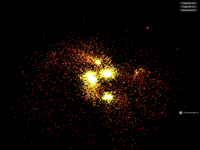
NBodyGravityCS11

<https://github.com/walbourn/directx-sdk-samples>

This is the DirectX SDK's Direct3D 11 sample updated to use Visual Studio 2012 and the Windows SDK 8.0 without any dependencies on legacy DirectX SDK content. This sample is a Win32 desktop DirectX 11.0 application for Windows 10, Windows 8.1, Windows 8, Windows 7, and Windows Vista Service Pack 2 with the DirectX 11.0 runtime.

**This is based on the legacy DirectX SDK (June 2010) Win32 desktop sample. This is not intended for use with Windows Store apps, Windows RT, or universal Windows apps.**

# Description



This is one of the Compute Shader demos shown at GDC09. This sample shows an N-Body particle system implemented using Compute Shader 4.0. Because of benefits brought by Compute Shader, it achieves significantly higher frames per second than the previous NBodyGravity sample under the Direct3D 10 section, which was implemented using Direct3D 10 Pixel Shader and Geometry Shader.

## How the Sample Works

The sample uses a structured buffer to store the position and velocity of each particle in the system. Each thread of the Compute Shader updates one of the particles in each frame. Compute Shader 4.0 can read and write to the structured buffer directly, which makes it easy to recirculate data between passes in this sample.

In order to update each single particle, the position of every other particles need to be accessed. This is very I/O intensive. To increase I/O efficiency, we use Compute Shader shared memory to cache a tile of particles, and then update against these cached particles instead of fetching them directly from video memory. Refer to NBodyGravityCS11.hlsl to see how this is done.

# Dependencies

DXUT-based samples typically make use of runtime HLSL compilation. Build-time compilation is recommended for all production Direct3D applications, but for experimentation and samples development runtime HLSL compilation is preferred. Therefore, the D3DCompile\*.DLL must be available in the search path when these programs are executed.

* When using the Windows 8.x SDK and targeting Windows Vista or later, you can include the D3DCompile\_46 or D3DCompile\_47 DLL side-by-side with your application copying the file from the REDIST folder.

%ProgramFiles(x86)%\Windows kits\8.0\Redist\D3D\arm, x86 or x64

%ProgramFiles(x86)%\Windows kits\8.1\Redist\D3D\arm, x86 or x64

%ProgramFiles(x86)%\Windows kits\10\Redist\D3D\arm, x86 or x64

# More Information

[Where is the DirectX SDK (2015 Edition)?](https://walbourn.github.io/where-is-the-directx-sdk-2015-edition/)

[DXUT for Win32 Desktop Update](https://walbourn.github.io/dxut-for-win32-desktop-update/)

[Games for Windows and DirectX SDK blog](https://walbourn.github.io/)