

<코드 및 동작 설명> - 정보융합학부 2020204097 윤가영

```
1 import * as THREE from 'three';
2 import {OrbitControls} from "../three/examples/jsm/controls/OrbitControls.js";
3
4
5
6 // Renderer Setting
7 const renderer = new THREE.WebGLRenderer();
8 renderer.setSize( window.innerWidth, window.innerHeight );
9 renderer.setViewport( 0, 0, window.innerWidth, window.innerHeight );
10
11 const container = document.getElementById( 'myContainer' );
12 container.appendChild( renderer.domElement );
13
14
15 // Camera Setting
16 const camera = new THREE.PerspectiveCamera( 90, window.innerWidth / window.innerHeight, 1, 1000); // Fov 값은 90
17 camera.position.set( 0, 0, 20 );
18 camera.up.set( 0, 1, 0 ); // 업벡터는 y = 1
19 camera.lookAt( 0, 0, 0 ); // 어디서든지 원점을 보기
20
21 const controls = new OrbitControls( camera, renderer.domElement );
22
```

결과물이 나오는 랜더러의 사이즈는 윈도우의 창 크기와 동일하게 했다.

카메라의 Fov값은 90, 비율은 동일하게 윈도우의 창 크기, near plane은 1, far plane은 1000으로 설정했다. 카메라는 z축의 20에 위치하며, y=1을 업벡터로 두고 있다. lookAt의 축을 모두 0으로 두어 카메라가 어디서든지 움직여도 원점을 볼 수 있도록 하였다.

또한 카메라가 움직일 수 있게끔, 'OrbitControls' 함수를 사용했다.

```
23
24 // Material & line Setting
25 const material = new THREE.MeshBasicMaterial( { color: 0xFFFFFF, wireframe: true } );
26 const geometry = new THREE.BoxGeometry( 10, 10, 10 );
27 const Cube = new THREE.Mesh( geometry, material );
28
29
30 // Resize Setting
31 function WindowResize(){
32     camera.aspect = window.innerWidth / window.innerHeight;
33     camera.updateProjectionMatrix();
34     renderer.setSize( window.innerWidth, window.innerHeight );
35 }
36 window.addEventListener('resize', WindowResize);
37
```

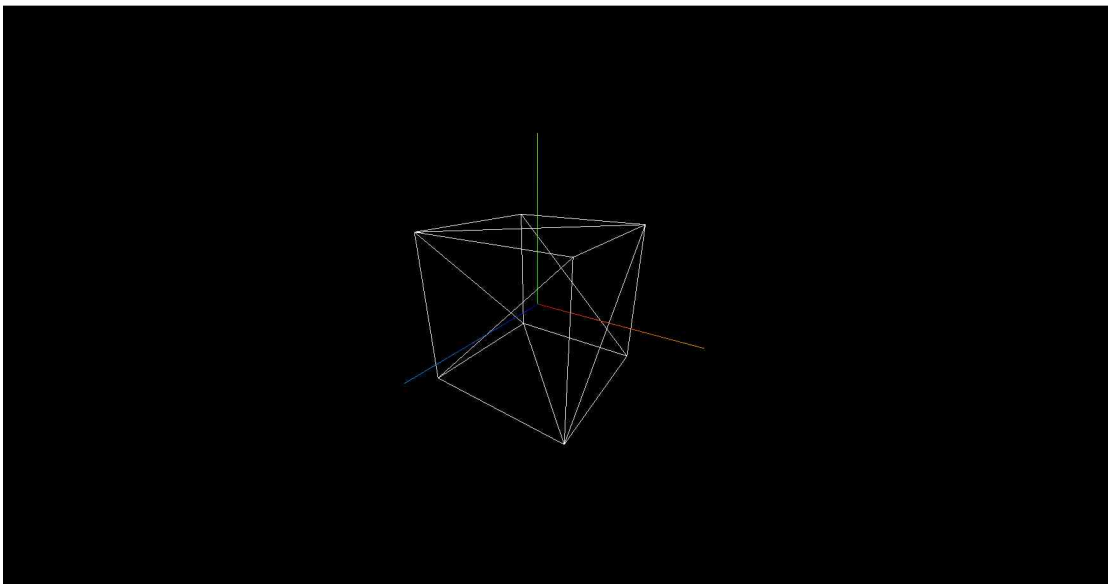
사용할 큐브는 MeshBasicMaterial로 하얀 모서리로 보이게끔 설정했으며, 사각형의 변의 길이는 10으로 두었다. 윈도우 창이 변해도 깨지지 않게끔 카메라와 랜더러를 윈도우 창만큼 설정했고, EventListener를 첨부해 함수를 실행시켰다.

```

38
39 // Scene Setting
40 const Scene = new THREE.Scene();
41 Scene.add( Cube );
42 Scene.add( new THREE.AxesHelper ( 10 ) ); // 3차원 좌표계 추가
43 animate();
44
45
46 function animate(){
47     requestAnimationFrame( animate );
48
49     controls.update();
50     renderer.render( Scene, camera );
51 }

```

Scene 세팅에 위에서 만들었던 큐브와 좌표계를 추가했다. 좌표계는 모든 축의 10까지 설정했다. 애니메이션 함수에는 OrbitControls가 항상 작동하며, 설정했던 Scene과 Camera가 WebGLRenderer를 통해 구축이 된다.



실행시 다음과 같은 화면이 뜬다.

* import의 저장 경로 때문에 부득이하게 Three.js와 함께 첨부하였습니다. Practice2를 GoLive 하면 실행됩니다.