案例查看地址：<http://www.wjceo.com/blog/threejs/2018-02-09/7.html>

学习three.js也有两天了，现在也在不间断的学习相关知识，了解three.js的规则。也做了两个简单的案例，感受到了three.js的强大，研究了一些别人的代码，现在想找到一种适合自己的书写代码。

今天，还真的找到了一种感觉特别适合自己的方法，自己写js的时候貌似也这么写。但是由于学了两个月的WebGL的基础，学的有点懵逼，感觉自己书写正常js的能力都快忘记了。今天，看到这个案例，感觉自己的灵感直接来到了。

每个three.js案例基本上都会包括渲染器、相机、场景、光源和模型。我以后书写相关代码会把它们封装成一个个的方法，然后等待页面加载完后，调用一下即可。

下面是我自己定义的方法名：

初始化渲染器：initRender

初始化相机：initCamera

初始化场景：initScene

初始化灯源：initLight

初始化模型：initModel

动画：animate

绘制：draw

下面是一个案例，生成了42条线，垂直交错：

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>Three框架</title>
6. <script src="build/three.js"></script>
7. <style type="text/css">
8. html, body {
9. margin: 0;
10. height: 100%;
11. }
13. div#canvas-frame {
14. border: none;
15. cursor: pointer;
16. width: 100%;
17. height: 100%;
18. background-color: #EEEEEE;
19. }
21. canvas {
22. display: block;
23. }
25. </style>
26. <script>
27. *//渲染器*
28. var renderer;
29. function initRender() {
30. width = document.getElementById('canvas-frame').clientWidth;
31. height = document.getElementById('canvas-frame').clientHeight;
32. renderer = new THREE.WebGLRenderer({
33. antialias: true
34. });
35. renderer.setSize(width, height);
36. document.getElementById('canvas-frame').appendChild(renderer.domElement);
37. renderer.setClearColor(0xFFFFFF, 1.0);
38. }
40. *//相机*
41. var camera;
42. function initCamera() {
43. camera = new THREE.PerspectiveCamera(70, width / height, 1, 1000);
44. camera.position.x = 0;
45. camera.position.y = 1000;
46. camera.position.z = 0;
47. camera.up.x = 0;
48. camera.up.y = 0;
49. camera.up.z = 1;
50. camera.lookAt({
51. x: 0,
52. y: 0,
53. z: 0
54. });
55. }
57. *//场景*
58. var scene;
59. function initScene() {
60. scene = new THREE.Scene();
61. }
63. *//光源*
64. var light;
65. function initLight() {
66. light = new THREE.DirectionalLight(0xFF0000, 1.0, 0);
67. light.position.set(100, 100, 200);
68. scene.add(light);
69. }
71. *//模型*
72. function initModel() {
73. var geometry = new THREE.Geometry();
74. geometry.vertices.push(new THREE.Vector3(-500, 0, 0));
75. geometry.vertices.push(new THREE.Vector3(500, 0, 0));
77. for (var i = 0; i <= 20; i++) {
79. var line = new THREE.Line(geometry, new THREE.LineBasicMaterial({color: randomColor(), opacity: 1}));
80. line.position.z = ( i \* 50 ) - 500;
81. scene.add(line);
83. var line = new THREE.Line(geometry, new THREE.LineBasicMaterial({color: randomColor(), opacity: 1}));
84. line.position.x = ( i \* 50 ) - 500;
85. line.rotation.y = 90 \* Math.PI / 180;
86. scene.add(line);
88. }
89. }
91. *//绘制*
92. function draw() {
93. initRender();
94. initCamera();
95. initScene();
96. initLight();
97. initModel();
98. renderer.clear();
99. renderer.render(scene, camera);
100. }
102. *//生成随机颜色*
103. function randomColor() {
104. var arrHex = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "a", "b", "c", "d", "e", "f"],
105. strHex = "0x",
106. index;
107. for (var i = 0; i < 6; i++) {
108. index = Math.round(Math.random() \* 15);
109. strHex += arrHex[index];
110. }
111. return eval(strHex);
112. }

115. </script>
116. </head>
118. <body onload="draw();">
119. <div id="canvas-frame"></div>
120. </body>
121. </html>