

threejs

OrbitControls

```
import { OrbitControls } from "@lib/three-r173/jsm/controls/OrbitControls.js";

// 
const controls = new OrbitControls(camera, renderer.domElement);
// // 
controls.enableRotate = false;
// // 
controls.mouseButtons.LEFT = THREE.MOUSE.RIGHT;

// 
controls.touches.ONE = THREE.TOUCH.PAN; // 
controls.touches.TWO = THREE.TOUCH.DOLLY_ROTATE; // 

// 
// controls.update();
```


MeshLine MeshLineMaterial three.js three.js THREE.Line

```
import { MeshLine, MeshLineMaterial } from "three.meshline";

// 
function createThickLine(points, color, lineWidth) {
  var geometry = new THREE.BufferGeometry();
  geometry.setFromPoints(points.map((point) => new THREE.Vector3(point.x, point.y, 0)));

  var line = new MeshLine();
  line.setGeometry(geometry);
```

```

var material = new MeshLineMaterial({
  color: color,
  lineWidth: lineWidth,
  transparent: true,
  opacity: 1.0,
  blending: THREE.NormalBlending,
  depthTest: false,
  depthWrite: false
});

var mesh = new THREE.Mesh(line.geometry, material);
return mesh;
}

// 创建
var line = createThickLine(points, 0x0099ff, 0.05);
line.position.set(object.x * scale, -(object.y * scale), 0);
line.renderOrder = 11;
group.add(line);

```



Ray

```

// 初始化
var raycaster = new THREE.Raycaster();
var mouse = new THREE.Vector2();
// 事件监听
window.addEventListener("click", onMouseClick, false);

// 点击事件
function onMouseClick(event) {

  let mouse = _mouse;
  let raycaster = _raycaster;

```

```
let camera = _camera;
let scene = _scene;

// 鼠标位置归一化到 [-1, 1] 范围
mouse.x = (event.clientX / window.innerWidth) * 2 - 1;
mouse.y = -(event.clientY / window.innerHeight) * 2 + 1;

// 初始化射线
raycaster.setFromCamera(mouse, camera);

// 射线与场景相交
var intersects = raycaster.intersectObjects(scene.children, true);

// 处理相交结果
if (intersects.length > 0) {
    intersects.forEach(intersected => {
        // 输出相交对象
    });
}
}
```

📄 #3

📄 18 🕒 2025 14:25:37

📄 18 🕒 2025 14:38:14