

🟡 8. 外部模型材质是否共享的问题

美术通过三维建模软件，比如Blender绘制好一个三维场景以后，一些外观一样的Mesh，可能会共享一个材质对象。

改变一个模型颜色其它模型跟着变化

由于楼房的Mesh共享了1号楼Mesh的材质，当你通过mesh1.material改变mesh1材质，本质上是改变所有楼Mesh的材质。

```
const mesh1 = gltf.scene.getObjectByName("1号楼");  
//1. 改变1号楼Mesh材质颜色  
mesh1.material.color.set(0xff0000);
```

js

.name 标记材质，判断两个mesh是否共享材质

通过 .name 标记材质，测试mesh1和mesh2是否共享了材质

```
const mesh1 = gltf.scene.getObjectByName("1号楼");  
mesh1.material.name = '楼房材质';//通过name标记mesh1对应材质  
const mesh2 = gltf.scene.getObjectByName("2号楼");  
//通过name相同，可以判断mesh1.material和mesh2.material共享了同一个材质对象  
console.log('mesh2.material.name', mesh2.material.name);
```

js

解决问题方向

改变一个模型颜色其它模型跟着变化，是因为多个模型对象共享了材质，如果单独改变一个模型的材质，比如颜色，下面两个方案，可以任选其一。

1. 三维建模软件中设置，需要代码改变材质的Mesh不要共享材质，要独享材质。
2. 代码批量更改：克隆材质对象，重新赋值给mesh的材质属性

代码方式解决多个mesh共享材质的问题

js

//用代码方式解决mesh共享材质问题

```
gltf.scene.getObjectByName("小区房子").traverse(function (obj) {  
    if (obj.isMesh) {  
        // .material.clone()返回一个新材质对象，和原来一样，重新赋值给.material属性  
        obj.material = obj.material.clone();  
    }  
});  
mesh1.material.color.set(0xffff00);  
mesh2.material.color.set(0x00ff00);
```

← [7. 递归遍历层级模型修改材质](#)

[9. 纹理encoding和渲染器](#) →