## Webgl教程

第一章 webGL开源框架—three.js介绍

WebGL是在浏览器中实现三维效果的一套规范。

使用WebGL原生的API来写3D程序是一件非常痛苦的事情，幸好，有很多同行花业余时间写了一些WebGL开源框架，其中three.js就是非常优秀的一个，它掩 盖了很多麻烦的细节，那么，就让我们一起来看看，什么是three.js吧

Threejs能做什么?

一言以蔽之，它能写出在浏览器上流畅运行的3D程序。

这些就是Threejs能做的事情，我向你保证，完成这些事情，会比c++用更少的代码。而且更容易，更酷。 在本课程中，我们将由浅入深的讲解这些效果的实现，这能让你迅速提高开发技能，在职场中处于不败之地。

1. three.js组件

在Three.js中，要渲染物体到网页中，我们需要3个组建：场景（scene）、相机（camera）和渲染器（renderer）。有了这三样东西，才能将物体渲染到网页中去。

1、场景

在Threejs中场景就只有一种，用THREE.Scene来表示，要构件一个场景也很简单，只要new一个对象就可以了，代码如下：

var scene = new THREE.Scene();

场景是所有物体的容器，如果要显示一个苹果，就需要将苹果对象加入场景中

2、相机

另一个组建是相机，相机决定了场景中那个角度的景色会显示出来。相机就像人的眼睛一样，人站在不同位置，抬头或者低头都能够看到不同的景色。

场景只有一种，但是相机却又很多种。和现实中一样，不同的相机确定了呈相的各个方面。比如有的相机适合人像，有的相机适合风景，专业的摄影师根据实际用途不一样，选择不同的相机。对程序员来说，只要设置不同的相机参数，就能够让相机产生不一样的效果。

在Threejs中有多种相机，这里介绍两种，它们是：

透视相机（THREE.PerspectiveCamera）、这里我们使用一个透视相机，透视相机的参数很多，这里先不详细讲解。定义一个相机的代码如下所示：

camera=newTHREE.PerspectiveCamera(75,window.innerWidth/window.innerHeight, 0.1, 1000);

3、渲染器

最后一步就是设置渲染器，渲染器决定了渲染的结果应该画在页面的什么元素上面，并且以怎样的方式来绘制。这里我们定义了一个WebRenderer渲染器，代码如下所示：

View Raw Code

var renderer = new THREE.WebGLRenderer();

renderer.setSize(window.innerWidth, window.innerHeight);

document.body.appendChild(renderer.domElement);

注意，渲染器renderer的domElement元素，表示渲染器中的画布，所有的渲染都是画在domElement上的，所以这里的appendChild表示将这个domElement挂接在body下面，这样渲染的结果就能够在页面中显示了。

4、添加物体到场景中

那现在，我们将一个物体添加到场景中：

View Raw Code

var geometry = new THREE.CubeGeometry(1,1,1);

var material = new THREE.MeshBasicMaterial({color: 0x00ff00});

var cube = new THREE.Mesh(geometry, material);

scene.add(cube);

代码中出现了THREE.CubeGeometry，THREE.CubeGeometry是什么东东，他是一个几何体，几何体由什么组成，已经不是本课的主要内容了，后面的课程我们会详细的学到。不过这里你只需要知道CubeGeometry是一个正方体或者长方体，究竟是什么，由它的3个参数所决定，cubeGeometry的原型如下代码所示：

View Raw Code

CubeGeometry(width, height, depth, segmentsWidth, segmentsHeight, segmentsDepth, materials, sides)

width：立方体x轴的长度

height：立方体y轴的长度

depth：立方体z轴的深度，也就是长度

想一想大家就明白，以上3个参数就能够确定一个立方体。

剩下的几个参数就要费解和复杂一些了，不过后面我们会自己来写一个立方体，到时候，你会更明白这些参数的意义，这里你可以将这些参数省略。