

ExecuteIndirect Sample

*This sample is compatible with the Microsoft Game Development Kit (June 2020)*

# Description

This sample demonstrates usage of D3D12’s ExecuteIndirect API, for asynchronously building rendering commands.

The sample creates a large number of mesh instances, distributed randomly in front of the camera. In Direct mode, each mesh instance is drawn using a separate draw call. In Indirect mode, the entire “scene” is drawn using a single ExecuteIndirect call.

The sample optionally performs frustum culling in either mode. In Direct mode, instances are culled one at a time on the CPU. In Indirect mode, instances are culled in parallel using GPU compute. The ExecuteIndirect call only ever sees those instances which pass culling. The other instances do not exist in the indirect command buffer.

# Building the sample

If using an Xbox One devkit, set the active solution platform to Gaming.Xbox.XboxOne.x64.

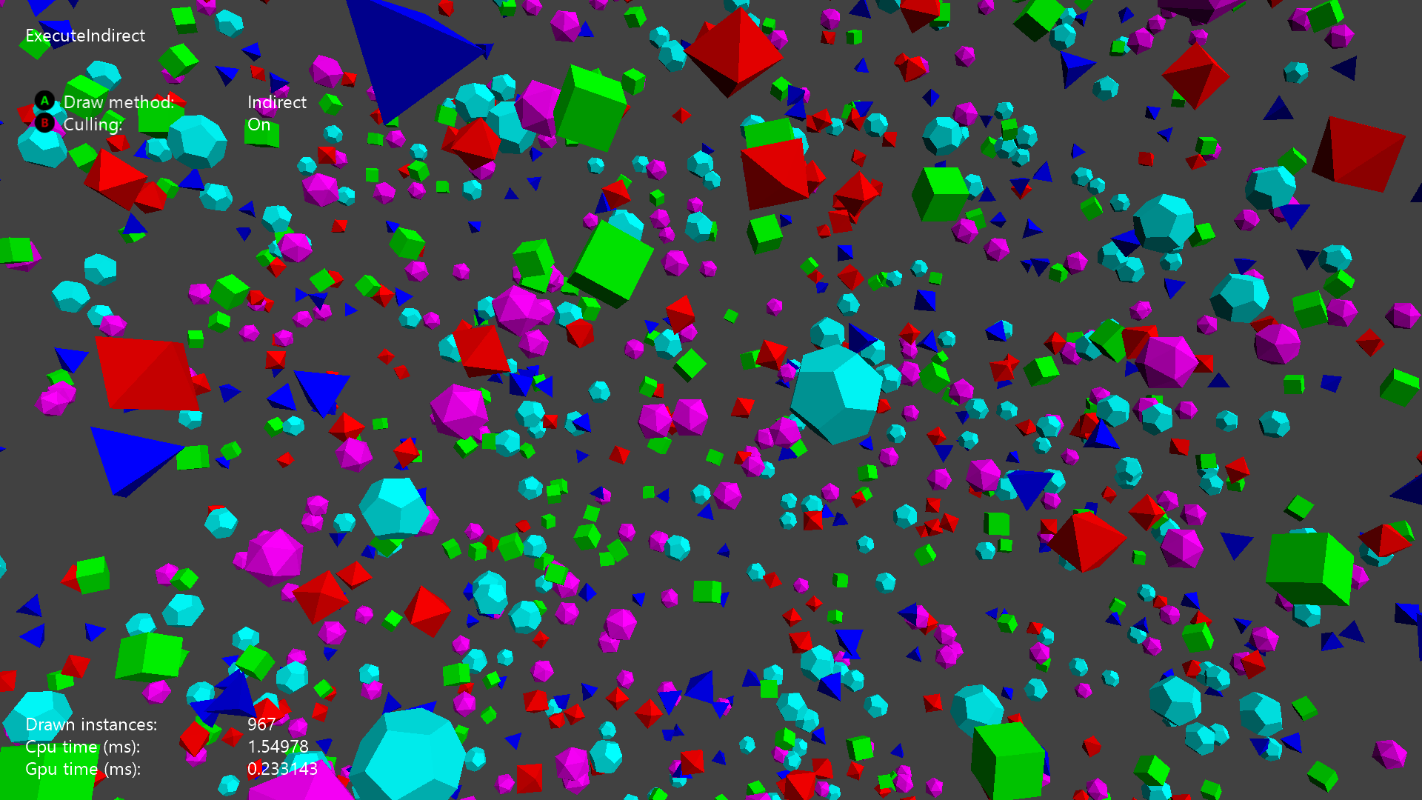
If using Project Scarlett, set the active solution platform to Gaming.Xbox.Scarlett.x64.

*For more information, see* Running samples*, in the GDK documentation.*

# Using the sample

This sample uses the following controls.

|  |  |
| --- | --- |
| Action | Gamepad |
| Toggle Direct/Indirect draws | A Button |
| Toggle culling on/off | B Button |
| Exit | View Button |

**

# Known issues

[none]

# Update history

Initial release for XDK August 2015

Updated for GDK and Scarlett April 2020

# Privacy Statement

When compiling and running a sample, the file name of the sample executable will be sent to Microsoft to help track sample usage. To opt-out of this data collection, you can remove the block of code in Main.cpp labeled “Sample Usage Telemetry”.

For more information about Microsoft’s privacy policies in general, see the [Microsoft Privacy Statement](https://privacy.microsoft.com/en-us/privacystatement/).