郭隆邦 2023-06-04

2. 物体下落动画(重力加速度)

本节课给大家讲解一个物体抛出去落在地面的动画效果，注意学习本节课内容之前，确保你已经掌握上节课关于匀速动画的讲解。

已知条件

```js
// 物体初始位置
mesh.position.set(0,100,0);
```

```js
//物体初始速度
const v = new THREE.Vector3(30,0,0);
```

重力加速度是y轴的负方向

```js
//重力加速度
const g = new THREE.Vector3(0, -9.8, 0);
```

物理加速度位移公式 `x = vt + 1/2gt^2` 计算位置

下面用three.js向量代码表示物理加速度的位移公式，如果你已经忘了高中物理知识，也没关系，也不用记忆，咱们的重点在于用threejs向量表达上面公式。

```js
const v = new THREE.Vector3(30, 0, 0);//物体运动速度
const clock = new THREE.Clock();//时钟对象
let t = 0;
const g = new THREE.Vector3(0, -9.8, 0);
const pos0 = mesh.position.clone();
// 渲染循环
function render() {
    const spt = clock.getDelta();//两帧渲染时间间隔(秒)
    t += spt;
    // 在t时间内，以速度v运动的位移量
```

```
    const dis = v.clone().multiplyScalar(t).add(g.clone().multiplyScalar(0.5 * t
    // 网格模型当前的位置加上spt时间段内运动的位移量
    const newPos = pos0.clone().add(dis);
    mesh.position.copy(newPos);
    renderer.render(scene, camera);
    requestAnimationFrame(render);
}
render();
```

物体接触地面后，停止运动，不在进行下落计算。

```js
// 渲染循环
function render() {
    if (mesh.position.y > 0) {
        ...
        mesh.position.copy(newPos);
    }
    renderer.render(scene, camera);
    requestAnimationFrame(render);
}
render();
```

速度 x 间隔时间，然后累加计算位移

- 速度 `v` = 加速度 `g` x 时间 `t`
- 位移 `x` = 速度 `v` x 时间 `t`

重力加速度乘以每次渲染时间，与原来的速度累加，可以更新当前的速度。

位移写法和上节课类似，就是当前速度乘以每次渲染时间，累加即可。

```js
const v = new THREE.Vector3(30, 0, 0);//物体初始速度
const clock = new THREE.Clock();//时钟对象
const g = new THREE.Vector3(0, -9.8, 0);
// 渲染循环
function render() {
    if (mesh.position.y > 0) {
        const spt = clock.getDelta();//两帧渲染时间间隔(秒)
        //spV:重力加速度在时间spt内对速度的改变
        const spV = g.clone().multiplyScalar(spt);
        v.add(spV);//v = v + spV  更新当前速度
```

```
            // 在spt时间内，以速度v运动的位移量
            const dis = v.clone().multiplyScalar(spt);
            // 网格模型当前的位置加上spt时间段内运动的位移量
            mesh.position.add(dis);
        }
        renderer.render(scene, camera);
        requestAnimationFrame(render);
    }
    render();
```

斜向上抛出去物体

你可以更改初速度，观察物体的运动轨迹效果

```js
//物体初始速度
const v = new THREE.Vector3(30, 20, 0);
```

← 1. 匀速动画(向量表示速度)

1. 向量点乘dot→