郭隆邦 2023-01-28

10. gltf模型更换.map(纹理.flipY)

下面给大家演示如何给gltf的网格模型Mesh更换颜色贴图 `.map` 。

加载颜色贴图 `.map`

注意单独加载的纹理贴图的 `.encoding` 和webgl渲染器的 `.outputEncoding` 保持一致。

```js
const texLoader = new THREE.TextureLoader();
const texture = texLoader.load('./黑色.png');// 加载手机mesh另一个颜色贴图
texture.encoding = THREE.sRGBEncoding; //和渲染器.outputEncoding一样值
```

更换gltf颜色贴图

执行 `mesh.material.map = texture;` 新的纹理对象 `Texture` 赋值给 `.material.map` 就可以更换材质贴图。

```js
loader.load("../手机模型.glb", function (gltf) {
    const mesh = gltf.scene.children[0]; //获取Mesh
    mesh.material.map = texture; //更换不同风格的颜色贴图
})
```

注意：如果你直接给gltf模型材质设置 `.map` 属性更换贴图，会出现纹理贴图错位的问题，这主要和纹理对象 `Texture` 的翻转属性 `.flipY` 有关。

纹理对象 `Texture` 翻转属性 `.flipY` 默认值

`.flipY` 表示是否翻转纹理贴图在Mesh上的显示位置。

纹理对象 `Texture` 翻转属性 `.flipY` 默认值是true。

```js
// 纹理对象texture.flipY默认值
console.log('texture.flipY', texture.flipY);
```

gltf的贴图翻转属性 `.flipY` 默认值

gltf的贴图翻转属性 `.flipY` 默认值是false。

```js
loader.load("../手机模型.glb", function (gltf) {
    const mesh = gltf.scene.children[0]; //获取Mesh
    console.log('.flipY', mesh.material.map.flipY);
})
```

如果更换单独加载的纹理贴图，比如颜色贴图 `.map` ，注意把纹理贴图.flipY的值设置给gltf中纹理的值false。

```js
//是否翻转纹理贴图
texture.flipY = false;
```

← 9. 纹理encoding和渲染器

1. PBR材质简介→