移动物体

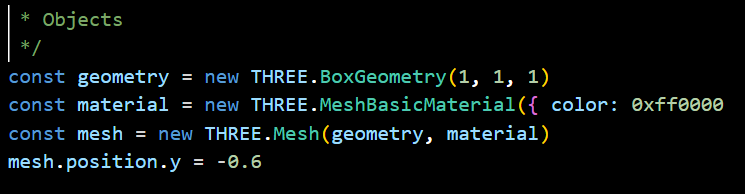
属性：

1. 位置
2. Scale刻度
3. Rotation 旋转
4. Quaternion

Perspectivecamera 、mesh

常规操作：打开终端输入npm install

npm run dev



用Mesh.position.x/y/x mesh.position.set(x1,y1,z1)

来移动。坐标为右手坐标系

这句代码在渲染前的任何位置都可以，当然，要在创建物体后面

THREE.Vector3(x1,y1,z1)坐标点（可以作为distanceTo()的参数）

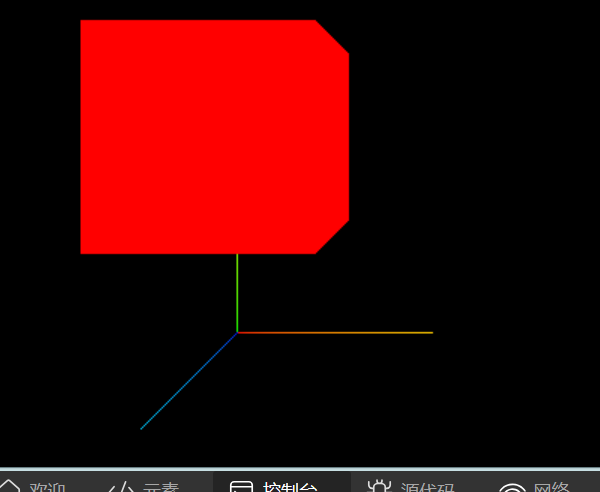
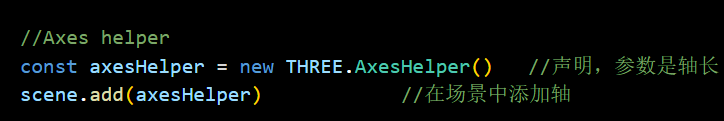
获取相机和物体的距离：

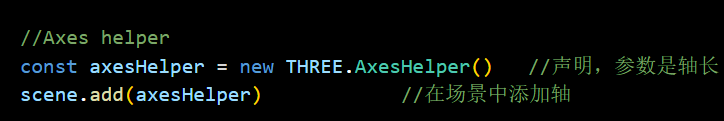
console.log(mesh.position.distanceTo(camera.position))

Nomalize()

将物体距相机距离减小为1

AxesHelper轴助手类：方便定位





Scale：改变物体形状



mesh.scale.set(2, 0.5, 0.5)

Rotation

mesh.rotation.x = Math.PI

注意：绕轴转

转半圈：3.14159（math.pi)

可能遇到的问题：

一个轴转的时候其他轴也会转方向，所以有的时候某个轴会莫名其妙失灵。（gimbal lock现象）

解决方法：

在rotation前更改旋转顺序

mesh.rotation.reorder('YXZ')

Quaternion（也是旋转，但是数学性更强）

Quaternion和rotation会相互影响

lookAt（）

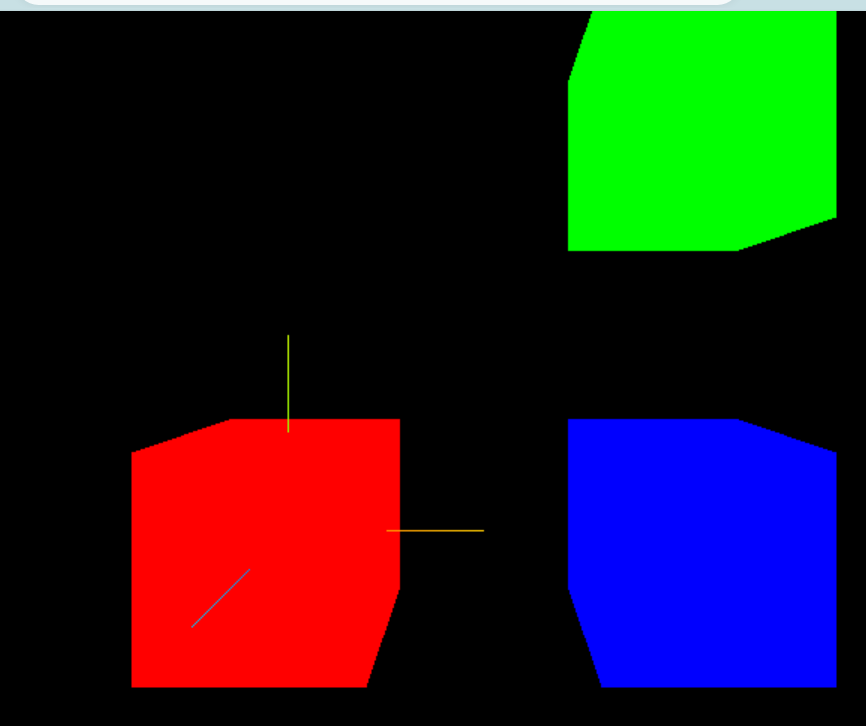
可以让一个物体“看”另一个物体！包括相机（很喜欢这个词感觉很有意思hhh

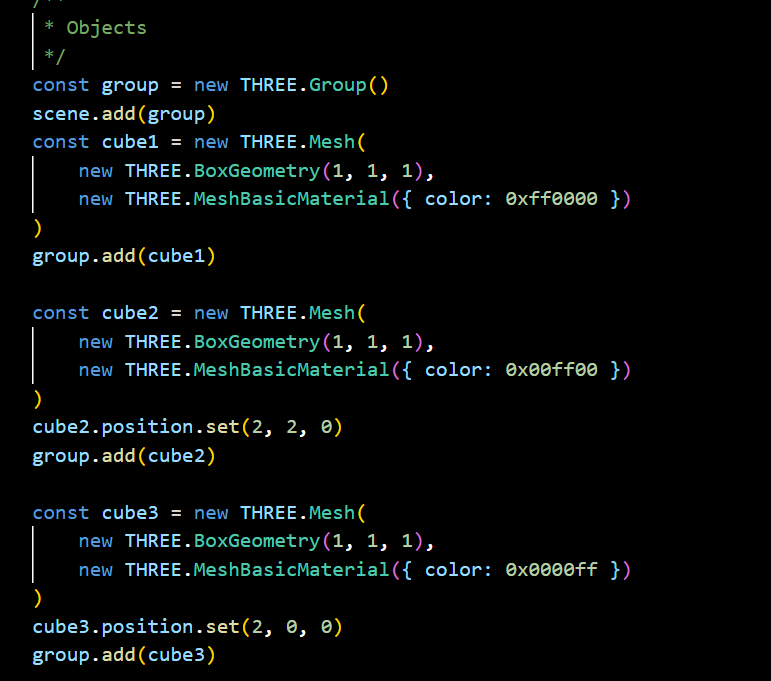
camera.lookAt(mesh.position)

语句的位置：

position,rotation(quaternion),scale可以放在任何位置

Create a group！Group() method





group.position.y = 1

可以让整个group移动

