用户登录

首页 Unity3D

Shader

.Net(C#)

英语

甚 其他

源码

最后编辑: 2014-12-24 20:15:07

A- A+

unity3d移动

unity3d摄像机

无线监控摄像机 微型摄像机



针孔摄像头

外企猎头公司

猎头公司排名

伪装微型摄像机

室内设计师

最新文章



【原创】C# 基础之 Lambda表达 式 - 907 次阅读



【原创】C#基础之 IEnumerable和 IEnumerator - 792次 阅读



【原创】C#基础之事 件 - 886 次阅读

【原

【原创】C#基础之委 托 - 912 次阅读

新王图片

【原创】C#基础之委托的

使用 - 856 次阅读



unity3d移动

unity3d摄像机

伪装微型摄像机

室内设计师

外企猎头公司

无线监控摄像机

防爆摄像机

找保姆照顾老人

随机阅读

【翻译】第二章节:RGB 立方体(关于顶点输出参数)

2014-11-20 08:34:00 2257 人阅读 Unity3D cg 顶点输出参数

文章内容 例子源码 网友评论

本文永久地址:http://www.omuying.com/article/87.aspx , 【文章转载请注明出处!】

原文链接: http://en.wikibooks.org/wiki/Cg_Programming/Unity/RGB_Cube

基于上一章节《最简单的着色器》,本教程将讨论顶点输出参数和片段输入参数。

在这个教程中,我们将编写一个着色器并应用到一个立方体上。立方体上每个点的颜色都是由它表面坐标决定的。一个点的位置信息 (x,y,z) 可以对应一个颜色值 $({\rm red,green,blue})=(x,y,z)$,例如:点 (x,y,z)=(0,0,1) 表示颜色值 $({\rm red,green,blue})=(0,0,1)$,即蓝色。

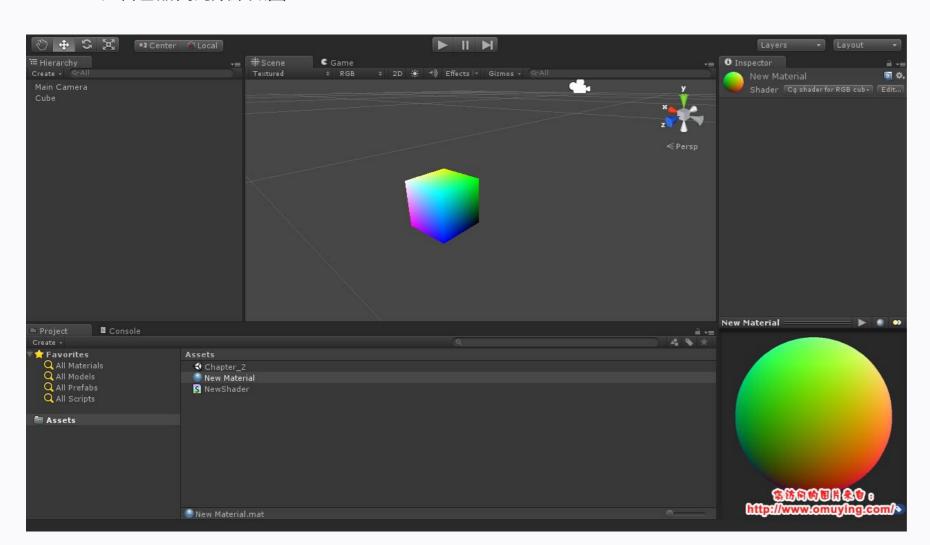
准备工作

如果我们想创建一个带有 RGB 颜色值的立方体,那我们应该首先创建一个立方体对象,你可以像在上一章节中创建球体那样创建一个立方体对象,然后我们还需要创建一个 Shader 以及一个 Material,并且还要给 Material 附加 Shader。

你可以复制/粘贴下面的代码到你的 Shader 代码编辑器中:

```
Shader "Cg shader for RGB cube"
  02
     {
         SubShader
  03
 04
         {
            Pass
  05
 06
               CGPROGRAM
  07
 08
               // vert function is the vertex shader
  09
               #pragma vertex vert
               // frag function is the fragment shader
10
               #pragma fragment frag
  11
 12
  13
               // for multiple vertex output parameters an output structure
               // is detined:
  15
               struct vertexOutput {
                  float4 pos : SV_POSITION;
  16
  17
                  float4 col : TEXCOORD0;
  18
               };
  19
               // vertex shader
  20
               vertexOutput vert(float4 vertexPos : POSITION)
  21
  22
                  vertexOutput output; // we don't need to type 'struct' here
  23
  24
                  output.pos = mul(UNITY MATRIX MVP, vertexPos);
  25
                  output.col = vertexPos + float4(0.5, 0.5, 0.5, 0.0);
  26
                  // Here the vertex shader writes output data
  27
                  // to the output structure. We add 0.5 to the
  28
                  // x, y, and z coordinates, because the
  29
                  // coordinates of the cube are between -0.5 and
  30
                  // 0.5 but we need them between 0.0 and 1.0.
  31
                  return output;
  32
  33
               // fragment shader
  34
               float4 frag(vertexOutput input) : COLOR
  35
  36
                  return input.col;
  37
                  // Here the fragment shader returns the "col" input
  38
                  // parameter with semantic TEXCOORD0 as nameless
  39
                  // output parameter with semantic COLOR.
  40
  41
               ENDCG
  42
  43
 44 }
```

如果你的立方体现在还没有颜色显示,你应该检查有没有错误消息,并且还应确保已经给立方体对象添加了 Material。着色器代码效果如图:



顶点与片段着色器之间的相互通信

着色器的主要任务是从顶点着色器中获得可用的顶点位置信息并通过片段着色器来设置颜色值(片段着色器的输出参数 COLOR)。但这也并不是完全正确的,在 Unity 立方体中顶点输入参数 POSITION 的坐标值范围在 -0.5 至 +0.5,而我们期望颜色分量的范围是 0.0 至 1.0。所以我们需要通过 vertexPos + float4(0.5, 0.5, 0.5, 0.0) 表达式分别给 x、y、z 分量加上 0.5。

然而,现在的主要问题是如何从顶点着色器获取值并传给片段着色器?唯一的方法就是顶点着色器的输出参数与片段着色器的输入参数具有相同语义词(本例为 TEXCOORD0)。事实上,只是为了确定顶点着色器的输出参数使用的语义词与片段着色器的哪个输入参数中的语义词对应。除了语义词 TEXCOORD0,我们也可以用语义词 COLOR,他们不同的地方只在于 COLOR 的取值范围为 0 至 1。但是 TEXCOORD0、TEXCOORD1、TEXCOORD2 参数是相同的,可以相互使用。

接下来的问题是,因为顶点着色器的返回指令只能返回一个值,如果需要指定多个输出参数,我们可以在顶点输出参数的结构体中声明。在这儿它被称为"vertexOutput":

```
1 struct vertexOutput {
2    float4 pos : SV_POSITION;
3    float4 col : TEXCOORD0;
4 };
```

在片段着色器函数中使用这个结构体作为参数,我们确保了语义匹配。注意在 Cg 语言中,我们用 vertexOutput 结构体声明变量时不需要再编写结构体,只要名称与 vertexOutput 相同就可以了。

另一种替代的解决方案是在顶点着色器函数的参数中使用 out 限定符,例如:

```
Shader "Cg shader for RGB cube"
02
       SubShader
03
04
05
          Pass
             CGPROGRAM
             // vert function is the vertex shader
08
09
             #pragma vertex vert
10
             // frag function is the fragment shader
             #pragma fragment frag
11
12
13
             void vert(float4 vertexPos : POSITION,
                out float4 pos : SV_POSITION,
14
15
                out float4 col : TEXCOORD0)
16
                pos = mul(UNITY_MATRIX_MVP, vertexPos);
17
18
                col = vertexPos + float4(0.5, 0.5, 0.5, 0.0);
19
                return;
20
21
             float4 frag(float4 pos : SV_POSITION,
22
```

【原创】Shader 内置 Shader 之 Diffuse Detail 学习 - 1460 次阅读

打

【翻译】第四章节:世界空间中的着色器(关于uniforms) - 2327次阅读

哲无图片

【原创】Shader 内置 Shader 之 Specular 学 习 - 2024 次阅读

新无图片

【翻译】第十章节:镜面高 光(关于每顶点光 照) - 2007次阅读

暂无图片

【翻译】概要: Unity Cg 编程(关于顶点与片段着 色器) - 7680次阅读

```
23
                float4 col : TEXCOORD0) : COLOR
24
                return col;
             ENDCG
28
29
30 }
```

但是,把输出定义成一个结构体的方式更加常见,因为它确保了顶点输出参数与片段输入参数具有相同的语 义匹配。

着色器变化

RGB 立方体表示一组可用的颜色集(即显示器的色域)。因此,它还可以用于显示颜色的变换效果。例 如:一个颜色的灰度变换将平均计算红、绿、蓝颜色分量,即 (red + green + blue)/3,然后将这个 值应用到片段所有颜色的三个颜色分量中来获得相同强度的灰度值,还有相对亮度,计算公式是 0.21 red + 0.72 green + 0.07 blue。当然颜色的其他任何变换(改变饱和度、对比度、色调等) 也都适用。

我们还可以用这个着色器来实现 CMY (青色、品红色、黄色) 立方体,根据位置信息 (x,y,z),你可以 从纯白色中减去 x (红色分量) 来产生青色,减去 y (绿色分量) 来产生品红色,减去 z (蓝色分量) 来产 生黄色。

如果还想弄点新花样,你可以计算一个HSV(色调,饱和度,亮度)柱面(cylinder)。因为 x 与 z 的坐 标值范围在 -0.5 至 +0.5, 在 Cg 中, 你可以通过公式 180.0 + degrees(atan2(z, x)) 来得到一个 0 至 360 度的夹角 H , 还可以通过公式 2.0 * $\mathsf{sqrt}(\mathsf{x} * \mathsf{x} + \mathsf{z} * \mathsf{z})$ 在 y 轴上得到一个值在 0 至 1 之间的距离 S 。 Unity 内置柱面 (cylinder) 坐标 y 的值范围是 -1 至 1 , 通过公式 (y+1.0)/2.0 转换之后 , 值 V 的 范围在 0 到 1。从 HSV 坐标计算 RGB 颜色可以查看《article on HSV in Wikipedia》。

顶点输出参数插值

关于顶点输出参数和片段输入参数的介绍还没完,选中立方体对象,你可以在 Scene 视图中看到立方体对 象有 12 个三角形和 8 个顶点组成,因此,顶点着色器只有 8 次called 以及只有 8 次不同的输出被写入到 顶点输出参数中,然而立方体有很多的颜色,这是什么情况呢?

事实上,顶点着色器只 called 每个三角形的每个顶点,然而,对于不同的顶点,顶点输出参数用不同的值 在三角形内进行插值,然后片段着色器把接收到的顶点输出参数中插值数据做为片段输入参数对三角形内的 每个像素进行 called。关于插值的详细信息可以查看《Rasterization》。

如果你想确保片段着色器接收到一个精确的、非插值的顶点着色器,你必须确保顶点着色器在顶点输出参数 中为三角形的所有顶点写入相同的值。

恭喜你,在本教程中,你已经学会了:

- 1、什么是 RGB 立方体。
- 2、什么是输出结构以及如何声明它。
- 3、使用输出结构如何确保顶点输出参数中与片段输入参数有相同的语义。
- 4、在写入片段着色器输入参数之前顶点输出参数如何在三角形内插值。

资源下载地址:点击下载 , 共下载 20 次。

前一篇: Unity3D 各种技能的释放效果

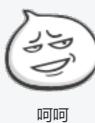
后一篇: Unity3D 使用 TextMesh 显示伤害效果



8人



0人



0人



0人 1人























7条评论

最新 最早 最热



乐乐

"而且这种输出结构在平时的使用中更加常见,因为它确保了顶点输出参数与片段输入参数具 有相同的语义匹配。"这句话翻译错了,跟原文的意思刚好相反,原文是说,"但是,把输出 定义成一个结构体的方式更加常见"~

2014年12月24日 回复 顶 参转发



〆.最终幻想.

回复 乐乐: EN, 正常使用应该是结构体比较多, 当时没注意, 非常感谢修正, 晚上调整一下, 再次感谢,

2014年12月24日 回复 顶 转发



tyt2008cn

谢谢博主的分享,这个讲shader的系列不错

2015年2月28日 回复 顶 转发



好好学习

有一点疑问,对于正方体,顶点着色器是不是24次call呢?

顶点着色器只有 8 次called 以及只有 8 次不同的输出被写入到顶点输出参数中

2015年7月14日 回复 顶 转发



爱奇趣

路过看一下!

2015年9月12日 回复 顶 转发



内涵段子

网站不错,雁过留痕,欢迎互访!

2015年10月2日 回复 顶 转发



changok88

学习了,谢谢楼主

2015年10月14日 回复 顶 转发

社交帐号登录: ♥️ 微信 🚮 微博 🚇 QQ 👢 人人 更多»



说点什么吧... 0 发布

最终幻想正在使用多说

7条评论

最新 最早 最热



乐乐

"而且这种输出结构在平时的使用中更加常见,因为它确保了顶点输出参数与片段输入参数具 有相同的语义匹配。"这句话翻译错了,跟原文的意思刚好相反,原文是说,"但是,把输出 定义成一个结构体的方式更加常见"~

2014年12月24日 回复 顶 参转发



〆.最终幻想.

回复 乐乐: EN,正常使用应该是结构体比较多,当时没注意,非常感谢修正,晚上调整一下, 再次感谢,

2014年12月24日 回复 顶 转发



tyt2008cn

谢谢博主的分享,这个讲shader的系列不错

2015年2月28日 回复 顶 转发



最终幻想 - 个人博客 | 关于网站 | 联系我们 | 友情链接 | 网站声明

Copyright © 2012-2016 最终幻想 - 个人博客 苏ICP备09017906号-5