

代码说明

本项目主要采用 WebGL 的 `three.js` 插件来实现一个球。

首先阐释 `sphere.js` 文件。

首先设置画布的大小。

```
var WIDTH = window.innerWidth - 30,  
    HEIGHT = window.innerHeight - 30;
```

然后设置一些有关 `camera` 的参数。 `camera` 用于定位我们的球。

```
//与相机有关的参数  
var angle = 45,  
    aspect = WIDTH / HEIGHT,  
    near = 0.1,  
    far = 3000;  
  
var camera = new THREE.PerspectiveCamera(angle, aspect, near, far);  
camera.position.set(0, 0, 0); //设置相机的位置
```

创建 `light` 对象，否则我们的球会一片漆黑，什么都看不见。

```
var light = new THREE.SpotLight(0xFFFFFF, 1, 0, Math.PI / 2, 1);  
light.position.set(4000, 4000, 1500);  
light.target.position.set(1000, 3800, 1000);  
scene.add(light);
```

上述代码的最后一步意味着将 `light` 加入到场景之中。

接着我们创建球体。

```
var sphereGeo = new THREE.SphereGeometry(30, 40, 40),  
    sphereMat = new THREE.MeshPhongMaterial();  
var sphereMesh = new THREE.Mesh(sphereGeo, sphereMat);  
sphereMesh.position.set(-100, 0, 0);  
sphereMesh.rotation.y = 5;  
scene.add(sphereMesh);
```

这里创建了一个 `Mesh` 对象，因为 `threejs` 封装得较好，这就是我们想要的球，最后将这个球加入到场景中。

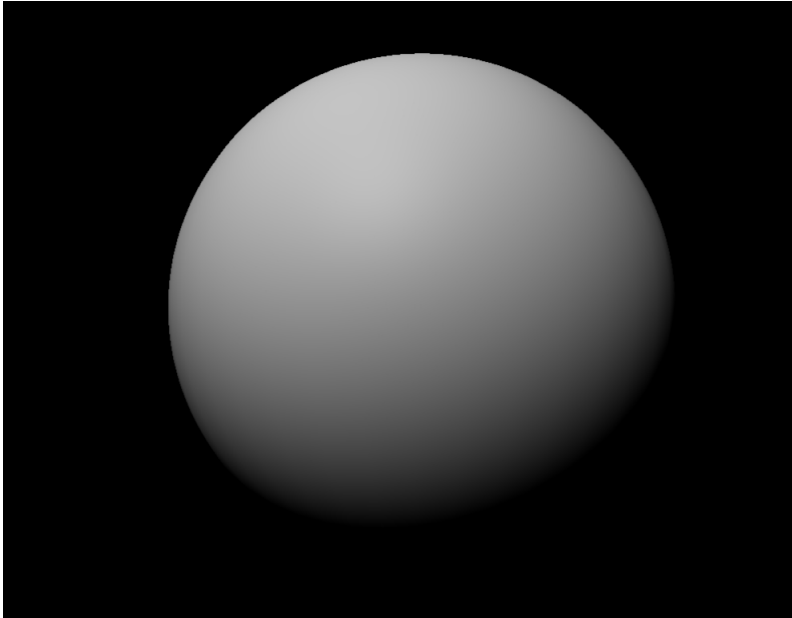
最后一步是渲染。

```
var renderer = new THREE.WebGLRenderer({ antialiasing: true });
renderer.setSize(WIDTH, HEIGHT);
renderer.domElement.style.position = 'relative';
container.appendChild(renderer.domElement);
renderer.autoClear = false;
renderer.shadowMapEnabled = true;
function render() {
    renderer.render(scene, camera);
}
render();
```

以上就是 `sphere.js` 文件的说明，有了 `js` 文件之后，再和 `html` 和 `css` 文件做交互就行。

结果示例

直接在 `Chrome` 浏览器打开 `index.html` 即可。



参考内容

- [1] <https://threejs.org/docs/#Reference/Lights/SpotLight>
- [2] <https://www.youtube.com/watch?v=EWY-SaHnxkM>
- [3] https://www.sitepoint.com/building-earth-with-webgl-javascript/?utm_source=html5weekly&utm_medium=email