

## 🟡 8. 长方体Box碰撞体(箱子下落)

前面给大家介绍过球形 `Sphere` 碰撞体,本节课给大家介绍一个长方体 `Box` 形状碰撞体,并用 `Box`来模拟一个箱子下落效果。

```
// 球形碰撞体
const body = new CANNON.Body({
  mass: 0.3,
  shape: new CANNON.Sphere(1)
});
```

js

下面代码是在前面小球下落基础上修改,其他不变。

### 长方体 `CANNON.Box`

`CANNON.Box` 表示一个长方体,参数是一个三维向量。三维向量的三个参数分别是表示长方体xyz方向的一半尺寸。

`Box`坐标原点默认与长方体几何中心重合。

```
// x、y、z三个方向的尺寸(长宽高),分别为1.0、0.4、0.6
new CANNON.Box(new CANNON.Vec3(0.5, 0.2, 0.3))
```

js

### 长方体Box形状Body表示物理箱子

课件代码已经预先提供了一个长方体网格Mesh,用来表示箱子的外观效果,箱子在x、y、z三个方向尺寸分别是1.0、0.4、0.6。

```
// 网格长方体
const geometry = new THREE.BoxGeometry(1.0,0.4,0.6);
const material = new THREE.MeshLambertMaterial({
  map: new THREE.TextureLoader().load('./箱子.jpg'),
});
const mesh = new THREE.Mesh(geometry, material);
```

js

```
mesh.position.y = 5;
```

Box形状的碰撞体Body表示物理箱子，高度与箱子Mesh一致设置为5m。

```
const world = new CANNON.World();
world.gravity.set(0, -9.8, 0);

const boxMaterial = new CANNON.Material()
// 物理长方体
const body = new CANNON.Body({
  mass: 0.3, // 碰撞体质量
  material: boxMaterial, // 碰撞体材质
  // x、y、z方向尺寸1.0、0.4、0.6
  shape: new CANNON.Box(new CANNON.Vec3(0.5, 0.2, 0.3))
});
body.position.y = 5;
world.addBody(body);
```

js

## 箱子与弹性碰撞设置

你可以对比下面两个弹性碰撞系数的反弹差异。

### 1. 原来乒乓球的弹性碰撞设置

```
new CANNON.ContactMaterial(groundMaterial, boxMaterial, {
  restitution: 0.7,
})
```

js

### 2. 如果长方体表示一个箱子，反弹恢复系数就低一点，不用像乒乓球那么高

```
new CANNON.ContactMaterial(groundMaterial, boxMaterial, {
  restitution: 0.2,
})
```

js

## 旋转长方体

箱子换个姿态角度下落，观察下落效果差异。

```
// 网格箱子旋转
// 设置箱子下落的初始姿态角度
mesh.rotation.set(Math.PI / 3, Math.PI / 3, Math.PI / 3);
```

js

箱子碰撞体body绕xyz方向分别旋转60度，这样就是箱子倾斜角着地。

```
body.quaternion.setFromEuler(Math.PI / 3, Math.PI / 3, Math.PI / 3);
```

js

## 同步箱子Mesh和Body姿态角度 `.quaternion`

在渲染循环中不仅仅要同步箱子网格模型Mesh与物理碰撞体Body的位置 `.position`，也同步两者的姿态角度 `.quaternion`。

```
// 渲染循环
function render() {
    world.step(1/60);
    // 网格mesh与body位置同步
    mesh.position.copy(body.position);
    //同步姿态角度
    mesh.quaternion.copy(body.quaternion);
    renderer.render(scene, camera);
    requestAnimationFrame(render);
}
render();
```

js

## 注意

除了箱子网格模型Mesh与物理碰撞体Body的位置 `.position` 一样之外，两者的几何中心也一致，这样才能保证两者重合。

如果长方体mesh与body不完全重合，比如位置不一致，比如姿态角度不一致，比如尺寸不一致，你通过下落测试就能观察出来。

```
// 网格长方体
const geometry = new THREE.BoxGeometry(1.0,0.4,0.6);
const mesh = new THREE.Mesh(geometry, material);
mesh.position.y = 5;
```

js

```
// 物理长方体
const body = new CANNON.Body({
  mass: 0.3, //碰撞体质量
  material: boxMaterial, //碰撞体材质
  // x、y、z方向尺寸1.0、0.4、0.6
  shape: new CANNON.Box(new CANNON.Vec3(0.5, 0.2, 0.3))
});
body.position.y = 5;
```

---

← 7. CannonJS碰撞事件, 碰撞声音

9. 练习题-外部glTF箱子模型→