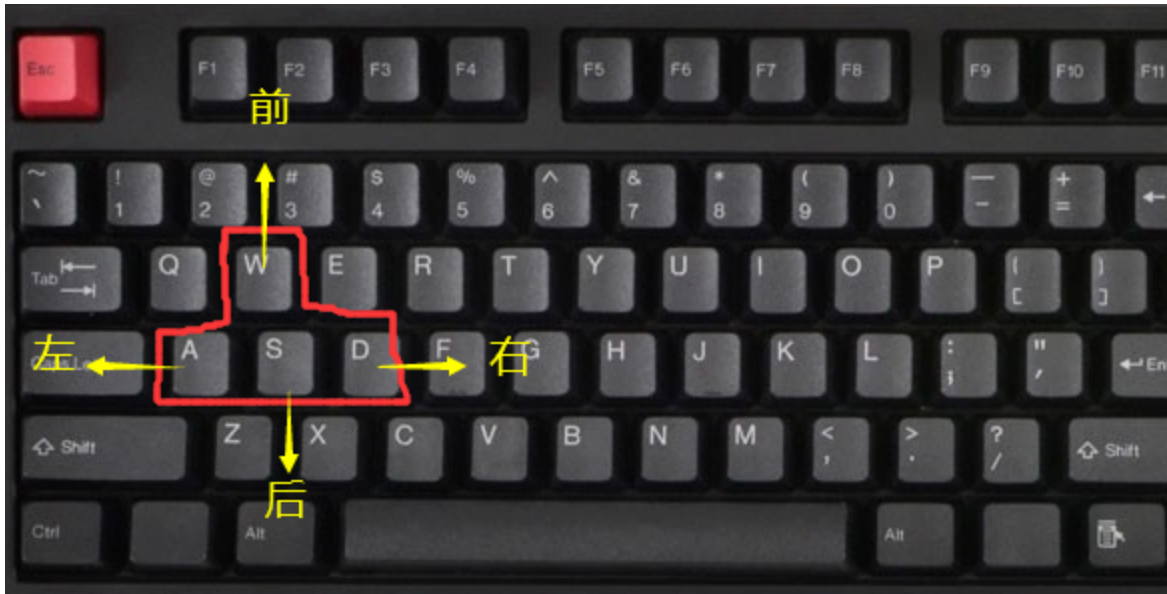


🔹 1. 键盘WASD按键状态记录

如果你玩过游戏，一般都知道，通过键盘的W、A、S、D按键可以控制玩家角色在3D场景中运动，比如控制一个人前后左右运动，比如控制一辆车前后左右运动。



键盘事件

如果你不熟悉HTML5前端鼠标、键盘事件，可以去学习下,下面主要说下思路，不在一行一行演示。

下面代码功能是当你按下随便一个键盘按键，就会触发参数2表示的函数执行，并log打印对应按键名字相关信息。

```
// 监听鼠标按下事件
document.addEventListener('keydown', (event) => {
  console.log('event.code', event.code);
})
```

js

执行上面代码后，你可以分别按下键盘W、A、S、D、空格键测试，你可以看到浏览器控制台打印输出对应键盘按键名字，也就是KeyW、KeyA、KeyS、KeyD、Space。

js

```
// 监听鼠标松开事件
document.addEventListener('keyup', (event) => {
  console.log('event.code', event.code);
})
```

记录键盘按键WASD状态

js

```
// 声明一个对象keyStates用来记录键盘事件状态
const keyStates = {
  // 使用W、A、S、D按键来控制前、后、左、右运动
  // false表示没有按下，true表示按下状态
  W: false,
  A: false,
  S: false,
  D: false,
};
// 当某个键盘按下设置对应属性设置为true
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;
});
// 当某个键盘抬起设置对应属性设置为false
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;
});
```

测试键盘状态

在循环执行的函数中查看键盘状态值。

js

```
// 循环执行的函数render
function render() {
  requestAnimationFrame(render);
}
render();
```

你可以执行下面代码，然后反复按下或松开W键，浏览器控制台查看 `keyStates.W` 的变化。

```
// 循环执行的函数中测试W键盘状态值
function render() {
  if(keyStates.W){
    console.log('W键按下');
  }else{
    console.log('W键松开');
  }
  requestAnimationFrame(render);
}
render();
```

js

其他写法(可以跳过)

批量记录所有键盘事件状态

```
// 声明一个对象keyStates用来记录键盘事件状态
const keyStates = {
  // // false表示没有按下，true表示按下状态
  // keyW:false,
  // keyA:false,
  // keyS:false,
  // keyD:false,
};

// 当某个键盘按下设置对应属性设置为true
document.addEventListener('keydown', (event) => {
  keyStates[event.code] = true;
});

// 当某个键盘抬起设置对应属性设置为false
document.addEventListener('keyup', (event) => {
  keyStates[event.code] = false;
});
```

js

