Three.js入门

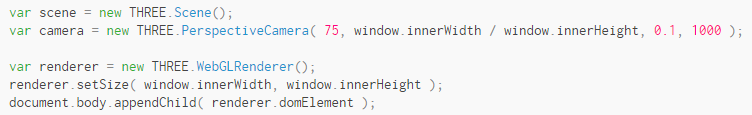
13302010015 庄文俊

three.js是使用Javascript编写的WebGL的第三方库，可以用来很方便地创建各种三维场景然而因为文档的匮乏导致对入门者的不友好，所以想简要写一些three.js的基础介绍。

首先需要一个html网页来显示场景。可以看到代码非常简单，只需要导入three.js文件即可。three.js地址：http://threejs.org/docs/index.html#Manual/Introduction/Creating\_a\_scene



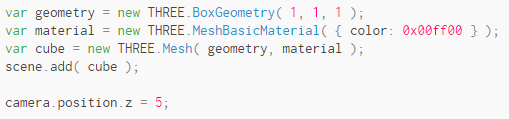
要真正创造场景，首先我们需要三样东西，首先就是场景，然后是一个相机也就是观察点，最后是渲染器。



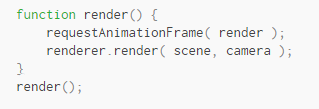
这样我们就架设好了三要素，其中我们Camera使用的是PerspectiveCamera，第一个参数就是视野。第二个参数是屏幕的高宽比，我们总是希望用实际屏幕的宽度除以实际屏幕的高度，这样就不会出现像是在宽屏电视上看老电影一样的被压扁了的效果。接下去两个参数是近端和远端的剪切平面，其实就是渲染范围，高于远端值或是低于近端值的物体都不会被渲染，这会提高应用的性能。

然后说说最神奇的渲染器。除了我们现在使用的WebGLRenderer以外，three.js还提供了很多其他的渲染方式以备老版本的浏览器或是不支持WebGL的浏览器使用。为了创造我们的渲染器实例，我们还需要设置应用里的渲染范围。通常情况下我们都会想撑满整个窗口不过我们也可以只渲染四分之一，比如把范围设置为window.innerWidth/2和window.innerHeight/2。如果我们不希望减少渲染范围，而是像得到低分辨率的效果，可以在setSize函数里设置第三个参数false，这样分辨率就只会有原来的一半。然后最后一句我们把渲染器元素加进了HTML网页文件。

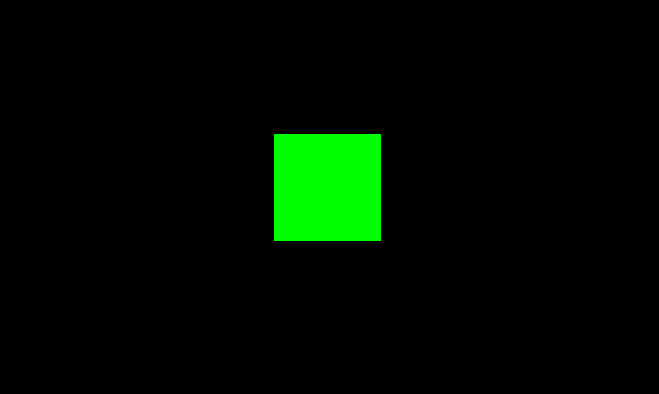
现在我们要在场景里加一个立方体了。首先需要一个BoxGeometry，这个对象里有立方体所需的全部定点和面，大小就设为1,1,1好了。只有一个框架是不行的，我们还要载入材质。three.js里附带了很多材质，不过我们先用MeshBasicMaterial，把颜色设成#00ff00，也就是绿的。然后我们通过Mesh（网格）把我们调好大小和材质的立方体放在这个场景里。最后调用scene.add，我们的立方体就会在立体直角坐标系的（0,0,0）处站定了，不过Camera也在那里，这样会出现在立方体内部的情况，所以我们稍稍把镜头移开一点camera.position.z=5。



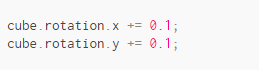
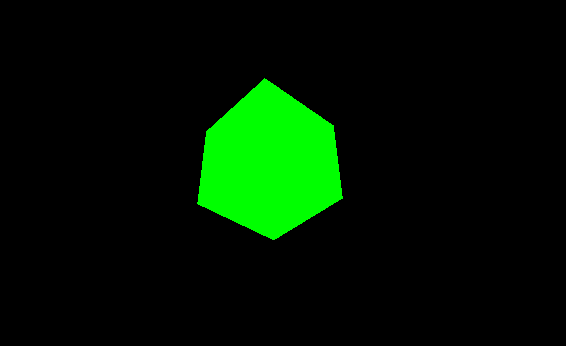
看到这里可能有人已经开始动手把代码贴上去尝试了，然而并不会出现什么立方体，只有一片漆黑，为什么呢？因为其实我们到现在为止还没有开始真正的渲染，我们需要一个名字叫渲染环（render loop）的东西。这个函数会创造一个渲染的循环，每秒重画60遍场景，而且相比与通常使用的setInterval，requestAnimationFrame在切换浏览器标签时会自动暂停，节省资源和电量。



现在立方体终于现出了原形，然而



看起来明明就只是一个翠绿翠绿的方形，因为我们把观察点放在z轴上，只能看到一个面。对于这样的结果完全说服不了自己学过three.js，所以我们想个办法，让立方体动起来，这样我们就可以肯定这的确是个立方体，自己也的确会了一点three.js了。

旋转听起来挺难的然而让我们看看刚刚加的渲染环，这不是每秒60帧已经摆在那里了吗。我们所要做的，只是在渲染环每次重画的时候加上一点点旋转的偏移量就行了，于是加上这两句话之后，我们的立方体就转起来了。

当然不仅仅是旋转，我们也可以加上x轴、y轴甚至z轴的偏移，这样我们就会看到一个一边旋转一边平移或者放大缩小的立方体了。