

🟡 6. 深度冲突(模型闪烁)

本节课通过一个比较简单的案例，来给大家展示一下，实际开发的过程中，你可能会遇到的模型闪烁问题。

对于模型闪烁的原因简单地说就是**深度冲突**，对应的英文关键词是 **Z-fighting** 。

Mesh面重合渲染测试

下面代码创建两个重合的矩形平面Mesh，通过浏览器预览，当你旋转三维场景的时候，你会发现模型渲染的时候产生闪烁。

这种现象，主要是两个Mesh重合，电脑GPU分不清谁在前谁在后，这种现象，可以称为深度冲突 **Z-fighting** 。

```
// 两个矩形平面Mesh重合，产生闪烁
// 闪烁原因：两个矩形面位置重合，GPU无法分清谁在前谁在后
const geometry = new THREE.PlaneGeometry(250, 250);
const material = new THREE.MeshLambertMaterial({
  color: 0x00ffff,
  side: THREE.DoubleSide,
});
const mesh = new THREE.Mesh(geometry, material);

const geometry2 = new THREE.PlaneGeometry(300, 300);
const material2 = new THREE.MeshLambertMaterial({
  color: 0xff6666,
  side: THREE.DoubleSide,
});
const mesh2 = new THREE.Mesh(geometry2, material2);
```

js

两个矩形Mesh拉开距离

适当偏移，解决深度冲突，偏移尺寸相对模型尺寸比较小，视觉上两个平面近似还是重合效果。

```
mesh2.position.z = 1;
```

js

建模注意

上面代码测试提醒我们，在三维软件中建模的时候，尽量避免两个Mesh完全重合，可以考虑适当偏移一定的距离。

间隙很小，深度冲突

当两个面间隙很小，也可能出现深度冲突。从纯理论的角度，你能分清0和0.0000...0000001的大小，但是实际上，电脑GPU精度是有限的。

```
// 当两个面间隙很小，也可能出现深度冲突。  
mesh2.position.z = 0.000000000000000000000001;
```

js

透视投影相机对距离影响(深度冲突)

第1步：设置两个Mesh平面的距离相差0.1，课件中案例源码你可以看到，没有深度冲突导致的模型闪烁问题

```
mesh2.position.z = 0;  
mesh2.position.z = 0.1;  
camera.position.set(292, 223, 185);
```

js

第2步：改变相机 `.position` 属性，你会发现当相机距离三维模型较远的时候，两个面也可能出现深度冲突，当然你也可以通过相机控件 `OrbitControls` 缩放功能，改变相机与模型的距离，进行观察。

```
camera.position.set(292*5, 223*5, 185*5)
```

js

透视投影相机的投影规律是远小近大，和人眼观察世界一样，模型距离相机越远，模型渲染的效果越小，两个mesh之间的间距同样也会变小。当两个Mesh和相机距离远到一定程度，两个模型的距离也会无限接近0。

webgl渲染器设置对数深度缓冲区

两个矩形平面距离比较近，相差0.1

```
mesh2.position.z = 0;
mesh2.position.z = 0.1;
camera.position.set(292*5, 223*5, 185*5);
```

js

当一个三维场景中有一些面距离比较近，有深度冲突，你可以尝试设置webgl渲染器设置对数深度缓冲区 `logarithmicDepthBuffer: true` 来优化或解决。 `logarithmicDepthBuffer: true` 作用简单来说，就是两个面间距比较小的时候，让threejs更容易区分两个面，谁在前，谁在后。

```
// WebGL渲染器设置
const renderer = new THREE.WebGLRenderer({
  // 设置对数深度缓冲区，优化深度冲突问题
  logarithmicDepthBuffer: true
});
```

js

有一点要注意，当两个面间隙过小，或者重合，你设置webgl渲染器对数深度缓冲区也是无效的。

```
mesh2.position.z = 0;
//当两个面重合，logarithmicDepthBuffer: true无效
mesh2.position.z = 0;
//当两个面间隙过小，logarithmicDepthBuffer: true无效
mesh2.position.z = 0.00001;
camera.position.set(292*5, 223*5, 185*5);
```

js

← 5. Three.js渲染结果保存为图片

7. 模型加载进度条 →