



唐老狮系列教程

前向渲染路径



唐老狮系列教程-前向渲染路径

| 主要讲解内容



唐老狮系列教程-前向渲染路径

主要讲解内容

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



唐老狮系列教程-前向渲染路径

前向渲染路径处理光照的方式



唐老狮系列教程-前向渲染路径

前向渲染路径处理光照的方式

前向渲染路径中会将光源分为以下3种处理方式：

- 1.逐像素处理（需要高等质量处理的光）
- 2.逐顶点处理（需要中等质量处理的光）
- 3.球谐函数（SH）处理（需要低等质量处理的光）

球谐函数处理光照的方式是将光照场景投影到球谐函数的空间中，通过一组球谐系数来表示光照。

内存换性能，细节表现效果差（不需要我们自己书写，Unity底层会帮助我们进行处理）

场景当中的各种光源将采用哪种方式处理？



唐老狮系列教程-前向渲染路径

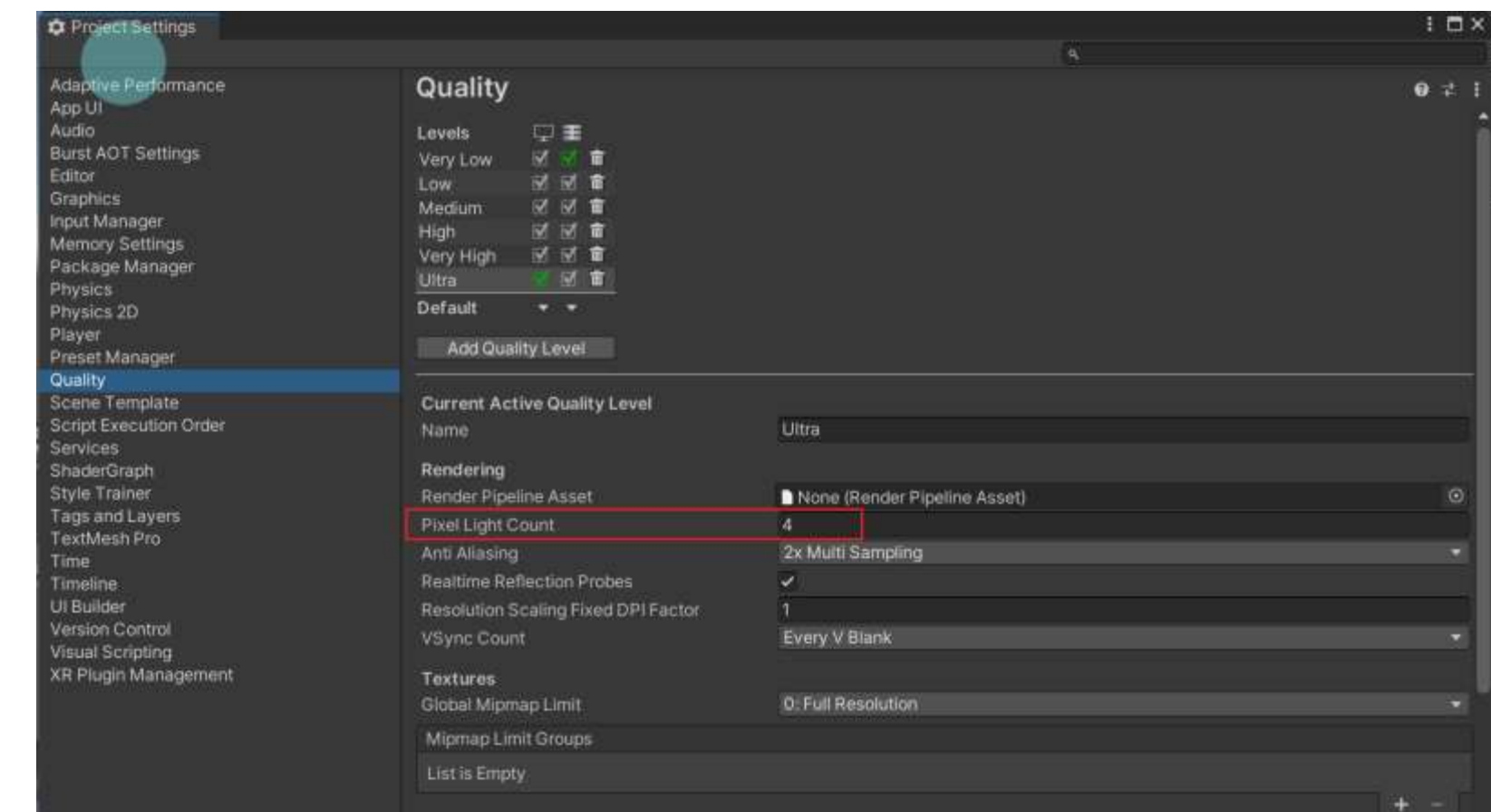
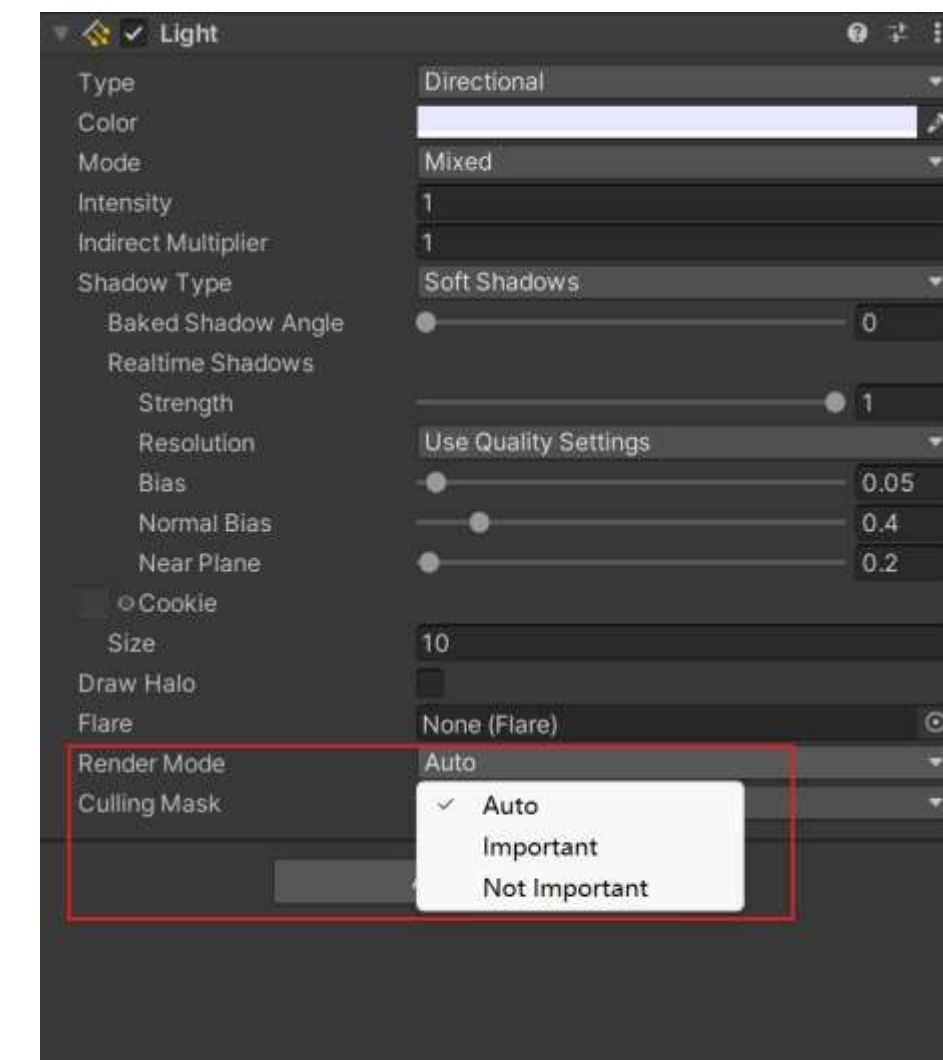
场景当中的各种光源将采用哪种方式处理？

在前向渲染中，一部分最亮的灯光以逐像素处理，然后4个点光源以逐顶点方式处理，其余的灯光以SH处理

一个光源是逐像素、逐顶点还是SH处理主要取决于以下几点：

1. 渲染模式设置为Not Important(不重要)的灯光始终以逐顶点或者SH的方式渲染
2. 渲染模式设置为Important(重要)的灯光始终是逐像素渲染
3. 最亮的平行光总是逐像素渲染
4. 如果逐像素光照的灯光数量少于项目质量设置中的Pixel Light Count(像素灯光计数)的数量，那么其余比较亮的灯光将会被逐像素渲染

注意：如果灯光渲染模式设置为Auto(自动)，Unity会根据灯光的亮度以及与物体的距离自动判断重要性





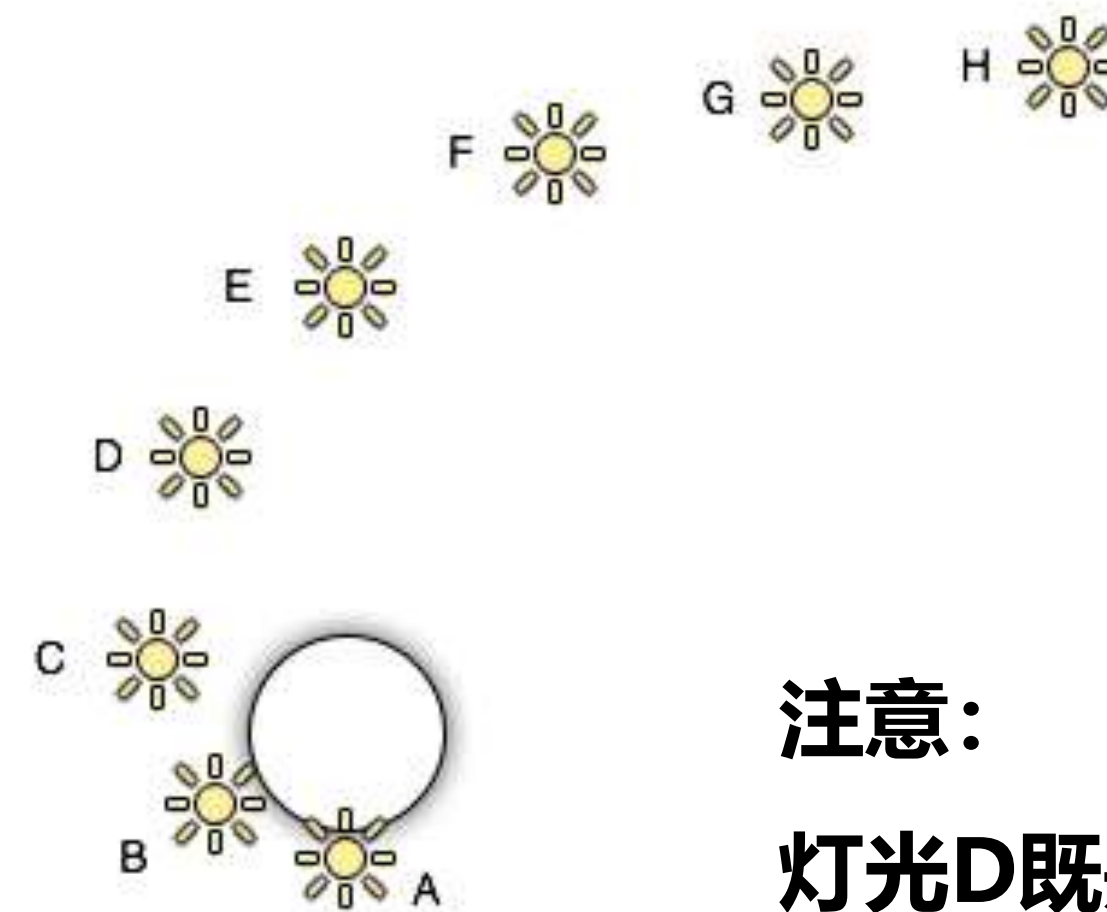
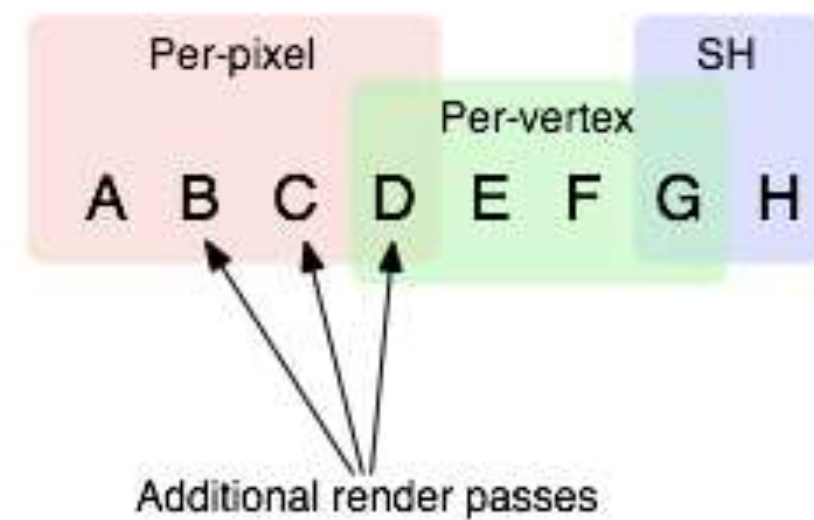
唐老狮系列教程-前向渲染路径

场景当中的各种光源将采用哪种方式处理？

举例说明：

如右图

A~H是8个具有相同的颜色和强度
并且光源渲染模式都是Auto(自动)的



注意：

灯光D既是逐像素也是逐顶点处理

灯光G既是逐顶点也是SH处理

是因为物体移动或灯光移动时，不同渲染模式的灯光交界处会出现明显瑕疵，为了避免该问题

Unity将不同灯光组之间进行了重叠

其中最亮的光源以**逐像素光照模式渲染**（A~D）

然后最多**4个光源**以**逐顶点光照模式渲染**（D~G）

最后剩下的光源以**SH模式渲染**（G~H）



唐老狮系列教程-前向渲染路径

前向渲染路径处理光照的方式

简而言之：

Unity当中有一套划分光源“三六九等”的规则

主要通过 灯光渲染模式、项目质量设置中的像素灯光计数的数量、光照强度、距离物体距离来综合判定

在前向渲染路径中

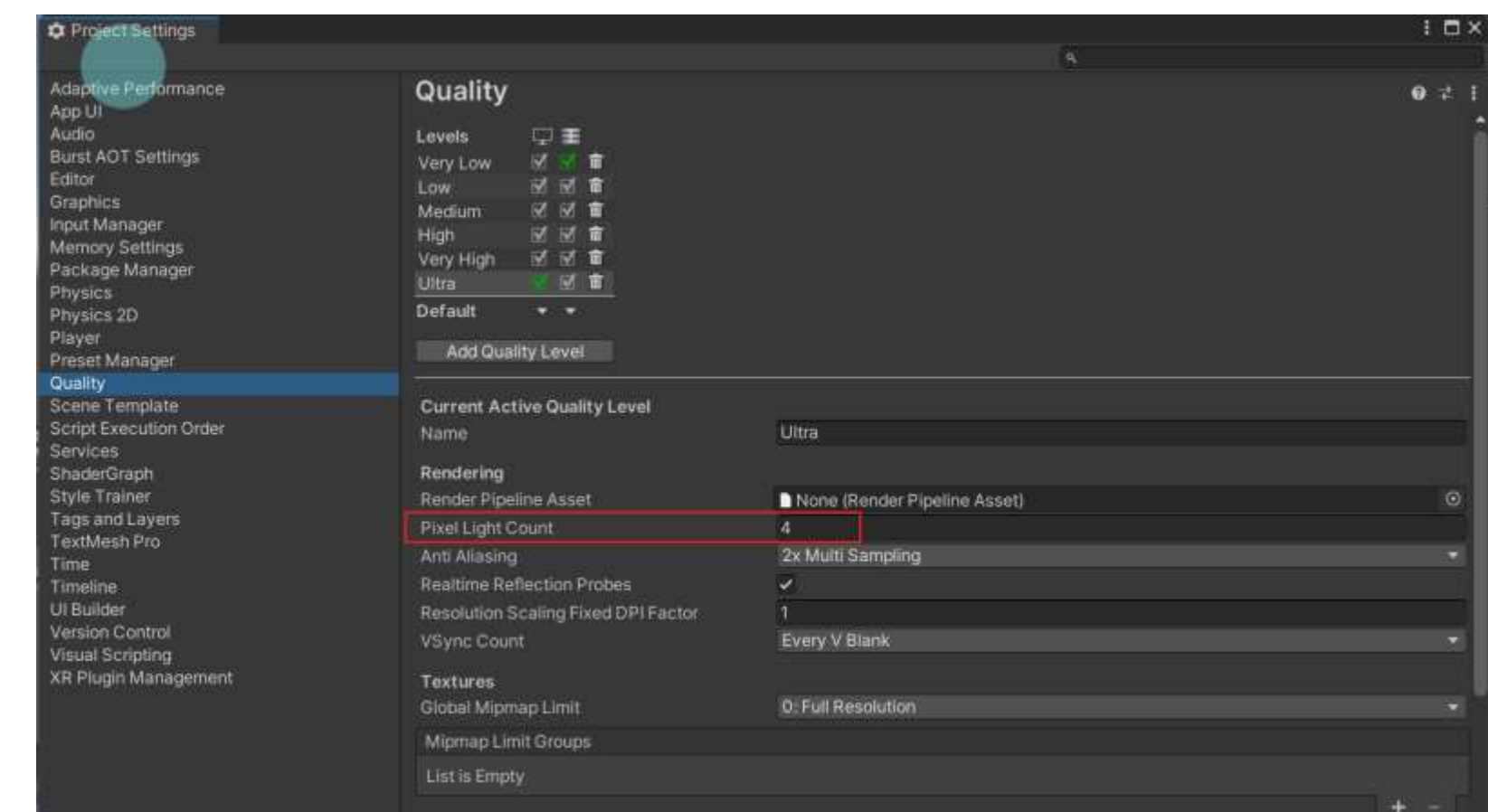
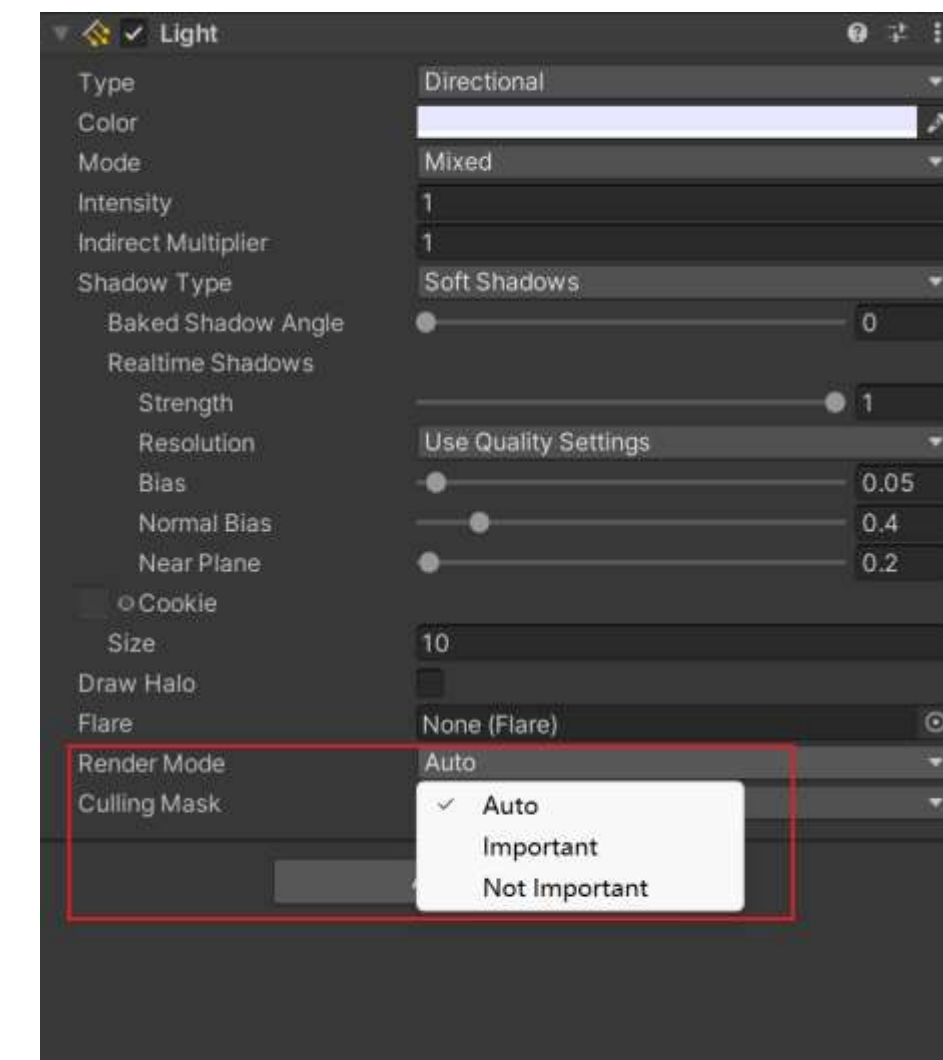
会将光源分成所谓的逐像素、逐顶点、SH三种处理类型

有了对光源的“高中低”的身份认知

Unity底层就可以将这些光源的数据存储到Shader中对应的内置变量中

我们就可以通过这些内置变量获取到对应“身份”的光源数据

从而进行差异化的处理





唐老狮系列教程-前向渲染路径

前向渲染路径在哪里进行光照计算



唐老狮系列教程-前向渲染路径

前向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**。

但是对于前向渲染来说，有两种Pass可以用来进行光照处理：

1.Base Pass（基础渲染通道）

渲染物体的主要光照通道，用于处理主要的光照效果

主要用于计算逐像素的平行光以及所有逐顶点和SH光源

可实现的效果：漫反射、高光反射、自发光、阴影、光照纹理等

```
Pass
{
    //Base Pass
    Tags{ "LightMode" = "ForwardBase" }
```

2.Additional Pass（附加渲染通道）

渲染物体额外的光照通道，用于处理一些附加的光照效果

主要用于计算其他影响物体的逐像素光源

每个光源都会执行一次该Pass

可实现的效果：描边、轮廓、辉光等

```
Pass
{
    //Additional Pass
    Tags{ "LightMode" = "ForwardAdd" }
    //用于和其他光照颜色进行混合叠加
    Blend One One
```




唐老狮系列教程-前向渲染路径

前向渲染路径在哪里进行光照计算

对于一个前向渲染路径下的Unity Shader

通常会**定义一个Base Pass（基础渲染通道）**以及**一个Additional Pass（附加渲染通道）**

每次渲染时

一个Base Pass仅会执行一次（多个Base Pass情况除外）主要用于渲染环境光或自发光等

而一个Additional Pass会根据影响该物体的其他逐像素光源的数量被多次调用

每个逐像素光源都会调用一次Additional Pass 由于开启了混合，渲染结果会和之前的光照颜色进行混合

注意：

1. Base Pass也可以有多个，比如需要双面渲染的情况

2. Base Pass默认支持阴影，Additional Pass默认不支持

可以通过添加#pragma multi_compile_fwdadd_fullshadows编译指令开启阴影

3. 这些Pass当中我们具体处理光照的方式是由我们自己决定的，使用逐顶点光照还是逐像素光照的计算方式

都根据我们的具体实现而定，前文提到的逐像素光源只是按照期望处理类型来分的而已

```
CGPROGRAM
#pragma vertex vert
#pragma fragment frag
//为点光源和聚光灯开启阴影效果
#pragma multi_compile_fwdadd_fullshadows
#include "UnityCG.cginc"
#include "Lighting.cginc"
//处理贴图对应的映射关系
```




唐老狮系列教程-前向渲染路径

前向渲染路径在哪里进行光照计算

简而言之：

Base Pass（基础渲染通道）：

主要用于处理影响该物体的一个高质量光源（如平行光）、所有中（逐顶点处理）低质量（SH处理）光源等

Additional Pass（附加渲染通道）：

主要用于处理影响该物体的除平行光以外的其它高质量光源（每个高质量光源都会调用）

```
Pass
{
    //Base Pass
    Tags{ "LightMode" = "ForwardBase" }
```

```
Pass
{
    //Additional Pass
    Tags{ "LightMode" = "ForwardAdd" }
    //用于和其他光照颜色进行混合叠加
    Blend One One
```

知道了在哪里处理对应等级的光照计算

接下来我们来学习，我们应该如何获取到对应“身份”的光源数据

从而进行差异化的处理



唐老狮系列教程-前向渲染路径

前向渲染路径的内置光照变量和函数



唐老狮系列教程-前向渲染路径

前向渲染路径的内置光照变量和函数

常用内置光照变量

变量名	类型	描述
_LightColor0	float4	该Pass当前处理的逐像素(高质量)光源的颜色
_WorldSpaceLightPos0	float4	_WorldSpaceLightPos0.xyz 表示该Pass当前处理的逐像素(高质量)光源的位置，如果该光源是平行光那么_WorldSpaceLightPos0.w是0 其他光源类型w为1
_LightMatrix0	float4x4	世界空间到光源空间的变换矩阵 可以用于采样光强衰减纹理和cookie
unity_4LightPosX0 unity_4LightPosY0 unity_4LightPosZ0	float4	仅用于Base Pass。前4个非重要的点光源在世界空间中的位置
unity_4LightAtten0	float4	仅用于Base Pass。存储了前4个非重要的点光源的衰减因子
unity_LightColor	half4[4]	仅用于Base Pass。存储了前4个非重要的点光源的颜色



唐老狮系列教程-前向渲染路径

前向渲染路径的内置光照变量和函数

常用内置光照函数

变量名	描述
float3 WorldSpaceLightDir(float4 v)	仅用于前向渲染路径中。输入模型空间中顶点位置，返回世界空间中从该点到光源的光照方向。结果没有被归一化
float3 UnityWorldSpaceLightDir(float4 v)	仅用于前向渲染路径中。输入一个世界空间中的顶点位置，返回世界空间中从该点到光源的光照方向。结果没有被归一化
float3 ObjSpaceLightDir(float4 v)	仅用于前向渲染路径中。输入一个模型空间中的顶点位置，返回模型空间中从该点到光源的光照方向。结果没有被归一化
float3 Shade4PointLights(float4 lightPosX, float4 lightPosY, float4 lightPosZ, float3 lightColor0, float3 lightColor1, float3 lightColor2, float3 lightColor3, float4 lightAttenSq, float3 pos, float3 normal)	仅用于前向渲染路径中。计算四个点光源的光照，它的参数非常多，主要用的就是刚才讲解的unity_相关的参数，都是非重要的点光源相关的数据。可以使用它来计算逐顶点光照。返回值是四个点光源的叠加颜色，之后用到再细讲参数
等等	以后用到再讲



唐老狮系列教程-前向渲染路径

前向渲染路径的内置光照变量和函数

简而言之：

通过这些内置光照变量，我们便可以获取到前向渲染路径当中，
对渲染质量有不同要求的光源们的相关数据。

我们只需要利用这些数据和函数在

Base Pass（基础渲染通道）和 Additional Pass（附加渲染通道）

中按照需求进行相关的逻辑处理即可



唐老狮系列教程-前向渲染路径

| 总结



唐老狮系列教程-前向渲染路径

主要讲解内容

1. 前向渲染路径处理光照的方式

有一套划分光源“三六九等”的规则，将光源分成了高中低三种身份

主要通过 灯光渲染模式、项目质量设置中的像素灯光计数的数量、光照强度、距离物体距离来综合判定

2. 前向渲染路径在哪里进行光照计算

Base Pass（基础渲染通道）：

主要用于处理影响该物体的一个高质量光源（平行光）、所有中（逐顶点处理）低质量（SH处理）光源 等

Additional Pass（附加渲染通道）：

主要用于处理影响该物体的除平行光以外的其它高质量光源（每个高质量光源都会调用）

3. 前向渲染路径的内置光照变量和函数

有了内置光照变量，我们便可以获取到前向渲染路径当中，对渲染质量有不同要求的光源们的相关数据

利用这些数据和函数在Base Pass（基础渲染通道）和 Additional Pass（附加渲染通道）中按照需求进行相关的逻辑处理即可



唐老狮系列教程

Thank
感谢您的聆听