Desenvolvedores DirectX - DirectX no Windows Phone

Este artigo tem como objetivo dar informações básicas sobre C++ nativo e desenvolvimento de jogos com DirectX no Windows Phone 8. Ele irá listar as grandes diferenças entre o DirectX no Windows 8 versus Windows Phone 8 e também irá dar sugestões sobre como fazer utilizar certas funcionalidades de uma forma alternativa por não estarem presentes em ambas plataformas.

Introdução

Com Windows Phone 8, desenvolvedores de jogos serão capazes de usar as APIs do C++ nativo e DirectX para criar seus jogos. Porém, a Microsoft fez algumas modificações em alguns aspectos do DirectX no Windows Phones em relação à sua versão Desktop para melhor otimizá-lo em ambientes móveis (mobile). Focaremos nas diferenças gerais.

Pré-requisitos

Este artigo é feito para desenvolvedores com algum conhecimento anterior sobre DirectX. Conhecimento básico de desenvolvimento de jogos também é recomendado.

DirectX no Windows Phone 8

Direct3D

DirectX no Windows Phone 8 usa a versão 11 do Direct3D com feature level acertado em 9_3 (Ao invés de 11_1 como o Windows 8 usa). Isso significa que funcionalidades presentes em versões mais novas que a 9_3 não podem ser usadas no Windows Phone 8, no momento. Fizemos uma tabela de commparação das diferenças entre as feature levels 11_1 e 9_3 usando a página MSDN sobre os feature levels do Direct3D como referência.

Característica	Feature level 11_1	Feature level 9_3
Shader Model	5.0	2.0
Geometry Shader	Sim	Não
Estágio Stream-Output	Sim	Não
DirectCompute/ComputerShader	Sim	N/A
Hull e Domain Shaders	Sim	Não
Texture Resource Arrays	Sim	Não
Cubemap Resource Arrays	Sim	Não
Compressão BC4/BC5	Sim	Não
Compressão BC6H/BC7	Sim	Não
Alpha-to-coverage	Sim	Não
Formato 10-bit XR High Color	Sim	N/A
Operações Lógicas(Output Merger)	Sim	Não
Rasterização Target-independent	Sim	Não
Render Target Múltiplo (MRT) with ForcedSampleCount 1	Sim	Não
UAV slots	64	N/A
Dimensão máxima de textura	16384	4096
Dinmensão máxima de Cubemap	16384	4096
Volume Extent Máximo	2048	256
Texture Repeat Máxima	16384	8192
Máxima Quantidade de Primitivas	2^32-1	1048575
Input Slots Máximos	32	16
Render Targets Simultâneos	8	4

Destes podemos ver que o feature level 9_3 tem muito menos características que o 11_1, mas isso também garante que as

(C) Copyright Nokia 2014. All rights reserved.

aplicações Direct3D rodarão mais fluidamente em hardware móvel.

Algumas APIs Direct3D que fazem parte do ambiente desktop **não podem ser usadas no WP8**. Essas APIs são:

- ID3D11CommandList
- ID3D11ComputeShader
- ID3D11DomainShader
- ID3D11GeometryShader
- ID3D11HullShader
- ID3D11UnorderedAccessView

Além destas, características específicas do Windows Phone 8 foram feitas na Swap Chain. Na prática, isso significa que algumas das características que estão disponíveis no Direct3D para desktop não podem ser usadas ou estão limitadas no WP8. Aqui está uma lista das modificações específicas do WP8:

- Multisampling não é permitido.
- Apenas single buffer pode ser usado.
- Scaling deve usar a opção de stretching.
- SwapEffect deverá ser setado para discard.

Compilação de Shader em Tempo de Execução

Runtime Shader Compilation no Direct3D do Windows Phone 8 está desabilitado e apenas compilação offline é possível devido a limitações no hardware móvel.

Outras Características do DirectX

Além das do Direct3D, outras características do DirectX no Windows Phone 8 foram desabilitadas:

- **Direct2D:** O que significa que o desenvolvedor terá que usar Direct3D mesmo que queira fazer um jogo 2D. O DirectX Toolkit é uma boa alternativa pra se usar no WP8. Ele contém a classe SpriteBatch provinda do XNA que é boa para render 2D.
- **DirectWrite:** Uma alternativa para escrita de texto, sugerimos a classe SpriteFont do DirectX Toolkit.
- **XInput:** Esta API é usada para obter input do Xbox Controller, então ser removida não é grande coisa se o desenvolvedor estiver fazendo um jogo apenas para mobile.
- **Windows Imaging Component (WIC):** WIC ser removida significa que carregar texturas de imagens como JPGs e PNGs é mais difícil e usar textura DDS torna-se recomendado. O DirectX Toolkit tem também uma função para fazer o carregamento de textura DDS mais fácil.
- **DirectAudio:** Não está disponível no WP8. O desenvolvedor pode usar as APIs XAudio2 ou WASAPI no lugar.

Requisitos de Ambiente de Desenvolvimento

Quando desenvolver jogos para WP8, todos os requisitos para um aplicativo comum WP8 se aplicam. O desenvolvedor irá precisar de um PC com Windows 8 e Visual Studio 2012. O SDK do Windows Phone 8 vem com o Visual Studio 2012 Express.

Para que o desenvolvedor seja capaz de usar o emulador para abrir aplicativos, o PC de desenvolvimento deverá reunir certos requisitos de hardware. Uma placa gráfica com suporte DirectX11 é necessária. É necessário também que a máquina tenha uma CPU capaz de usar HyperV e SLAT (series i3, i5 e i7 da intel) e ao menos 4GB de RAM. Isso faz com que o desenvolvedor possa escolher tanto entre desenvolver com o emulador quanto abrir a aplicação desenvolvida no dispositivo desbloqueado do desenvolvedor, ou ambos. Uma conta Windows Phone Dev Center é necessária para desbloquear o dispositivo para desenvolvimento.

Se o emulador não puder rodar na máquina do desenvolvedir, a aplicação poderá ser também aberta e debugada num dispositivo com Windows Phone 8 conectado ao PC de desenvolvimento via USB.

Forma Mais Fácil de Começar: o Template do Visual Studio

O Visual Studio ajuda o desenvolvedor a começar um aplicativo Direct3D para WP8 com o **template de projeto "Windows Phone Direct3D App"**. Este template possui algumas das funcionalidades de aplicação como um device D3D e input de toque já implementadas. O template também possui um objeto de renderização D3D implementado para um cubo 3D que gira. Isso livra o desenvolvedor de começar tudo por si próprio e facilita o início do desenvolvimento.

Ler Mais

(C) Copyright Nokia 2014. All rights reserved.

Para melhor adentrar no desenvolvimento C++/DirectX em Windows Phone você pode verificar os artigos Windows Phone Native C++ and DirectX - First Direct3D App, setting up Touch and Sensors e DirectX on Windows Phone: 2D Game Example using DirectX Toolkit.

(C) Copyright Nokia 2014. All rights reserved.