

🎮 5. 按键S退后运动

这些内容非常简单，算是一个练习题，你可以在前面几节课基础上，增加一个S按键，玩家角色模型的退后运动，学习视频之前，可以自己先动手写下。

S键退后

```
function render() {  
  if (v.length() < vMax) { //限制最高速度  
    if (keyStates.W) {  
      //先假设W键对应运动方向为z  
      const front = new THREE.Vector3(0, 0, 1);  
      v.add(front.multiplyScalar(a * deltaTime));  
    }  
    if (keyStates.S) {  
      // 与W按键相反方向  
      const front = new THREE.Vector3(0, 0, -1);  
      v.add(front.multiplyScalar(a * deltaTime));  
    }  
  }  
  v.addScaledVector(v, damping); //阻尼减速  
  //更新玩家角色的位置  
  const deltaPos = v.clone().multiplyScalar(deltaTime);  
  player.position.add(deltaPos);  
}
```

js

下面是更多具体代码

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

js

```

    if (event.code === 'KeyW') keyStates.W = false;
    if (event.code === 'KeyA') keyStates.A = false;
  });

```

js

```

// 用三维向量表示玩家角色(人)运动漫游速度
const v = new THREE.Vector3(0, 0, 0); // 初始速度设置为0
const a = 12; // WASD按键的加速度: 调节按键加速快慢
const damping = -0.04; // 阻尼 当没有WASD加速的时候, 人、车等玩家角色慢慢减速停下来
const vMax = 5; // 限制玩家角色最大速度
// 渲染循环
const clock = new THREE.Clock();
function render() {
  const deltaTime = clock.getDelta();
  if (v.length() < vMax) { // 限制最高速度
    if (keyStates.W) {
      const front = new THREE.Vector3(0, 0, 1); // 先假设W键对应运动方向为z
      v.add(front.multiplyScalar(a * deltaTime));
    }
    if (keyStates.S) {
      // 与W按键相反方向
      const front = new THREE.Vector3(0, 0, -1);
      v.add(front.multiplyScalar(a * deltaTime));
    }
  }
  //  $v * (1 + \text{damping}) = v * (1 - 0.04) = v * 0.96$ 
  // 多次循环乘以0.96( $v * 0.96 * 0.96 * 0.96 \dots$ ), v就会无限逼近于0。
  //  $v * (1 + \text{damping}) = v + v * \text{damping}$ 
  v.addScaledVector(v, damping); // 阻尼减速

  // 更新玩家角色的位置 当v是0的时候, 位置更新也不会变化
  const deltaPos = v.clone().multiplyScalar(deltaTime);
  player.position.add(deltaPos);

  mixer.update(deltaTime);
  renderer.render(scene, camera);
  requestAnimationFrame(render);
}
render();

```

