

🟡 7. 三维软件导出PBR材质属性

实际开发的时候PBR材质的属性，很多时候是可以在三维建模软件中设置的，然后通过glTF导出即可，这样就不用在三.js代码设置。

通常美术对三维场景渲染的了解也比大部分前端程序员多的多，只要美术在三维建模软件设置好并导出包含pbr材质属性的glTF即可。

three.js与建模软件对接的问题

1. glTF能否存储3D建模软件的某个材质属性：有些三维软件特有的材质属性，不一定能通过glTF导出，也谈不上three.js解析
2. 三维建模能否导出PBR材质：能导出的话，能导出哪些属性，不能导出哪些属性

如果你的三维建模不能导出pbr材质，或者部分pbr材质属性无法导出，那你通常需要用代码方式添加材质，这样就麻烦些。

Blender导出PBR材质演示

首先Blender最新版导出glTF模型时候，是可以把PBR材质的很多属性导出的，比如金属度 `metalness` 、粗糙度 `roughness` 、清漆 `.clearcoat` 、透光率(透射度) `.transmission` 等等。课件源码中提供了blender导出的glTF模型你可以浏览器控制台打印测试，这些PBR材质属性能否解析渲染。

Blender中设置PBR材质

你可以在Blender中设置车外壳、车玻璃的材质属性

1. 车外壳：清漆、清漆粗糙度
2. 车玻璃：透光率(透射度)

three.js解析glTF材质规则

大家都知道, `MeshPhysicalMaterial` 是 `MeshStandardMaterial` 的子类, 具有更多的PBR材质属性和功能。

所以, threejs解析glTF模型, 会用两种材质PBR材质去解析, 一个是标准网格材质 `MeshStandardMaterial`, 一个是物理网格材质 `MeshPhysicalMaterial`, 如果能用 `MeshStandardMaterial` 表示就用, 不能就换 `MeshPhysicalMaterial`。

具体说就是, threejs解析glTF模型材质的时候, 一般默认使用标准网格材质 `MeshStandardMaterial`, 如果glTF有的材质具有 `.clearcoat`、`.transmission` 等属性, 标准网格材质 `MeshStandardMaterial` 无法表达的时候, 会用物理网格材质 `MeshPhysicalMaterial` 来解析glTF材质。

查看threejs解析的PBR材质

```
glTF.scene.traverse(function(obj) {  
    if (obj.isMesh) {  
        console.log('obj.material',obj.material);  
    }  
});  
console.log('外壳',mesh1.material);  
console.log('玻璃',mesh2.material);
```

js

设置环境贴图

这时候清漆、清漆粗糙度、透光率(透射度)等属性Bledner都已经设置好了, threejs可以自动解析渲染,不用在代码中麻烦设置了, 只要配上环境贴图即可。

```
const mesh1 = glTF.scene.getObjectByName('外壳01');  
mesh1.material.envMap = textureCube; //环境贴图  
mesh1.material.envMapIntensity = 1.0; /////环境贴图对Mesh表面影响程度  
const mesh2 = glTF.scene.getObjectByName('玻璃01');  
mesh2.material.envMap = textureCube; //环境贴图  
mesh2.material.envMapIntensity = 1.0; /////环境贴图对Mesh表面影响程度
```

js

