THREE.PerspectiveCamera(75,sizes.width/sizes.height,1,1000) 透视摄像机

75:视野（垂直方向的度数）

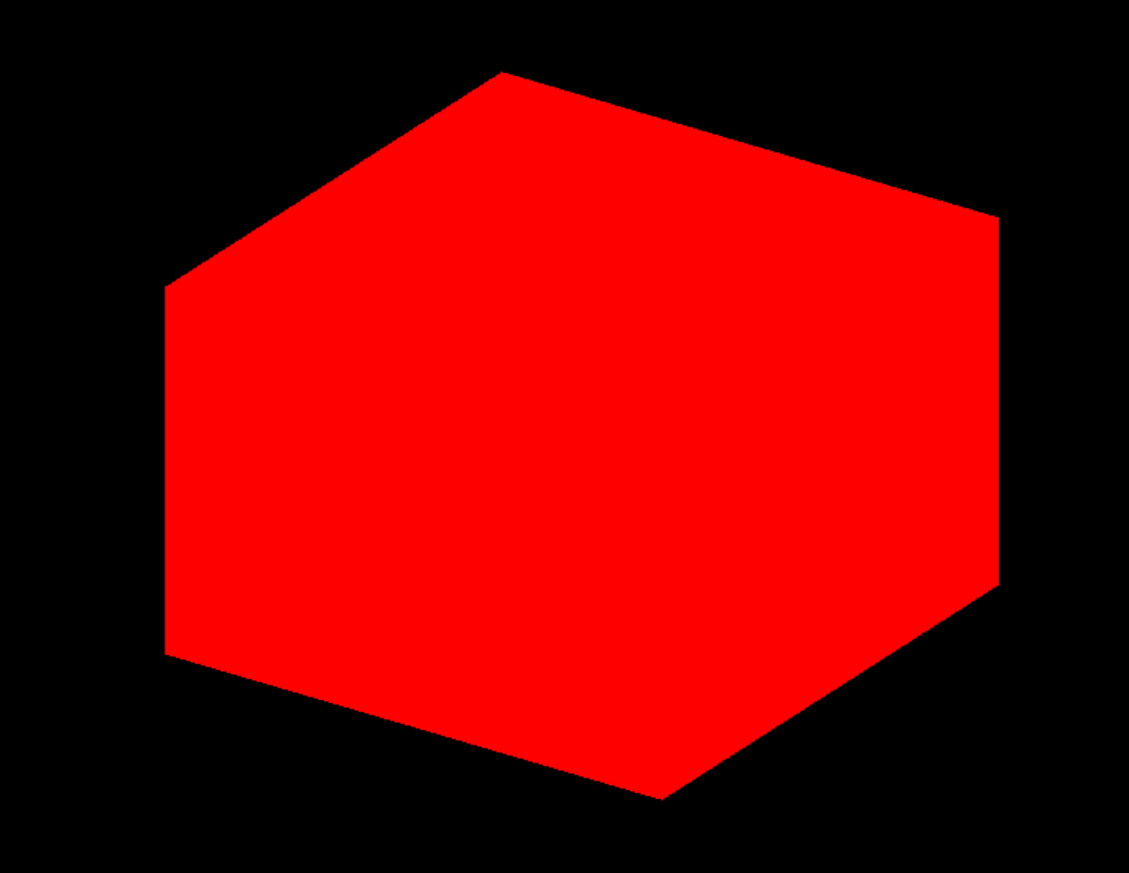
sizes.width/sizes.height：宽高比（效果界面）

1,1000：近景远景参数，任何小于/大于这个距离的部分会变得不可见 不要设置成0.0001和99999的类似数字，会产生bug（模型离太近会卡模型

OrthographicCamera

视角类似立方体，后面的参数左右上下，远近（可缺省）

const camera = new THREE.OrthographicCamera(-1, 1, 1, -1)



有点怪。可以用左\*渲染器比例，右\*渲染器比例，但还是有点奇怪

所以还是用透视摄像机

用鼠标移动camera视角

1. 获取鼠标光标的x坐标

//Cuisor

window.addEventListener('mousemove',(event)=>{

    console.log(event.clientX)

})



建议将大小控制在0-1

const cursor = {

x:0,

y:0

}

window.addEventListener('mousemove', (event) => {

cursor.x=event.clientX/sizes.width-0.5 //可以有一个中间值

console.log(cursor.x)

})

1. 在tick中更新camera位置

camera.position.x=cursor.x\*3

注意，在window里，y向上是负，向下是正，所以要修改y

cursor.y = -(event.clientY / sizes.height - 0.5)

1. 让camera对准画面中心：camera.lookAt(new THREE.Vector3())

问题：看不到立方体背面

改变x、z

1. 添加上y

//update camera

camera.position.x = Math.sin(cursor.x \* Math.PI \* 2) \* 3

camera.position.z = Math.cos(cursor.x \* Math.PI \* 2) \* 3

camera.position.y = cursor.y \* 5

camera.lookAt(mesh.position)

注意：xzy设定要放在lookat前

controls

1. device orientation controls

设备（手机）移动时，视角会跟着移动

1. fly controls

飞船控制

1. first controls

不能控制高度的fly controls

1. pointer lock controls

真正意义上的第一人称，能够用键盘前进后退，隐藏鼠标，跳跃

1. orbit controls 轨道

平滑地控制视角，有移动距离限制

1. trackball controls

没有限制的orbit controls

1. transform controls

与camera无关，能够移动物体，有点像编辑器

1. drag controls

拖动控制物体

使用orbit controls

该类存在于webpack

需要import { OrbitControls } from 'three/examples/jsm/Addons.js'

声明

const controls = new OrbitControls(camera, canvas)

就可以用鼠标拖动视角了，还可以缩放。不过我觉得这比起移动视角更像拖动物体

默认对准canvas中心

controls.target.y = 1

controls.update()

可以修改，但是一定要加上update才会生效

移动太灵敏了？添加阻尼

controls.enableDamping = true

同时要在tick中更新控件，在渲染前

controls.update() 不更新也可以，但是停止拖动就会立即停止。更新后停止时会有惯性移动