NVIDIA®

Getting Started with Cg

Release 1.0

December 2002

What This Presentation Contains

- What's Included in the Cg Distribution
- How to Learn Cg Quickly
- A Brief Introduction to Cg
- Using Cg in Your Applications
- Using the Cg Runtime
- Transitioning to the New Cg Runtime
- Frequently Asked Questions about Cg

What's Included in the Cg Distribution

- Cg Compiler
- Cg Runtime
- Cg User's Manual
- Cg Browser
- Sample Cg Shaders
- Runtime Transition Document and Utility

Where to Find Information About Cg

- http://developer.nvidia.com/Cg
  - Whitepapers
  - Presentations
  - Cg User's Manual
  - Cg Language Specification
  - Cg Toolkit Downloads
  - Bug Reporting
- www.CgShaders.org
  - Forums
  - Shader Repository (Freeware)

How to Learn Cg Quickly

- **Read this presentation**
  - Understand the high-level picture
- **Download the Cg Toolkit**
  - Cg_Toolkit.zip contains the complete toolkit
  - If bandwidth is an issue, just get Cg_Compiler.exe
- **Check out the Cg User's Manual**
  - Read the Introduction to the Cg Language chapter
    - This explains the language syntax and constructs
  - Try A Brief Tutorial
    - Gives you a chance to try Cg first-hand
    - Everything is already set up for you

How to Learn Cg Quickly (Cont'd)

- **Try the Cg Tutorials**
  - Available at http://developer.nvidia.com/view.asp?IO=cg_tutorials
- **Learn about the Cg Runtime**
  - Read **Using the Cg Runtime** (in the Cg User's Manual)
- **Don't Hesitate to Ask Questions**
  - Try the forums on www.CgShaders.org to talk to others who are learning or have learned Cg
  - Send mail to cgSupport@nvidia.com

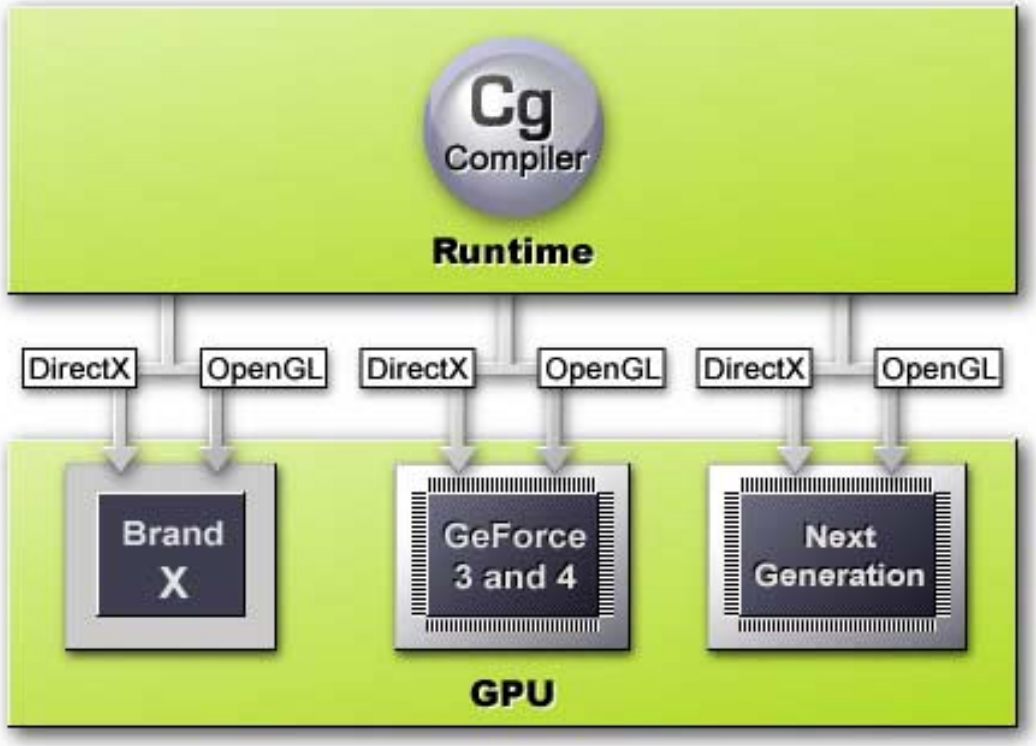
A Brief Introduction to Cg

- Cg is "C for Graphics," a high-level, cross-platform language for graphics programming
- Cg replaces tedious assembly coding with a C-like language and a compiler that generates assembly for you
- Cg is a cross-API and cross-platform language:
  - It works with both OpenGL and DirectX
  - It runs on Windows and Linux (more in the works)
  - It supports hardware from NVIDIA, ATI, Matrox, and any other programmable hardware that supports OpenGL or DirectX
- The Cg Runtime simplifies parameter passing from your application to the vertex and fragment programs

NVIDIA.

A Brief Introduction to Cg (Cont'd)

- Ensures increasing optimization through forward compatibility
- Works with ALL programmable GPUs supporting DirectX 8/9 or OpenGL 1.4
- Works with future versions of DirectX and OpenGL



NVIDIA.

What Does Cg Look Like?

Assembly

```
...
RSQR  R0.x, R0.x;
MULR  R0.xyz, R0.xxxx, R4.xyzz;
MOVR  R5.xyz, -R0.xyzz;
MOVR  R3.xyz, -R3.xyzz;
DP3R  R3.x, R0.xyzz, R3.xyzz;
SLTR  R4.x, R3.x, {0.000000}.x;
ADDR  R3.x, {1.000000}.x, -R4.x;
MULR  R3.xyz, R3.xxxx, R5.xyzz;
MULR  R0.xyz, R0.xyzz, R4.xxxx;
ADDR  R0.xyz, R0.xyzz, R3.xyzz;
DP3R  R1.x, R0.xyzz, R1.xyzz;
MAXR  R1.x, {0.000000}.x, R1.x;
LG2R  R1.x, R1.x;
MULR  R1.x, {10.000000}.x, R1.x;
EX2R  R1.x, R1.x;
MOVR  R1.xyz, R1.xxxx;
MULR  R1.xyz, {0.900000, 0.800000,
     1.000000}.xyzz, R1.xyzz;
DP3R  R0.x, R0.xyzz, R2.xyzz;
MAXR  R0.x, {0.000000}.x, R0.x;
MOVR  R0.xyz, R0.xxxx;
ADDR  R0.xyz, {0.100000, 0.100000,
     0.100000}.xyzz, R0.xyzz;
MULR  R0.xyz, {1.000000, 0.800000,
     0.800000}.xyzz, R0.xyzz;
ADDR  R1.xyz, R0.xyzz, R1.xyzz;

...
```

Cg

```
…
float3 cSpec = pow(max(0, dot(Nf, H)), phongExp).xxx;
float3 cPlastic = Cd * (cAmbi + cDiff) + Cs * cSpec;
…
```

Shading Language (RenderMan™)

```
…
color cSpec = phong(Nf,V,phongExp);
Ci = Oi * (FinalColor = DiffuseColor *
    (AmbientLight + DiffuseLight)) + SpecularColor
    * cSpec;
…
```

NVIDIA.

Advantages of Cg

- Cg greatly simplifies developing OpenGL and DirectX applications with programmable shading
  - Cg is easier than assembly
  - Managing parameters is simplified with Cg
  - Adds abstraction from hardware and graphics API
- Cg is flexible—you can use as little or as much of it as you want
  - Cg language only
  - API-independent libraries
  - API-dependent libraries

Choose from Many Levels of Support

- Cg [corresponds to DirectX 9's HLSL language]
  - The shading language itself (cross-platform, multi-API)
  - Compile to assembly and use it directly
- Cg Runtime
  - Parameter management
  - Optional specialized runtimes for OpenGL and Direct3D
- CgFX [corresponds to DirectX 9's .fx format]
  - Shading code + render state encapsulation
  - OpenGL and DirectX
- CgFX Runtime [corresponds to DirectX 9's runtime]
  - API to access CgFX files
- Cg Plugins
  - Use Cg in 3ds max, Maya and Softimage|XSI

NVIDIA.

Using Cg in Your Applications

- Simply replace your vertex and fragment shaders with equivalent Cg shaders
- Two options:
  - Use Cg for development, and compile to assembly code for final product
    - Cg compiler output is assembly code
    - Use your preferred graphics API to load and run it
  - Use the Cg Runtime
    - Compiles Cg code and passes it to the GPU for you

Using the Cg Runtime

- The Cg Runtime helps you by:
  - Loading programs
  - Compiling programs
  - Managing program parameters
  - Managing texture units
  - Making your programs future-proof (you can now compile them at run-time instead of compile-time)
- Read Using the Cg Runtime (in the Cg User's Manual)

Transitioning to the New Cg Runtime

- The Cg Runtime has been updated from Public Beta 2 to Release 1.0
  - Changes based on developer feedback
  - More intuitive
  - Cleaner interface
  - New features
- We've created a Runtime Transition Document to help you convert applications to use the new runtime
  - The document is in Start… NVSDK… Cg Toolkit
  - An automatic search-and-replace utility is also included
  - Instructions are in the Runtime Transition Document

Frequently Asked Questions About Cg

- To see some of the most frequent questions and answers about Cg, please visit:
  - http://www.cgshaders.org/articles/interview_davidkirk02.php
  - http://www.cgshaders.org/articles/interview_nvidia-jul2002.php
  - http://www.nvidia.com/view.asp?IO=cg_faq