

🔗 2. 八叉树与胶囊Capsule交叉计算

Capsule 表示胶囊形状的几何体，具体说就是上面一个半球、中间一个圆柱、下面一个半球拼接构成的胶囊形状几何体。

下面给大家讲解胶囊形状几何体apsule与八叉树表示的3D模型进行交叉计算，你可以类比以前的射线交叉计算，虽然不同，都是都是交叉相关的计算。

项目引入胶囊碰撞体 **Capsule.js**

Three.js在目录 `/examples/jsm/math/` 下提供了一个胶囊形状的几何体 **Capsule.js**。

npm安装threejs情况下，Capsule.js扩展库引入路径。

```
// 引入/examples/jsm/math/目录下胶囊扩展库Capsule.js
import { Capsule } from 'three/examples/jsm/math/Capsule.js';
```

js

本课程案例源码 `.html` 里面自定义了新的路径。

```
import { Capsule } from 'three/addons/math/Capsule.js';
```

js

创建胶囊几何体

创建胶囊几何体：让胶囊底部半球与 $y=0$ 的平面刚好相切即可。

```
const R = 0.4; // 胶囊半径
const H = 1.7; // 胶囊总高度
const start = new THREE.Vector3(0, R, 0); // 底部半球球心坐标
const end = new THREE.Vector3(0, H - R, 0); // 顶部半球球心坐标
const capsule = new Capsule(start, end, R);
console.log('capsule', capsule);
```

js

练习：Mesh可视化上面胶囊几何体

```
// 可视化胶囊几何体
const capsuleHelper = CapsuleHelper(R, H);
model.add(capsuleHelper);
```

js

```
function CapsuleHelper(R, H) {
  const group = new THREE.Group();
  const material = new THREE.MeshLambertMaterial({
    color: 0x00ffff,
    transparent: true,
    opacity: 0.5,
  });
  // 底部半球
  const geometry = new THREE.SphereGeometry(R, 25, 25, 0, 2 * Math.PI, 0, Math.PI);
  geometry.rotateX(Math.PI);
  const mesh = new THREE.Mesh(geometry, material);
  mesh.position.y = R;
  group.add(mesh)
  // 顶部半球
  const geometry2 = new THREE.SphereGeometry(R, 25, 25, 0, 2 * Math.PI, 0, Math.PI);
  const mesh2 = new THREE.Mesh(geometry2, material);
  mesh2.position.set(0, H - R, 0);
  group.add(mesh2)
  // 中间圆柱
  const h = H - 2 * R;
  const geometry3 = new THREE.CylinderGeometry(R, R, h, 32, 1, true);
  geometry3.translate(0, h / 2 + R, 0);
  const mesh3 = new THREE.Mesh(geometry3, material);
  group.add(mesh3)
  return group;
}
```

js

平移胶囊几何体

平移胶囊碰撞体,使底部半球位于 $y=0$ 的平面以下。

```
capsule.translate(new THREE.Vector3(0, -R, 0));
```

js

可视化胶囊的模型对象同步平移

```
capsuleHelper.position.y += -R;
```

js

换一种平移方式

```
capsuleHelper.position.copy(capsule.start)
capsuleHelper.position.y -= R;
```

js

交叉计算

`Octree.capsuleIntersect(capsule)` 可以计算Octree表示的3D模型与胶囊几何体capsule是否重合交叉，如果有重合交叉，返回交叉相关的信息，具体说就是在某个方向上交叉重合的深度是多少。

```
// 碰撞检测：几何体交叉计算
// Octree表示的3D模型和Capsule交叉计算
const result = worldOctree.capsuleIntersect(capsule);
console.log('碰撞检测结果', result);
```

js

- `.depth` 交叉重合的深度
- `.normal` 深度对应的方向

先与八叉树里面的包围盒子节点进行交叉计算，在与相交叉的包围盒包含的三角形进行交叉计算。借助八叉树，相比较，for循环所有模型所有三角形分别进行交叉计算，更节约时间。

胶囊放在斜面上

平移胶囊放在斜面上，查看交叉重合计算结果。

```
capsule.translate(new THREE.Vector3(0, 0, -3*H));
capsuleHelper.position.z += -3 * H;
```

js

你可以看到交叉方向不再是垂直于平面

- `.depth` 交叉重合的深度

- `.normal` 深度对应的方向

根据交叉碰撞数据，平移碰撞体

根据交叉碰撞数据，平移碰撞体，让胶囊碰撞体不在于八叉树对应模型重合。

- `.depth` 交叉重合的深度
- `.normal` 深度对应的方向

总结：`.normal` 数据的特点就是让胶囊沿着 `.normal` 方向，平移 `.depth` 距离，就能刚好确保交叉重合深度为0

```
// 根据碰撞结果平移胶囊碰撞体，使交叉重合深度为0
capsule.translate(result.normal.multiplyScalar(result.depth));
capsuleHelper.position.copy(capsule.start);
capsuleHelper.position.y -= R;
```

js

胶囊与楼梯交叉

```
// 根据碰撞结果平移胶囊碰撞体，使交叉重合深度为0
capsule.translate(result.normal.multiplyScalar(result.depth));
capsuleHelper.position.copy(capsule.start);
capsuleHelper.position.y -= R;
```

js

平移后，不在于楼梯交叉，上升偏离地面了(如果运动起来，这个特点可以让胶囊产生上楼梯的效果，下节课会讲解)

胶囊向上平移，与地面没有接触，查看交叉计算结果

胶囊向上平移，与地面没有接触

```
// 胶囊向上平移，与地面不交叉情况，查看计算结果
capsule.translate(new THREE.Vector3(0, R, 0));
capsuleHelper.position.y += R;
```

js

交叉计算结果返回值 `false`

```
const result = worldOctree.capsuleIntersect(capsule);  
console.log('碰撞检测结果', result);
```

js

← 1. 八叉树Octree扩展库介绍

1. 物理引擎CannonJS简介和引入 →