BasicHLSL10 Sample

Collapse All

Path

Source	*SDK root*\Samples\C++\Direct3D10\BasicHLSL10
Executable	*SDK root*\Samples\C++\Direct3D10\Bin*x86 or x64*\BasicHLSL10.exe

Sample Overview

This sample simply loads a mesh, creates an effect from a file, and then uses the effect 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 10 codepaths. Only the Direct3D 10 codepath will be described here. The Direct3D 9 codepath is similar and can be found in the Direct3D 9 documentation.

How the Sample Works

First the sample calls D3DX10CreateEffectFromFile to create an effect from the supplied FX file.

Secondly, the sample loads the effect techniques and effect variables from the FX file and stores them for later use.

Third, 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 CDXUTMesh10. After this the sample loads the geometry using the CDXUTMesh10 class. Internally this calls OptimizeInplace() to optimize the mesh for the vertex cache, then loads the textures using D3DX10CreateTextureFromFile.

In OnD3D10FrameRender, sets variables used by the technique such as the World*View*Projection matrix and the time variable with various ID3DX10Effect::Set* calls. Then the sample sets the current input layout to be used by the device using ID3D10Device::IASetInputLayout. Next, the number of passes in the selected rendering technique are determined by calling ID3D10Technique::GetDesc. For each pass contained within the technique, the pass is applied to the device with ID3D10EffectPass::Apply and the mesh is rendered.