

## Report 2: 真实感静态景物

16307130065 仇均易

---

### 一、文件目录

- 1、code/index.html: 主要页面
- 2、code/js/pic.js: 绘制东方明珠的javascript文件
- 3、code/js/OrbitControls.js、stats.js、three.min.js: 程序所需库文件
- 4、code/src: 绘制所需图片、材质资源
- 5、Report\_2.pdf: 项目报告

### 二、开发环境

Javascript (Three.js) +Visual Studio Code

### 三、实现细节

在本次Project中选择使用**WebGL**进行真实感静态景物的绘制，而其中**three.js**就是利用javascript编写的WebGL开源框架，因此选择采用**three.js**绘制东方明珠作为主要任务。

- **绘制准备**

three.js的图形绘制主要需要以下的准备工作（具体代码见pic.js）：

- 1、创建渲染器：**用于绘制图形的渲染工作

```
function initRender() {  
    ...  
    renderer.shadowMap.enabled = true;  
    ...  
    document.body.appendChild(renderer.domElement);  
}
```

其中，**renderer.shadowMap.enabled**为设置阴影，否则渲染得到的页面将不会出现阴影；**document.body.appendChild(renderer.domElement)**用于向页面添加canvas进行显示，否则仅仅进行渲染而不能显示图形；

- 2、创建相机：**用于观察绘制物体

```
function initCamera() {  
    camera = new THREE.PerspectiveCamera(45,  
        window.innerWidth / window.innerHeight, 0.1, 1000);  
    ...  
    camera.lookAt(new THREE.Vector3(0, 0, 0));  
}
```

其中，**PerspectiveCamera**为创建透视相机，参数分别为视角、宽高比、近平面距离、远平面距离。再加上相机的位置和相机照相的方向（**lookAt**）就可以在三维空间中唯一确定一个可视空间，出现在该空间内的物体最后才能够被渲染出；

- 3、创建场景：**用于放置绘制物体

```
function initScene() {
    scene = new THREE.Scene();
}
```

在一般的场景搭建中，可以采用天空盒形式搭建，即利用CubeTextureLoader进行创建，构造出真实的天空场景，而此项目中无此必要，故仅搭建简单场景；

#### 4、创建光源：用于照射绘制物体

```
function initLight() {
    ambientLight = new THREE.AmbientLight("#111111");
    ...
    spotLight = new THREE.SpotLight("#ffffff");
    ...
    spotLight.castShadow = true;
    ...
}
```

在该部分中，创建两种不同的光源：ambientLight为环境光，用于增加场景的亮度；spotLight为聚光光源，用于照射绘制物体，使物体可视&形成阴影；

#### 5、创建控制器：用于控制场景

```
function initControls() {
    controls = new THREE.OrbitControls(camera, renderer.domElement);
    ...
}
```

在模型搭建的过程中，往往需要旋转缩放等操作进行多方位的观察，因此利用OrbitControls插件进行实现，使得场景能够事件鼠标左键旋转、鼠标右键拖拽、鼠标中键缩放的功能。

#### • 模型绘制

东方明珠重要分为三个部分进行绘制（代码见函数GetOrientalPearl）：

底部：圆台x2+斜向圆柱x3+底部带球支柱x3

中部：直立圆柱x3+最下大球x1+中层中球x1+塔体圆环x7

顶部：顶部圆柱x4+顶部圆环x2+顶部小球x1

采用three.js中的Object3D对象进行拼接，构成完成的东方明珠模型，进而添加到场景中；此外，还需要额外绘制一块平面plane，用于承载东方明珠&投射东方明珠的阴影。

针对于真实感方面，three.js中可以利用材质、贴图的方式完成。对于所有的物体均采用MeshPhongMaterial材质（该材质能够制造光滑的表面，并且感光效果好能够接受光线投射）；并且对于柱体、环体、球体均采用贴图进行装饰，构建真实感的同时增加美感，最终得到四中图所示的绘制效果。

### 四、程序说明

- 由于three.js中图片加载存在跨域请求的问题，因此不能直接通过打开index.html查看绘制效果，需要在源文件目录利用 `http-server -c-1` 搭建本地端服务器，进而访问 <http://localhost:8080/> 即可查看东方明珠的绘制效果，如下图所示：

（由于在页面中添加了OrbitControls控制，可以使用鼠标左键旋转、鼠标右键拖拽、鼠标中键缩放）

