

SimpleCompute 示例

此示例与 Microsoft 游戏开发工具包预览版（2019 年 11 月）兼容

# 说明



SimpleCompute 演示了如何将 DirectCompute™（即 Direct3D 计算着色器）与 DirectX 12 结合使用。此示例演示如何将计算工作提交到两个图形命令列表，以及如何使用 D3D12\_COMMAND\_LIST\_TYPE\_COMPUTE 接口提交异步计算着色器工作负载。它通过使用计算着色器计算曼德布洛特集合来更新纹理。

# 构建示例

如果使用 Xbox One 开发工具包，请将活动解决方案平台设置为 Gaming.Xbox.XboxOne.x64。

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

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 |
| 切换异步计算 | A 按钮 |
| 将视区重置为默认值 | Y 按钮 |
| 平移视区 | 左摇杆 |
| 缩放视区 | 右摇杆 |
| 增加缩放速度 | 右扳机键 |
| 退出 | “视图”按钮 |
| 菜单 | 显示/隐藏帮助 |

# 实现说明

本示例的主要目的是让读者熟悉如何创建和使用简单的计算着色器。

* CreateDeviceDependentResources：用于加载已编译的计算着色器，并创建各种 Direct3D 呈现资源。着色器是由 Visual Studio 编译的。
* Render：如果示例未使用异步计算，则在需要分派结果的绘制调用之前分派计算着色器。这将更新每个帧的纹理。
* AsyncComputeProc：如果示例使用异步计算，那么一旦告知它开始处理，就会立即从此线程分派计算着色器。在执行从属绘制调用之前，呈现器将等待，直到告知其异步任务已经完成。

# 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/)。