郭隆邦 2023-01-27

1. 创建纹理贴图

通过纹理贴图加载器 `TextureLoader` 的 `load()` 方法加载一张图片可以返回一个纹理对象 `Texture` ，纹理对象 `Texture` 可以作为模型材质颜色贴图 `.map` 属性的值。

```js
const geometry = new THREE.PlaneGeometry(200, 100);
//纹理贴图加载器TextureLoader
const texLoader = new THREE.TextureLoader();
// .load()方法加载图像，返回一个纹理对象Texture
const texture = texLoader.load('./earth.jpg');
const material = new THREE.MeshLambertMaterial({
    // 设置纹理贴图：Texture对象作为材质map属性的属性值
    map: texture,//map表示材质的颜色贴图属性
});
```

颜色贴图属性 `.map`

也可以通过颜色贴图属性 `.map` 直接设置纹理贴图，和材质的参数设置一样。

```js
material.map = texture;
```

颜色贴图和color属性颜色值会混合

材质的颜色贴图属性 `.map` 设置后，模型会从纹理贴图上采集像素值，这时候一般来说不需要再设置材质颜色 `.color` 。 `.map` 贴图之所以称之为颜色贴图就是因为网格模型会获得颜色贴图的颜色值RGB。

颜色贴图map和color属性颜色值会混合。如果没有特殊需要，设置了颜色贴图.map,不用设置color的值，color默认白色0xffffff。

```js
const material = new THREE.MeshLambertMaterial({
    // color: 0x00ffff,
    // 设置纹理贴图：Texture对象作为材质map属性的属性值
```

```
    map: texture,//map表示材质的颜色贴图属性
});
```

测试不同几何体添加纹理贴图的效果

你可以尝试把颜色纹理贴图映射到不同的几何体上查看渲染效果，至于为什么映射效果不同，其实和UV坐标相关，具体可以关注下节课关于UV坐标的讲解。

```js
const geometry = new THREE.BoxGeometry(100, 100, 100); //长方体
```

```js
const geometry = new THREE.SphereGeometry(60, 25, 25); //球体
```

← 6. 模型隐藏或显示

2. 自定义顶点UV坐标→