郭隆邦 2023-09-03

13. 骨骼动画与运动状态关联

学习本节课之前，首先确保你已经熟悉前面基础内容16. 关键帧动画，要不然大部分代码你也看不懂，也跟不上视频。

骨骼动画有休息、步行、跑步等动作，本节课大家简单讲解下，根据玩家角色的运动状态来执行对应的骨骼动画。

查看模型所有骨骼动画数据

```js
const gltf = await loader.loadAsync("../../人.glb");
const player = gltf.scene;//玩家角色模型

//包含关键帧动画的模型作为参数创建一个播放器
const mixer = new THREE.AnimationMixer(player);
console.log('所有骨骼动画数据', gltf.animations);
// 骨骼动画动画名字和对应含义，名字是可以在bledner中随意命名的
//Idle  休息
//Run   跑步
//Walk  走路
//休息动作
const IdleAction = mixer.clipAction(gltf.animations[5]);
//步行动作
const WalkAction = mixer.clipAction(gltf.animations[13]);
//跑步动作
const RunAction = mixer.clipAction(gltf.animations[0]);
```

查看`gltf.animations`，你可以看到不同clip动画的名字等数据

```js
console.log('所有骨骼动画数据', gltf.animations);
// 骨骼动画动画名字和对应含义，名字是可以在bledner中随意命名的
//Idle  休息
//Run   跑步
//Walk  走路
```

不同clip动画数据的索引值，可以在浏览器控制台打印`gltf.animations`查看

不同模型不同，你根据自己项目模型，灵活应对即可。

```js
//休息动作
const IdleAction = mixer.clipAction(gltf.animations[5]);
//步行动作
const WalkAction = mixer.clipAction(gltf.animations[13]);
//跑步动作
const RunAction = mixer.clipAction(gltf.animations[0]);
```

站着休息和步行两个动作切换

具体思路：你参考前面基础课程11. 骨骼动画不同动作切换 里面案例2即可。

```js
//休息动作
const IdleAction = mixer.clipAction(gltf.animations[5]);
//步行动作
const WalkAction = mixer.clipAction(gltf.animations[13]);
IdleAction.play();
WalkAction.play();
IdleAction.weight = 1.0;//默认休息状态
WalkAction.weight = 0.0;
```

在站着休息和步行两个动作之间切换

```js
function changeAction(name){
    if (name=='Idle'){
        IdleAction.weight = 1.0;//休息状态
        WalkAction.weight = 0.0;
    }else if (name == 'Walk') {
        IdleAction.weight = 0.0;
        WalkAction.weight = 1.0;//步行状态
    }
}
```

根据玩家速度 `v` 的大小 `v.length()` ,控制使用休息和步行那个动作。

```js
function playerUpdate(deltaTime) {
    const vL = v.length();
    if (vL < 0.2) {//速度小于0.2切换到站着休息状态
        changeAction('Idle');
```

```
    } else if (vL >= 0.2) {//步行状态
        changeAction('Walk');
    }
}
```

阻尼和加速度调节

当不在使用WASD加速的时候，如果你希望玩家快速减速，可以适当提升阻尼。

```js
const damping = -0.1;
v.addScaledVector(v, damping);//阻尼减速
```

WASD按键加速的同时，也会通过阻尼减速，所以阻尼的存在会限制最大速度。如果达不到自己想要的最大速度，可以把加速度a的值提升。

```js
const a = 30;//WASD按键的加速度：调节按键加速快慢
const damping = -0.1;
const vMax = 10;//限制玩家角色最大速度

const vL = v.length();
console.log('vL', vL);//浏览器控制查看速度大小

v.addScaledVector(v, damping);//阻尼减速
```

练习题：休息、步行、跑步三个动作

```js
//休息动作
const IdleAction = mixer.clipAction(gltf.animations[5]);
//步行动作
const WalkAction = mixer.clipAction(gltf.animations[13]);
//跑步动作
const RunAction = mixer.clipAction(gltf.animations[0]);
IdleAction.play();
WalkAction.play();
RunAction.play();
IdleAction.weight = 1.0;//默认休息状态
WalkAction.weight = 0.0;
RunAction.weight = 0.0;


function changeAction(name) {
```

```
    if (name == 'Idle') {
        IdleAction.weight = 1.0;
        WalkAction.weight = 0.0;
        RunAction.weight = 0.0;
    } else if (name == 'Walk') {
        IdleAction.weight = 0.0;
        WalkAction.weight = 1.0;
        RunAction.weight = 0.0;
    } else if (name == 'Run') {
        IdleAction.weight = 0.0;
        WalkAction.weight = 0.0;
        RunAction.weight = 1.0;
    }
}
```

不同速度设置三个动画状态，测试上面骨骼动画效果

```js
function playerUpdate(deltaTime) {
    const vL = v.length();
    if (vL < 0.2) {//速度小于0.2切换到站着休息状态
        // 注释如果当前就是Idle状态，不要再次执行changeAction
        changeAction('Idle');
    } else if (vL > 0.2 && vL < 4) {//步行状态
        changeAction('Walk');
    } else if (vL >= 4) {//跑步状态
        changeAction('Run');
    }
}
```

作业练习：批量解析全部骨骼动画动作

如果动作多了你会发现一个一个写，比较麻烦，其实你也可以通过for循环批量解析所有骨骼动画动作。

```js
const clipArr = gltf.animations;//所有骨骼动画
const actionObj = {};//包含所有动作action
for (let i = 0; i < clipArr.length; i++) {
    const clip = aniArr[i];//休息、步行、跑步等动画的clip数据
    const action = mixer.clipAction(clip);//clip生成action
    action.name = clip.name;//action命名name
    // 批量设置所有动画动作的权重
    if (action.name === 'Idle') {
```

```
            action.weight = 1.0;//这样默认播放Idle对应的休息动画
        } else {
            action.weight = 0.0;
        }
        action.play();
        // action动画动作名字作为actionObj的属性
        actionObj[action.name] = action;
    }
```

动作切换函数

```js
let currentAction = actionObj['Idle'];//记录当前播放的动作
// 切换不同动作
function changeAction(actionName) {
    currentAction.weight = 0.0;//原来动作权重为0，不播放
    const action = actionObj[actionName];//新的需要播放的动作
    action.weight = 1.0;//将要播放的动作权重为1
    currentAction = action;//替换记录的动作
}
```

根据玩家角色速度v设置休息和步行动作。

```js
function playerUpdate(deltaTime) {
    const vL = v.length();
    if (vL < 0.2) {//速度小于0.2切换到站着休息状态
        // 如果当前就是Idle状态，不需要再次执行changeAction
        if (currentAction.name != 'Idle') changeAction('Idle');
    } else {//步行状态
        if (currentAction.name != 'Walk') changeAction('Walk');
    }
}
```

← 12. 快捷键切换第一、第三人称

1. 八叉树Octree扩展库介绍→