



唐老狮系列教程

延迟渲染路径



唐老狮系列教程-延迟渲染路径

| 主要讲解内容



唐老狮系列教程-延迟渲染路径

主要讲解内容

1. 延迟渲染路径处理光照的方式
2. 延迟渲染路径在哪里进行光照计算
3. 延迟渲染路径的内置光照变量



唐老狮系列教程-延迟渲染路径

延迟渲染路径处理光照的方式



唐老狮系列教程-延迟渲染路径

延迟渲染路径处理光照的方式

延迟渲染路径**对光照的数量没有任何限制，并且所有灯光都可以采用逐像素渲染**。理论上来说，即使场景中有成百上千个实时灯光，依然可以保持比较流畅的渲染帧率。

它支持法线纹理、阴影等等效果的处理；但是它不能处理半透明物体，并且不支持真正的抗锯齿。这些会自动使用前向渲染路径。

延迟渲染的效率不依赖于场景的复杂度，而是和我们使用的屏幕空间的大小有关。

这是因为延迟渲染路径中除了使用颜色缓冲和深度缓冲外，还会利用一个叫做G缓存的额外缓存区。它会存储我们关心的表面（通常是离摄像机最近的表面）的其他信息，比如表面的法线、位置、材质属性等等。总之我们需要的信息都存储到缓冲区中，而这些缓冲区可以理解为一一张张的2D图片，我们实际上是在这些图像空间中进行处理的。



唐老狮系列教程-延迟渲染路径

延迟渲染路径在哪里进行光照计算



唐老狮系列教程-延迟渲染路径

延迟渲染路径在哪里进行光照计算

要进行光照计算，那肯定是在Shader当中的**Pass渲染通道中进行计算**。

延迟渲染路径中主要包含两个Pass

第一个Pass（对于每个物体，该Pass只会执行一次，通常无需我们自己实现）

主要判断哪些片元可见，并且将可见片元的相关信息存储到G缓冲区中

比如：表面法线、视角方向、漫反射系数等等数据

第二个Pass

```
Pass
{
    Tags{ "LightMode" = "Deferred" }
```

利用G缓冲区中各个片元的相关信息进行真正的相关计算，最终将颜色写入颜色缓冲区

注意：

在第二个Pass中计算光照时，默认情况下只能用Unity内置的标准(Standard)光照模型计算



唐老狮系列教程-延迟渲染路径

| 延迟渲染路径的内置光照变量



唐老狮系列教程-延迟渲染路径

顶点照明渲染路径的内置光照变量

常用内置光照变量

变量名	类型	描述
sampler2D _CameraGBufferTexture0; sampler2D _CameraGBufferTexture1; sampler2D _CameraGBufferTexture2; sampler2D _CameraGBufferTexture3;	sampler2D	这些变量是自定义变量，我们一般无需实现延迟渲染路径中的第一个Pass，G缓冲中的相关数据Unity会帮助我们存储到这些自定义变量中，我们可以直接使用他们来参与计算 _CameraGBufferTexture0:漫射颜色 (RGB)，遮挡 (A)。 _CameraGBufferTexture1:镜面反射颜色 (RGB)，粗糙度 (A)。 _CameraGBufferTexture2:世界空间法线 (RGB)，未使用 (A)。 _CameraGBufferTexture3:发射 + 光照 + 光照贴图 + 反射探针缓冲区 _CameraGBufferTexture4:光照遮挡值 (RGBA)
_LightColor	float4	光源颜色（#include "UnityDeferredLibrary.cginc"）
_LightMatrix0	float4x4	世界空间到光源空间的变换矩阵 （#include "UnityDeferredLibrary.cginc"）

```
#include "UnityGBuffer.cginc"

sampler2D _CameraGBufferTexture0;
sampler2D _CameraGBufferTexture1;
sampler2D _CameraGBufferTexture2;
```

```
// unpack Gbuffer
half4 gbuffer0 = tex2D (_CameraGBufferTexture0, uv);
half4 gbuffer1 = tex2D (_CameraGBufferTexture1, uv);
half4 gbuffer2 = tex2D (_CameraGBufferTexture2, uv);
UnityStandardData data = UnityStandardDataFromGbuffer(gbuffer0, gbuffer1, gbuffer2);
```



唐老狮系列教程-延迟渲染路径

| 总结



唐老狮系列教程-延迟渲染路径

主要讲解内容

1. 延迟渲染路径处理光照的方式

对光照的数量没有任何限制,并且所有灯光都可以采用逐像素渲染

它不能处理半透明物体, 并且不支持真正的抗锯齿

2. 延迟渲染路径在哪里进行光照计算

第一个Pass (对于每个物体, 该Pass只会执行一次, 通常无需我们自己实现)

主要判断哪些片元可见, 并且将可见片元的相关信息存储到G缓冲区中

第二个Pass

利用G缓冲区中各个片元的相关信息进行真正的相关计算, 最终将颜色写入颜色缓冲区

3. 延迟渲染路径的内置光照变量

Unity会自动填充自定义变量 `sampler2D_CameraGBufferTexture0~4` , 其中存储的就是G缓冲区中的数据

我们可以通过他们获取数据来进行逻辑处理



唐老狮系列教程

Thank
感谢您的聆听