



泰然
移动开发专家

 用QQ帐号登录

只需一步，快速开始

用户名

密码

☐ 自动登录

找回密码

登录

注册泰然

请输入搜索内容

帖子

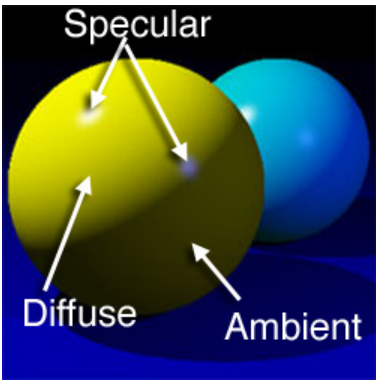
热搜: OpenGL Cocos2d 泰然教程

[转载]从零开始学习OpenGL ES之五 - 材质

2011-9-14 23:42 | 发布者: Iven | 查看: 8231 | 评论: 16

摘要: 图形图像, 编程,编程, OpenGL ES, 教程,OpenGL ES 3D

在 [上一篇文章](#)，我们讨论了光效的设定以及光效的各种属性。我们还讨论了光的三要素：**散射光**, **环境光** 和**高光**。如果你还不是完全清楚，那么我们来复习一下，在定义材质时大量的用到这些要素。

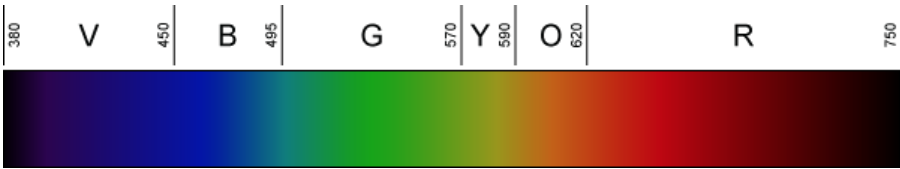


作为本文的起点，我们使用了[原文中球体绘制](#)的项目文件。我们不再使用二十面体而是转向球体是因为球体是展示光和材质不同要素之间相互作用的最佳形状。

颜色是什么

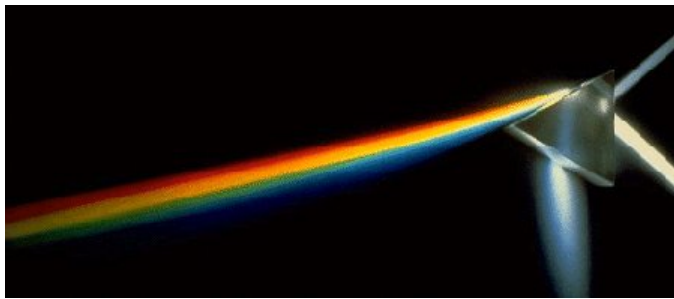
这可能是对小学美术课的复习。为什么现实世界会有颜色？是什么造成的？

我们看得见的光被称为 **光的可见频谱**。根据不同的波长我们可以感知到不同的颜色。在可见光谱的一端是低波长高频率的紫色和蓝色，而在另一端是低频高波长的橘色和红色：



电磁波在这个范围之外，因此不是“可见光”，尽管这只是人工的区分方法，它们的唯一不同在频率和波长，在于人眼的感知。尽管有各种方法感知各种电磁波的能力，但从OpenGL的角度出发，我们只关心可见光谱。

“白色光”包括等量的所有波长。换言之，白光之所以是白色的是因为它包括了所有（或至少大部分）可见光的频率。如果你曾经做过棱镜试验，那么你可能看过如下效果：



棱镜反射白光，各种波长被分离出来。这就是彩虹产生的原理。

如果你看到一个物体呈蓝色，那么实际上是该物体吸收了大部分可见光谱低频部分。它吸收了红，橘，黄和绿光。根据蓝色的不同色度，它还可能吸收一些紫色和蓝色。

但大部分蓝色的波长都被反射到你的眼睛。因为一些可见光被吸收了，由于它不再包括可见光谱中的所有波长所以反射到你眼中的光不再是白色。

简单吧？让我们看看这些是怎样运用在OpenGL的。

OpenGL 材质

我们通过定义材质的反射光来定义OpenGL ES中的材质，正如现实世界中一样。如果一个材质定义为反射红光，那么在正常的白光下，它将显示红色。

在OpenGL中 (至少在使用光滑着色处理和光效时)，材质是没有颜色的。OpenGL具有分别定义材质是怎样反射OpenGL光效三要素（环境，散射和高光）的能力。另外，它还具有指定材质 自发光（**emissive**）属性的能力，关于这点我们稍后再讨论。

指定材质

要在OpenGL创建一个材质，我们需要一次或多次调用 `glMaterialf()` 或者 `glMaterialfv()`。类似于上一篇文章中光效的定义，由于各属性或元素需要分别通过这些调用了指定，所以我们通常必须多次调用这些函数以完全定义材质。所有未定义的元素或属性默认值为0，或者以颜色来说为黑色。

传递给 `glMaterialf()` 或者 `glMaterialfv()` 的第一个参数总是用于指定是否材质影响多边形的前，后或两者的GL_ENUM。实际上除了为了与OpenGL兼容，第一个参数在OpenGL ES中没有什么意义，因为只有一个有效的选项：`GL_FRONT_AND_BACK`，它简单地表示材质适用于任何绘制的多边形。如果你还记得[第一部分](#)，那么你应该知道三角形的正面和背面是由winding（顶点的绘制次序）决定的。默认情况下，只有三角形的正面被绘制出来，但是有可能让OpenGL也绘制背面，或甚至只绘制背面，常规 OpenGL 允许你通过传递`GL_FRONT`，`GL_BACK`，或者 `GL_FRONT_AND_BACK`来为正面和背面指定不同的材质。但是OpenGL ES仅支持`GL_FRONT_AND_BACK`。

`glMaterialf()` 或 `glMaterialfv()`的第二个参数是指示正在设定材质的哪个元素或属性的 GL_ENUM。它们像传递给`glLightfv()`的值一样，比如 `GL_AMBIENT`，另外还有些新的值我们稍后再谈。

最终值是 `GL_FLOAT`或包括了实际属性或元素的GL_FLOAT数组的指针。

材质的最重要元素是环境光和散射光，因为它们决定了材质是怎样反射大量光线的。我们今天使用的项目代码定义了正如太阳光或白炽灯产生的白色，它具有平均分布的各种波长和颜色的光。如果光不是白色，球体看上去会有不同的外观。例如，反射至红色材质的蓝光将产生紫色阴影。简单起见，我们只使用白色光。当然，你可以随

意改变光的颜色进行试验看看光和材质是怎样交互作用的。大部分时候，它们在OpenGL ES中的表示与现实生活中完全一样。

下面是项目在添加材质前运行时的样子：



如你所见，它具有一些环境光和更为显著的散射光。

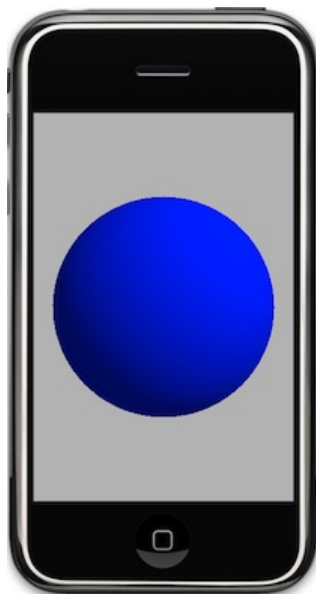
环境光和散射光

当讨论OpenGL的材质时，我们需要同时讨论环境光和散射光，这是因为这两个元素是一起工作从而决定物体被感知的颜色的。记住，散射光处于本文第一个图片中球体的顶部（亮黄色），环境光则是下方的暗黄色。材质怎样反射这两个元素决定了物体被感知的颜色。上图的效果可以通过不止一种方法获得。同样地，黄色球以同样比例反射环境光和散射光，但是场景中具有较少的环境光。

大约 90% 或更多的情况下，将材质的环境光和散射光参数设定成一样。这样做，使它们成为决定物体阴影和外观的因素。实际上，有一种方法通过一个调用`glMaterialfv()`同时设定材质环境光和散射光。下面是定义材质为蓝色的示例：

```
GLfloat ambientAndDiffuse[] = {0.0, 0.1, 0.9, 1.0};  
glMaterialfv(GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE, ambientAndDiffuse);
```

正如 `glColor4f()`，设置材质指示随后所有物体绘制的方式直到另一个材质被指定。如果将上叙代码放置于我们的绘制代码之前，那么运行时你将看到球体变为蓝色。由于环境光没有散射光强，其下方只是稍暗。



有时你希望更多地控制并希望分别设定材质对环境光和散射光的反射方式。例如，下例中，材质从环境光反射蓝色，从散射光反射红色：

```
GLfloat ambient[] = {0.0, 0.1, 0.9, 1.0};  
glMaterialfv(GL_FRONT_AND_BACK, GL_AMBIENT, ambient);  
GLfloat diffuse[] = {0.9, 0.0, 0.1, 1.0};  
glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUSE, diffuse);
```

运行结果：



在此情况下，它看上去像我们投射了有色光到球体上，实际上却不是这样。原因是我们反射了与环境光不一样的定向光线。

大部分情况下，如果你希望产生彩色光的效果，你只需创建彩色光线然后使用GL_AMBIENT_AND_DIFFUSE来指定材质颜色。但是有时却希望分别设置它们产生特殊效果或在不引起创建额外光线开销的情况下假造一个分离的彩色点光源。记住：你每增加一个光源，也就增加了每秒钟的运算量，所以有时候欺骗并不完全是一件坏事。

高光和光泽

你还可以单独设置场景中高光元素的反射方式，从而控制高光“热点”的亮度。一个叫 GL_SHININESS的参数与材

质的高光元素一起定义了高光热点的大小。如果你设定了材质的 `GL_SPECULAR` 值，你还应该定义其反光度。反光度越高，高光反射越小，所以默认值 `0.0` 几乎完全淹没了散射光，因此看上去很糟糕。

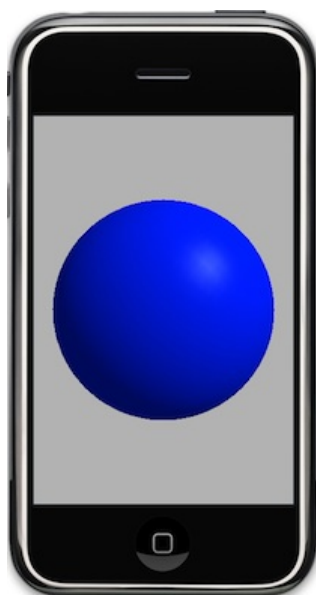
让我们回到蓝色球体，增加高光热点：

```
GLfloat ambientAndDiffuse[] = {0.0, 0.1, 0.9, 1.0};  
glMaterialfv(GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE, ambientAndDiffuse);  
GLfloat specular[] = {0.3, 0.3, 0.3, 1.0};  
glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, specular);  
glMaterialf(GL_FRONT_AND_BACK, GL_SHININESS, 25.0);
```

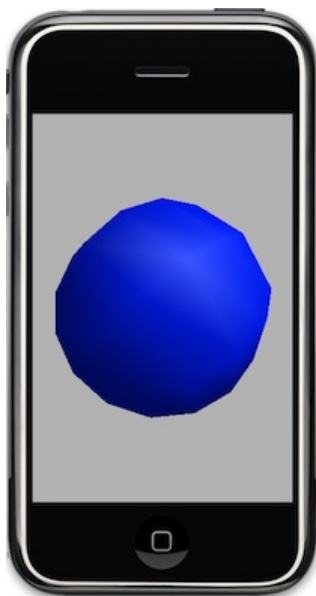
我们使用了一个较暗的白色作为球体的高光值。这个值看上去有些小，但就是这么小的值也能产生显著的效果。光线高光元素的值与材质的高光元素相乘所产生的光集中在材质反光度指定的区域。下面是上叙代码产生的结果：



现在，球体上有一个小的区域具有更强的反射光。我们可以通过增强光或光的高光元素，或者通过增加材质的反光度使这个点更亮。我们还可以通过调整反光度改变高光的大小。材质反光度越高，高光越集中。例如，如果我们将反光度从 `25.0` 改为 `50.0`，我们将得到一个更小的热点，它使得球体显得更具有光泽。



但是，有一点要小心。本文之所以改为使用球体的部分原因以及为什么球体使用较高的顶点数目的原因是高光在一个低面数的物体上看上去实在糟糕。注意一下，如果我减少球体的顶点数会发生什么：



低面片

高光使三角形边缘突出，高光通常在我们的游戏中经常使用的低面片的物体上表现不佳。在常规OpenGL中，有一个称为 **着色器 (shader)** 的机制可以用来为低面片物体产生较为理想的结果，但目前iPhone上的OpenGL ES并不支持此功能（译者注：iPhone 3GS支持OpenGL ES 2.0，有shader功能。但有一个问题就是OpenGL ES 1.1与OpenGL ES 2.0并不完全兼容）在游戏中如果你想使低面片物体漂亮的唯一方法就是完全摒弃高光元素而使用纹理映射，这将在下一篇文章中谈到。

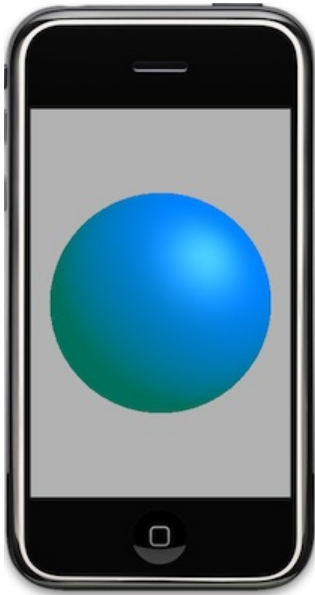
自发光

还剩最后一个材质的重要属性，它称为自发光元素。通过设定自发光元素，使得材质看上去会发射我们指定的颜色。它并不是真正在发光。例如，其周边物体并不会被发射的光线影响。如果你希望一个物体像灯泡一样发光照亮其他物体，由于在OpenGL ES中只有光源会发光（听上去像废话），你需要将自发光元素和与物体同一位置处的实际光源结合起来。但是自发光元素可以使物体漂亮地发光。

例如，我们可以为蓝色球体添加绿色的光泽：

```
GLfloat emission[] = {0.0, 0.4, 0.0, 1.0};  
glMaterialfv(GL_FRONT_AND_BACK, GL_EMISSION, emission);
```

结果如下：



自发光元素影响整个材质，因此 GL_EMISSION的值将与落入物体指定区域的任何类型的光相叠加。请注意，甚至上图中的高光部分也成了一点蓝-绿色而不是纯白色。在高光点处其效果是很微小的，但在只有环境光被反射的底部效果更为明显。它实际影响整个物体。

20

21

鲜花握手雷人路过鸡蛋

刚表态过的朋友 (23 人)

HexBlues

Burning_s...

々枫々

45553283...

jiangmingz

度娘818

伊然

yuxiang11...

iam_gaowei

riseliang

yjh4866

usb748

ccboby

sen

红沙尘

halley

myhapi

fspinach

286820549

fox

胡建平

bulice

apiter

邀请

收藏

最新评论

发表评论

吸进阳光 〰 吐出 2014-1-27 21:52

引用

学习中...

lotusr039 2013-10-23 10:40

引用

项目我传到csdn了 <http://download.csdn.net/detail/u012543342/6440561>
不用积分，也不是圆饼:)

flyzhg 2013-8-7 16:13

引用

demo, 我放在了 csdn 上一份。
这个demo, 在xcode 4.5上还需要修改一下， 否则显示是一个圆饼。 具体怎么改，可以问我。给我留言。

samoo 2013-1-27 11:22

引用

项目的文件链接似乎失效了，无法下载。