

简易实例样本

*此示例与Windows 10 Anniversary Update SDK兼容（14393）*

# 

# 描述

此示例演示如何使用Direct3D 11 API进行实例化。

# 使用样本



|  |  |  |
| --- | --- | --- |
| 操作 | 游戏手柄 | 键盘 |
| 旋转相机 | 左控制棒 | WASD/鼠标左键和鼠标左键 |
| 更改实例计数 | LB / RB | Q/E |
| 居中 | n/a | 主页 |
| 重置模拟 | A | 空间 |
| 退出 | 查看按键 | 退出 |

# 实施说明

渲染实例化几何体时需要考虑四个方面：

1. **几何数据**

在此样本的情况下，这包括顶点和索引缓冲区，其中包含描述立方体表面的顶点和索引。还包括将此立方体渲染至世界中所需的着色器和状态。所有这些组件的设置和操作方式与非实例化几何体的设置和操作方式相同。（请参阅 SimpleInstancing12.cpp 中的*CreateDeviceDependentResources*）

1. **实例数据**

对于标准 D3D11 实例化渲染，每个实例数据通过一个或多个顶点缓冲区提供。这些顶点缓冲区的创建方式与任何其他顶点缓冲区的创建方式相同。此样本使用了两个顶点缓冲区。一个为静态（使用 *D3D11\_USAGE\_IMMUTABLE* 标记创建），包含每个实例的颜色数据（在样本使用期内保持不变）。另一个为动态（使用 *D3D11\_USAGE\_DYNAMIC* 和 *D3D11\_CPU\_ACCESS\_WRITE* 标记创建），包含每个实例的位置和方向信息。

1. **实例化布局**

要渲染几何实例，D3D需要有关如何解释所提供的顶点数据的信息。这是使用数组完成的 *D3D11\_INPUT\_ELEMENT\_DESC* 结构，与标准渲染完成的方式非常相似。但是，此结构中添加了额外的元素。几何数据标记为 *D3D11\_INPUT\_PER\_VERTEX\_DATA* 的价值 *InputSlotClass* 像往常一样，元素，但每个实例数据使用 *D3D11\_INPUT\_PER\_INSTANCE\_DATA* 值。该 *InputSlot* element也用于表示从中提取每个数据的顶点流。

顶点着色器使用的顶点结构被定义为将几何和每个实例数据都集中在一起（这反映了在 *D3D11\_INPUT\_ELEMENT\_DESC* 数组中进行描述的布局）。

***注意：****此样本使用*D3D11\_APPEND\_ALIGNED\_ELEMENT *元素的* AlignedByteOffset *常量在输入布局中自动正确对齐数据。这仅在所讨论的顶点缓冲区的结构包含正确对齐的数据时才有效。如果要跳过（或忽略）顶点数据中的元素，则需要精确对齐偏移。*

1. **渲染**

一旦先前的点被钉死，渲染实例化数据就很简单。*ID3D11DeviceContext::IASetVertexBuffers*API 用于设置用作输入的顶点缓冲区（在本例中为输入缓冲区），而 *ID3D11DeviceContext::DrawIndexedInstanced* API 则用于渲染。渲染设置的其余部分与标准非实例渲染的执行方式相同。

## 系统需求

此示例与Windows 8或更高版本兼容。

若已安装 DirectX 11.1 Runtime（[KB 2670838](https://support.microsoft.com/zh-cn/help/2670838/platform-update-for-windows-7-sp1-and-windows-server-2008-r2-sp1)），此样本也可在 Windows 7 Service Pack 1上运行。看到 [这篇博文](https://blogs.msdn.microsoft.com/chuckw/2013/02/26/directx-11-1-and-windows-7-update/) 了解更多信息。

# 更新历史记录

初次发布于 2018 年 4 月