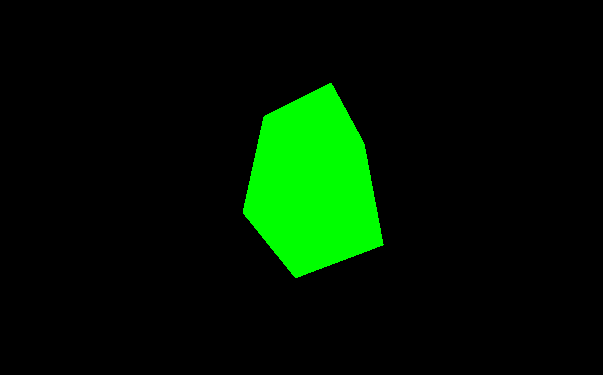
**基于WebGL的web3D的实践和开发**

主要还是一些three.js的实践

显示简单实现了一个旋转的立方体。



实现效果：



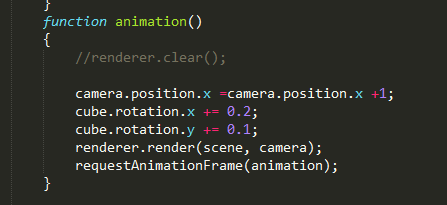
从这个例子大概了解了scene、camera、renderer的一些基础知识。

然后尝试做一个动画。

如果我们改变了物体的位置或者颜色之类的属性，就必须重新调用render()函数，才能够将新的场景绘制到浏览器中去。不然浏览器是不会自动刷新场景的。

如果不断的改变物体的颜色，那么就需要不断的绘制新的场景，所以我们最好的方式，是让画面执行一个循环，不断的调用render来重绘，这个循环就是渲染循环，在游戏中，也叫游戏循环。

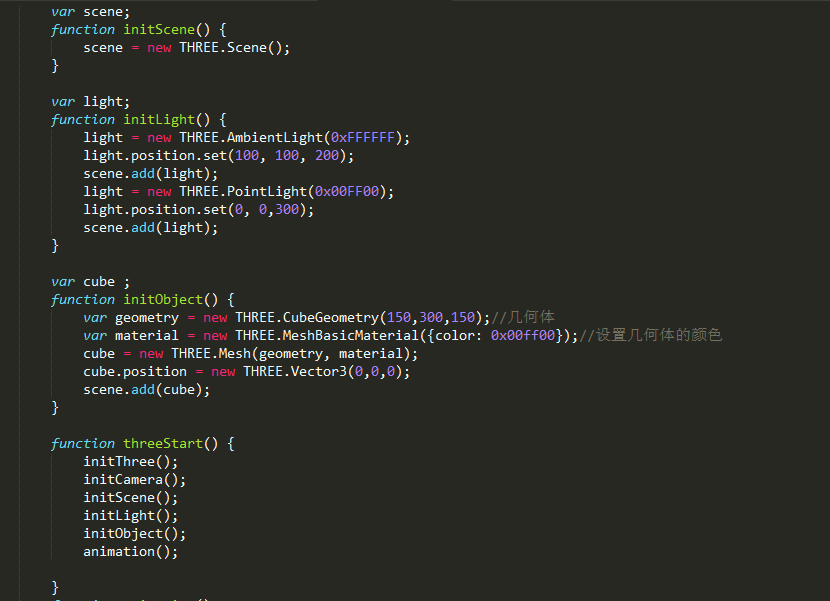
其实就是在刚才的代码基础上让camera进行了一个移动。

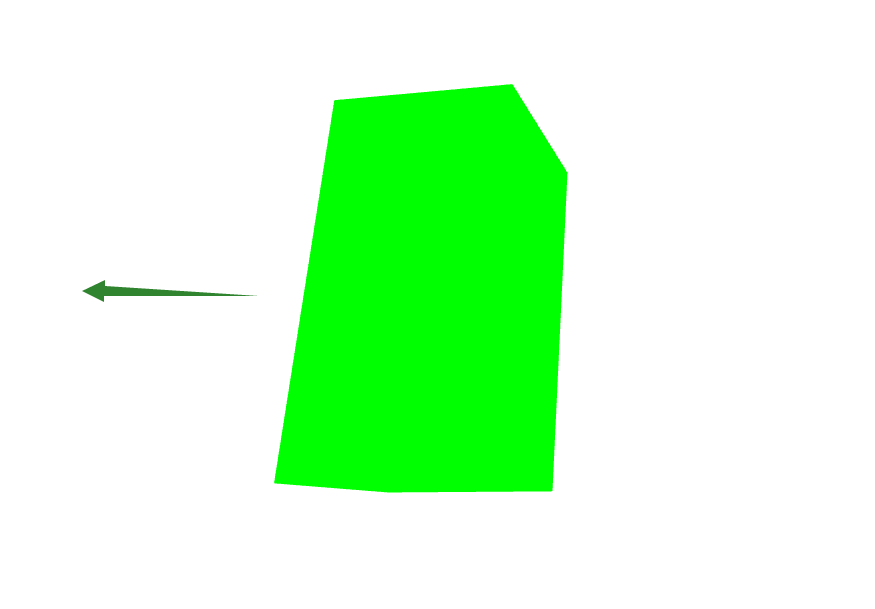


实现效果就是立方体旋转着离开了我的视线。

然后这一次的代码结构比较清晰我还加了light终于不是黑色背景了……

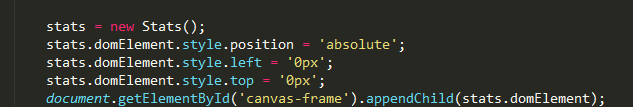


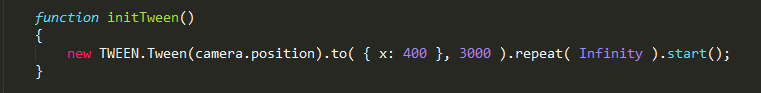




在这个基础上，了解到了tween.js这样一个动画引擎，用来制作稍微复杂一点的动画。

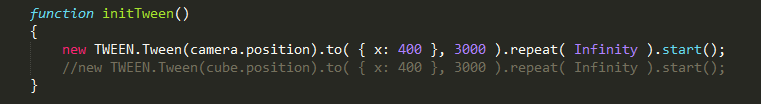
添加了这些代码。







然后就变成了一个无限循环移动camera位置的小动画。



然后也尝试了移动物体的循环。

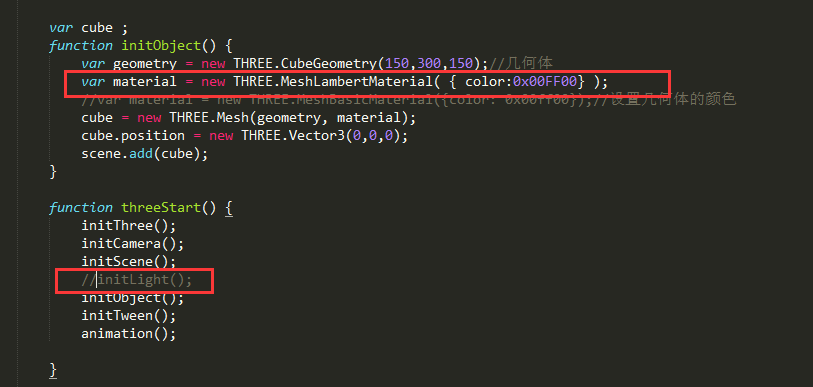
当然两个同时的话在我们看来物体是在原地翻转，实际是物体和镜头一起移动了。

当然除了直线运动，tween还可以实现一些曲线运动。

然后我注意到一开始的test文件背景是漆黑的，其他的背景就是白色，我本来觉得是不是因为没有加light……后来发现是renderer少写了



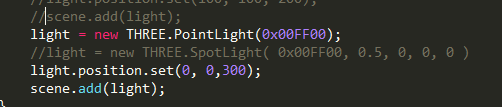
然后我就去实验了材质和灯光。



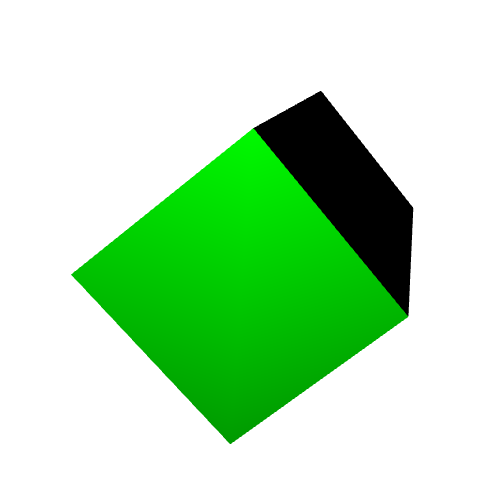
当没有光时，兰伯特材质没有可反射的光全部呈现黑色。

然后我尝试去掉了环境光。

在只有



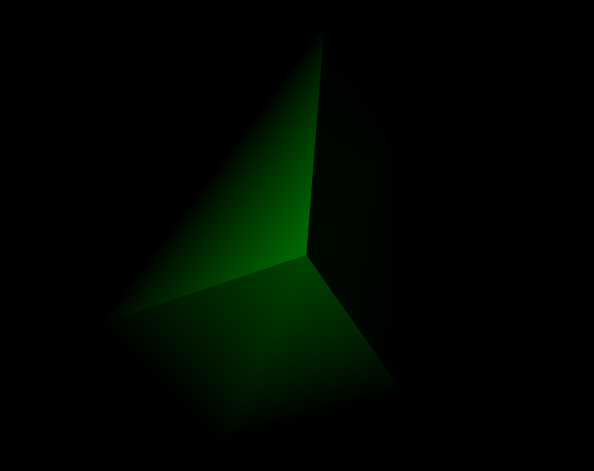
这样一个点光源的时候。点光源应该是会随着离它比较远衰减。



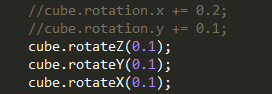
可见环境光无处不在且不会衰减。

然后我又尝试了聚光灯。因为在很多的demo里面应该都是聚光灯我觉得很好玩。

用的反光材质然后渲染器背景色黑色就有一种很好玩的效果。



最后，之前是用改变cube的x，y坐标来实现的物体旋转。我们还可以用其他方式来实现。



然后以上就是three.js的一些基本知识的实践和学习。

在网上当然也搜到了非常多关于WebGL的好看的例子，比如非常经典的那个水立方。