

🟡 8. 获取玩家(相机)正前方方向

实际开发，**玩家角色的视角**或者说**相机的视角**，会随着鼠标左右移动变化的，不过前面几节课，为了降低学习难度，代码给的是固定方向。

```
function render() {  
  if (keyStates.W) {  
    //先假设W键对应运动方向为z  
    const front = new THREE.Vector3(0, 0, 1);  
    // 改变玩家速度  
    v.add(front.multiplyScalar(a * deltaTime));  
  }  
}
```

.getWorldDirection()

`Object3D` 类有一个获取模型局部z轴方向相关的方法 `.getWorldDirection()`。

`obj.getWorldDirection()` 表示的获取obj对象自身z轴正方向在世界坐标空间中的方向。

模型没有任何旋转情况，`.getWorldDirection()` 获取的结果 `(0,0,1)`

```
const mesh = new THREE.Mesh();  
const dir = new THREE.Vector3();  
mesh.getWorldDirection(dir);  
console.log('dir', dir);
```

模型绕y旋转90度情况，`.getWorldDirection()` 获取的结果 `(1,0,0)`

```
const mesh = new THREE.Mesh();  
mesh.rotateY(Math.PI / 2);  
const dir = new THREE.Vector3();  
mesh.getWorldDirection(dir);  
// 模型没有任何选择打印结果(1,0,0)
```

```
console.log('dir', dir);
```

`.getWorldDirection()` 获取玩家角色正前方

注意：threejs加载的玩家角色glTF模型，自身 `.rotation` 没有任何旋转的情况下，注意玩家角色正前方方向最好和z轴方向一致,这样就可以直接用 `.getWorldDirection()` 获取的结果表示人的正前方。

```
// 按下W键，实时计算当前玩家角色的正前方向
if (keyStates.W) {
    const front = new THREE.Vector3();
    //获取玩家角色(相机)正前方
    player.getWorldDirection(front);
}
```

js

S键运动方向

注意S键运动方向与W的正前方相反，这时候很简单，可以计算方向的时候，把front取反，或者最简单加速度设置一个负号 `front.multiplyScalar(- a * deltaTime)`

```
function render() {
    if (v.length() < vMax) { //限制最高速度
        if (keyStates.W) {
            const front = new THREE.Vector3();
            player.getWorldDirection(front); //获取玩家角色(相机)正前方
            v.add(front.multiplyScalar(a * deltaTime));
        }
        if (keyStates.S) {
            const front = new THREE.Vector3();
            player.getWorldDirection(front);
            // - a: 与W按键反向相反
            v.add(front.multiplyScalar(- a * deltaTime));
        }
    }
}
```

js

