

SimplePBR 样本

# *此样本与 Windows 10 October 2018 Update SDK (17763) 兼容*

# */2016 年 8 月 Xbox One XDK。*

# 描述

此样本演示了使用 DirectX 12 在 UWP 和 Xbox One XDK 上进行基于物理的渲染 (PBR)。样本将正向渲染的迪士尼风格 PBR 实现为具有以下参数的独立着色器：

1. 反照率：未照明的基本 RGB 颜色
2. 法线贴图：未压缩的 3 通道法线贴图（Y 轴截距为正），

以及 RMA 贴图指定：

1. 平整度：[0..1]，正态分布意味着镜面高光的尺寸和形状。根据迪士尼说明文件，平整度可进行缩放。
2. 金属：（通常为 0 或 1，可混合），控制折射率、反射率的镜面反射与漫反射分布。
3. 环境光遮蔽：一个值[0..1]，测量镜面反射和漫反射作用。

参数可仅表示为常数，也可仅表示为纹理（但非混合）。着色器支持基于图像的照明（具有预先计算的漫反射和镜面反射贴图）和平行光。

有关 PBR 的更多详细信息，请参阅本文档末尾的实现/参考部分。



# 构建示例

构建并运行 SimplePBRUWP12 或 SimplePBRXbox12 解决方案。所有资产都在“资产”子目录中共享，共享代码（包括着色器源）位于“共享”子目录中。着色器由 Visual Studio 2015 项目编译，无需外部构建步骤。

# 使用样本

在 UWP 上，使用鼠标（鼠标滚轮，右键单击）和键盘（W、A、S、D 键）在场景中移动相机。

UWP 和 Xbox One 也可以使用游戏手柄相机控制：使用左右手柄来改变相机方向和半径。

# 实施说明

PBREffect 类覆盖着色器的实现。着色器有两种配置 - 常量和纹理。常量配置主要用于调试。在纹理配置中，输入参数（反照率和平整度、金属、AO）被指定为纹理。

要创建纹理着色器，请使用 EffectFlags 枚举：

m\_effect = std::make\_unique<ATG::PBREffect>(device, EffectFlags::Texture, pipelineState);

如需设置纹理参数，只需为每个纹理和采样器传入描述符：

m\_effect->SetSurfaceTextures(m\_descriptors->GetGpuHandle(AlbedoIndex),

m\_descriptors->GetGpuHandle(NormalIndex),

m\_descriptors->GetGpuHandle(RoughnessMetallicAOIndex),

commonStates->AnisotropicWrap());

着色器作为 Visual Studio 项目的一部分进行编译，并分为三个文件

1. PBREffect\_VSConstant - 共享顶点着色器
2. PBREffect\_PSConstant - 常量参数像素着色器
3. PBREffect\_PSTextured - 纹理参数像素着色器

还有两个 HLSL 包括

1. PBREffect\_Math - BRDF 等共享数学函数
2. PBREffect\_Common - 根签名、常量和共享照明功能“PBR\_LightSurface”。

## 照明

PBREffect 支持定向和基于图像的照明。呼叫者必须提供预先计算的辐照度纹理（用于漫射环境照明）和辐射纹理（用于镜面环境照明）。 纹理应为 HDR 格式的 DDS 立方体贴图。

呼叫者还必须指定辐射纹理中的 MIP 级别数。如需了解 PBR 生成预先计算的贴图的更多详细信息，请参阅“[基于物理进行渲染的 AMD Cubemapgen](https://seblagarde.wordpress.com/2012/06/10/amd-cubemapgen-for-physically-based-rendering/)”。

m\_effect->SetIBLTextures(

m\_descriptors->GetGpuHandle(m\_radTexDescIndex),

m\_radianceTexture->*GetDesc*().*MipLevels*,

m\_descriptors->GetGpuHandle(m\_irrTexDescIndex),

m\_commonStates->AnisotropicClamp());

可选的是，呼叫者还可以使用 SetLight\* 方法指定定向照明。着色器将混合定向和图像照明。

## 参考资料

<https://www.allegorithmic.com/system/files/software/download/build/PBR_Guide_Vol.1.pdf>

<https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf>

<http://blog.selfshadow.com/publications/s2015-shading-course/>

<http://graphicrants.blogspot.com/2013/08/specular-brdf-reference.html>

<https://github.com/dariomanesku/cmftStudio>

## Xbox 上的 UWP

如需在 Xbox One 上支持 DirectX 12，UWP 应用必须将其应用类型设置为“游戏”。否则，只有开发人员主机上的软件设备 (WARP12) 可用，不支持零售主机。在开发期间，可以通过 DevHome 设置应用程序类型。此样本使用了默认情况下实现此功能的 [expandedResources](https://msdn.microsoft.com/zh-cn/library/windows/desktop/mt808808.aspx) 限制功能，但可在将打包应用类型设置为“游戏”后将其删除。请注意，如果使用此限制功能，那么提交到 Windows Store 的应用将无法通过验证。

# 已知的问题

* 此样本中的着色器不支持 Alpha 通道

注意：PBREffect 和此样本中的着色器已整合至 DirectX 具包用于 [DX11](https://github.com/Microsoft/DirectXTK/wiki/PBREffect) / [Dx12](https://github.com/Microsoft/DirectXTK12/wiki/PBREffect)，包括支持反照率纹理中的 alpha 通道以及可选的放射性纹理。

# 更新历史记录

初始发布 2016 年 10 月。于 2017 年 6 月更新，以利用 [DirectX 工具包音调映射器](https://github.com/Microsoft/DirectXTK12/wiki/PostProcess)。

于 2017 年 10 月更新，以支持 Xbox One S / Xbox One X 上的 HDR10 输出以及配置了 HDR 输出的 Windows 10 Creators Update PC。

2018 年 2 月，RMA 纹理的频道顺序更改为匹配 [glTF2](https://github.com/KhronosGroup/glTF) 规格：*金属性贴图*为蓝色，*平整度*为绿色，*环境光遮蔽*为红色。

2019 年 1 月更新了法线纹理的着色器来重建 .z 组件，以支持对法线纹理使用 BC5 压缩。

# 隐私声明

编译和运行示例时，示例可执行文件的文件名将发送给Microsoft以帮助跟踪示例使用情况。要选择退出此数据收集，您可以删除Main.cpp中标记为“Sample Usage Telemetry”的代码块。

有关 Microsoft 隐私政策的更多信息，请参阅 [Microsoft 隐私声明](https://privacy.microsoft.com/zh-cn/privacystatement/)。