采用正方体六面贴图方式实现全景

1. 目前知道的有三种办法, push(),texturesloader() 与 uv 映射;

已经尝试前两种办法,后面的暂时未尝试：

2. 关于 push();

```
var materials = [];
for(var i = 1;i<7;i++)
{
    materials.push(new THREE.MeshBasicMaterial(
        {map :THREE.ImageUtils.loadTexture('./textures3/'+i+'.jpg', {},function()
            {
                renderer.render(scene,camera);
            })}
    ));
}
mesh = new THREE.Mesh(new THREE.CubeGeometry(500,500,500),new THREE.MeshFaceMaterial(materials) );

scene.add( mesh );
```

实验结果上：push()方法有它自己顺序,后续对单面操作可能会造成影响;

3. 关于 texturesloader()

```
//cubemap
var path = "textures/";
var format = '.jpg';
var urls = [
    path + 'posx' + format, path + 'negx' + format,
    path + 'posy' + format, path + 'negy' + format,
    path + 'posz' + format, path + 'negz' + format
];

var reflectionCube = new THREE.CubeTextureLoader().load( urls );
reflectionCube.format = THREE.RGBFormat;

var refractionCube = new THREE.CubeTextureLoader().load( urls );
refractionCube.mapping = THREE.CubeRefractionMapping;
refractionCube.format = THREE.RGBFormat;

scene = new THREE.Scene();
scene.background = reflectionCube;
```

在这个方案中, 还使用了相机跟随鼠标相关操作 js (OrbitControls.js),非常方便全景的浏览;