

OrbitControls

참고 문헌 : <https://threejs.org/docs/#examples/en/controls/OrbitControls>

OrbitControls을 사용하려면 따로 import를 해주어야한다.(webpack으로 설치함)

```
import {OrbitControls} from 'three/examples/jsm/controls/
```

**** 반드시 camera 세팅 코드 이후에 작성해주자!**

```
const controls = new OrbitControls( camera,
  renderer.domElement );

//controls.update() must be called after any manual
changes to the camera's transform
camera.position.set( 0, 20, 100 );
controls.update();

function animate() {

  requestAnimationFrame( animate );

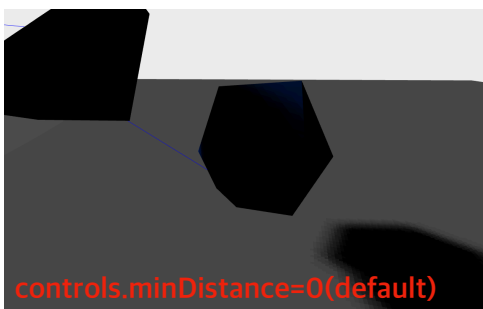
  // required if controls.enableDamping or
controls.autoRotate are set to true
  controls.update();

  renderer.render( scene, camera );

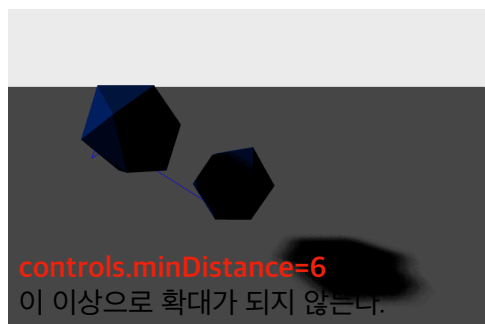
}
```

그리고 animate 함수가 필요하니 참고

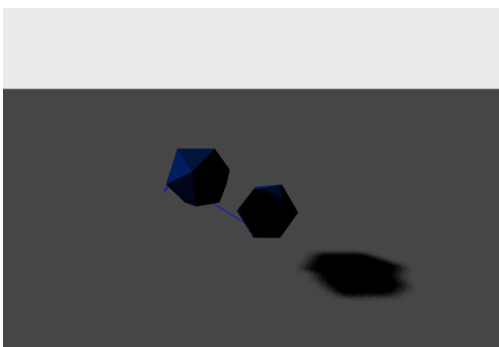
최대 줌 / 최소 줌



화면을 최대한 가까이 해도 멈추지 않아서 피사체들이 점점 안보이기 시작한다.



controls.minDistance=6
이 이상으로 확대가 되지 않는다.



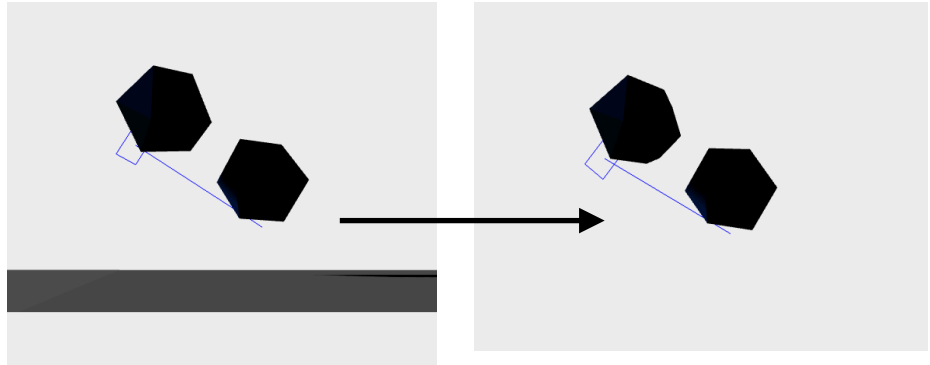
controls.maxDistance=6
더이상 멀어지지 않는다.

최대각도 / 최소각도

마우스로 이리저리 화면을 돌리다보면 내가 원하지 않은 바닥이 보인다거나 안보였으면 하는 각도가 있으면 지정해줄 수 있다.

- 바닥부터 밑으로의 각도 :

```
controls.maxPolarAngle = 2
```



바닥에서부터 밑으로 2정도 내려가는 것을 볼 수 있다.

- 정확히 바닥까지만 보여주기 :

```
controls.maxPolarAngle = Math.PI/2
```

- 바닥부터 위로의 각도 :

```
controls.minPolarAngle = 2
```

(숫자가 커질수록 위로 올려다보는 각도가 작아진다.) 특별한 경우가 아니고선은 사용안할 듯

팁 ✨

shift + 드래그하면 카메라의 위치를 손 쉽게 바꿀 수 있다.