

## 🟡 13. stats查看threejs渲染帧率

视频讲解 [🔗](#)

three.js每执行WebGL渲染器 `.render()` 方法一次，就在canvas画布上得到一帧图像，不停地周期性执行 `.render()` 方法就可以更新canvas画布内容，一般场景越复杂往往渲染性能越低，也就是每秒钟执行 `.render()` 的次数越低。

通过stats.js库可以查看three.js当前的渲染性能，具体说就是计算three.js的渲染帧率(FPS),所谓渲染帧率(FPS)，简单说就是three.js每秒钟完成的渲染次数，一般渲染达到每秒钟60次为最佳状态。

stats.js下载链接：<https://github.com/mrdoob/stats.js>

### 引入Stats

```
//引入性能监视器stats.js
import Stats from 'three/addons/libs/stats.module.js';
```

js

### Stats使用

```
//创建stats对象
const stats = new Stats();
//stats.domElement:web页面上输出计算结果,一个div元素,
document.body.appendChild(stats.domElement);
// 渲染函数
function render() {
  //requestAnimationFrame循环调用的函数中调用方法update(),来刷新时间
  stats.update();
  renderer.render(scene, camera); //执行渲染操作
  requestAnimationFrame(render); //请求再次执行渲染函数render，渲染下一帧
}
render();
```

## stats方法setMode(mode)

可以通过 `setMode()` 方法的参数mode的数值设置首次打开页面，测试结果的显示模式，鼠标单击可以更换不同的显示模式。

```
// stats.domElement显示：渲染帧率 刷新频率,一秒渲染次数
stats.setMode(0); //默认模式
//stats.domElement显示：渲染周期 渲染一帧多长时间(单位：毫秒ms)
stats.setMode(1);
```

## 性能测试

控制长方体模型数量，你可以逐渐增加或减少,看看帧率变化，电脑性能不同结果不同。

```
// 随机创建大量的模型,测试渲染性能
const num = 1000; //控制长方体模型数量
for (let i = 0; i < num; i++) {
  const geometry = new THREE.BoxGeometry(5, 5, 5);
  const material = new THREE.MeshLambertMaterial({
    color: 0x00ffff
  });
  const mesh = new THREE.Mesh(geometry, material);
  // 随机生成长方体xyz坐标
  const x = (Math.random() - 0.5) * 200
  const y = (Math.random() - 0.5) * 200
  const z = (Math.random() - 0.5) * 200
  mesh.position.set(x, y, z)
  scene.add(mesh); // 模型对象插入场景中
}
```

