郭隆邦 2023-10-01

3. 练习-threejs可视化cannon计算结果

本节课是一个练习题，就是用threejs把cannonjs计算的小球位置可视化表示出来。

- CannonJS：负责物理计算，比如计算出来一个小球的下落位置
- three.js：负责可视化渲染，比如用Mesh渲染一个下落的小球视觉效果

一个负责物理模拟计算，一个负责3D场景的渲染

知识点回顾

参考上节课知识点，创建一个body表示小球，与threejs的网格小球mesh对应。

```js
const world = new CANNON.World();
// 设置物理世界重力加速度
world.gravity.set(0, -9.8, 0);

// 物理小球：对应threejs的网格小球
const body = new CANNON.Body({
    mass: 0.3,//碰撞体质量
    shape:new CANNON.Sphere(1),
});
body.position.y = 100;
world.addBody(body);

// 网格小球
const geometry = new THREE.SphereGeometry(1);
const material = new THREE.MeshLambertMaterial({
    color: 0x00ffff,
});
const mesh = new THREE.Mesh(geometry, material);
mesh.position.y = 100;
```

渲染循环更新小球位置

执行 `world.step()` 会更新计算物理小球body的下落位置，这时候如果你想看到threejs小球Mesh下落动画，就需要把body位置同步到mesh上面，非常简单，直接复制即可 `mesh.position.copy(body.position)` 。

```js
const fixedTimeStep = 1/60;
// 渲染循环
function render() {
    world.step(fixedTimeStep);
    // 渲染循环中，同步物理球body与网格球mesh的位置
    mesh.position.copy(body.position);
    renderer.render(scene, camera);
    requestAnimationFrame(render);
}
render();
```

调节物理世界加速度

你可以尝试改变物理世界的重力加速度，对比不同加速度，threejs小球Mesh下落动画视觉效果差异。

```js
world.gravity.set(0, -9.8, 0);
```

```js
world.gravity.set(0, -50, 0);
```

你实际设置，重力加速度不一定就是设置为9.8，也可以根据需要设置不同的重力加速度大小，开发游戏或元宇宙项目，物理效果追求的是感知正确，不是物理正确，所谓感知正确，就是你的眼睛看着正常就行，并不一定非要与现实100%一致，cannonjs一般就是近似计算位置、速度。当然咱们这节课还没有结合threejs，大家看不到视觉效果，后面咱们都会结合threejs网格Mesh给大家演示。

← 2. CannonJS自由落体计算　　4. CannonJS模拟乒乓球下落反弹→