

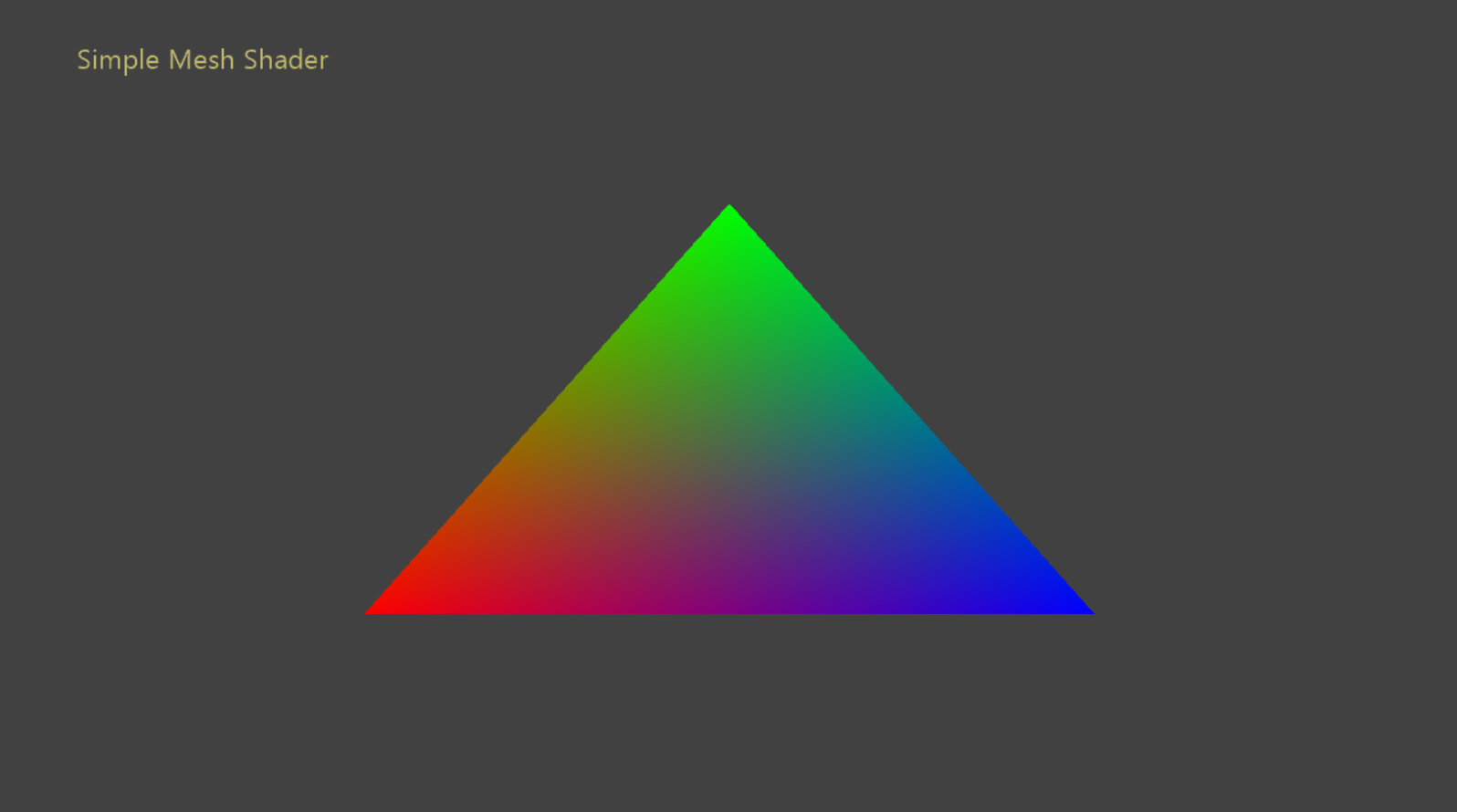
简单的网格着色器示例

此示例与 Microsoft 游戏开发工具包预览版（2019 年 11 月）和 Windows 10“20H1”Insider for PC 兼容

# 说明

本示例与 SimpleTriangle 这个成熟的示例相辅相成，只不过这里使用的是网格着色器。它是为了简单演示初始化网格着色器管道并在电脑和 Scarlett GXDK 上运行的所有必备组件。

注意：Xbox One 主机系列产品不支持任何网格着色器，因此没有适用于该平台的生成配置。



# 构建示例

如果使用 Project Scarlett，请将活动解决方案平台设置为 Gaming.Xbox.Scarlett.x64。

如果使用电脑且结合使用相应的硬件和 Windows 10 版本，请将活动解决方案平台设置为 Gaming.Desktop.x64。

本示例不支持 Xbox One。

有关详细信息，请参阅 GDK 文档中的“运行示例”。

# 使用示例

|  |  |
| --- | --- |
| 操作 | 游戏手柄 |
| 退出 | “视图”按钮 |

# 实现说明

本示例演示以下几个步骤：

1. 初始化 ID3D12Device 并请求 API 对象，以便使用 DirectX12 进行呈现。
2. 使用 ID3D12Device::CheckFeatureSupport() 函数检查网格着色器功能支持情况。
3. 使用 ID3D12Device2::CreatePipelineState() 函数创建网格着色器管道。
4. 将根签名、管道状态和资源绑定到命令列表。使用 ID3D12GraphicsCommandList6::DispatchMesh() 函数和必备参数调度网格绑定管道。

# 更新历史记录

2019 年 10 月 31 日 - 创建示例。

# 隐私声明

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

有关 Microsoft 的一般隐私策略的详细信息，请参阅《[Microsoft 隐私声明](https://privacy.microsoft.com/en-us/privacystatement/)》。