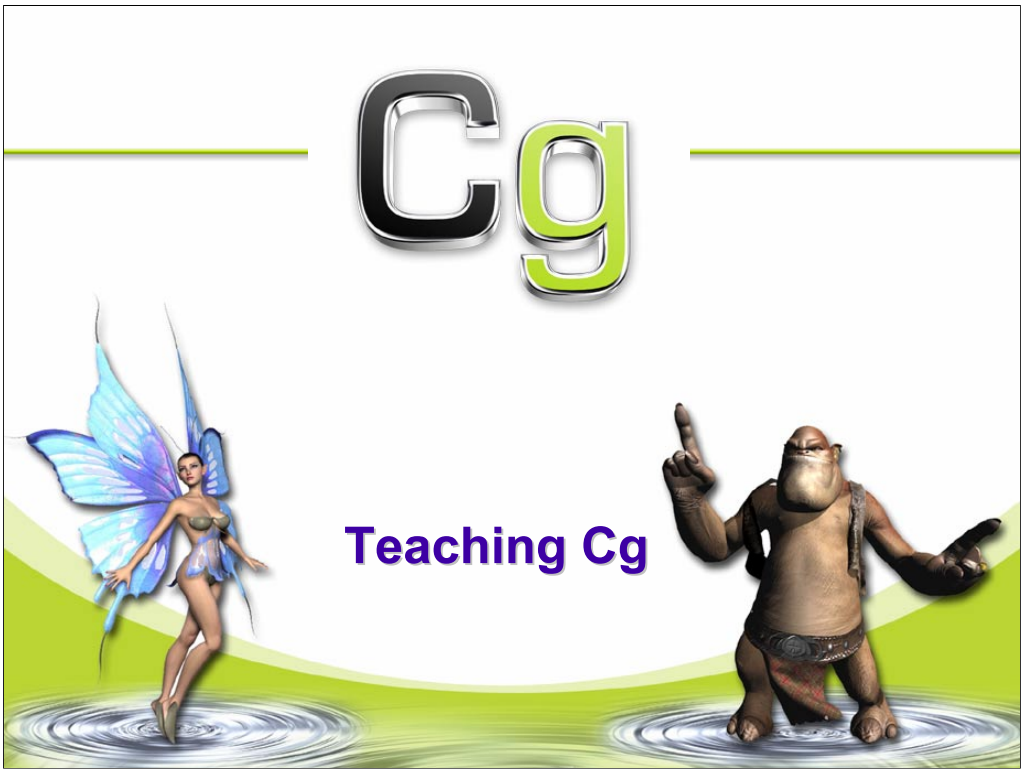


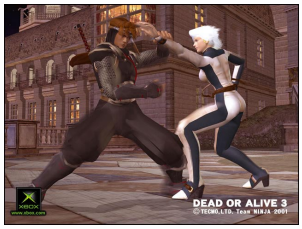
This presentation introduces Cg (“C for graphics”) and explains why it would be useful when teaching a computer graphics course.

Real-Time Graphics Has Come a Long Way

Virtua Fighter
(SEGA Corporation)

NV1
50K triangles/sec
1M pixel ops/sec

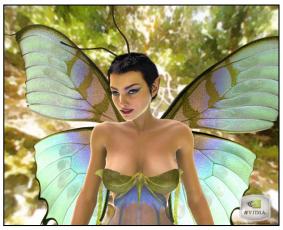
1995



Dead or Alive 3
(Tecmo Corporation)

Xbox (NV2A)
100M triangles/sec
1G pixel ops/sec

2001



Dawn
(NVIDIA Corporation)

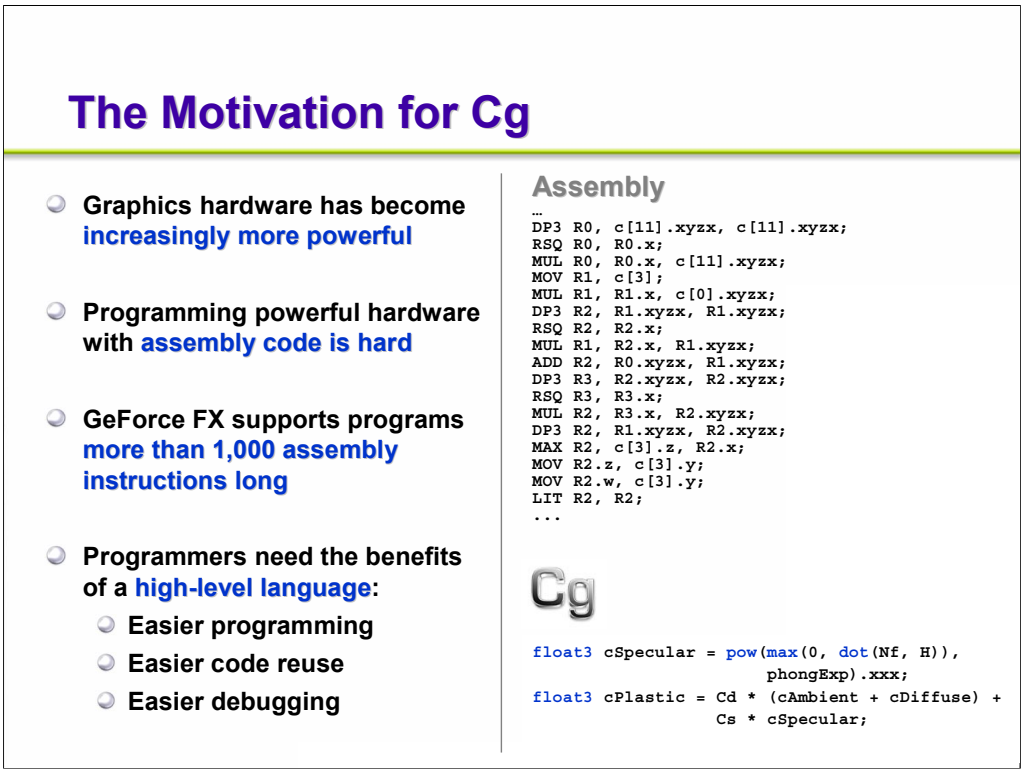
GeForce FX (NV30)
200M triangles/sec
2G pixel ops/sec

2003

NVIDIA.

Real-time graphics has progressed dramatically over the past several years. Today's consumer-level graphics boards can deliver many times the performance of million-dollar graphics workstations of a decade ago.

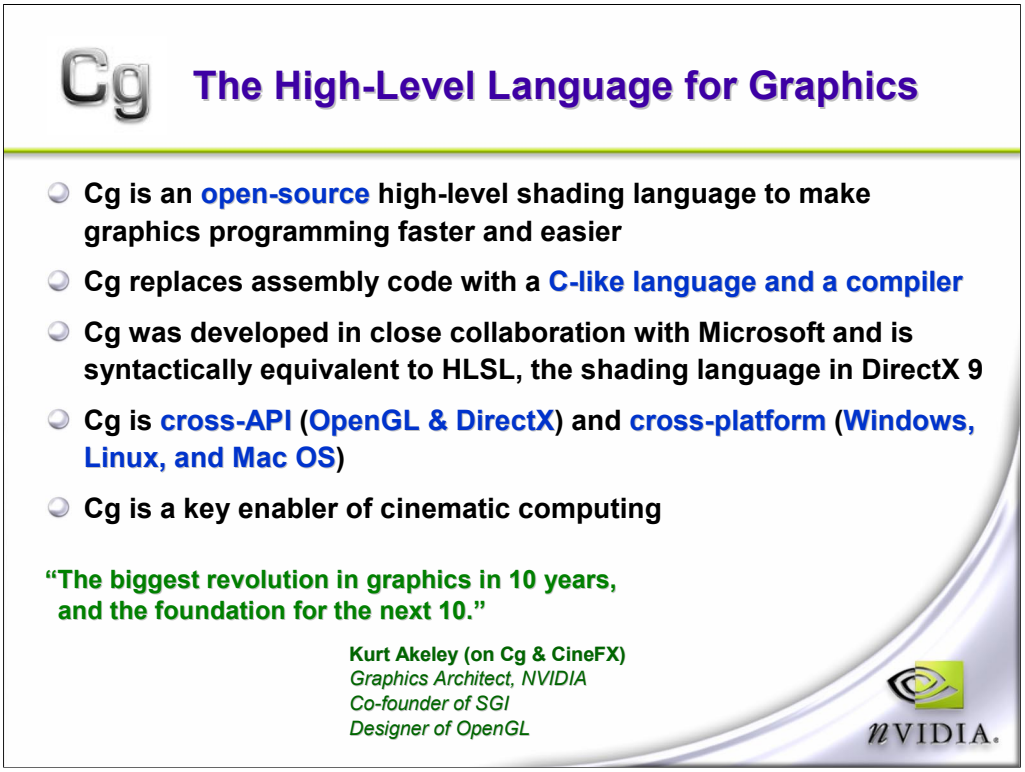
Modern graphics processing units (GPUs) are also massively programmable, allowing the creation of a truly infinite number of new effects and algorithms.



Before Cg, the only way to access the programmability of recent GPUs was to write assembly code and perform complex operations to correctly configure the graphics pipeline. With recent GPUs such as the GeForce FX, programs can be thousands of instructions long, making assembly programming undesirable.

With Cg, these same algorithms can be written in a high-level language, with the Cg compiler and runtime handling the task of creating assembly code suitable for the GPU.

Cg offers the same advantages over GPU assembly code that C offers over CPU assembly code. Algorithms written with high-level languages are easier to express, reuse, understand, and debug.



Cg was developed in conjunction with Microsoft, and uses the same syntax and constructs as DirectX 9's High Level Shading Language (HLSL).

Cg is also fully compatible with today's OpenGL 1.4 environment – and supports ARB standards, like ARB_vertex_program and ARB_fragment_program.

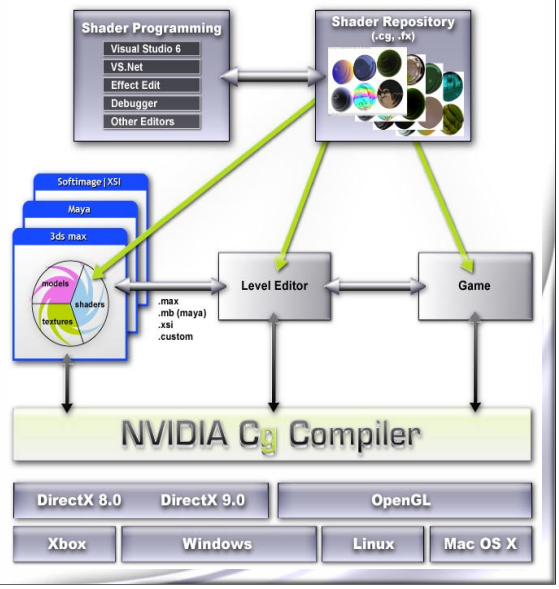
Cg works with both OpenGL and DirectX, with a variety of major platforms (Windows, Linux, and Mac OS), and with hardware from different graphics vendors, as long as the hardware supports OpenGL or DirectX.

NVIDIA also has released an open-source Cg compiler, to rapidly enable other vendors to implement Cg for their platforms.

Kurt Akeley, one of the founders of SGI and creator of OpenGL, said that the combination of the CineFX architecture and Cg is the "biggest revolution in graphics in 10 years, and the foundation for the next 10." He even said that not since the Reality Engine and texture mapping has there been such an advancement in graphics technology.

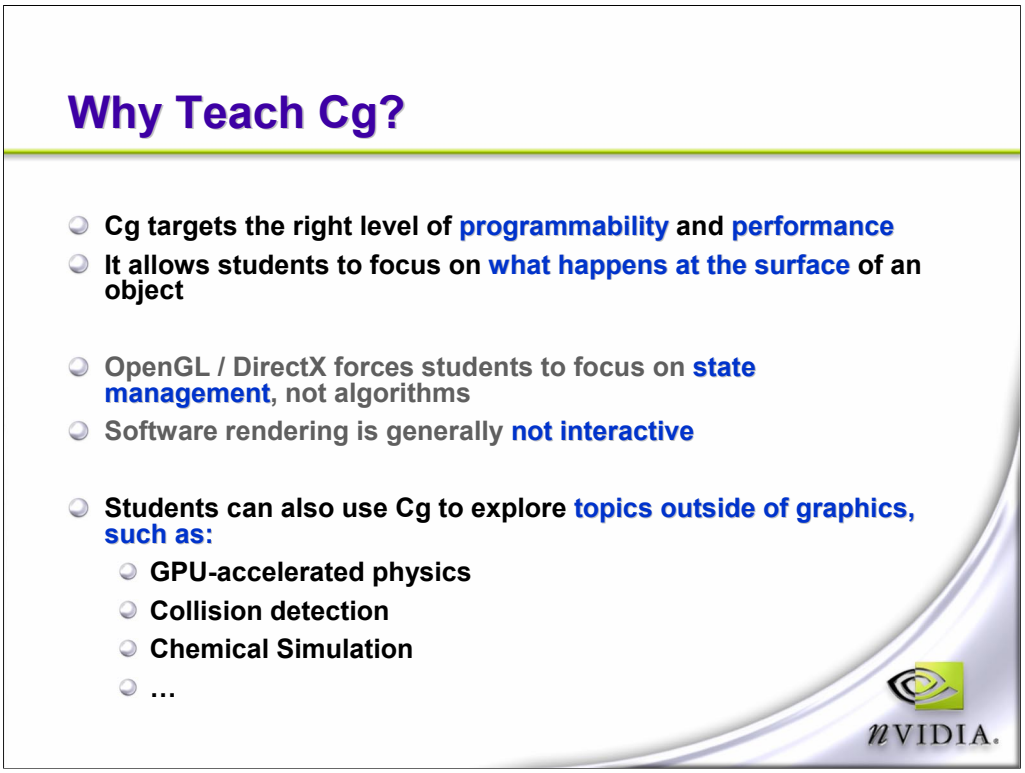
How Cg Works

- **Shaders are created (from scratch, from a common repository, or modified from other shaders)**
- **These shaders are used for modeling in Digital Content Creation (DCC) applications or rendering in other applications**
- **The Cg compiler compiles the shaders to a variety of target platforms, including APIs, OSes, and GPUs**



This slide shows the workflow when using Cg. The main advantage that Cg provides is that hardware-accelerated shaders can be applied during content creation, so that DCC applications can preview objects just as they will appear in the final application.

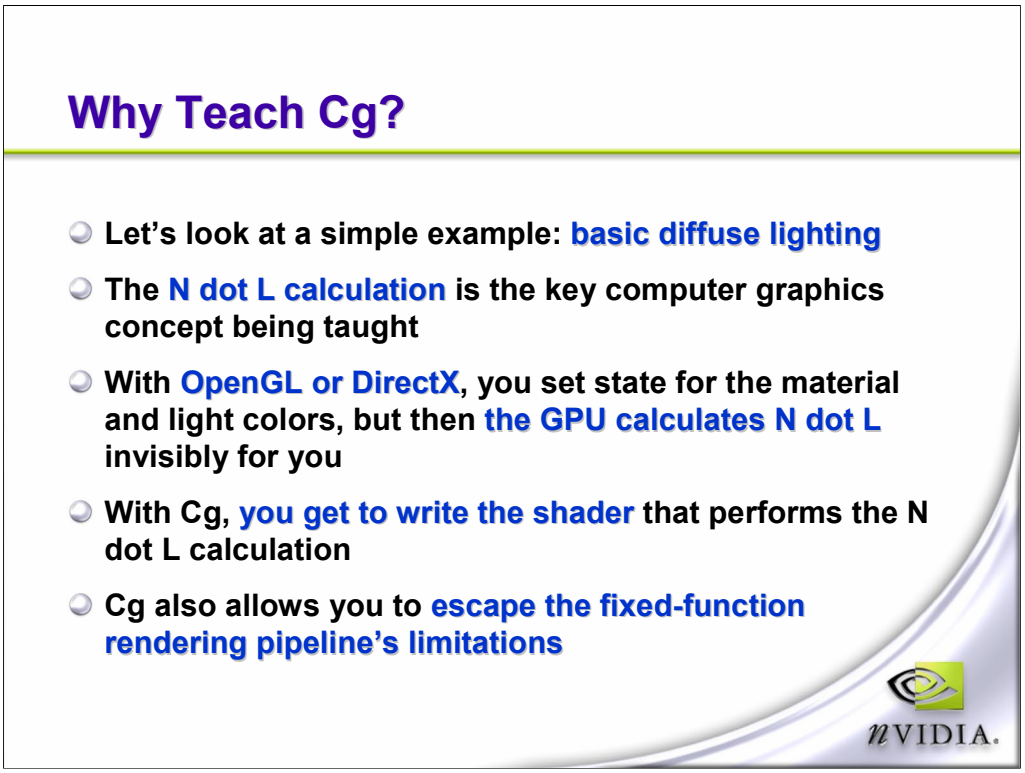
The Cg compiler can then translate the same shaders to a variety of platforms and APIs.



One major reason to teach Cg is that it allows students to focus exactly on the key graphics algorithms. In particular, Cg makes it easy to focus on what happens at the surface of an object.

Using just OpenGL or DirectX to teach a graphics course means that students will spend all their time on state management, while restricted to the capabilities of the fixed-function pipeline.

Using a software rendered to teach a graphics course means that students will not be able to interact with their applications at real-time rates. This stifles their creativity, because programming is really an iterative process of discovery and development.

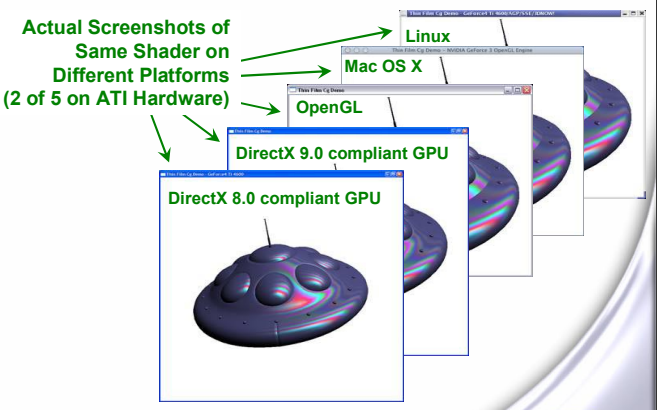


As a simple example, take diffuse lighting. If you were to use OpenGL or DirectX to teach this concept, your students would make a series of function calls to get the material and light properties. The API would then transparently calculate N dot L for them, when in fact this is the key concept that you want to teach.

With Cg, your students can now have complete control of what happens at the surface. After implementing the basic N dot L calculation, they can easily extend it to more advanced models. This ability to extend the model is not available with the fixed-function pipeline.

Why Teach Cg?

- Students who learn Cg can apply their skills in a variety of situations
  - Graphics APIs
    - OpenGL
    - DirectX
  - Operating Systems
    - Windows
    - Linux
    - Mac OS
  - Graphics Hardware
    - NVIDIA GPUs
    - ATI GPUs
    - Other GPUs that support OpenGL and DirectX 9

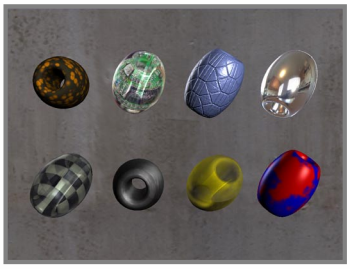


Cg is a highly transferable skill. Students who are proficient in Cg will be able to leverage their skills in a wide range of scenarios (different graphics APIs, operating systems, and graphics hardware).

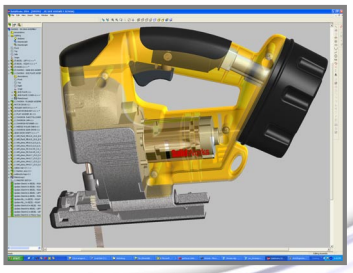
The figure in this slide shows a thin film shader running on several combinations of platform/API/GPU.

Why Teach Cg?

- Cg is integrated with the major DCC and CAD applications
  - 3ds max 5
  - Maya 4.5
  - SOFTIMAGE|XSI 3.0
  - SolidWorks
  - Digital Immersion
  - Mental Images



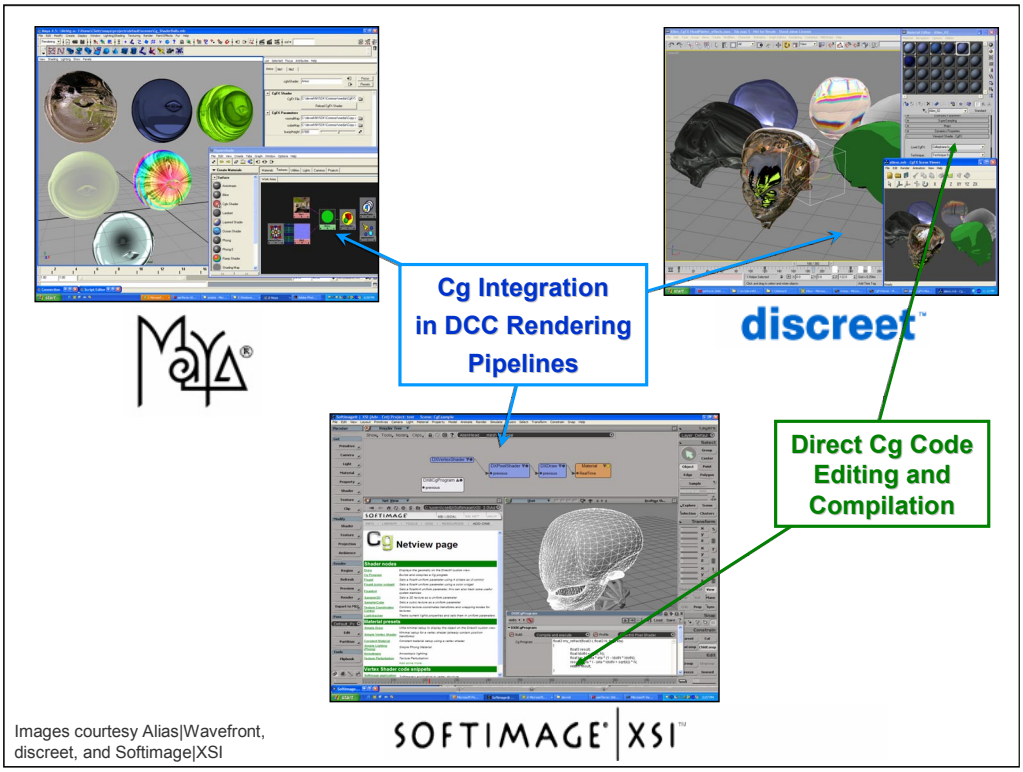
Rendered with Cg (Courtesy of Alias|Wavefront)



Next Version of SolidWorks, Using Cg

NVIDIA.

Cg is also integrated into a number of major DCC and CAD applications. Understanding Cg and hardware shading will make students more capable in industry.



This slide shows Cg's integration with the three major DCC applications.

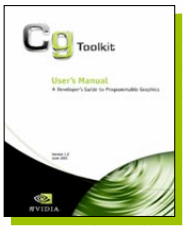
Top Left: CgFX integrated with Maya's hypershade – a node-based shader editor. Sliders control key real-time parameters (for example, bump depth).

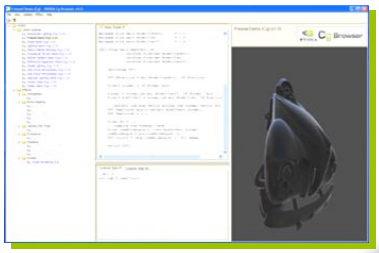
Top Right: The CgFX Viewport manager enables the use of real-time shaders in 3ds max 5.

Bottom: Cg integration in Softimage|XSI's Render Tree allows direct Cg code editing and compilation. Net View also contains help, samples, and documentation.

Comprehensive Documentation

- NVIDIA Cg Compiler
- Cg Standard Library
- Cg Runtime Libraries for DirectX and OpenGL
- NVIDIA Cg Browser
- Cg Language Specification
- Cg User's Manual
- Cg Shaders (assorted pre-written programs)

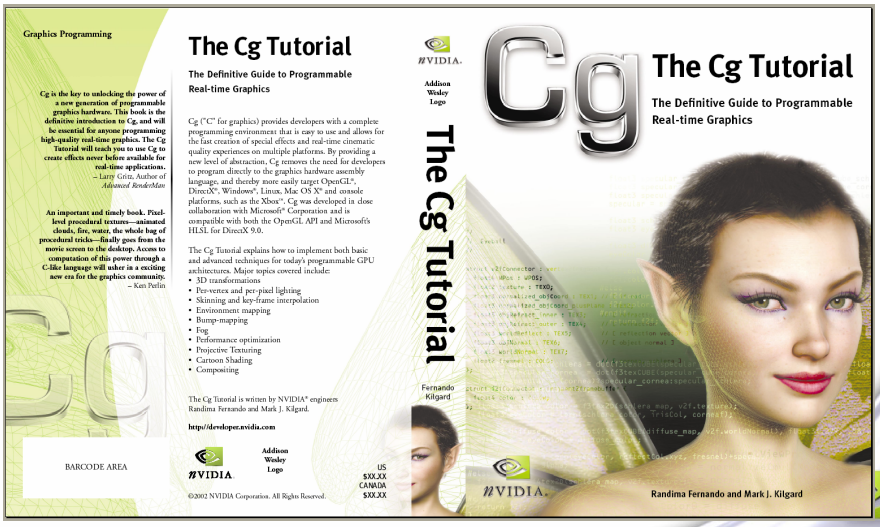




NVIDIA

The Cg toolkit comes with comprehensive documentation, including a complete User's Manual that is available electronically and in hardcopy. It also comes with several sample shaders to learn from.

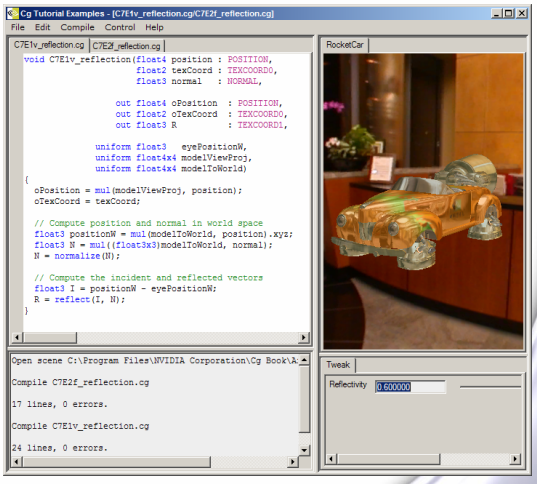
The Cg Tutorial Book



The Cg Tutorial book is an ideal resource for teaching Cg.

The Cg Tutorial Book

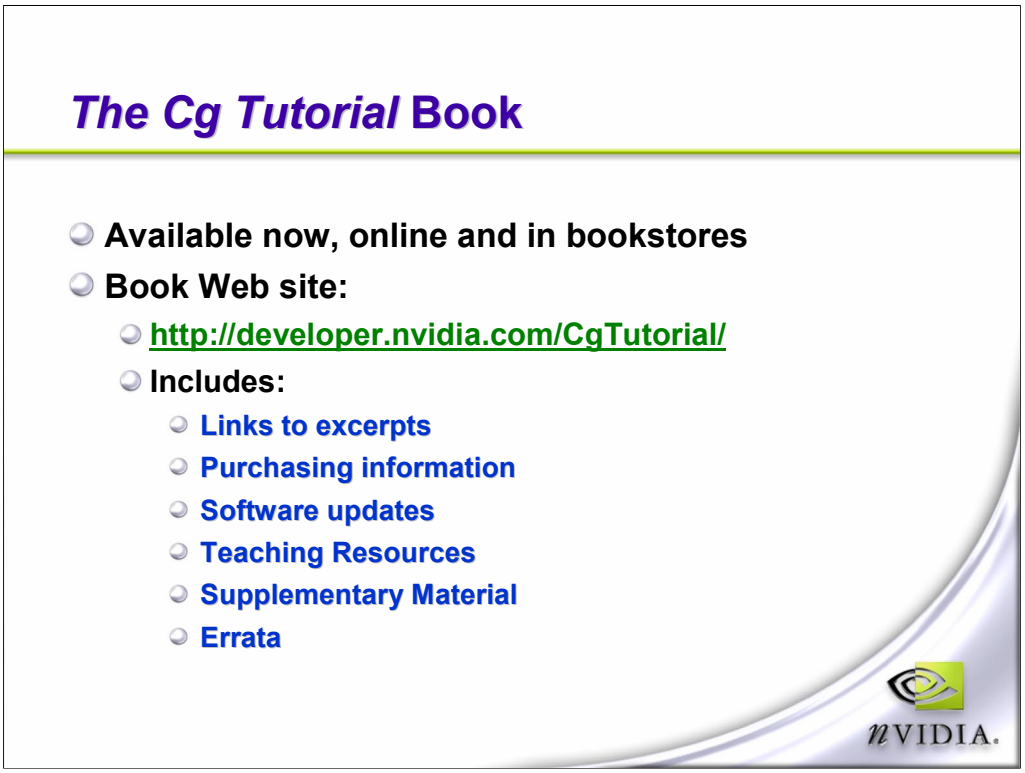
- The first book about hardware shading to:
  - Discuss graphics concepts thoroughly
  - Provide complete examples
  - Provide a complete hands-on framework to try and modify the examples, out-of-the-box
- Includes end-of-chapter exercises and further reading



The Cg Tutorial is the first book about real-time hardware shading to combine theoretical discussion with a clear hands-on approach.

The book is geared towards courses, with end-of-chapter exercises to try out, and further reading for topics of interest.

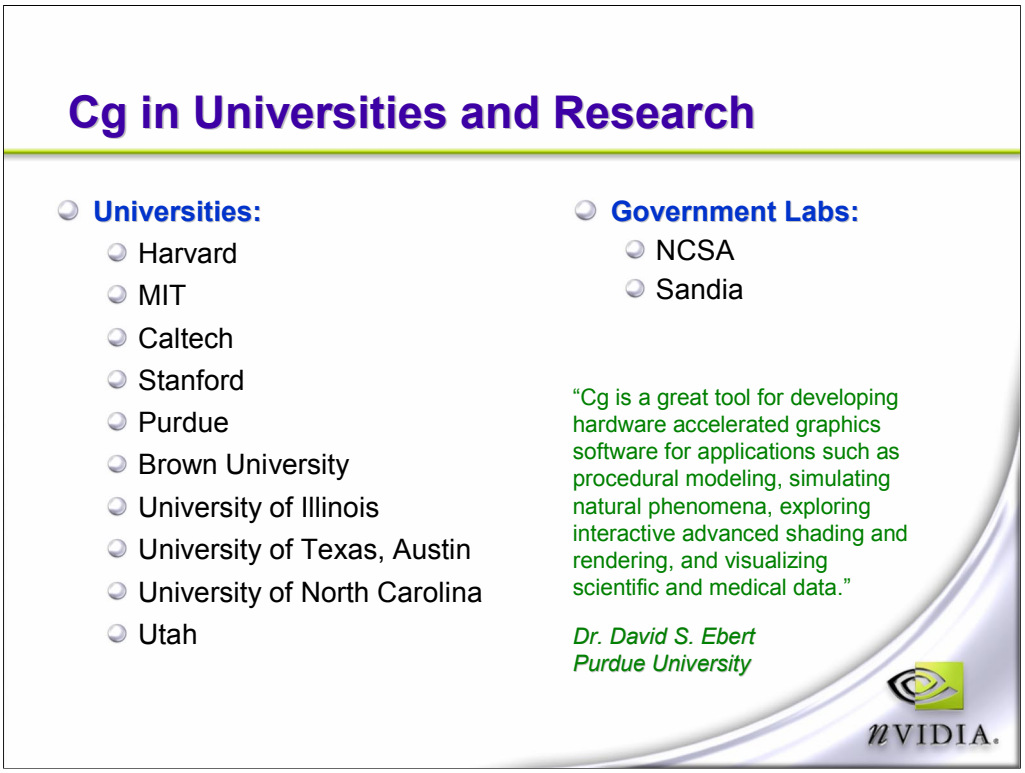
The book also comes with a complete hands-on framework (a standalone application) that allows students to try the various examples in the book, even if they don't already know OpenGL, DirectX, or even C/C++.



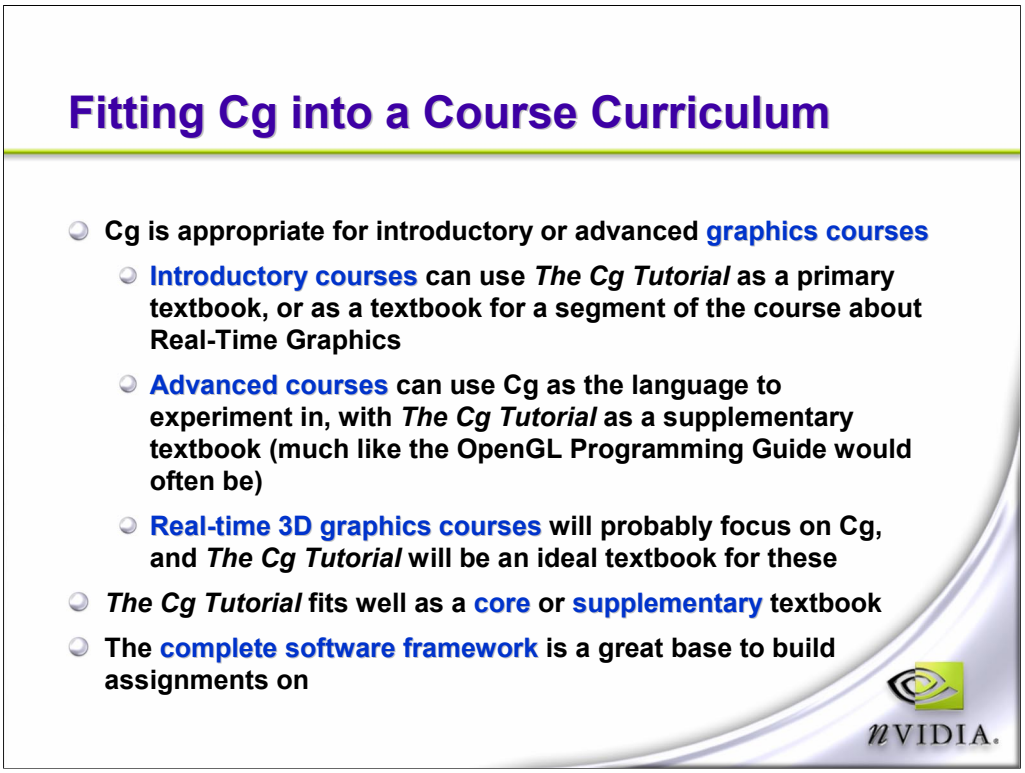
The Cg Tutorial is now available online and in bookstores around the United States. The book's Web site, http://developer.nvidia.com/CgTutorial, contains a wealth of information about the book, as well as related resources.

You can order the book from Addison-Wesley at the link indicated, or you can find it at other sites, such as Amazon.com and BarnesandNoble.com.

The URL indicated also contains sample material from the book.



Several notable universities and government labs are currently involved with Cg.



Because of Cg's cross-platform nature and industry support, it's appropriate to teach in a wide variety of situations. In some cases, it may be best suited as an optional textbook, while in other cases, it may be a primary resource. The standalone software framework that accompanies the book allows students to try and modify the examples that come with the book, without writing any C, C++, OpenGL, or DirectX code. This software is regularly updated and available on the book's Web site.