Technical Brief

NVIDIA GeForce FX GPUs and Microsoft's DirectX 9.0 API

The Art and Science of Visual Effects

NVIDIA GeForce FX GPUs and Microsoft DirectX 9.0 API

The NVIDIA® GeForce™ FX graphics processing units (GPUs) incite a revolution in the graphics world. Teamed with new application programming interfaces (APIs), such as Microsoft® DirectX™ 9.0 (DX9), the highly-anticipated GeForce FX GPUs fuel a new generation of interactive content and brings real-time cinematic shading to the desktop.

The Microsoft DirectX 8.0 (DX8) API was the first DirectX version to enable programmable vertex shading hardware while adding significant flexibility to advanced pixel shading hardware. However, DX8 and DX8 hardware still imposed many restrictions on developers. Microsoft DX9 frees developers from those restrictions. With the most complete DX9 implementation in hardware, the GeForce FX GPUs deliver the performance and power required to overcome three major DX8 limitations:

- ***Pixel shader programmability***: The DX8 pixel shader specifications could appropriately be termed "configurable" rather than programmable because the DX8 spec needed the general programmability associated with a full instruction set and flexible programming structure. The GeForce FX GPUs and DX9 allow much longer shader programs, and give developers a greatly expanded pixel shading instruction set.
- ***Vertex shader programmability***: The DX8 vertex shader specification gave developers very little control over program flow. DX8 vertex shader programs are executed linearly, with no early termination for performance optimization. DX9 and the GeForce FX GPUs support conditional branching for greatly improved program flow control.
- ***Unified pixel shading specifications***: The addition of Pixel Shader 1.4 to the DX8 specification created an alternate pixel shader structure and programming structure that followed a completely different implementation philosophy. Many developers were forced to re-write shader programs for different hardware, rather than writing strictly to the API. The DX9 specification eliminates this problem with the new unified Pixel Shader 2.0 definition.

The combination of DX9 and GeForce FX GPUs narrows the gap between state-of-the-art PC graphics and state-of-the-art rendered movies such as the Disney/Pixar *Monsters, Inc.* and Columbia Pictures *Final Fantasy: The Spirit Within* productions. The DX9 API is a critical enabling technology for this powerful new combination of hardware and software. This paper explains the basic features and benefits of the Direct3D component of the DX9 specification.

New DirectX Features and Effects

The DX9 specification includes three major new features:

- ***Pixel Shader 2.0*:** DX9 exposes true programmability of the pixel shading engine. This makes procedural shading on a GPU possible for the first time.
- ***Vertex Shader 2.0*:** DX9 dramatically enhances the power of the previous DirectX vertex shader by increasing the length and flexibility of vertex programs.
- ***High-precision, floating-point color*:** DX9 breaks the mathematical precision barrier that has limited PC graphics in the past. Precision, and therefore visual quality, is increased with 128-bit floating-point color per pixel.

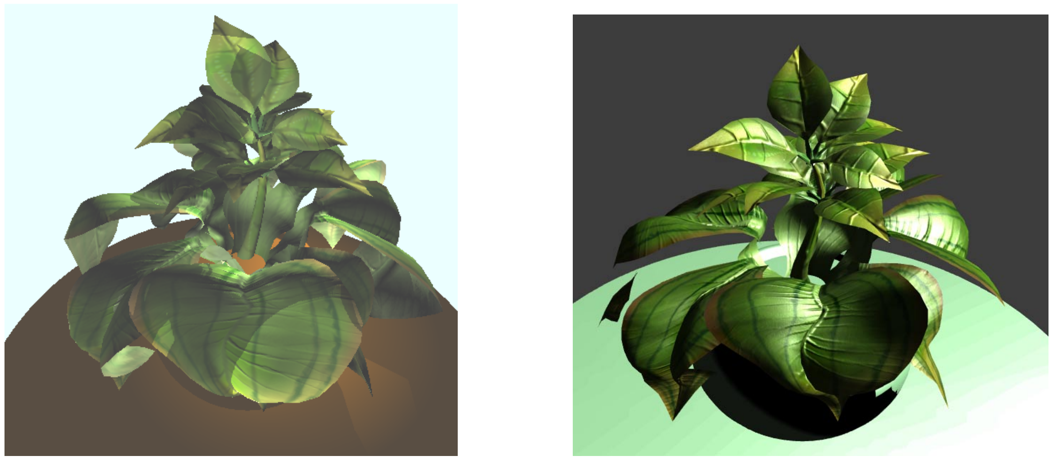
To take advantage of these new features in DX9, NVIDIA has developed the NVIDIA Cg Developer's Toolkit. When combining the NVIDIA Cg Developer Toolkit with the GeForce FX GPU, developers have the ability to take full advantage of the API to develop stunning visual effects. For more information on the Cg language and NVIDIA solutions, please review the NVIDIA technical paper titled "The NVIDIA Cg Compilier: C for Graphics" available at

http://www.nvidia.com/view.asp?IO=Cg.

Stunning Effects

The versatility and power of the GeForce FX architecture combined with DX9 enable new digital worlds with stunning visual effects that were simply not achievable with DX8. Through its programmability and higher precision, DX9 brings sophisticated effects into the world of real-time graphics for the first time ever. Plants, metallic paint, skin, and eyes are great examples of objects that can be visually stunning with this new platform. Side-by-side comparisons highlight the leap in image quality made possible by DX9 (see Figures 1 and 2).

DX8 DX9



(© 2002 NVIDIA Corporation)

Figure 1. DX8 compared to DX9 vegetation

DX8 DX9

(© 2002 NVIDIA Corporation)

Figure 2. Standard glossy paintcompared to metallic paint effect

Faster Effects

As an example of the performance enhancing capabilities of DX9, consider the Wolfman character in one of the NVIDIA GeForce4 Ti family demos (Figure 3).

(© 2002 NVIDIA Corporation)

Figure 3. The Wolfman character

Using DX8-style rendering techniques, the fur for Wolfman required eight passes for every pixel. The technique involved modeling the fur as several layers of geometry, and stepping through each layer to calculate its contribution to the current pixel. This procedure is explained in detail in the NVIDIA technical brief titled "nfiniteFX II Engine: From Research to Real-Time" available at

http://www.nvidia.com/view.asp?PAGE=pg_20020201142843.

Using DX9, GeForce FX GPU is capable of rendering the Wolfman's fur in a single pass. Collapsing the rendering from eight passes down to a single pass delivers huge performance gains, without compromising image quality.

Programmability Using DX9

DX9 offers a richer programming language with more commands, as well as a longer and more flexible program structure. The specific enhancements for pixel shader programs and vertex shader programs are discussed separately in following sections. Table 1 summarizes the fundamental limits for DX8, DX9, and the special capability bits (also known as "cap bits") that extend the DX9 shader specifications to take better advantage of the capabilities inherent to the GeForce FX architecture. Note that the move from DX8 to DX9 results in an increase of various resources such as constants and programming registers to store temporary data. Longer programs will lead to more sophisticated programming, which in turn requires more of these resources.

Table 1. Quick-look specifications for DX8, DX9, and beyond

	DX8 Specification	**DX9 Specification**	**The GeForce FX Implementation of DX9**
Pixel Shader	Version 1.4	Version 2.0	Version 2.0+
Instructions	32	64	Thousands
Constants	8	32	Thousands
Registers	6	12	32
Vertex Shader	Version 1.1	Version 2.0	Version 2.0+
Instructions	128	256	256
Constants	96	256	256
Registers	12	12	16

Note: Table entries are the maximum numbers of instructions, constants, or registers that can be used in a valid shading program.

Pixel Shader 2.0 Benefits

The Pixel Shader 2.0 specification in DX9 enables stunning new levels of image quality. The new specification exposes true programmability for pixel shader hardware such as that of the GeForce FX architecture. Programmers and artists can now create new, distinctive images—raising the art and science of images to a higher level. The programmable nature of the GeForce FX GPUs and DX9 also result in more and a broader range of tools such as procedural shading.

Artistic Control

Before DX9 and GeForce FX GPUs, developers were limited to very simple pixel shader effects. Previous GPUs and APIs offered pixel shader engines that were configurable rather than programmable. The distinction is important. The DX8 Pixel Shader 1.4 specification (and, consequently, DX8 hardware) has a limited list of valid pixel instructions and those instructions are focused on various texture-related operations (texture fetching, texture blending, etc.). Additionally, DX8 Pixel Shader programs are limited to 32 instructions. Contrast this with DX9 that introduces a full instruction set and a generalized programming model for hardware pixel shader engines.

The Pixel Shader 2.0 instruction set (part of the DX9 specification) has all of the commands available in the Vertex Shader 2.0 instruction set (also part of DX9), plus a few more that deal specifically with texturing and therefore have no context in vertex shader programs. These new pixel commands include math functions and flow-control functions.

DX9 pixel shader programs and GeForce FX GPUs break down the barriers between an artistic vision and a final product that can be shipped to and enjoyed by end users. If the displayed effect doesn't match the original vision, the developer or artist has the creative control to tune it and tweak it until it does.

Scientific Instructions

The Pixel Shader 2.0 math instructions allow the pixel shader engine to solve arbitrary math algorithms. Developers can define their own mathematical formula as part of a special effect. As a result, developers can create distinctive and stylized effects like the multi-colored alien (see Figure 4). Instead of the plain blue alien image, they can create a unique and stylized look such as depicted in the pink and brown alien, where the color of the object is a function of the angle of light hitting it.

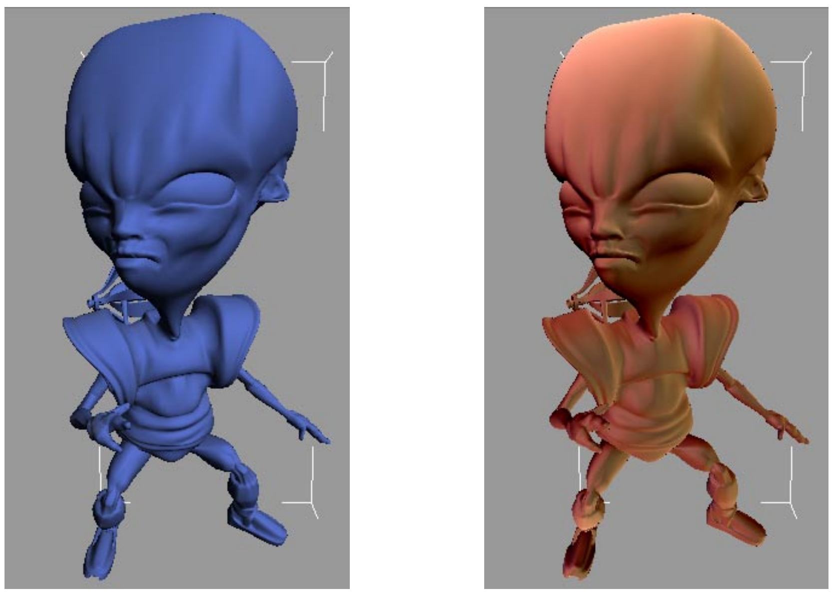


Figure 4. DX9 offers the ability to interpret mathematical functions for achieving unique effects, in contrast to the limitations of DX8

Flow Control

DX9 is the first Direct3D specification that enables loops and branches in pixel shader programs. These flow control capabilities offer two distinct benefits. They give a developer the freedom to structure their shader program in the most efficient manner for the problem being solved. The Pixel Shader specifications in DX8 were very restrictive in terms of which commands could follow others and how many levels of texture dependency could be used. DX9 flow control also enables code re-use through conditional branching and other similar techniques. Effects can be grouped into categories or families, creating the opportunity to re-use lots of code that can be written and debugged in a generalized way—resulting in a generalized wood-grain shader, a generalized vegetation shader, a generalized eyeball shader, etc. This enhances productivity for developer teams and organizations, which can have the effect of reducing time-to-market, reducing costs, or leaving extra time for developers and artists to create even more distinctive effects than otherwise possible within the scheduled development phase.

Procedural Shading

The new level of programmability in DX9 also provides the developer with a wider array of tools in the artist's virtual toolbox. The flexibility to perform arbitrary math in the pixel shader engine means that there are more ways to solve a problem. Developers can use procedural shading to create sophisticated 2D or 3D patterns on the fly. For example, the image shown in Figure 5 was created without any pre-computed textures.

(Image courtesy of artist Goran Kocov))

Figure 5. The wood grains and surface of the rough-looking ceramic vase were created mathematically using procedural shading techniques; no textures were used to create this scene

The benefit to the developer is the freedom to solve the problem multiple ways. Procedural shading uses mathematics to calculate what the pixel color should be and offers a solution to the classic texturing problem of mapping a 2D image to a curved object.

Vertex Shader 2.0 Benefits

Vertex Shader 2.0 introduces true programmability to the DX9 vertex shader specification. Two key changes, compared to the DX8 specification, are longer vertex programs and flow control using conditional branching and subroutine commands. Vertex Shader 2.0 has also extended the instruction set with new math functions.

Longer Vertex Programs

With DX9, developers can write longer vertex programs and create more sophisticated and visually compelling effects (see Figure 6). The longer programs can have a huge impact on the apparent complexity of the objects in the scene.

(© 2002 NVIDIA Corporation)

Figure 6. DX9 vertex programs also simplify the animation of complex geometries such as water

These same techniques can be used to animate complex geometry for rendering waves in ocean water. While calculating the parameters, the pixel shading engine will need to do per-pixel calculations for how much light shines through the wave as well as how much reflects off the surface of the wave.

Flow Control

Program flow control gives the developer the freedom to create sophisticated programs that use subroutines to iterate on mathematical calculations as well as provide early termination by branching to the end of the program based on pre-determined conditions. The opportunity for early termination is important for overall vertex program performance. Sometimes the vertex shading program can determine that additional calculations are unnecessary. Under those conditions, a DX9 vertex shading engine can end the program and pass the results to the next stage of the graphics pipeline, while a DX8 vertex shading engine would need to step through the remainder of the program, despite the fact that additional calculations did not improve the overall image quality. The DX9 vertex shading engine will be able to start working on the next vertex (or load the next vertex program) instead of waiting to march through the unnecessary calculations that are required under DX8.

High-Precision Floating-Point Color

To create more sophisticated special effects, high-precision operations become critical. By combining the NVIDIA GeForce FX GPUs and DX9 API, developers can now utilize the same high standards of color precision used by the film industry today. The inherent 16- and 32-bit floating point formats (FP16 and FP32) of the GeForce FX architecture give developers the flexibility to create the highest-quality graphics. FP32 offers the ultimate image quality, delivering true 128-bit color. FP16 provides an optimal balance of image quality and performance.

Developers are free to move back and forth between these formats in their application, using the format that is best suited to a particular computation. For instance, some actions such as indexing into a texture can only be optimally accomplished using a 32-bit floating-points format. If the texture is larger than 1024 x 1024 (2^{10} x 2^{10}, requiring at least 20 bits), the developer needs FP32 to access all of the data. Other computations can be accurately accomplished using FP16, and can benefit from the maximized execution speed afforded by this level of precision.

The accuracy levels and choice afforded by these higher-precision formats make it possible for developers to produce cinematic graphics in real time.

For a more comprehensive discussion of the issues and benefits of high mathematical precision in graphics, see the NVIDIA white paper titled "High-Precision Graphics: Studio-Quality Color on the PC" that can be found at

http://www.nvidia.com/view.asp?PAGE=techbriefs111802

Conclusion

The Microsoft DX9 specification is a critical element for raising the quality of real-time graphics. This API, paired with the capabilities in GeForce FX GPUs, enables a stunning new class of imagery in real-time applications. The combination of the API and the GeForce FX platform brings the best programmability, flexibility, and precision for cinematic graphics rendering systems to the desktop. Developers are empowered to express their artistic vision, and end users will enjoy unprecedented visual experiences.

NVIDIA Corporation
2701 San Tomas Expressway
Santa Clara, CA 95050
www.nvidia.com