

## 🎮 6. 相机跟着玩家走(第三人称漫游)

前面案例，通过按键控制玩家角色模型运动的时候，并没有控制相机移动。下面给大家讲解怎么控制相机运动产生漫游的感觉。

```
const camera = new THREE.PerspectiveCamera();
camera.position.set(8, 10, 14);
camera.lookAt(0, 0, 0);

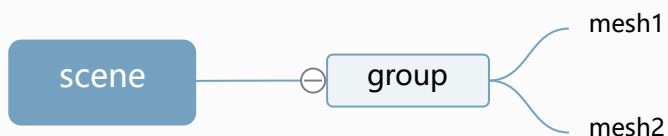
function render() {
  ...
  //更新玩家角色的位置
  player.position.add(deltaPos);
}
render();
```

js

## 层级模型知识点回顾

### 4.1 层级模型🔗

### 4.3. 本地坐标和世界坐标🔗



```
const mesh1 = new THREE.Mesh();
const group = new THREE.Group();
group.add(mesh1);
scene.add(group);
```

js

mesh1的父对象group移动，mesh1会跟着移动

```
group.position.y = 10;
```

js

`mesh1.position` 表示mesh1局部坐标，也就是相对父对象group的位置。mesh1在三维场景scene中的实际位置(世界坐标)就是 `group.position` 的 `mesh1.position` 叠加。

```
mesh1.position.y = 10;
```

js

## 相机对象父类 `Object3D`

相机对象Camera的父类和mesh、group一样，都是 `Object3D`，这意味着，如果你把相机作为某个模型的子对象，相机的位置和姿态同样受到模型的影响。

```
const group = new THREE.Group();
group.add(camera); // 相机作为group子对象
// 父对象group平移，相机跟着平移
group.position.y = 10;
// 父对象group旋转，相机跟着旋转
group.rotateY(Math.PI/6);
```

js

## 注释相机空间OrbitControls代码

注释相机空间OrbitControls代码，避免影响相机W、A、S、D对相机的控制，原来用OrbitControls只是为了方便观察测试3D场景。

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

js

## 相机作为玩家角色子对象

相机作为玩家角色子对象，可以实现相机对玩家角色模型的跟随运动，使相机运动，模拟人漫游3D场景的感觉。

```
// 把相机作为玩家角色的子对象，这样相机的位置和姿态就会跟着玩家角色改变
player.add(camera); // 相机作为人的子对象，会跟着人运动
```

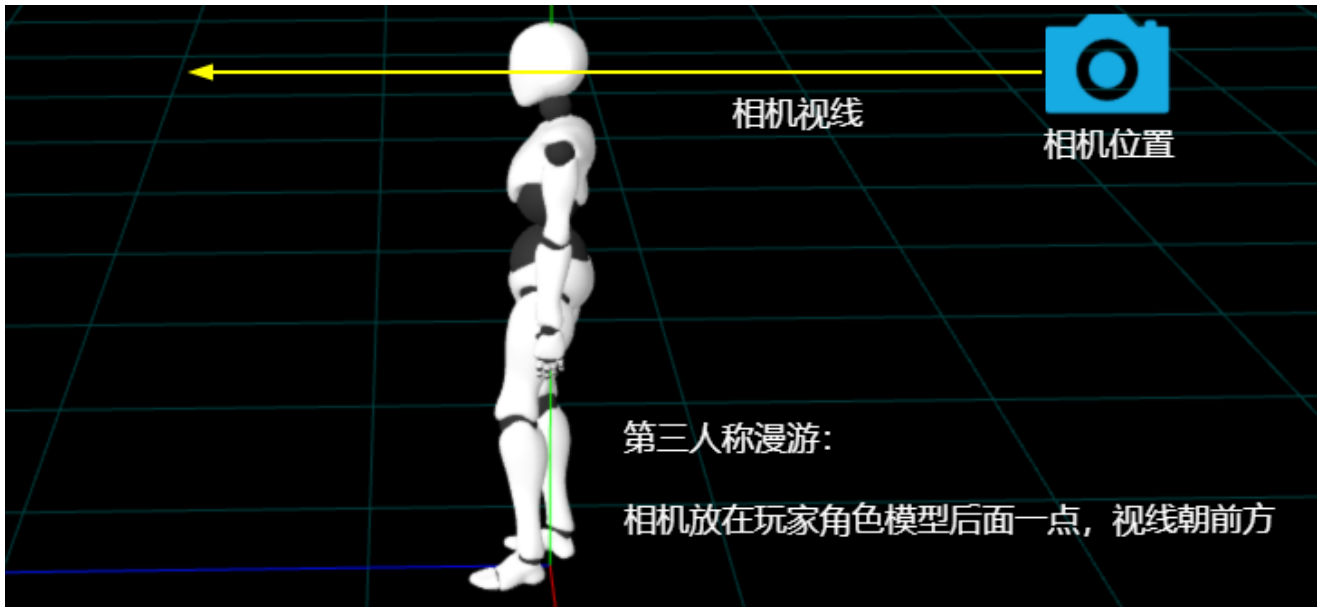
js

你可以根据需要放相机相对玩家角色的位置，比如我这里相机与人高度相近(你可以在blender中测量下人的高度)，把相机放在人后脑勺，拉开一定距离，然后相机镜头对准人的后脑勺。

下面尺寸是以相机的父对象玩家角色模型的局部坐标系坐标原点为参照的。

```
camera.position.set(0, 1.6, -5.5); // 玩家角色后面一点  
camera.lookAt(0, 1.6, 0); // 对着人身上某个点 视线大致沿着人的正前方
```

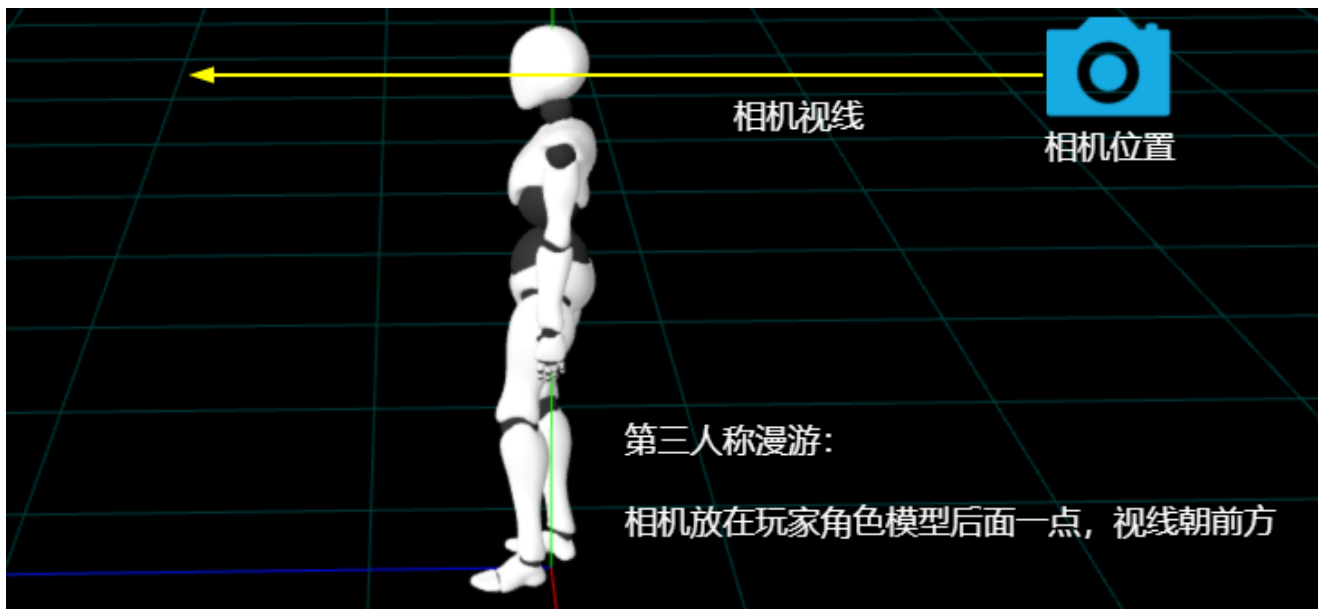
js



我这里是以一个人为例写的相机位置，如果你换成一个车，模拟人在驾驶位上的感觉，也可以把相机高度设置在车辆高度附近。

## 第三人称漫游

这里所谓的**第三人称**，你可以简单的理解为相机在运动漫游的过程中，你可以看到玩家角色模型，比如你能看到运动的人、车等角色模型，就是上面咱们写的代码，把相机放在玩家角色模型后面一点即可。



## 补充：相机视角参数fov影响相机位置设置！！

```
const camera = new THREE.PerspectiveCamera(30,...);  
player.add(camera); // 相机作为人的子对象  
// 玩家角色后面一点 对应fov 30度  
camera.position.set(0, 1.6, -5.5);  
camera.lookAt(0, 1.6, 0); // 对着人身上某个点
```

js

根据透视投影相机规律，fov变大，能够看到的视野范围角度更大。

```
const camera = new THREE.PerspectiveCamera(70,...);  
// 玩家角色后面一点 对应fov 70度  
camera.position.set(0, 1.6, -2.3);
```

js

← 5. 按键S退后运动

7. 鼠标左右拖动改变玩家视角 →