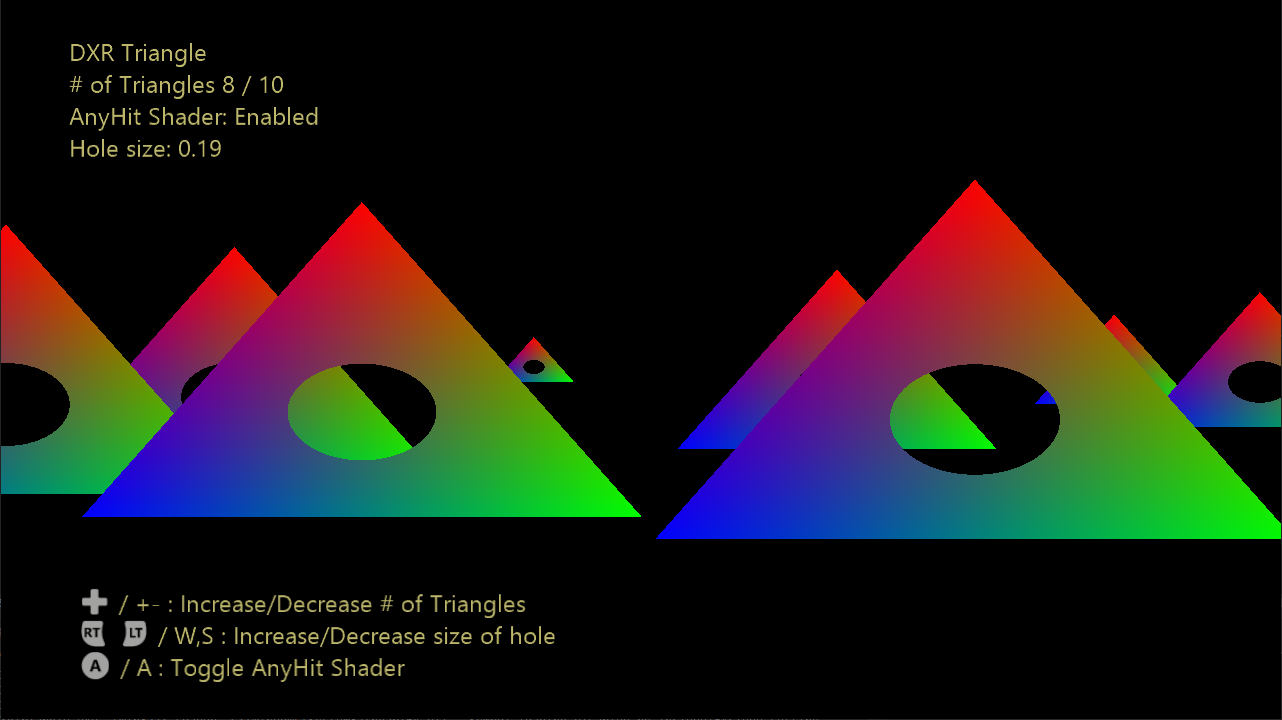


DXR 三角形示例

\* 此示例与 2019 年 11 月 Scarlett GXDK 兼容。

# 说明

此示例演示 DXR API 的基本用法。它将创建光线追踪管道状态对象 (RTPSO)、着色器表格以及顶层/底层加速结构。此外，它还演示了“光线生成”、“最接近的命中”、“任意命中”和“未命中”着色器的用法，以及底层加速结构的实例化。



# 构建示例

此示例仅支持 Project Scarlett。

有关详细信息，请参阅 GDK 文档中的“运行示例”。

# 使用示例

通过采用经典的“SimpleTriangle”并将其放入光线跟踪纪元，数量可配置的三角形即可在屏幕上滑动。通过按键盘/游戏手柄上的 A，可打开使用“任意命中”着色器或将其关闭 – 在每个三角形上打一个孔以显示其后面的三角形。

# 控件

|  |  |  |
| --- | --- | --- |
| 操作 | 游戏手柄 | 键盘 |
| 切换“任意命中”着色器 | A | A |
| 增加三角形的数量 | 向上方向键 | + |
| 减少三角形的数量 | 向下方向键 | - |
| 增大孔的大小 | 右扳机键 | W |
| 减小孔的大小 | 左扳机键 | S |
| 退出 | “视图”按钮 | ESC 键 |

# 实现说明

示例首先使用“CreateStateObject”API 创建光线跟踪管道状态对象 (RTPSO)。RTPSO 包含一个命中组，该组由一个“最接近的命中”着色器和一个“任意命中”着色器组成。还指定了其他配置属性，例如递归级别和光线有效负载大小。

底层加速结构 (BLAS) 由一个三角形组成，并在顶层加速结构 (TLAS) 中实例化 1 到 10 次，具体取决于当前所选的三角形数量。可在场景中为每个实例提供自己的位置和大小。默认情况下，每个帧都会生成 TLAS 和 BLAS，以便它们出现在你要采用的 PIX 捕获中。但是，在运输标题中，通常只需重新生成每个帧的 TLAS，并长时间（可能永远）重复使用 BLAS。

“光线生成”着色器可设置一个简单的正交投影（沿 Z 轴方向看），并为在 DispatchRays 调用中启动的每个线程发射一束光线。

将为每个未命中每个三角形的光线调用“未命中”着色器，并将黑色写到 UAV（这是指替代在 DispatchRays 调用前将 UAV 预先清除为黑色的操作）。

“最接近的命中”着色器计算重心坐标，并将其用作像素的最终颜色。

“任意命中”着色器计算在重心坐标空间中到三角形中心 (0.333, 0.333, 0.333) 的简单距离，并且如果该距离小于某个公差，则要求忽略命中。然后，允许光线穿过三角形，并有可能沿 z 轴进一步命中其他三角形。

# 更新历史记录

2019/11/1 - 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 的一般隐私政策的详细信息，请参阅《[Microsoft 隐私声明](https://privacy.microsoft.com/en-us/privacystatement/)》。