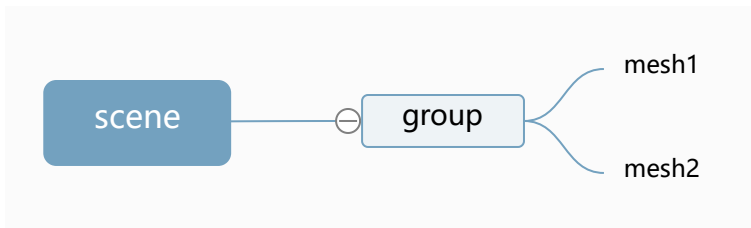


🟡 1. Vector3与模型位置、缩放属性

本节课给通过组对象Group📦 给大家讲解一下threejs层级模型或树结构的概念。

Group层级模型(树结构)案例



下面代码创建了两个网格模型mesh1、mesh2，通过 `THREE.Group` 类创建一个组对象group，然后通过 `add` 方法把网格模型mesh1、mesh2作为设置为组对象group的子对象，然后在通过执行 `scene.add(group)` 把组对象group作为场景对象的scene的子对象。也就是说场景对象是scene是group的父对象，group是mesh1、mesh2的父对象。这样就构成了一个三层的层级结构，当然了你也可以通过 `Group` 自己创建新模型节点作为层级结构中的一层。

```
//创建两个网格模型mesh1、mesh2
const geometry = new THREE.BoxGeometry(20, 20, 20);
const material = new THREE.MeshLambertMaterial({color: 0x00ffff});
const group = new THREE.Group();
const mesh1 = new THREE.Mesh(geometry, material);
const mesh2 = new THREE.Mesh(geometry, material);
mesh2.translateX(25);
//把mesh1型插入到组group中，mesh1作为group的子对象
group.add(mesh1);
//把mesh2型插入到组group中，mesh2作为group的子对象
group.add(mesh2);
//把group插入到场景中作为场景子对象
scene.add(group);
```

js

查看子对象 `.children`

Threejs场景对象Scene、组对象Group都有一个子对象属性 `.children` ,通过该属性可以访问父对象的子对象,子对象属性 `.children` 的值是数组,所有子对象是数组的值,你可以在浏览器控制台打印测试上面案例代码。

父对象执行 `.add()` 方法的本质就是把参数中的子对象添加到自身的子对象属性 `.children` 中。

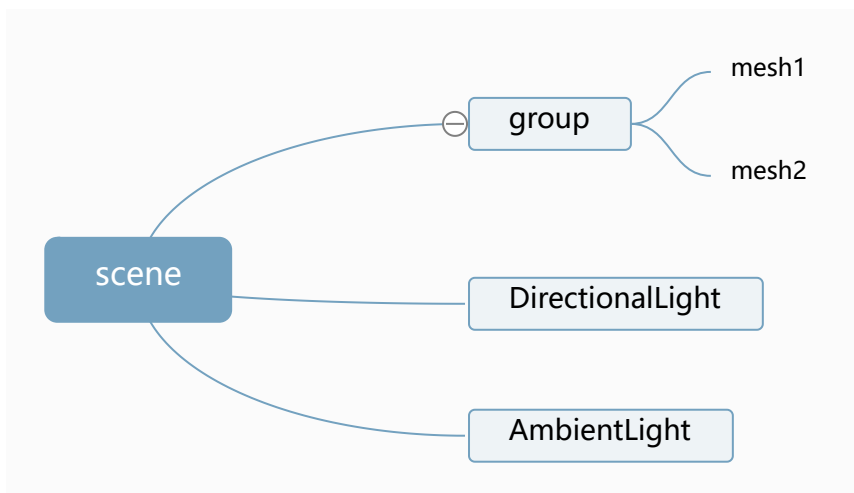
```
console.log('查看group的子对象',group.children);
```

js

场景对象结构

```
console.log('查看Scene的子对象',scene.children);
```

js



场景对象Scene的子对象,除了组对象 `Group` 之外,还可以看到环境光 `AmbientLight` 、平行光 `DirectionalLight` 、辅助坐标对象 `AxesHelper` 。

场景对象对象scene构成的层级模型本身是一个树结构,场景对象层级模型的第一层,也就是树结构的根节点,一般来说网格模型Mesh、点模型Points、线模型Line是树结构的最外层叶子结点。构建层级模型的中间层一般都是通过Threejs的 `Group` 类来完成, `Group` 类实例化的对象可以称为组对象。

`.add()` 方法总结

场景对象 `Scene` 、组对象 `Group` 的 `.add()` 方法都是继承自它们共同的基类(父类) `Object3D` 。

```
group.add(mesh1);
group.add(mesh2);
```

js

.add() 方法可以单独插入一个对象，也可以同时插入多个子对象。

```
group.add(mesh1,mesh2);
```

js

父对象旋转缩放平移变换，子对象跟着变化

网格模型mesh1、mesh2作为设置为父对象group的子对象，如果父对象group进行旋转、缩放、平移变换，子对象同样跟着变换，就像你的头旋转了，眼睛会跟着头旋转。

```
//沿着Y轴平移mesh1和mesh2的父对象，mesh1和mesh2跟着平移
group.translateY(100);
```

js

```
//父对象缩放，子对象跟着缩放
group.scale.set(4,4,4);
```

js

```
//父对象旋转，子对象跟着旋转
group.rotateY(Math.PI/6)
```

js

Object3D 表示模型对象节点

受threejs历史影响，你会在很多别的代码中看到 Object3D 作为 Group 来使用，如果看到不用奇怪。某种程度上，你可把两者画等号，只是 Group 更加语义化，Object3D本身就是表示模型节点的意思。

```
const mesh1 = new THREE.Mesh(geometry, material);
const mesh2 = new THREE.Mesh(geometry, material);
const obj = new THREE.Object3D();//作为mesh1和mesh2的父对象
obj.add(mesh1,mesh2);
```

js

mesh也能添加mesh子对象

threejs默认mesh也可以添加子对象,其实原因很简单, mesh和Group父类都是Object3D, 本质上也可以认为都是Object3D。

```
//threejs默认mesh也可以添加子对象,mesh基类也是Object3D  
mesh1.add(mesh2);
```

js

← 6. 克隆.clone()和复制.copy()

2. 遍历模型树结构、查询模型节点 →