

## 【翻译】第二十四章节：表面反射（关于反射贴图）

2014-12-22 08:43:00 1441 人阅读 Unity3D cg 表面反射

A- A+

|   |      |      |                          |
|---|------|------|--------------------------|
| 文章内容  | 例子源码 | 网友评论 | 最后编辑：2014-12-22 21:02:05 |
| 本文永久地址： <a href="http://www.omuying.com/article/115.aspx">http://www.omuying.com/article/115.aspx</a> ，【 <a href="#">文章转载请注明出处！</a> 】 |      |      |                          |

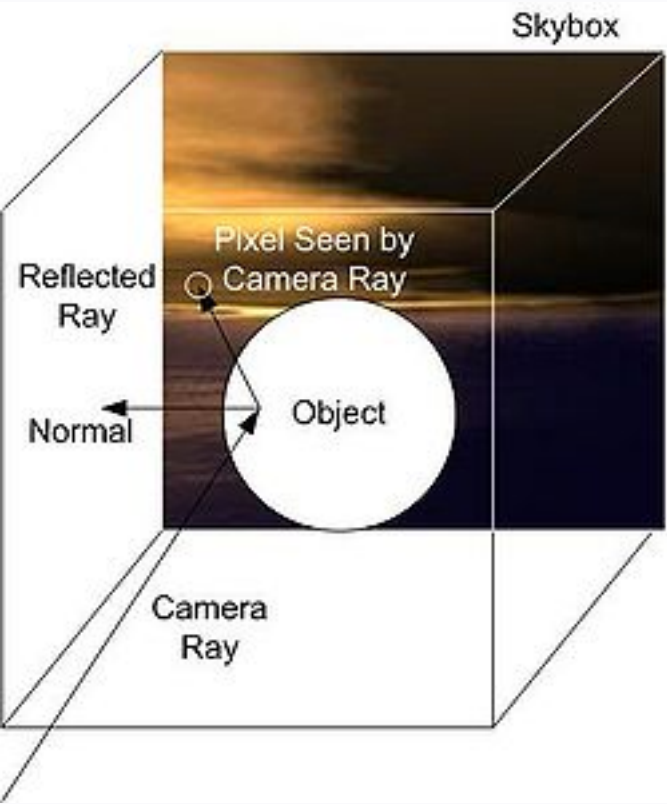
原文链接：[http://en.wikibooks.org/wiki/Cg\\_Programming/Unity/Reflecting\\_Surfaces](http://en.wikibooks.org/wiki/Cg_Programming/Unity/Reflecting_Surfaces)

本教程介绍表面反射（用立方体贴图来实现它）。

本章节是系列教程中第一篇介绍在 Unity 中使用立方体贴图作为环境贴图的教程，本篇教程基于《[平滑镜面高光](#)》章节和《[纹理球](#)》章节。

### 天空盒反射贴图

下面的图片展示了静态天空盒反射贴图的概念：



视图射线被对象表面的点反射，然后通过被反射的射线与天空盒相交来决定对象表面相应点像素的颜色，天空盒只是整个场景周围一个大的立方体纹理面，需要指出的是天空盒通常是静态的并且不包含场景中的任何动态对象，不过如果一个天空盒作为环境贴图使用的话，那可能会在渲染的时候包含场景，不过这点已超出了本教程的范围。

此外，本教程只计算天空盒的反射，并不包含天空盒的渲染，关于天空盒的渲染可以查看《[Skyboxes](#)》章节，对于在一个对象上反射天空盒，我们必须渲染对象和反射从摄像机到表面点的表面法线向量的射线，这个反射的数学公式与在表面法线向量反射光射线相同，详情可以参考《[镜面高光](#)》章节。

一旦我们有了反射射线，我们必须计算它与天空盒的相交，如果这个天空盒无限大，那么这个计算实际上就会很容易：在这种情况下，对象表面点的位置并不重要，因为与天空盒的大小相比，对象表面所有点与坐标系原点的距离是无限小的，因此只有反射射线的方向是重要的，而反射位置并不重要，其实我们还可以认为一个射线是在一个小天空盒的中心开始的，而不是在一个无限大的天空盒的某个地方开始（如果你不熟悉这样的想法，那么你可能需要花点时间来理解这种想法），根据反射射线的方向，它将与天空盒纹理六个面中的一个相交，然后根据计算，我们可以知道它与哪个面相交，并且还可以知道它与这个面的哪个位置相交，然后我们可以在已相交面的纹理图像中查找纹理，然而 Cg 提供支持根据一个方向向量就可以在立方体的六个面中查找纹理，因此我们需要做的就是着色器的属性中提供一个立方体贴图作为环境贴图，然后根据反射方向并使用 texCUBE 指令在立方体贴图的相应位置获得颜色。

### 立方体贴图



#### 最新文章

- 暂无图片

【原创】C# 基础之 Lambda 表达式 - 907 次阅读
- 暂无图片

【原创】C#基础之 IEnumerable 和 IEnumerator - 792 次阅读
- 暂无图片

【原创】C#基础之事件 - 886 次阅读
- 暂无图片

【原创】C#基础之委托 - 912 次阅读
- 暂无图片

【原创】C#基础之委托的使用 - 856 次阅读

#### 随机阅读

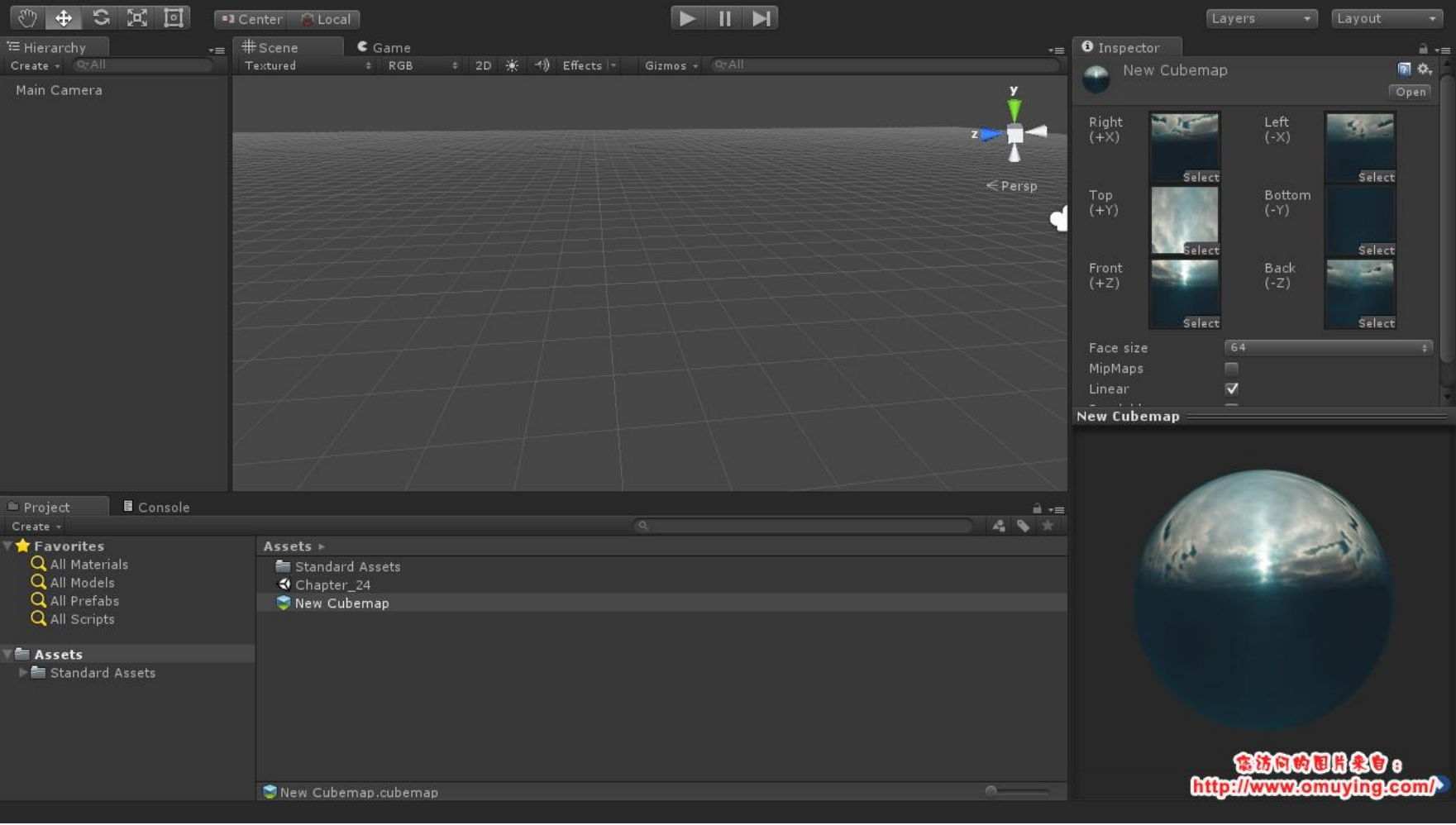
我们可以在 Unity 的着色器中声明一个名为 \_Cube 的立方体贴图属性：

```
1 Properties
2 {
3     _Cube ("Reflection Map", Cube) = "" {}
4 }
```

相应的 uniform 变量可以在 Cg 着色器中像下面这样声明：

```
1 uniform samplerCUBE _Cube;
```

创建一个立方体贴图，在 Project 视图选择 Create -> Cubemap，然后你必须在 Inspector 视图中指定六个纹理图片来作为立方体面的纹理，我们可以从 Standard Assets -> Skyboxes -> Textures 查看这种纹理，如图：



此外对于立方体贴图，你应该在 Inspector 视图中检查 MipMaps，这样做是为反射贴图产生更好的视觉效果。

顶点着色器必须计算视图方向 viewDir 和法线方向 normalDir，这在《[镜面高光](#)》章节中有讨论，我们可以在片段着色器中使用 Cg 函数 reflect 来反射视图方向，这在《[镜面高光](#)》章节中也有讨论：

```
1 float3 reflectedDir = reflect(input.viewDir, normalize(input.normalDir));
```

然后我们可以像下面那样在立方体贴图中执行纹理查找和在片段着色器存储颜色结果：

```
1 return texCUBE(_Cube, reflectedDir);
```

完成着色器代码

最后着色器的代码如下：

```
01 Shader "Cg shader with reflection map"
02 {
03     Properties
04     {
05         _Cube("Reflection Map", Cube) = "" {}
06     }
07     SubShader
08     {
09         Pass
10         {
11             CGPROGRAM
12
13             #pragma vertex vert
14             #pragma fragment frag
15
16             #include "UnityCG.cginc"
17
18             // User-specified uniforms
19             uniform samplerCUBE _Cube;
20 }
```

暂无图片

【原创】Shader 内置 Shader 之 Specular 学习 - 2024 次阅读

暂无图片

【翻译】第十四章节：多个灯（关于在一个 pass 中遍历处理多个光源） - 1932 次阅读

暂无图片

【翻译】第二章节：RGB 立方体（关于顶点输出参数） - 2256 次阅读

暂无图片

【翻译】第十一章节：双面表面（关于双面每顶点光照） - 1186 次阅读

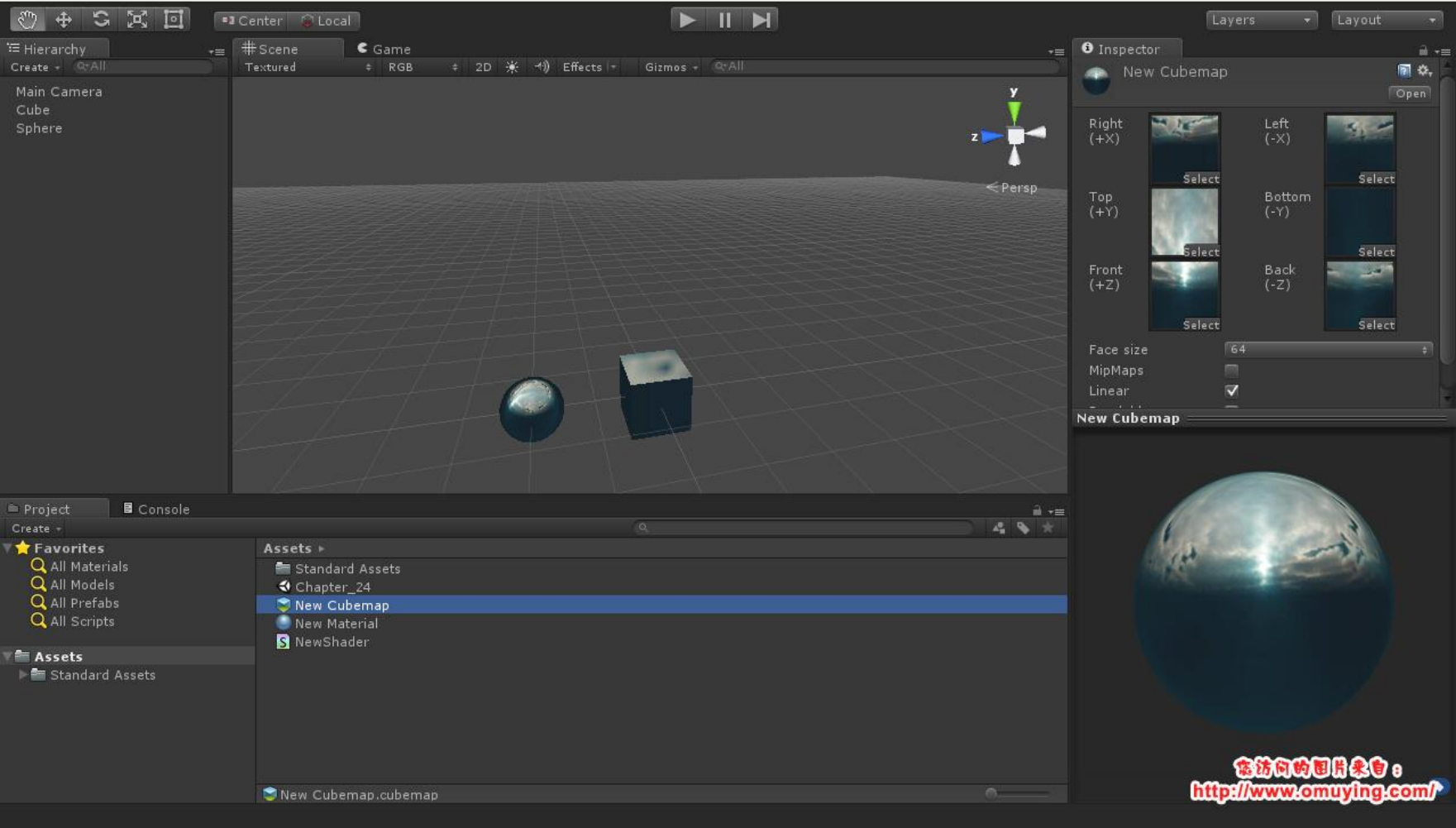
暂无图片

【原创】Shader 内置 Shader 之 Vertex Lit 学习 - 3365 次阅读



```
21 struct vertexInput
22 {
23     float4 vertex : POSITION;
24     float3 normal : NORMAL;
25 };
26 struct vertexOutput
27 {
28     float4 pos : SV_POSITION;
29     float3 normalDir : TEXCOORD0;
30     float3 viewDir : TEXCOORD1;
31 };
32
33 vertexOutput vert(vertexInput input)
34 {
35     vertexOutput output;
36
37     float4x4 modelMatrix = _Object2World;
38     float4x4 modelMatrixInverse = _World2Object;
39     // multiplication with unity_Scale.w is unnecessary
40     // because we normalize transformed vectors
41
42     output.viewDir = mul(modelMatrix, input.vertex).xyz -
    _WorldSpaceCameraPos;
43     output.normalDir = normalize(mul(float4(input.normal, 0.0),
    modelMatrixInverse).xyz);
44     output.pos = mul(UNITY_MATRIX_MVP, input.vertex);
45     return output;
46 }
47
48 float4 frag(vertexOutput input) : COLOR
49 {
50     float3 reflectedDir = reflect(input.viewDir,
    normalize(input.normalDir));
51     return texCUBE(_Cube, reflectedDir);
52 }
53 ENDCG
54 }
55 }
56 }
```

这个着色器的效果如图：



恭喜你，在本章节中你应该了解：

- 1、怎样计算一个对象的天空盒反射。
- 2、如何在 Unity 中生成立方体贴图以及如何把它作为反射贴图来使用。

资源下载地址：[点击下载](#)，共下载 25 次。

前一篇：[第二十三章节：投影（关于使用投影纹理贴图实现投影）](#)

后一篇：[第二十五章节：弧形玻璃（关于折射贴图）](#)



赞

5 人

打酱油

0 人

呵呵

0 人

鄙视

0 人

正能量

0 人







0

0条评论


最新 最早 最热

还没有评论，沙发等你来抢

社交帐号登录:  微信  微博  QQ  人人 [更多»](#)



说点什么吧...





 发布


最终幻想正在使用多说

0条评论


最新 最早 最热

还没有评论，沙发等你来抢

社交帐号登录:  微信  微博  QQ  人人 [更多»](#)



说点什么吧...

 发布

最终幻想正在使用多说