郭隆邦 2023-01-26

2. 遍历模型树结构、查询模型节点

上节课说过Threejs场景对象Scene和各种子对象构成的层级模型就是一个树结构。如果你有一定的算法基础对树结构肯定会非常了解，如果你了解前端的DOM树结构也非常有助于本节课的学习，如果这些都不了解也没有关系，直接体验本节课的案例源码。

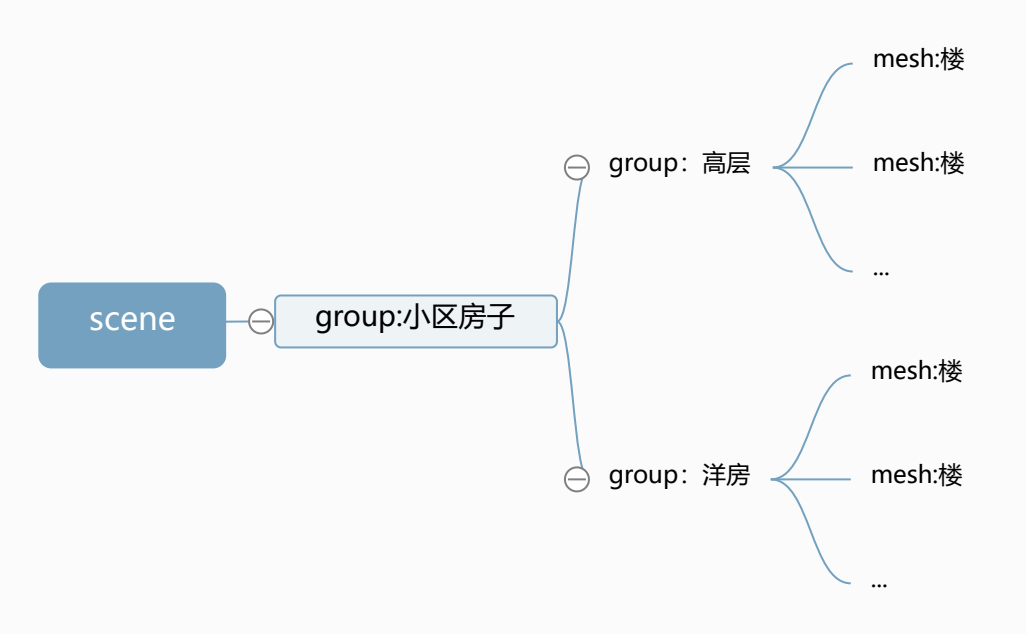
模型命名(.name 属性)

在层级模型中可以给一些模型对象通过 .name 属性命名进行标记。

```js
const group = new THREE.Group();
group.name='小区房子';
const mesh = new THREE.Mesh(geometry, material);
mesh.name='一号楼';
```

树结构层级模型设置 .name 属性

下面是通过代码创建了一个层级模型，一般实际开发的时候，会加载外部的模型，然后从模型对象通过节点的名称 .name 查找某个子对象。



```js
// 批量创建多个长方体表示高层楼
const group1 = new THREE.Group(); //所有高层楼的父对象
group1.name = "高层";
for (let i = 0; i < 5; i++) {
    const geometry = new THREE.BoxGeometry(20, 60, 10);
    const material = new THREE.MeshLambertMaterial({
        color: 0x00ffff
    });
    const mesh = new THREE.Mesh(geometry, material);
    mesh.position.x = i * 30; // 网格模型mesh沿着x轴方向阵列
    group1.add(mesh); //添加到组对象group1
    mesh.name = i + 1 + '号楼';
    // console.log('mesh.name',mesh.name);
}
group1.position.y = 30;


const group2 = new THREE.Group();
group2.name = "洋房";
// 批量创建多个长方体表示洋房
for (let i = 0; i < 5; i++) {
    const geometry = new THREE.BoxGeometry(20, 30, 10);
    const material = new THREE.MeshLambertMaterial({
        color: 0x00ffff
    });
    const mesh = new THREE.Mesh(geometry, material);
    mesh.position.x = i * 30;
    group2.add(mesh); //添加到组对象group2
    mesh.name = i + 6 + '号楼';
}
group2.position.z = 50;
group2.position.y = 15;

const model = new THREE.Group();
model.name='小区房子';
model.add(group1, group2);
model.position.set(-50,0,-25);
```

递归遍历方法 .traverse()

Threejs层级模型就是一个树结构，可以通过递归遍历的算法去遍历Threejs一个模型对象包含的所有后代。

```js
// 递归遍历model包含所有的模型节点
model.traverse(function(obj) {
    console.log('所有模型节点的名称',obj.name);
    // obj.isMesh：if判断模型对象obj是不是网格模型'Mesh'
    if (obj.isMesh) {//判断条件也可以是obj.type === 'Mesh'
        obj.material.color.set(0xffff00);
    }
});
```

查找某个具体的模型 .getObjectByName()

看到Object3D的 `.getObjectByName()` 方法，如果已有前端基础，很容易联想到DOM的一些方法。

Threejs和前端DOM一样，可以通过一个方法查找树结构父元素的某个后代对象，对于普通前端而言可以通过name或id等方式查找一个或多个DOM元素，Threejs同样可以通过一些方法查找一个模型树中的某个节点。更多的查找方法和方法的使用细节可以查看基类Object3D。

```js
// 返回名.name为"4号楼"对应的对象
const nameNode = scene.getObjectByName ("4号楼");
nameNode.material.color.set(0xff0000);
```

← 1. Vector3与模型位置、缩放属性

3. 本地坐标和世界坐标→