郭隆邦 2023-09-28

1. 物理引擎CannonJS简介和引入

JavaScript物理模拟引擎还是比较多的，比如Ammo.js、Physi.js、Cannon.js，这些引擎虽然语法细节有差异，但是在3D应用中开发思路是相似的。

本课程就以Cannon.js为例给大家讲解**three.js**和**物理引擎**的结合。

github资源cannon.js

github资源cannon.js：https://github.com/schteppe/cannon.js

cannon.js**文档**：可以在本地静态服务器打开`cannon.js\docs\index.html`预览Canonjs引擎的文档。

cannon.js**案例**：`cannon.js\examples\`和`\demos\`目录下可以看到一些cannonjs和three.js结合的一些小例子。

github资源cannon-es

cannon-es对**cannon.js**进行了重写，补充支持了ES6和Typescript语法。

不过除了es语法版本问题，也要注意一点就是**cannon-es**也改变了**cannon.js**部分API写法，这一点提醒大家，你查看别人文档或代码一定注意，别人用的**cannon.js**还是**cannon-es**。

github资源cannon-es：https://github.com/dreammonkey/cannon-es

cannon-es在线文档：https://pmndrs.github.io/cannon-es/docs/index.html

cannon-es在线案例：https://pmndrs.github.io/cannon-es/

cannon-es**案例**：`cannon-es\examples\`目录下可以看到一些cannonjs和three.js结合的一些小例子。

本课程使用**cannon-es**给大家讲解CannonJS的使用。

cannon-es安装和引入

在工程化开发的时候可以通过npm命令行安装cannon.js模块。

```
npm install --save cannon-es
```

```
// 某个API
import {World, Vec3} from "cannon-es";
// 全部API一次性引入
import * as CANNON from "cannon-es";
```

咱们课件中是在 `.html` 文件中直接引入的cannon-es，实际开发，用上面npm安装方式引入即可。

```html
<script type="importmap">
    {
            "imports": {
                "cannon-es": "../cannon-es/dist/cannon-es.js"
            }
        }
    </script>
  <script type="module">
    import * as CANNON from 'cannon-es';
    // 测试是否引入成功
    console.log('CANNON', CANNON.World);

    import { World } from 'cannon-es';
    console.log('World', World);
  </script>
```

← 2. 八叉树与胶囊Capsule交叉计算

2. CannonJS自由落体计算→