郭隆邦 2023-10-01

4. CannonJS模拟乒乓球下落反弹

接着前面小球自由落体的讲解，这节课给物理世界设置一个物理地面，用来阻止小球的下落，同时通过CannonJS模拟乒乓球下落反弹的效果。

创建物理地面

创建一个物理地面，这样小球碰到地面就不会继续下落。

为了让地面不受重力或其他物体碰撞影响，按照CannonJS的规则，可以把质量设置为0就行。

提醒：创建物理地面Plane姿态角度改变可以参考threej的矩形平面网格模型。

```js
// 物理地面
const groundBody = new CANNON.Body({
    mass: 0, // 质量为0，始终保持静止，不会受到力碰撞或加速度影响
    shape:new CANNON.Plane()
});
// 改变平面默认的方向，法线默认沿着z轴，旋转到平面向上朝着y方向
//旋转规律类似threejs 平面
groundBody.quaternion.setFromEuler(-Math.PI / 2, 0, 0);
world.addBody(groundBody);
```

设置body材质，并关联

给物理地面、物理小球body分别创建一个材质。

```js
const sphereMaterial = new CANNON.Material()
// 物理小球
const body = new CANNON.Body({
    mass: 0.3,//碰撞体质量
    material:sphereMaterial//碰撞体材质
});
// 物理地面
const groundMaterial = new CANNON.Material()
const groundBody = new CANNON.Body({
```

```
    mass: 0, // 质量为0，始终保持静止，不会受到力碰撞或加速度影响
    material: groundMaterial,//地面材质
});
```

通过材质设置物理地面与物理球的碰撞特点，比如碰撞的反弹恢复系数

`restitution` 的范围一般是 `0~1` 之间选择一个值，一般弹性越大 `restitution` 的值也大，比如乒乓球相比橡皮泥反弹能力就更强。

```js
const contactMaterial = new CANNON.ContactMaterial(groundMaterial, sphereMateria
    restitution: 0.7, //反弹恢复系数
})
// 把关联的材质添加到物理世界中
world.addContactMaterial(contactMaterial)
```

完整代码

```js
import * as THREE from 'three';
import {
    OrbitControls
} from 'three/addons/controls/OrbitControls.js';

// 引入cannon-es
import * as CANNON from 'cannon-es';

const world = new CANNON.World();
// 设置物理世界重力加速度
// world.gravity.set(0, -9.8, 0);
world.gravity.set(0, -50, 0);

const sphereMaterial = new CANNON.Material()
// 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
    material: sphereMaterial,//碰撞体材质
    shape: new CANNON.Sphere(1.5)
});

body.position.y = 100;
world.addBody(body);

// 物理地面
```

```
const groundMaterial = new CANNON.Material()
const groundBody = new CANNON.Body({
    mass: 0, // 质量为0，始终保持静止，不会受到力碰撞或加速度影响
    shape: new CANNON.Plane(),
    material: groundMaterial,
});
// 改变平面默认的方向，法线默认沿着z轴，旋转到平面向上朝着y方向
groundBody.quaternion.setFromEuler(-Math.PI / 2, 0, 0);//旋转规律类似threejs 平面
world.addBody(groundBody);

// 设置地面材质和小球材质之间的碰撞反弹恢复系数
const contactMaterial = new CANNON.ContactMaterial(groundMaterial, sphereMateria
    restitution: 0.7, //反弹恢复系数
})
// 把关联的材质添加到物理世界中
world.addContactMaterial(contactMaterial)

// 网格小球
const geometry = new THREE.SphereGeometry(1.5);
const material = new THREE.MeshLambertMaterial({
    color: 0xffff00,
});
const mesh = new THREE.Mesh(geometry, material);
mesh.position.y = 100;

// 网格地面
const planeGeometry = new THREE.PlaneGeometry(200, 200);
const texture = new THREE.TextureLoader().load('./瓷砖.jpg');
texture.wrapS = THREE.RepeatWrapping;
texture.wrapT = THREE.RepeatWrapping;
texture.repeat.set(3, 3);
const planeMaterial = new THREE.MeshLambertMaterial({
    color:0x777777,
    map: texture,
});
const planeMesh = new THREE.Mesh(planeGeometry, planeMaterial);
planeMesh.rotateX(-Math.PI / 2);

//场景
const scene = new THREE.Scene();
scene.add(mesh, planeMesh); //模型对象添加到场景中


//辅助观察的坐标系
const axesHelper = new THREE.AxesHelper(100);
scene.add(axesHelper);
```

```
//光源设置
const directionalLight = new THREE.DirectionalLight(0xffffff, 1);
directionalLight.position.set(100, 60, 50);
scene.add(directionalLight);
const ambient = new THREE.AmbientLight(0xffffff, 0.4);
scene.add(ambient);


//相机
const width = window.innerWidth;
const height = window.innerHeight;
const camera = new THREE.PerspectiveCamera(30, width / height, 1, 3000);
camera.position.set(292, 223, 185);
camera.lookAt(0, 0, 0);

// WebGL渲染器设置
const renderer = new THREE.WebGLRenderer({
    antialias: true, //开启优化锯齿
});
renderer.setPixelRatio(window.devicePixelRatio); //防止输出模糊
renderer.setSize(width, height);
document.body.appendChild(renderer.domElement);


const clock = new THREE.Clock();
// 渲染循环
function render() {
    world.step(1/60);//更新物理计算
    mesh.position.copy(body.position);// 网格小球与物理小球位置同步
    renderer.render(scene, camera);
    requestAnimationFrame(render);
}
render();


const controls = new OrbitControls(camera, renderer.domElement);

// 画布跟随窗口变化
window.onresize = function () {
    renderer.setSize(window.innerWidth, window.innerHeight);
    camera.aspect = window.innerWidth / window.innerHeight;
    camera.updateProjectionMatrix();
};
```

← 3. 练习-threejs可视化cannon计算结果

5. 练习-修改小球参数→