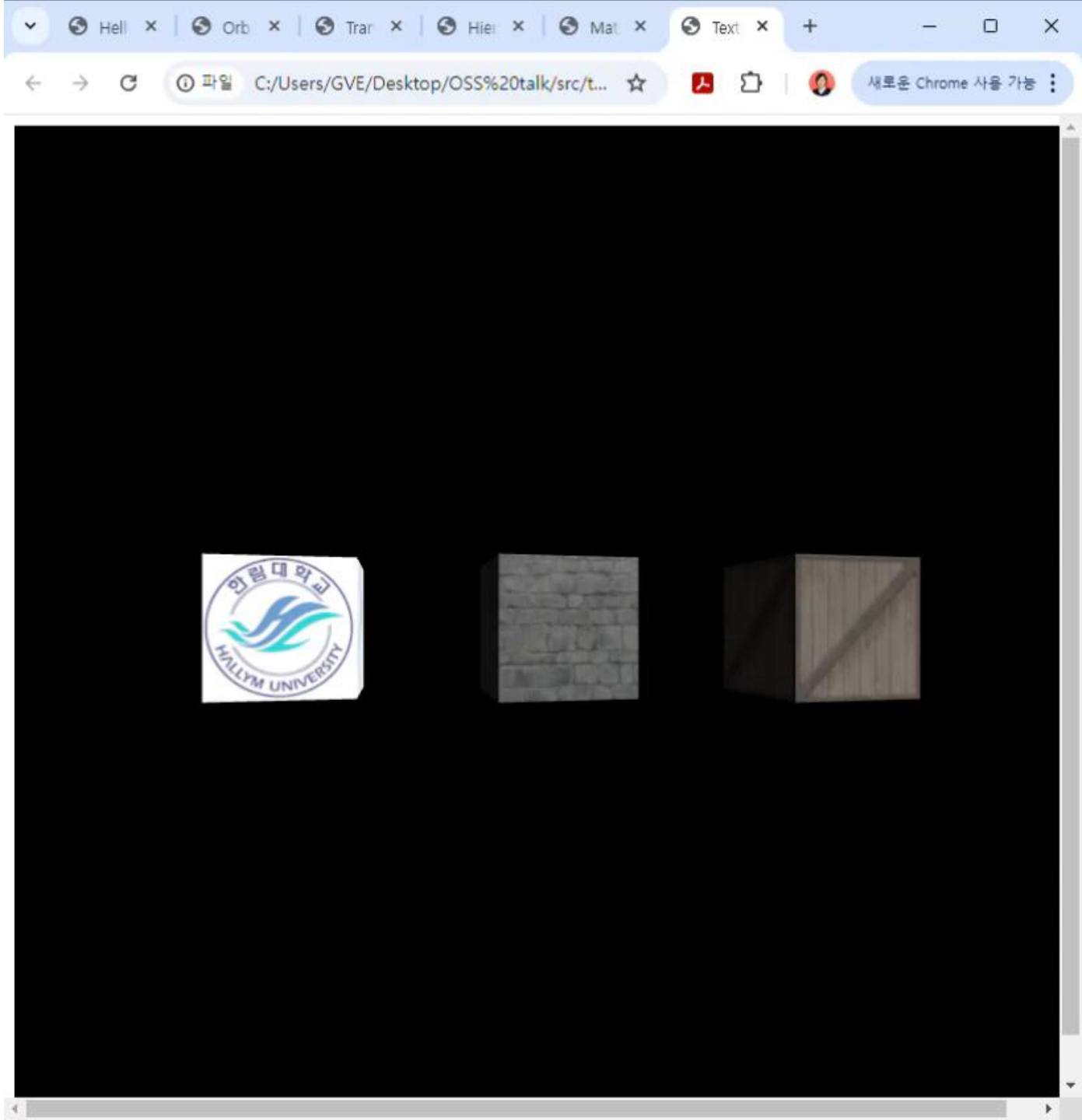
```
33
34          const geometry = new THREE.BoxGeometry();
35          const loader = new THREE.TextureLoader();
36          const tex = loader.load("https://raw.githubusercontent.com/ProfSunKim/threejs/main/image/logo.bmp");
37          materials.push(new THREE.MeshBasicMaterial({map: tex}));
38          materials.push(new THREE.MeshPhongMaterial({
39              map: tex, flatShading: false, specular: 0xffffff, shininess: 90.0,
40              wireframe: false, emissive: 0x000000
41          }));
42          materials.push(new THREE.MeshLambertMaterial({map: tex}));
43
44          const light1 = new THREE.DirectionalLight(0xffffff, 1.0);
45          light1.position.set(2, 2, 2);
46          scene.add(light1);
47          const light2 = new THREE.AmbientLight(0x303030);
48          scene.add(light2);
49
50          for(var idx=0; idx<3; idx++) {
51              cubes.push(new THREE.Mesh(geometry, materials[idx]));
52              scene.add(cubes[idx]);
53          }
54          cubes[0].position.x = -2;
55          cubes[2].position.x = 2;
56
57          var angle = 0.0;
58          const animate = function() {
59              requestAnimationFrame(animate);
60
61              angle += 0.01;
62              for(var idx=0; idx<3; idx++) {
63                  cubes[idx].rotation.y = angle;
64              }
```

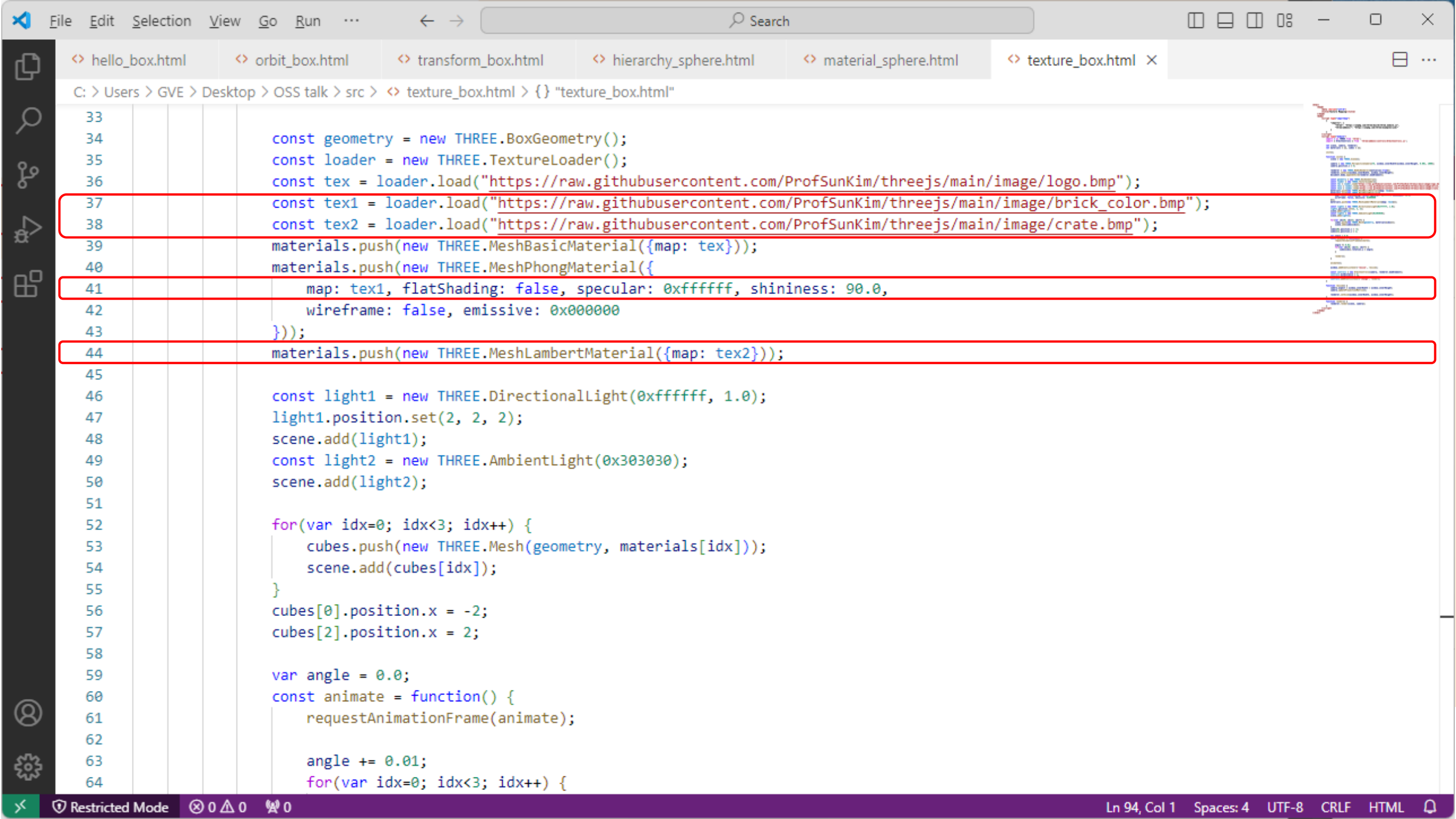

File Edit Selection View Go Run
Search
hello_box.html
orbit_box.html
transform_box.html
hierarchy_sphere.html
material_sphere.html
texture_box.html
C: > Users > GVE > Desktop > OSS talk > src > texture_box.html > {} "texture_box.html"
65 render();
66 }
67
68 animate();
69
70 window.addEventListener('resize', resize);
71
72 const controls = new OrbitControls(camera, renderer.domElement);
73 controls.minDistance = 2;
74 controls.maxDistance = 10;
75 controls.addEventListener('change', render);
76 }
77
78 function resize() {
79 camera.aspect = window.innerWidth / window.innerHeight;
80 camera.updateProjectionMatrix();
81
82 renderer.setSize(window.innerWidth, window.innerHeight);
83 }
84
85 function render() {
86 renderer.render(scene, camera);
87 }
88 </script>
89 </body>
90 </html>
91
92
Restricted Mode
0 0
0
Ln 92, Col 1
Spaces: 4
UTF-8
CRLF
HTML


Hello B
Orbit B
Transfo
Hierarc
Materi
Tex
파일
C:/Users/GVE/Desktop/OSS%20tal...
새로운 Chrome 사용 가능
한림대학교
HALLYM UNIVERSITY

연습 문제 (6)

- 세 Box에 서로 다른 이미지로 텍스처 매핑 시키시오.




File Edit Selection View Go Run
Search
hello_box.html
orbit_box.html
transform_box.html
hierarchy_sphere.html
material_sphere.html
texture_box.html
C: > Users > GVE > Desktop > OSS talk > src > texture_box.html > {} "texture_box.html"
const geometry = new THREE.BoxGeometry();
const loader = new THREE.TextureLoader();
const tex = loader.load("https://raw.githubusercontent.com/ProfSunKim/threejs/main/image/logo.bmp");
const tex1 = loader.load("https://raw.githubusercontent.com/ProfSunKim/threejs/main/image/brick_color.bmp");
const tex2 = loader.load("https://raw.githubusercontent.com/ProfSunKim/threejs/main/image/crate.bmp");
materials.push(new THREE.MeshBasicMaterial({map: tex}));
materials.push(new THREE.MeshPhongMaterial({
map: tex1, flatShading: false, specular: 0xffffff, shininess: 90.0,
wireframe: false, emissive: 0x000000
}));
materials.push(new THREE.MeshLambertMaterial({map: tex2}));
const light1 = new THREE.DirectionalLight(0xffffff, 1.0);
light1.position.set(2, 2, 2);
scene.add(light1);
const light2 = new THREE.AmbientLight(0x303030);
scene.add(light2);
for(var idx=0; idx<3; idx++) {
cubes.push(new THREE.Mesh(geometry, materials[idx]));
scene.add(cubes[idx]);
}
cubes[0].position.x = -2;
cubes[2].position.x = 2;
var angle = 0.0;
const animate = function() {
requestAnimationFrame(animate);
angle += 0.01;
for(var idx=0; idx<3; idx++) {
Restricted Mode
Ln 94, Col 1
Spaces: 4
UTF-8
CRLF
HTML

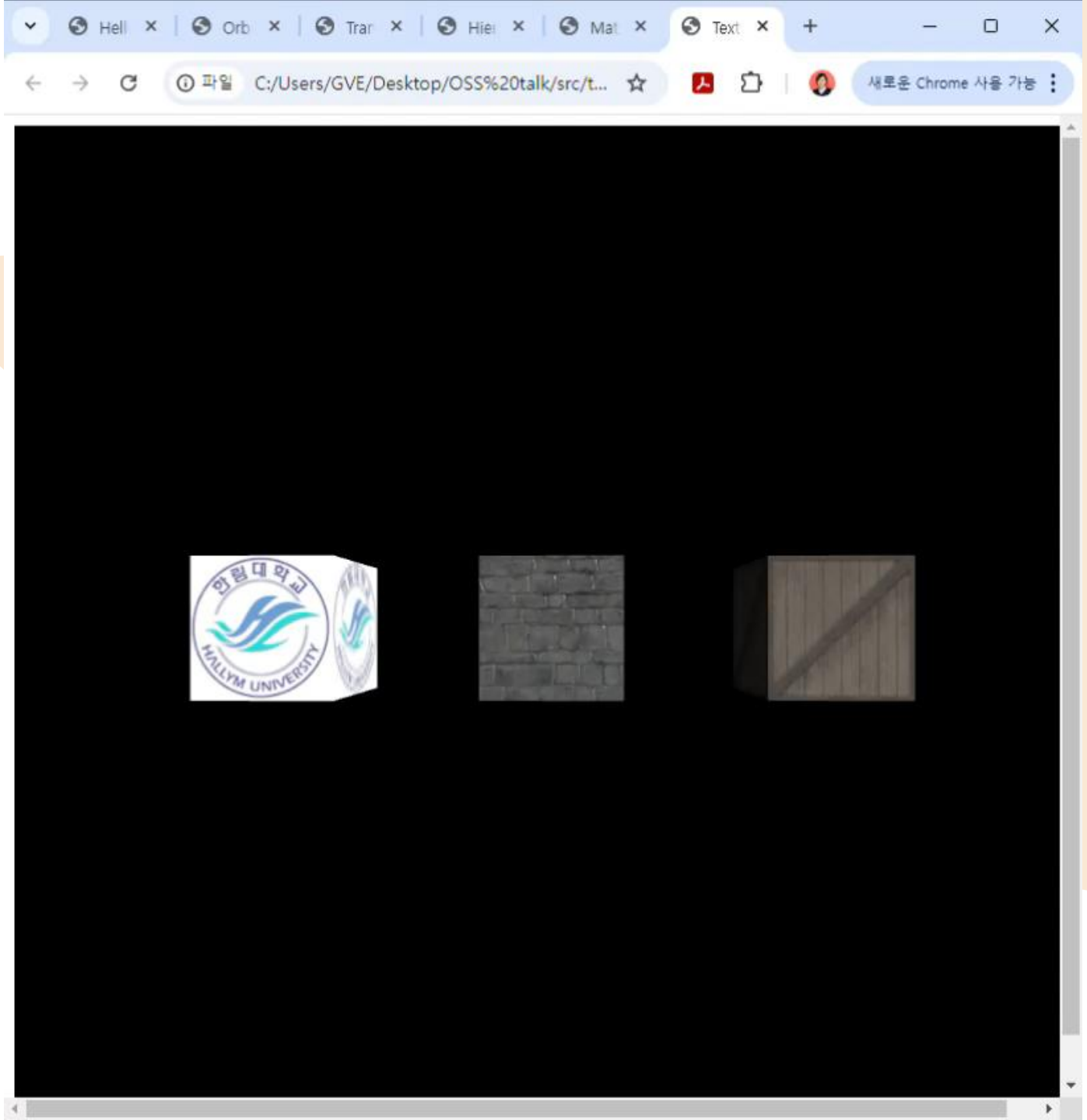
범프 매핑

File Edit Selection View Go Run ··· Search

hello_box.html | orbit_box.html | transform_box.html | hierarchy_sphere.html | material_sphere.html | texture_box.html

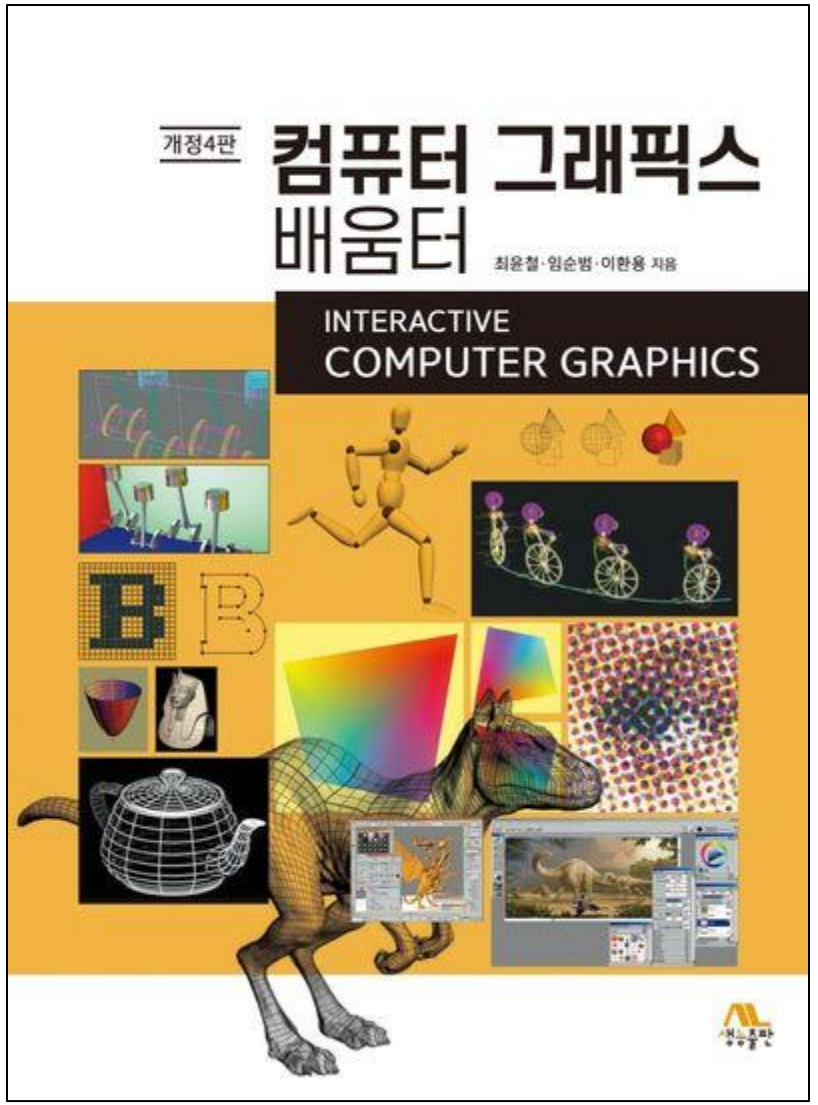
C: > Users > GVE > Desktop > OSS talk > src > texture_box.html > {} "texture_box.html"

```
33
34              const geometry = new THREE.BoxGeometry();
35              const loader = new THREE.TextureLoader();
36              const tex = loader.load("https://raw.githubusercontent.com/ProfSunKim/threejs/main/image/logo.bmp");
37              const tex0 = loader.load("https://raw.githubusercontent.com/ProfSunKim/threejs/main/image/brick_normal.bmp");
38              const tex1 = loader.load("https://raw.githubusercontent.com/ProfSunKim/threejs/main/image/brick_color.bmp");
39              const tex2 = loader.load("https://raw.githubusercontent.com/ProfSunKim/threejs/main/image/crate.bmp");
40              materials.push(new THREE.MeshBasicMaterial({map: tex}));
41              materials.push(new THREE.MeshPhongMaterial({
42                  map: tex1, bumpMap: tex0, bumScale: 0.02,
43                  flatShading: false, specular: 0xffffff, shininess: 90.0,
44                  wireframe: false, emissive: 0x000000
45              }));
46              materials.push(new THREE.MeshLambertMaterial({map: tex2}));
47
48              const light1 = new THREE.DirectionalLight(0xffffff, 1.0);
49              light1.position.set(2, 2, 2);
50              scene.add(light1);
51              const light2 = new THREE.AmbientLight(0x303030);
52              scene.add(light2);
53
54              for(var idx=0; idx<3; idx++) {
55                  cubes.push(new THREE.Mesh(geometry, materials[idx]));
56                  scene.add(cubes[idx]);
```


Hell
Orb
Trar
Hier
Mat
Text
파일
C:/Users/GVE/Desktop/OSS%20talk/src/t...
새로운 Chrome 사용 가능
한림대학교
HALLYM UNIVERSITY

참고 교재

- **컴퓨터 그래픽스 배움터** (개정 4판)
  - 저자: 최윤철, 임순범, 이환용
  - 생능출판사
  - 2022년 3월



three.js examples

threejs.org/examples/#webgl_animation_keyframes

새로운 Chrome 사용 가능

three.js docs examples

webgl

animation / keyframes

animation / skinning / blending

animation / skinning / additive / blending

60 FPS (1-61)

three.js webgl - animation - keyframes

Model: Littlest Tokyo by Glen Fox, CC Attribution.

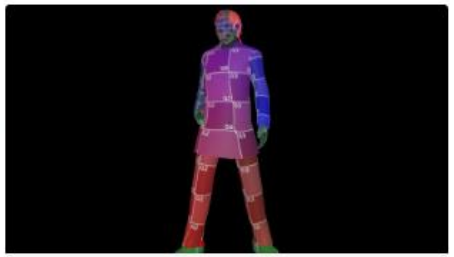
three.js examples

threejs.org/examples/#webgl_loader_obj_mtl

새로운 Chrome 사용 가능

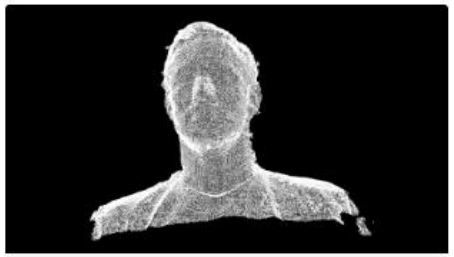
three.js docs examples

loader / nrrd



loader / obj

loader / obj / mtl



loader / pcd

three.js - OBJLoader + MTLLoader

Files
master
Go to file t
.github
build
docs
editor
examples
files
fonts
ies
jsm
luts
models
screenshots
sounds
textures
css2d_label.html
css3d_molecules.html
css3d_orthographic.html
css3d_periodictable.html
css3d_sandbox.html
css3d_sprites.html
css3d_youtube.html

three.js / examples / webgl_loader_obj_mtl.html

WestLangley Clean up non-breaking spaces (#29080) 67bdee6 · 2 months ago History

Code Blame 125 lines (82 loc) · 2.71 KB Code 55% faster with GitHub Copilot Raw

```
<!DOCTYPE html>
<html lang="en">
	<head>
		<title>three.js webgl - OBJLoader + MTLLoader</title>
		<meta charset="utf-8">
		<meta name="viewport" content="width=device-width, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
		<link type="text/css" rel="stylesheet" href="main.css">
	</head>

	<body>
		<div id="info">
		<a href="https://threejs.org" target="_blank" rel="noopener">three.js</a> - OBJLoader + MTLLoader
		</div>

		<script type="importmap">
			{
				"imports": {
					"three": "../build/three.module.js",
					"three/addons/": "./jsm/"
				}
			}
		</script>

		<script type="module">

			import * as THREE from 'three';

			import { MTLLoader } from 'three/addons/loaders/MTLLoader.js';
			import { OBJLoader } from 'three/addons/loaders/OBJLoader.js';
			import { OrbitControls } from 'three/addons/controls/OrbitControls.js';

			let camera, scene, renderer;
```

File Edit Selection View Go Run ··· Search

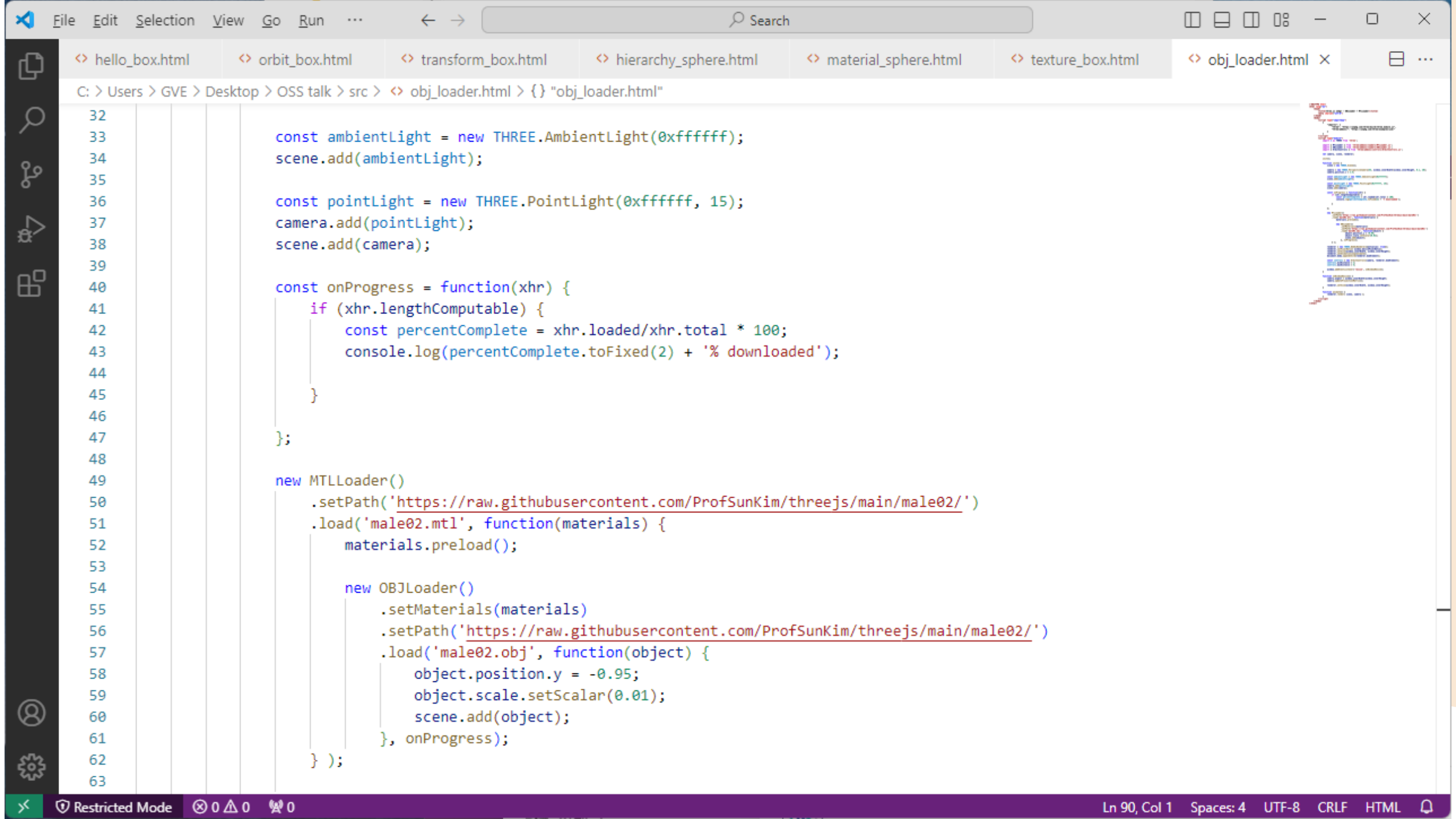
hello_box.html | orbit_box.html | transform_box.html | hierarchy_sphere.html | material_sphere.html | texture_box.html | obj_loader.html

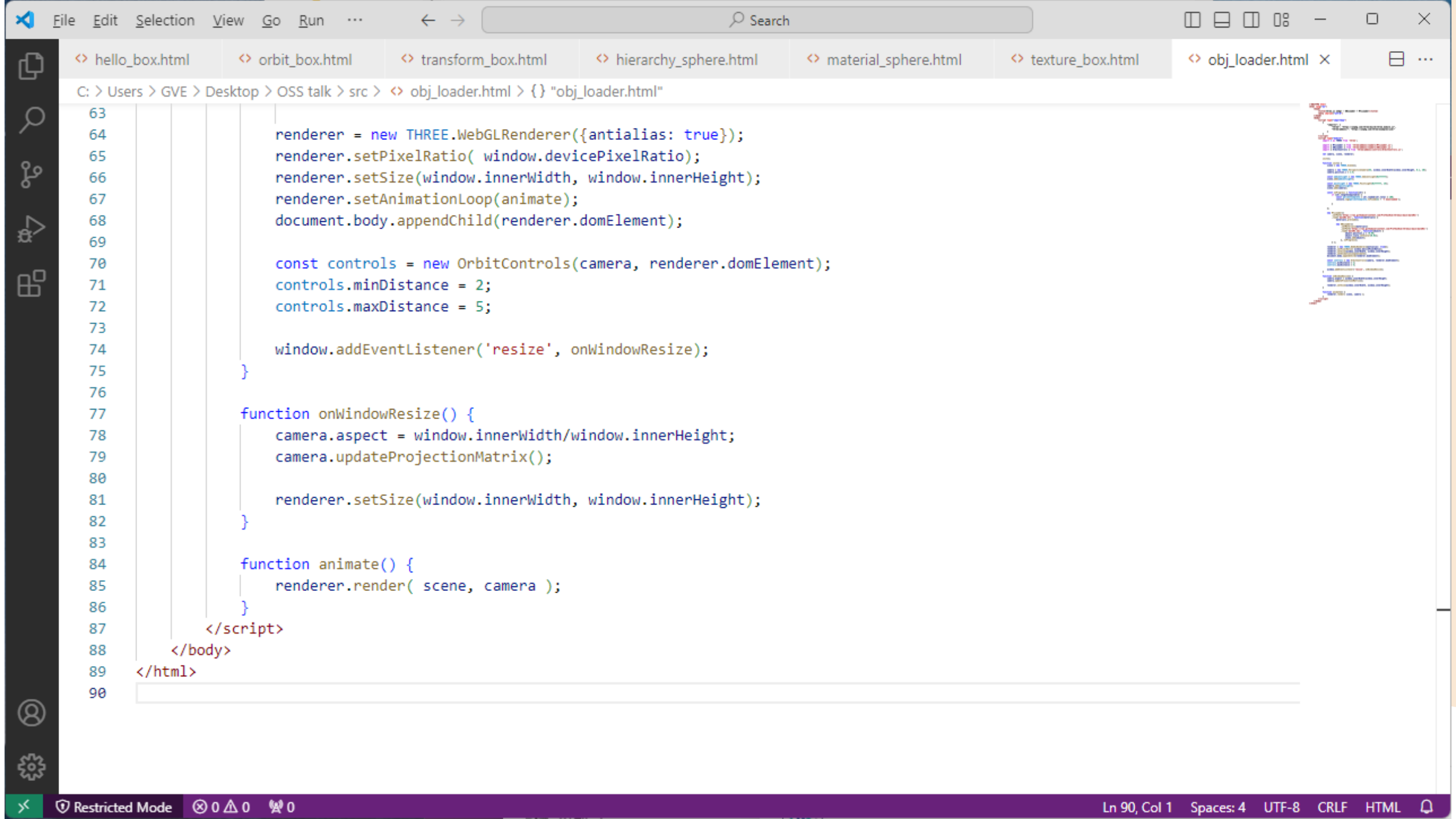
C: > Users > GVE > Desktop > OSS talk > src > obj_loader.html > {} "obj_loader.html"

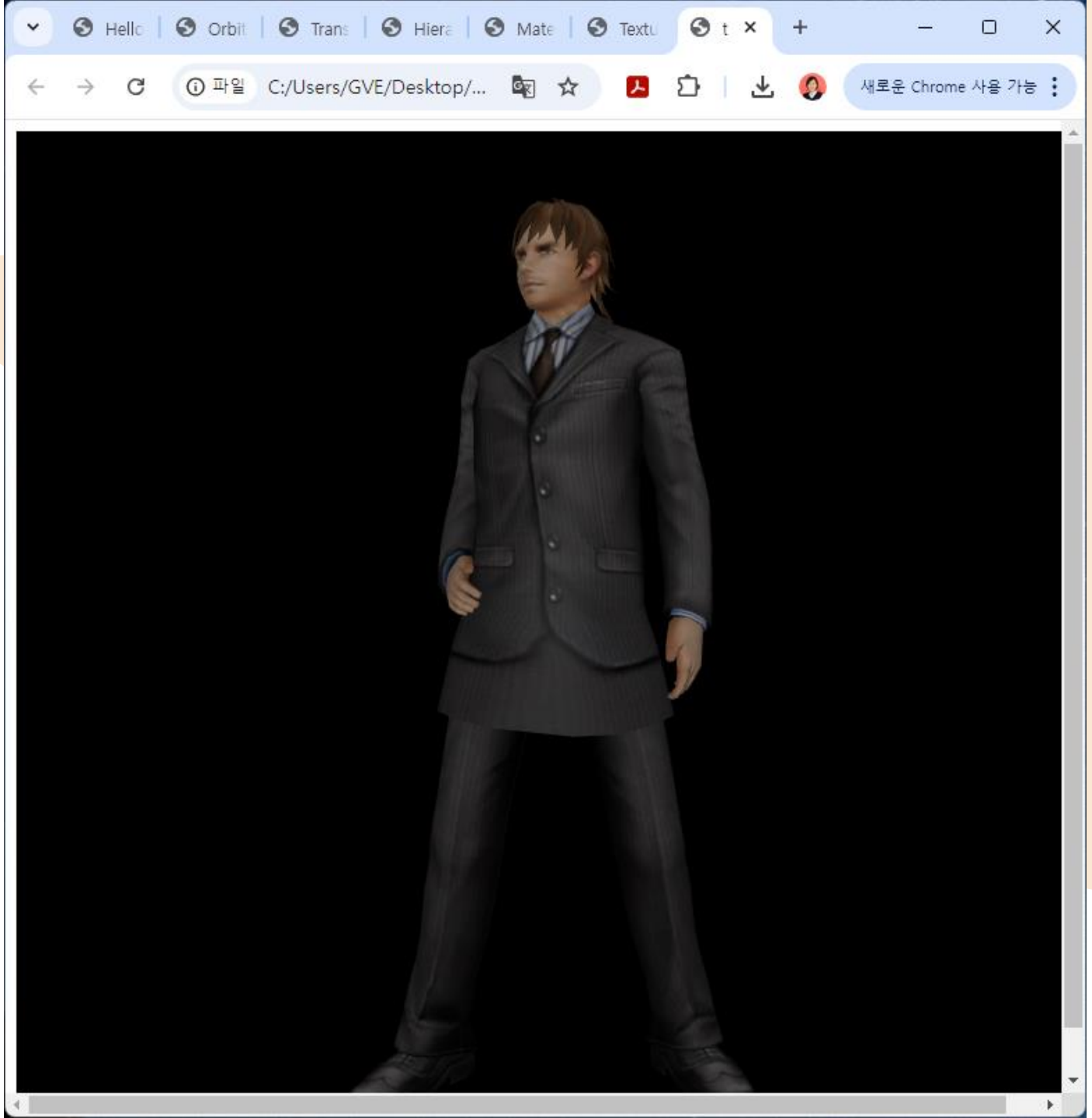
https://github.com/ProfSunKim/threejs

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>three.js webgl - OBJLoader + MTLLoader</title>
        <meta charset="utf-8">
    </head>
    <body>
        <script type="importmap">
            {
                "imports": {
                    "three": "https://unpkg.com/three/build/three.module.js",
                    "three/addons/": "https://unpkg.com/three/examples/jsm/"
                }
            }
        </script>
        <script type="module">
            import * as THREE from 'three';

            import { MTLLoader } from 'three/addons/loaders/MTLLoader.js';
            import { OBJLoader } from 'three/addons/loaders/OBJLoader.js';
            import { OrbitControls } from 'three/addons/controls/OrbitControls.js';

            let camera, scene, renderer;

            init();

            function init() {
                scene = new THREE.Scene();

                camera = new THREE.PerspectiveCamera(45, window.innerWidth/window.innerHeight, 0.1, 20);
                camera.position.z = 2.5;

```

Restricted Mode ⊗ 0 ⚠ 0 0 Ln 90, Col 1 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run
Search
hello_box.html
orbit_box.html
transform_box.html
hierarchy_sphere.html
material_sphere.html
texture_box.html
obj_loader.html
C: > Users > GVE > Desktop > OSS talk > src > obj_loader.html > {} "obj_loader.html"
32
33 const ambientLight = new THREE.AmbientLight(0xffffff);
34 scene.add(ambientLight);
35
36 const pointLight = new THREE.PointLight(0xffffff, 15);
37 camera.add(pointLight);
38 scene.add(camera);
39
40 const onProgress = function(xhr) {
41 if (xhr.lengthComputable) {
42 const percentComplete = xhr.loaded/xhr.total * 100;
43 console.log(percentComplete.toFixed(2) + '% downloaded');
44
45 }
46
47 };
48
49 new MTLLoader()
50 .setPath('https://raw.githubusercontent.com/ProfSunKim/threejs/main/male02/')
51 .load('male02.mtl', function(materials) {
52 materials.preload();
53
54 new OBJLoader()
55 .setMaterials(materials)
56 .setPath('https://raw.githubusercontent.com/ProfSunKim/threejs/main/male02/')
57 .load('male02.obj', function(object) {
58 object.position.y = -0.95;
59 object.scale.setScalar(0.01);
60 scene.add(object);
61 }, onProgress);
62 });
63
Restricted Mode
0 0
0
Ln 90, Col 1
Spaces: 4
UTF-8
CRLF
HTML


File Edit Selection View Go Run
Search
hello_box.html
orbit_box.html
transform_box.html
hierarchy_sphere.html
material_sphere.html
texture_box.html
obj_loader.html
C: > Users > GVE > Desktop > OSS talk > src > obj_loader.html > {} "obj_loader.html"
63
64 renderer = new THREE.WebGLRenderer({antialias: true});
65 renderer.setPixelRatio(window.devicePixelRatio);
66 renderer.setSize(window.innerWidth, window.innerHeight);
67 renderer.setAnimationLoop(animate);
68 document.body.appendChild(renderer.domElement);
69
70 const controls = new OrbitControls(camera, renderer.domElement);
71 controls.minDistance = 2;
72 controls.maxDistance = 5;
73
74 window.addEventListener('resize', onWindowResize);
75 }
76
77 function onWindowResize() {
78 camera.aspect = window.innerWidth/window.innerHeight;
79 camera.updateProjectionMatrix();
80
81 renderer.setSize(window.innerWidth, window.innerHeight);
82 }
83
84 function animate() {
85 renderer.render(scene, camera);
86 }
87 </script>
88 </body>
89 </html>
90
Restricted Mode
0 0
0
Ln 90, Col 1
Spaces: 4
UTF-8
CRLF
HTML


Hello
Orbit
Trans
Hiera
Mate
Textu
t
파일
C:/Users/GVE/Desktop/...
새로운 Chrome 사용 가능

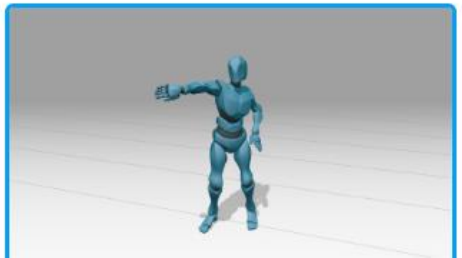
three.js examples

threejs.org/examples/#webgl_loader_fbx

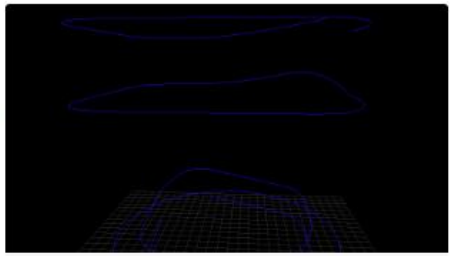
새로운 Chrome 사용 가능

three.js docs examples

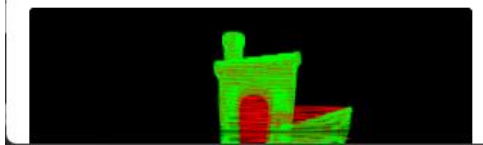
loader / draco



loader / fbx



loader / fbx / nurbs



60 FPS (22-60)

three.js - FBXLoader

Character and animation from Mixamo

Controls

asset Samba Dancing

Files
master
Go to file t
.github
build
docs
editor
examples
files
fonts
ies
jsm
luts
models
screenshots
sounds
textures
css2d_label.html
css3d_molecules.html
css3d_orthographic.html
css3d_periodictable.html
css3d_sandbox.html
css3d_sprites.html
css3d_youtube.html

three.js / examples / webgl_loader_fbx.html

3 people FBXLoader: Support more texture formats. (#28515) 6abb47c · 5 months ago History

Code Blame 226 lines (146 loc) · 5.32 KB Code 55% faster with GitHub Copilot Raw

```
<!DOCTYPE html>
<html lang="en">
	<head>
		<title>three.js webgl - FBX loader</title>
		<meta charset="utf-8">
		<meta name="viewport" content="width=device-width, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
		<link type="text/css" rel="stylesheet" href="main.css">
	</head>

	<body>
		<div id="info">
			<a href="https://threejs.org" target="_blank" rel="noopener">three.js</a> - FBXLoader<br />
			Character and animation from <a href="https://www.mixamo.com/" target="_blank" rel="noopener">Mixamo</a>
		</div>

		<script type="importmap">
			{
				"imports": {
					"three": "../build/three.module.js",
					"three/addons/": "./jsm/"
				}
			}
		</script>

		<script type="module">

			import * as THREE from 'three';

			import Stats from 'three/addons/libs/stats.module.js';

			import { OrbitControls } from 'three/addons/controls/OrbitControls.js';
			import { FBXLoader } from 'three/addons/loaders/FBXLoader.js';
```

File Edit Selection View Go Run ··· Search

orbit_box.html | transform_box.html | hierarchy_sphere.html | material_sphere.html | texture_box.html | obj_loader.html | fbx_loader.html

C: > Users > GVE > Desktop > OSS talk > src > fbx_loader.html > {} "fbx_loader.html"

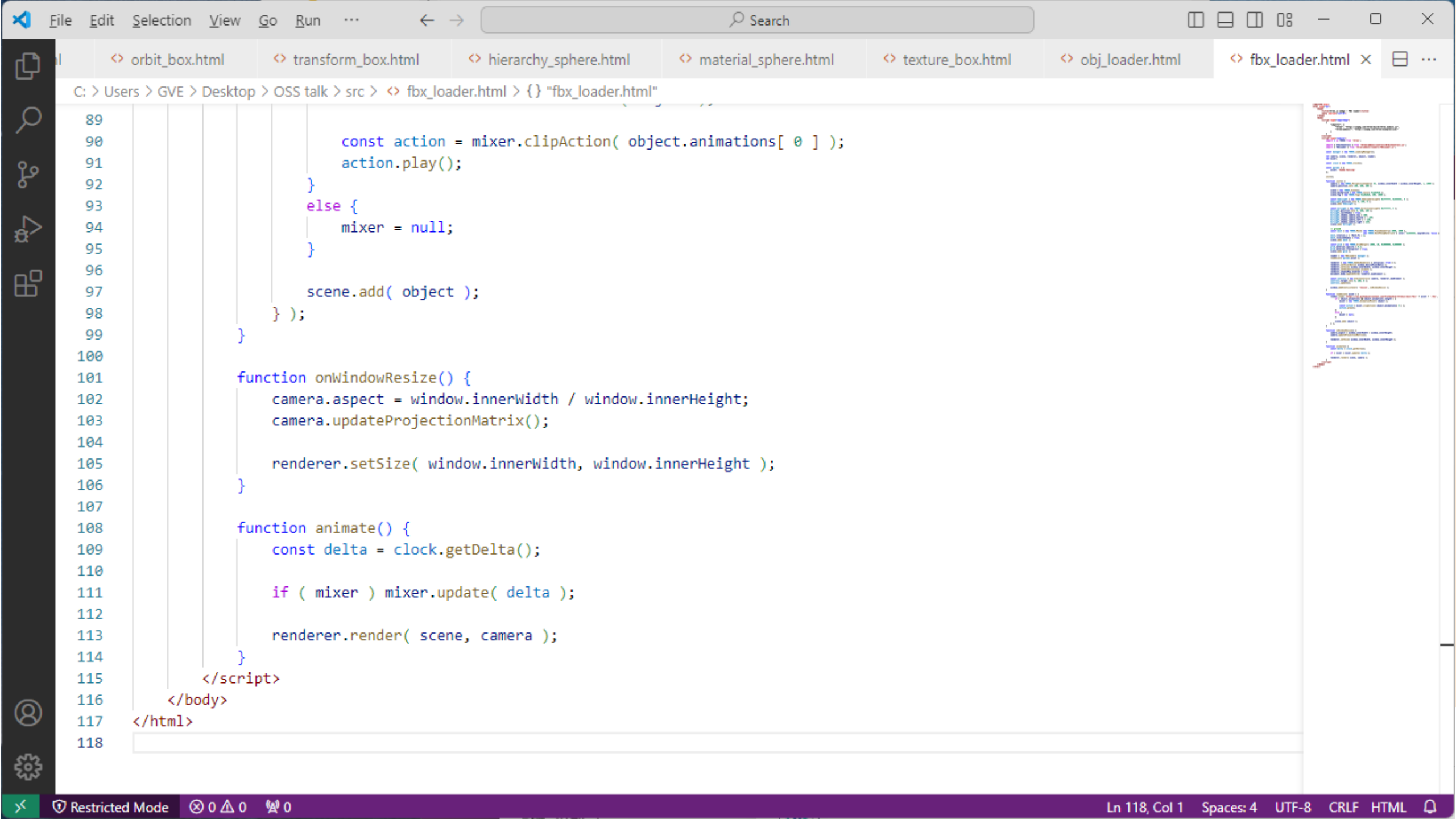
https://github.com/ProfSunKim/threejs

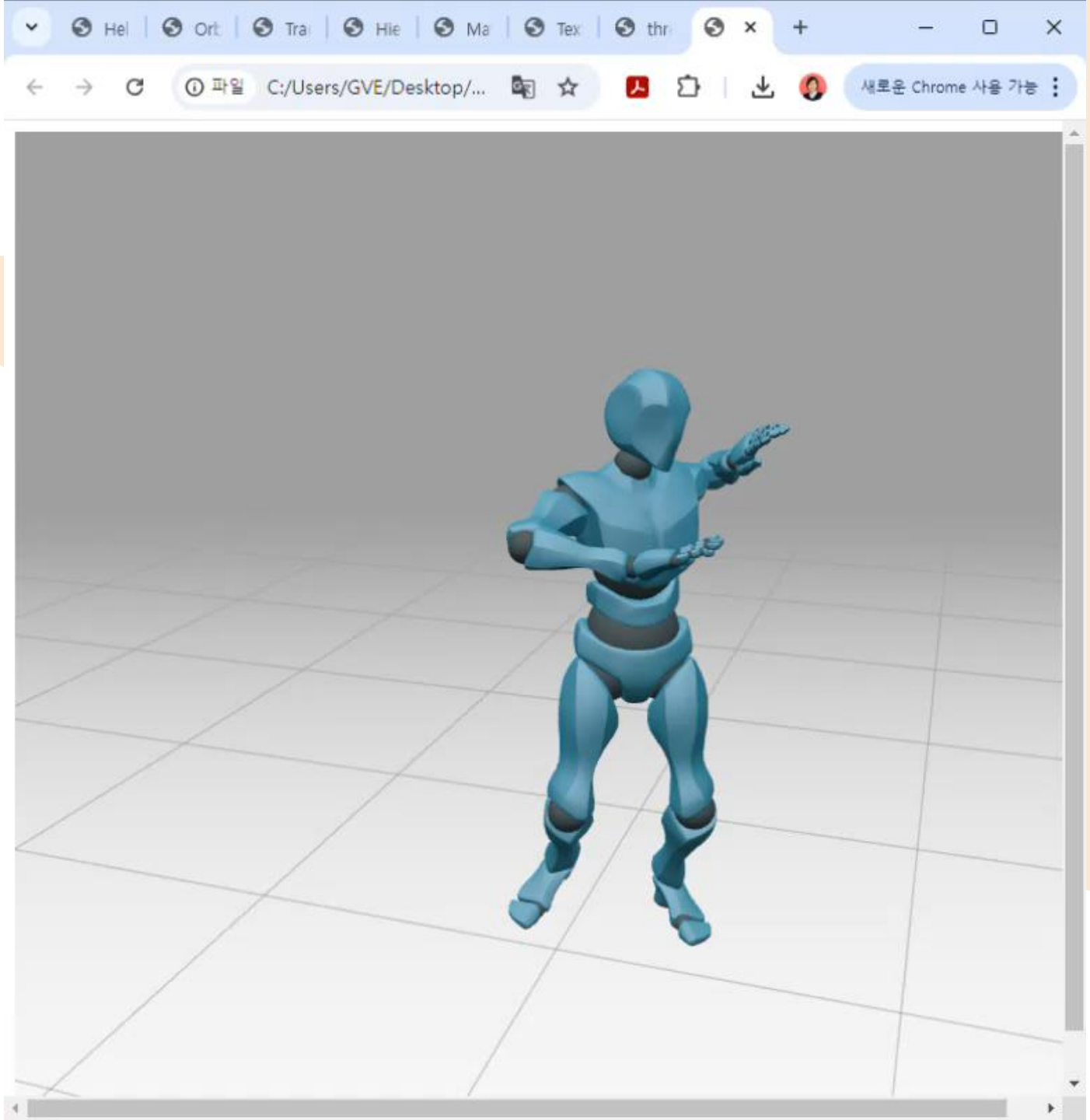
```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>three.js webgl - FBX loader</title>
        <meta charset="utf-8">
    </head>
    <body>
        <script type="importmap">
            {
                "imports": {
                    "three": "https://unpkg.com/three/build/three.module.js",
                    "three/addons/": "https://unpkg.com/three/examples/jsm/"
                }
            }
        </script>
        <script type="module">
            import * as THREE from 'three';

            import { OrbitControls } from 'three/addons/controls/OrbitControls.js';
            import { FBXLoader } from 'three/addons/loaders/FBXLoader.js';

            const manager = new THREE.LoadingManager();

            let camera, scene, renderer, object, loader;
            let mixer;

            const clock = new THREE.Clock();

            const params = {
                asset: 'Samba Dancing'
            };

```

Restricted Mode ⊗ 0 ⚠ 0 0 Ln 118, Col 1 Spaces: 4 UTF-8 CRLF HTML

```
32
33          init();
34
35          function init() {
36              camera = new THREE.PerspectiveCamera( 45, window.innerWidth / window.innerHeight, 1, 2000 );
37              camera.position.set( 100, 200, 300 );
38
39              scene = new THREE.Scene();
40              scene.background = new THREE.Color( 0xa0a0a0 );
41              scene.fog = new THREE.Fog( 0xa0a0a0, 200, 1000 );
42
43              const hemiLight = new THREE.HemisphereLight( 0xffffff, 0x444444, 5 );
44              hemiLight.position.set( 0, 200, 0 );
45              scene.add( hemiLight );
46
47              const dirLight = new THREE.DirectionalLight( 0xffffff, 5 );
48              dirLight.position.set( 0, 200, 100 );
49              dirLight.castShadow = true;
50              dirLight.shadow.camera.top = 180;
51              dirLight.shadow.camera.bottom = - 100;
52              dirLight.shadow.camera.left = - 120;
53              dirLight.shadow.camera.right = 120;
54              scene.add( dirLight );
55
56              // ground
57              const mesh = new THREE.Mesh( new THREE.PlaneGeometry( 2000, 2000 ),
58                                           new THREE.MeshPhongMaterial( { color: 0x999999, depthWrite: false } ) );
59              mesh.rotation.x = -Math.PI / 2;
60              mesh.receiveShadow = true;
61              scene.add( mesh );
62
63              const grid = new THREE.GridHelper( 2000, 20, 0x000000, 0x000000 );
```

```
63              const grid = new THREE.GridHelper( 2000, 20, 0x000000, 0x000000 );
64              grid.material.opacity = 0.2;
65              grid.material.transparent = true;
66              scene.add( grid );
67
68              loader = new FBXLoader( manager );
69              loadAsset( params.asset );
70
71              renderer = new THREE.WebGLRenderer( { antialias: true } );
72              renderer.setPixelRatio( window.devicePixelRatio );
73              renderer.setSize( window.innerWidth, window.innerHeight );
74              renderer.setAnimationLoop( animate );
75              renderer.shadowMap.enabled = true;
76              document.body.appendChild( renderer.domElement );
77
78              const controls = new OrbitControls( camera, renderer.domElement );
79              controls.target.set( 0, 100, 0 );
80              controls.update();
81
82              window.addEventListener( 'resize', onWindowResize );
83          }
84
85          function loadAsset( asset ) {
86              loader.load( 'https://raw.githubusercontent.com/ProfSunKim/threejs/main/fbx/' + asset + '.fbx', function (object) {
87                  if ( object.animations && object.animations.length ) {
88                      mixer = new THREE.AnimationMixer( object );
89
90                      const action = mixer.clipAction( object.animations[ 0 ] );
91                      action.play();
92                  }
93                  else {
94                      mixer = null;
```


File Edit Selection View Go Run
Search
orbit_box.html
transform_box.html
hierarchy_sphere.html
material_sphere.html
texture_box.html
obj_loader.html
fbx_loader.html
C: > Users > GVE > Desktop > OSS talk > src > fbx_loader.html > {} "fbx_loader.html"
89
90 const action = mixer.clipAction(object.animations[0]);
91 action.play();
92 }
93 else {
94 mixer = null;
95 }
96
97 scene.add(object);
98 });
99 }
100
101 function onWindowResize() {
102 camera.aspect = window.innerWidth / window.innerHeight;
103 camera.updateProjectionMatrix();
104
105 renderer.setSize(window.innerWidth, window.innerHeight);
106 }
107
108 function animate() {
109 const delta = clock.getDelta();
110
111 if (mixer) mixer.update(delta);
112
113 renderer.render(scene, camera);
114 }
115 </script>
116 </body>
117 </html>
118
Restricted Mode
0 0
0
Ln 118, Col 1
Spaces: 4
UTF-8
CRLF
HTML


파일
C:/Users/GVE/Desktop/...
새로운 Chrome 사용 가능