郭隆邦 2023-03-03

4. Raycaster(鼠标点击选中模型)

在实际开发中，射线投射器`Raycaster`经常会使用到，本节课先通过一个简单的小案例来给大家展示射线投射器`Raycaster`的射线拾取功能，简单说就是鼠标点击，选中一个模型对象。

下面代码的功能是鼠标单击threejs的canvas画布，通过射线投射器`Raycaster`射线拾取网格模型，被选中拾取到的网格模型改变颜色。

```
renderer.domElement.addEventListener('click', function (event) {
    // .offsetY、.offsetX以canvas画布左上角为坐标原点,单位px
    const px = event.offsetX;
    const py = event.offsetY;
    //屏幕坐标px、py转WebGL标准设备坐标x、y
    //width、height表示canvas画布宽高度
    const x = (px / width) * 2 - 1;
    const y = -(py / height) * 2 + 1;
    //创建一个射线投射器`Raycaster`
    const raycaster = new THREE.Raycaster();
    //.setFromCamera()计算射线投射器`Raycaster`的射线属性.ray
    // 形象点说就是在点击位置创建一条射线，射线穿过的模型代表选中
    raycaster.setFromCamera(new THREE.Vector2(x, y), camera);
    //.intersectObjects([mesh1, mesh2, mesh3])对参数中的网格模型对象进行射线交叉计算
    // 未选中对象返回空数组[],选中一个对象，数组1个元素，选中两个对象，数组两个元素
    const intersects = raycaster.intersectObjects([mesh1, mesh2, mesh3]);
    console.log("射线器返回的对象", intersects);
    // intersects.length大于0说明，说明选中了模型
    if (intersects.length > 0) {
        // 选中模型的第一个模型，设置为红色
        intersects[0].object.material.color.set(0xff0000);
    }
})
```

射线拾取网格模型步骤

- 1.坐标转化(鼠标单击的屏幕坐标转标准设备坐标)

- 2.射线计算(通过鼠标单击位置+相机参数计算射线值)
- 3.射线交叉计算

1. 坐标转化(屏幕坐标转标准设备坐标)

```
// .offsetY、.offsetX以canvas画布左上角为坐标原点,单位px
const px = event.offsetX;
const py = event.offsetY;
//屏幕坐标px、py转WebGL标准设备坐标x、y
//width、height表示canvas画布宽高度
const x = (px / width) * 2 - 1;
const y = -(py / height) * 2 + 1;
```

2. 计算射线(`.setFromCamera()` 方法)

把鼠标**单击位置坐标**和**相机**作为 `.setFromCamera()` 方法的参数， `.setFromCamera()` 就会计算射线投射器 `Raycaster` 的射线属性 `.ray` ,形象点说就是在点击位置创建一条射线，用来选中拾取模型对象。

```
//创建一个射线投射器`Raycaster`
const raycaster = new THREE.Raycaster();
//.setFromCamera()计算射线投射器`Raycaster`的射线属性.ray
// 形象点说就是在点击位置创建一条射线，用来选中拾取模型对象
raycaster.setFromCamera(new THREE.Vector2(x, y), camera);
```

3. 射线交叉计算(`.intersectObjects()` 方法)

通过 `.intersectObjects()` 方法可以计算出来与射线相交的网格模型(语法参考上节课14.2讲解)。

```
const intersects = raycaster.intersectObjects([mesh1, mesh2, mesh3]);
if (intersects.length > 0) {
    // 选中模型的第一个模型，设置为红色
    intersects[0].object.material.color.set(0xff0000);
}
```

← 3. 屏幕坐标转标准设备坐标 5. Canvas尺寸变化(射线坐标计算)→