

🟡 2. W键控制角色模型运动

WASD的综合控制比较复杂，本节课，先给大家演示下，怎么通过W监控人物模型沿着Z轴运动。

当W键一直处于按下状态时，人物模型沿着Z轴向前运动，松开的时候，不在运动(你可以测试课件的源码)。

演示文件

演示文件里面给大家提供了一个基本的文件，加载了一个人物模型，设置一个简单网格线地面，关于人物模型的骨骼动画，你可以参考前面基础课程章节16讲解：[16.4. 解析外部模型关键帧动画](#)🔗；

知识回顾

上节课给大家讲解过，怎么通过一个对象的属性，记录W、A、S、D四个按键的状态。

```
const keyStates = {  
  W: false,  
  A: false,  
  S: false,  
  D: false,  
};  
document.addEventListener('keydown', (event) => {  
  if (event.code === 'KeyW') keyStates.W = true;  
  if (event.code === 'KeyA') keyStates.A = true;  
  if (event.code === 'KeyS') keyStates.S = true;  
  if (event.code === 'KeyD') keyStates.D = true;  
});  
document.addEventListener('keyup', (event) => {  
  if (event.code === 'KeyW') keyStates.W = false;  
  if (event.code === 'KeyA') keyStates.A = false;  
  if (event.code === 'KeyS') keyStates.S = false;  
  if (event.code === 'KeyD') keyStates.D = false;  
});
```

js

W键控制人物模型运动

你先回顾下，本课程[章节2](#)关于位移、速度、加速的讲解，更好理解接下来要讲解的内容。

```
// 用三维向量表示玩家角色(人)运动漫游速度
//按下W键对应的人运动速度
const v = new THREE.Vector3(0, 0, 3);
```

js

渲染循环里面，通过 `时间*速度` ,来更新人模型位置。

```
// 渲染循环
const clock = new THREE.Clock();
function render() {
    const deltaTime = clock.getDelta();
    if (keyStates.W) {
        // 在间隔deltaTime时间内，玩家角色位移变化计算(速度*时间)
        const deltaPos = v.clone().multiplyScalar(deltaTime);
        player.position.add(deltaPos); //更新玩家角色的位置
    }
    mixer.update(deltaTime); // 更新播放器相关的时间
    renderer.render(scene, camera);
    requestAnimationFrame(render);
}
render();
```

js

← [1. 键盘WASD按键状态记录](#)

[3. 加速度\(按键给玩家加速\)](#) →