BasicHLSL11 Sample

Collapse All

Illustrates how to load a mesh, create vertex and pixel shaders from files, and then use the shaders to render the mesh.

Path

Source	*SDK root*\Samples\C++\Direct3D11\BasicHLSL11
Executable	*SDK root*\Samples\C++\Direct3D11\Bin*x86 or x64*\BasicHLSL11.exe

Sample Overview

This sample loads a mesh, creates vertex and pixel shaders from files, and then uses the shaders to render the mesh. The effect that is used is a simple vertex shader that animates the vertices based on time. It uses DXUT to switch between Direct3D 9 and Direct3D 11 codepaths. Only the Direct3D 11 codepath is described here. The Direct3D 9 codepath is similar and can be found in the Direct3D 9 documentation.

How the Sample Works

First the sample checks for the currently supported feature level and compiles and creates vertex and pixel shaders from a file.

Next, the sample creates an input layout that matches the input layout of the mesh that will be loaded. This will be the same for all meshes loaded through CDXUTSDKMesh.

Next, the sample loads the geometry using the CDXUTSDKMesh class.

In OnD3D11FrameRender, the sample updates dynamic state contained in constant buffers such as the World*View*Projection matrix, then sets both static and dynamic state such as the current input layout and samplers to be used for rendering using the immediate context. The mesh is rendered using DrawIndexed calls called in a loop over the mesh subsets.