这是three.js的一个组件，需要额外的引入文件，文件的地址是在官方下载的案例的examples/js/controls/TrackballControls.js。

只需要和案例里面一样设置相关的属性，并在实例化的时候讲相机传入。就可以实现交互效果。

可以实现的效果：

鼠标按住左键可以旋转模型

鼠标按住右键拖拽可以移动模型

鼠标滚轮可以缩放模型

案例查看地址：<http://www.wjceo.com/blog/threejs/2018-02-09/10.html>

案例代码：

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="UTF-8">
5. <title>Title</title>
6. <style type="text/css">
7. html, body {
8. margin: 0;
9. height: 100%;
10. }
12. canvas {
13. display: block;
14. }
16. </style>
17. </head>
18. <body onload="draw();">
20. </body>
21. <script src="build/three.js"></script>
22. <script src="examples/js/controls/TrackballControls.js"></script>
23. <script>
24. var renderer;
25. function initRender() {
26. renderer = new THREE.WebGLRenderer({antialias:true});
27. renderer.setSize(window.innerWidth, window.innerHeight);
28. document.body.appendChild(renderer.domElement);
29. }
31. var camera;
32. function initCamera() {
33. camera = new THREE.PerspectiveCamera(45, window.innerWidth/window.innerHeight, 1, 10000);
34. camera.position.set(0, 0, 400);
35. }
37. var scene;
38. function initScene() {
39. scene = new THREE.Scene();
40. }
42. var light;
43. function initLight() {
44. scene.add(new THREE.AmbientLight(0x404040));
46. light = new THREE.DirectionalLight(0xffffff);
47. light.position.set(1,1,1);
48. scene.add(light);
49. }
51. function initModel() {
52. var map = new THREE.TextureLoader().load("examples/textures/UV\_Grid\_Sm.jpg");
53. var material = new THREE.MeshLambertMaterial({map:map});
55. var cube = new THREE.Mesh(new THREE.BoxGeometry(100, 200, 100, 1, 1, 1), material);
56. scene.add(cube);
57. }
59. *//用户交互插件 鼠标左键按住旋转，右键按住平移，滚轮缩放*
60. var controls;
61. function initControls() {
62. controls = new THREE.TrackballControls( camera );
63. *//旋转速度*
64. controls.rotateSpeed = 5;
65. *//变焦速度*
66. controls.zoomSpeed = 3;
67. *//平移速度*
68. controls.panSpeed = 0.8;
69. *//是否不变焦*
70. controls.noZoom = false;
71. *//是否不平移*
72. controls.noPan = false;
73. *//是否开启移动惯性*
74. controls.staticMoving = false;
75. *//动态阻尼系数 就是灵敏度*
76. controls.dynamicDampingFactor = 0.3;
77. *//未知，占时先保留*
78. *//controls.keys = [ 65, 83, 68 ];*
79. controls.addEventListener( 'change', render );
80. }
82. function render() {
83. renderer.render( scene, camera );
84. }
86. *//窗口变动触发的函数*
87. function onWindowResize() {
89. camera.aspect = window.innerWidth / window.innerHeight;
90. camera.updateProjectionMatrix();
91. controls.handleResize();
92. render();
93. renderer.setSize( window.innerWidth, window.innerHeight );
95. }
97. function animate() {
98. *//更新控制器*
99. controls.update();
100. render();
101. requestAnimationFrame(animate);
102. }
104. function draw() {
105. initRender();
106. initScene();
107. initCamera();
108. initLight();
109. initModel();
110. initControls();
112. animate();
113. window.onresize = onWindowResize;
114. }
115. </script>
116. </html>