郭隆邦 2023-10-01

2. CannonJS自由落体计算

正式使用物理引擎CannonJS与Threejs结合之前，先用CannonJS模拟一个小球自由落体运动的物理计算。

物理引擎概念解释

所谓**物理引擎**，就是通过代码模拟物理世界。举个简单例子，比如你初高中都学过物理学，其中**力**、**速度**、**加速度**、**位移**都是比较常见的物理量，咱们通过CannonJS等物理引擎，都可以辅助你计算生活中物体的**速度**、**位移**，比如计算一个小球在地球重力的作用下，下落的速度和位置。

引入物理引擎cannon-es

上节课给大家讲解过，怎么引入物理引擎cannon-es。

```
import * as CANNON from "cannon-es";
```

碰撞体 Body

通过 `CANNON.Body` 类，可以创建一个用于物体物理模拟计算，比如用 `Body` 表示一个球、一个箱子、一张桌子，你可以计算 `Body` 的位置、速度等物理量。

你可以也把 `Body` 称为碰撞体collider。

```
const body = new CANNON.Body();
```

设置 Body 的物理属性

设置 `Body` 的一些物理属性，比如质量 `mass`

```
const body = new CANNON.Body({
    mass: 0.3, // 碰撞体质量0.3kg
});
```

设置物体body的位置，CannonJS的三维向量`Vec3`和threejs名称不同，不过使用方式相似。

```
const body = new CANNON.Body({
    mass: 0.3,
     // 碰撞体的三维空间中位置
    position: new CANNON.Vec3(0, 100, 0)
});
```

碰撞体`Body`几何形状

第一次接触`Body`，你可以类比threejs的Mesh去联想记忆，网格模型表示一个物体，需要通过几何体Geometry定义Mesh的几何外形，对于Body同样道理，你需要设置物体`Body`的几何形状。

CannonJS定义几何体形状的API有很多种，比如比如长方体`Box`、球体`Sphere`等等，本节课先给大家介绍球体`Sphere`。

```
// 1m半径球体
const bodyShape = new CANNON.Sphere(1);
// 可以把Body称为碰撞体,用来模拟生活中的物体
const body = new CANNON.Body({
    mass: 0.3,
    position: new CANNON.Vec3(0, 100, 0),
    shape: bodyShape,//碰撞体的几何体形状
});
```

`CANNON.World`创建一个物理世界

通过`CANNON.World`类创建一个物理世界。

```
// CANNON.World创建物理世界对象
const world = new CANNON.World();
```

定义物理世界的物理属性，比如设置重力加速度。

重力加速度的属性`gravity`类似body的位置，是一个三维向量`Vec3`。

重力加速度x、y、z分量值，实际开发根据自己项目和坐标系设置即可，咱们假设小球所在的场景，y轴竖直向上，这样重力就速度就是y方向负方向。

```
const world = new CANNON.World();
// 设置物理世界重力加速度
world.gravity.set(0, -9.8, 0); //单位：m/s²
```

`.addBody()`把物体添加到物理世界

物理球body添加到物理世界中，这样body就会受到物理世界加速度的影响World。

```
const world = new CANNON.World();
world.addBody(body);
```

`world.step()`物理世界更新计算

最后不要忘记周期性执行`world.step()`方法，`world.step()`方法第一个参数表示固定时间步长，一般可以设置为`1/60`秒，用于近似计算。

```
function render() {
    world.step(1/60);//更新物理计算
    requestAnimationFrame(render);
}
render()
```

```
//固定的时间步长1/60秒
const fixedTimeStep = 1/60;
function render() {
    world.step(fixedTimeStep);
}
```

浏览器控制台查看计算结果

你可以查看物体对象body的位置`.position`、速度`.velocity`属性。

```
function render() {
    console.log('球位置', body.position);
    console.log('球速度', body.velocity);
    console.log('y方向球位置', body.position.y);
    world.step(1/60);//更新物理计算
    requestAnimationFrame(render);
}
```

浏览器控制辅助开发

除了查看文档，你还可以通过浏览器控制台，查看CannonJS某个类的属性或方法。

`Body`具有位置`.position`、重量`.mass`、几何形状`.shapes`等属性

```
const body = new CANNON.Body();
console.log('body', body);
```

```
const body = new CANNON.Body({
    mass: 0.3,
    position: new CANNON.Vec3(0, 100, 0),
    shape: new CANNON.Sphere(1),
});
```

比如`body.position`位置属性的属性值是三维向量Vec3，Vec3具有`x`、`y`、`z`属性和`set`等多个方法。

```
const body = new CANNON.Body();
console.log('body.position', body.position);
```

语法总结：访问或设置`Body`属性

通过函数选项设置body对象

```
const body = new CANNON.Body({
    mass: 0.3,
    position: new CANNON.Vec3(0, 100, 0),
```

```
    shape: new CANNON.Sphere(0.1),
});
```

部分属性也可以直接访问设置。

```
const body = new CANNON.Body();
body.mass = 0.3;
body.position = new CANNON.Vec3(0, 100, 0);
```

body没有 `.shape` 属性，而是 `.shapes` 属性， `Body` 类提供了方法 `.addShape()` 设置几何体，执行 `.addShape()` 方法会改变 `.shapes` 属性。

```
const body = new CANNON.Body();
body.addShape(new CANNON.Sphere(1));
```

← 1. 物理引擎CannonJS简介和引入

3. 练习-threejs可视化cannon计算结果→