Technical Brief

Microsoft® DirectX® 7: What's New for Graphics

Microsoft DirectX 7: What's New for Graphics?

Microsoft's introduction of DirectX 7 introduces a wide range of new 3D graphics features for the personal computer. The most interesting of these new features include support for hardware acceleration of transformation and lighting, cube environment mapping, vertex blending, and particle systems. NVIDIA's GeForce 256™ GPU (Graphics Processing Unit) fully supports the entire suite of DirectX 7 for graphics. By providing the world's best implementation for acceleration of these key new features, NVIDIA will enable a new class of visual and entertainment experience on the personal computer.

About DirectX

Microsoft® DirectX® is a group of technologies designed by Microsoft to make Windows-based computers an ideal platform for running and displaying applications rich in multimedia elements such as full-color graphics, video, 3D animation, and surround sound. Built directly into the Microsoft® Windows® family of operating systems, DirectX is an integral part of Windows 98 and Windows 2000, as well as Microsoft® Internet Explorer. DirectX components may also be automatically installed on your system by advanced multimedia games and applications written for Windows 95. (Early releases of Windows 95 may not include DirectX. If you think your Windows 95 operating system does not include DirectX, you can download it from the www.microsoft.com/directx.)

Microsoft's goal in developing DirectX was to provide developers with a common set of instructions and components that would accomplish two things. First, DirectX would allow developers to be confident that their multimedia applications would run on any Windows-based PC, no matter what the hardware, and at the same time ensure that their products take full advantage of high-performance hardware capabilities to achieve the best possible performance. This was accomplished through the DirectX Foundation layer. Second, DirectX would make life easier for developers by giving them tools that simplify the creation and playback of multimedia content, while at the same time making it easier to integrate a wide range of multimedia elements. DirectX does this with the DirectX Media layer.

DirectX provides developers with new opportunities for creativity and innovation by allowing them to focus on building unique features for their application without having to worry about which display adapter, sound card, or 3D accelerator chip is installed in your PC. And because DirectX was designed to support future innovations in software and hardware, developers and consumers can be confident that they will continue to get the best possible performance from their applications as technology advances.

DirectX Foundation

DirectX Foundation gives developers a single set of APIs (application programming interfaces) that provides them with improved access to the advanced features of high-performance hardware such as 3D graphics acceleration chips and sound cards. These APIs control what are called "low-level functions," including 2D graphics acceleration; support for input devices such as joy sticks and keyboards, and mice; and control of sound mixing and sound output. The low-level functions are supported by the components that make up the DirectX Foundation layer: Microsoft DirectDraw®, Microsoft Direct3D®, Microsoft DirectInput®, Microsoft DirectSound®, Microsoft DirectPlay®, and Microsoft DirectMusic®.

Before DirectX, developers creating multimedia applications for Windows machines had to customize their products so that they would work well on the wide variety of hardware devices and configurations available on Windows machines. DirectX Foundation provides something called a "hardware abstraction layer" (HAL for short) that uses software drivers to communicate between game software and computer hardware. As

a result, developers can write a single version of their product that utilizes DirectX without worrying about the wide range of hardware devices and configurations in existence.

DirectX Foundation also provides developers with tools that help you get the best possible performance from the machine you're using. It automatically determines the hardware capabilities of your computer and then sets the application's parameters to match. DirectX also allows you to run multimedia applications that require support for features that their system doesn't offer by simulating certain hardware devices through a "hardware emulation layer" (called HEL) that provides software-based drivers that act like hardware. For example, a DirectX game that makes use of 3D imagery can run on a machine that doesn't have a 3D acceleration card because DirectX simulates the services of a 3D card.

DirectX Media Layer

DirectX Foundation handles the low-level functions. The DirectX Media layer sits on top of DirectX Foundation and provides high-level services that support animation, media streaming (transmission and viewing of audio and video as it downloads over the Internet), and interactivity. Automatic integration of the low-level services of DirectX Foundation and the high-level services of DirectX Media simplifies the process of creating and playing multimedia elements, allowing developers to energize and enhance applications and Web pages with cutting-edge interactive content that was previously impossible.

Like DirectX Foundation, DirectX Media is made up of several integrated components. These components include Microsoft DirectShow™, DirectAnimation™, and DirectX Transform. Support for DirectShow and DirectAnimation is built into the latest versions of Microsoft Internet Explorer. As a result, software developers and web site creators now have unprecedented control over graphics, animation, audio, video, and other online multimedia elements.

The DirectX Media layer helps developers solve one of the most daunting tasks they face in developing great games for Windows-based machines – coordinating different types of multimedia effects. It does so in two ways. First, it offers a set of APIs that makes it possible for different types of effects to work together as if they were a single application. (Prior to DirectX, developers often had to use an API from one software vendor for audio, another for video, and yet another for animation, making it difficult to combine elements.)

DirectX Media layer also helps developers synchronize the playback of multimedia elements. DirectX provides a time-based approach to coordinating multimedia, allowing game developers to anchor media elements of an application (2D and 3D animation, video, audio, etc.) along a single timeline. That means a developer can take a 2D character, layer it over a video clip, and then add sound, and ensure that all three multimedia elements will run smoothly and in the right relationship to each other.

A Common Library and More

Taken together, DirectX Foundation and DirectX Media provide developers with another important benefit. DirectX contains a library of components that they can draw on to create the underlying elements of a game. This library simplifies the development process by giving programmers a large group of pre-existing interfaces and elements to use in their products. That means they can concentrate their efforts on making their products more original and creative, and spend less time struggling to figure out how to make an effect work in the first place.

The components that make up DirectX also provide hardware makers with a flexible platform that is designed to get the most out of advances in technology. Because DirectX accesses a computer's

hardware through software drivers, hardware developers can create special DirectX drivers that ensure you get the most out of advances in hardware technology – now and in the future.

Direct3D

The graphics programming interface portion of DirectX is called Direct3D. Direct3D serves two main purposes. The first is to provide a standardized programming interface for application developers (typically game developers) to interface with 3D acceleration hardware. The second purpose is to provide 3D hardware acceleration manufacturers with a method to expose the functionality of their acceleration device to application developers in a standardized way.

These standardized programming interfaces provide a more unified and readily supportable method for making things work together. This reduces development time by giving developers of games and other applications access to hardware acceleration without having to hand-code to a specific piece of hardware.

What does all this mean for end users? DirectX and Direct3D enable applications to work seamlessly with high performance graphics acceleration hardware, like NVIDIA's award-winning RIVA TNT2™ and GeForce 256 GPU, to provide the richest, most visually stunning experience possible.

DirectX and Direct3D were introduced several years ago, and have gone through steady evolution and advancement, offering increased levels of performance, functionality and ease of use. DirectX, version 6, introduced many improvements in the area of graphics, most notably – multitexturing.

Multitexturing

Multitexturing is the ability to combine two or more texture maps to achieve realistic special effects in a 3D world. The most common example of multitexturing is the use of light maps to provide more realistic per-pixel lighting effects. An example is illustrated below.

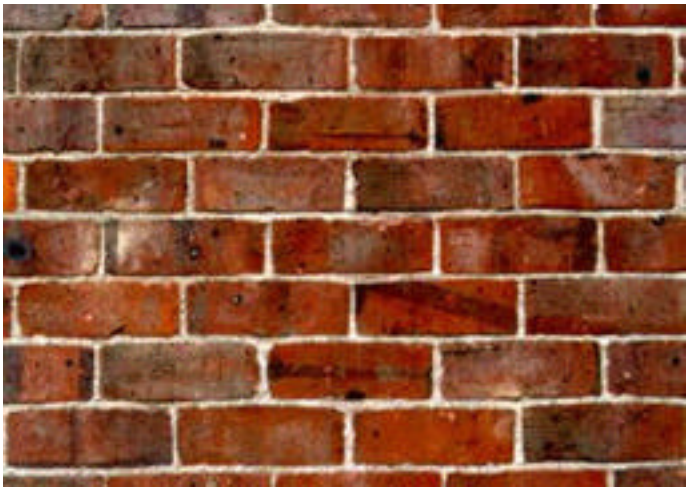

Figure 1

Figure 1 represents a traditional two-dimensional texture map of a brick texture. This picture could be created in a variety of ways, from a scanned photograph, to an artist's creation. The two-dimensional picture is mapped onto a geometric surface to provide that geometric surface with additional realism and detail. Note how the brick texture appears to be uniformly lit, without any dramatic light effects or shadows. Figure 1 will be used as the base texture map for our multitexturing example.

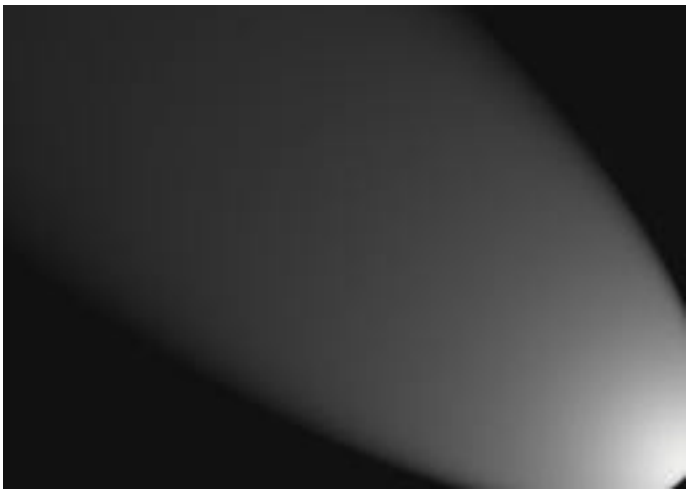

Figure 2

Figure 2 represents a sample light map. A light map is nothing more than another simple two-dimensional texture map designed to provide lighting effects to another texture. As can be seen from Figure 2, this light map represents a single, directional, soft-edged spotlight. The multitexturing process will create increased shadow or darkness from the dark areas of this texture map, and lighter areas from the highlights when applied to a base texture map.

Figure 3

Figure 3 represents the resulting multitexturing combination of the base texture and the light map texture, to create a more realistic and dramatic lighting effect. Using sophisticated multitexturing effects like these allows artists to create more realistic and engrossing environments, dramatically improving the visual quality of a gaming experience.

Advanced graphics acceleration hardware like NVIDIA's RIVA TNT2 and GeForce 256 processors are capable of implementing these types of multitexturing effects in a single pass, that is without having to render the polygons twice, and write the image to the frame buffer twice. This "single-pass" multitexturing results in improved image quality and performance, and enables the most realistic and complete implementations of texture effects that graphics API's like DirectX support. While multitexturing provided a

key advance towards the creation of more realistic and exciting imagery on the personal computer, we still have a long way to go before the personal computer is capable of reproducing reality in real time. Tremendous advancements are required in both hardware and software. Both Microsoft and NVIDIA are pushing the evolution of graphics at fierce pace, Microsoft with the introduction of DirectX 7 and NVIDIA with the introduction of the GeForce 256 GPU.

What's New in DirectX 7?

DirectX 7 will be released in the fall of 1999, introducing a number of breakthrough new technologies. We will focus on the graphics portion of DirectX – Direct3D.

Cube Environment Mapping

Imagine nature without reflections. What if you looked into a pool of water and saw nothing reflecting back? While the goal of the PC graphics industry is to perfect the realism of the visual experience, certain rendering tasks such as accurate, real-time reflections have proven too difficult to implement practically. It often only takes a single artifact or incorrect surface detail to make a computed scene look fake and objectionable. Creating photorealistic "virtual environments" requires that surfaces are capable of reflecting light as they would in the natural world. Yet using today's hardware and common software tricks, it is impossible to create accurate reflections without difficult software implementations, objectionable artifacts, and disastrous performance hits. As a result, computer-synthesized environments are often simple and less interesting as developers have been forced to avoid reflective surfaces.

DirectX 7 introduces cube environment mapping to address the challenge of creating photorealistic reflections and lighting in real time. Using DirectX 7 in combination with GeForce 256's cube environment mapping, mainstream PC users will be able to experience stunning real-time, accurate reflections and specular lighting effects that previously were prohibitively difficult to create. With cube environment mapping, developers will be able to utilize correct reflections to enrich 3D scenes and enhance game play. Correct, real-time reflections will allow characters to gain insight into the surrounding 3D scenes, perhaps revealing the next step in game play. Cube environment mapping offers a quantum leap in visual realism that is easy to implement and fully supported by DirectX 7 and GeForce 256. Figure 4 is an example of using cube environment mapping to reflect an environment on a mirrored sphere.

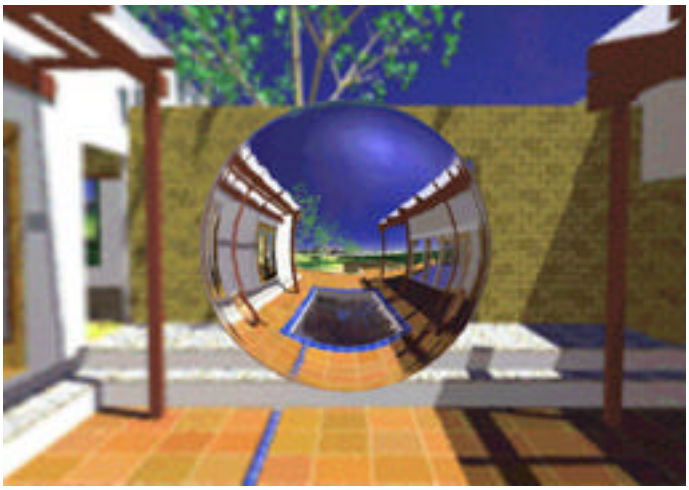

Figure 4

Hardware Transformation and Lighting

The basic building block of all real-time 3D graphics applications is the polygon, typically a triangle. A triangle is built by connecting three vertices together. These vertices carry information about the position, color, lighting, texture and other parameters necessary to construct a picture from those triangles. In order to produce increasingly realistic scenes, more triangles (and hence more vertices) must be employed. Today's advanced graphics hardware, such as NVIDIA's award-winning RIVA TNT2, is capable of producing hundreds of millions of pixels from millions of triangles every second. Unfortunately, the process of creating that many triangles, manipulating them, and transmitting them from the host processor to the graphics hardware has been beyond the capability of the personal computer. The computational resources required to do that would completely swamp today's fastest processors from Intel® or AMD™, preventing anything else from being done.

This is unfortunate as applications like games need to do much more than just generate triangles to draw pictures on the display. They need to run sophisticated artificial intelligence (AI) algorithms to provide characters in the game with life, complex physics to detect collisions between objects and characters in the game, advanced networking to provide multiplayer interactive gaming, and a host of other computationally challenging tasks. Game developers, and hence game players have been forced to make compromises in all aspects of game design, as no one part of the game should consume all the available processing power of the host.

With hardware support for transformation and lighting enabled in DirectX 7, NVIDIA has been able to provide the industry's first mainstream GPU with hardware acceleration for transformation and lighting in its revolutionary GeForce 256 GPU, breaking through the largest barrier to a more interactive and immersive experience.

Transform and lighting are the first two of the four major steps in the 3D graphics pipeline, or series of steps required to produce 3D graphics. Both steps are mathematically intensive and specific in the mathematics they require. They are also highly repetitive, with the same set of instructions performed millions of times per second. Thus, both transformation and lighting are perfect candidates for hardware acceleration.

The process of describing and displaying 3D graphics objects and environments is very complex. To reduce this complexity, it is useful to describe the 3D data according to different frames of reference, or different coordinate systems, at different times. These different frames of reference are referred to as "spaces" such as world space, eye space and screen space. Each of these spaces is convenient for one or more operations that must be performed as a 3D image is created. World space is convenient for building the 3D objects that are part of the 3D world. Eye space is convenient for lighting and culling and screen space is convenient for storing the scene in the graphics frame buffer. However, because these spaces use different coordinate systems, 3D data must be converted or "transformed" from one space to another as it moves through the 3D pipeline. The transform engine performs all of these mathematical transformations.

The lighting engine is similar to the transform engine because it has a set of mathematical functions that it must perform. This engine calculates vectors from lights to objects in the 3D scene as well as vectors from objects to the viewer's eyes. A vector, by definition, contains information about direction and distance. The lighting engine must also separate the length or distance information from the direction information because that simplifies future steps in the 3D pipeline. Lighting calculations are used for

vertex lighting, such as advanced fog effects, that are based on the eye-to-object distance rather than just the Z-value of the object.

The simple fact is that geometry transform and lighting performance has now become the most significant barrier to more sophisticated 3D graphics on a typical PC. This bottleneck has forced software developers to limit the geometric complexity of their 3D characters and environments, sacrificing image quality and elaborate 3D environments in order to maintain expected performance. Dedicated transform engines are the most cost-effective way to alleviate the geometry performance bottleneck and open up new opportunities for software developers to add much more geometric detail to their 3D worlds. As Figure 5 demonstrates, the ability to use more polygons can have a dramatic impact on the realism of an object. The image on the left is generated from a model representative of the best that today's games could offer. The entire model is composed of 998 polygons. The image on the right is representative of what the next generation of model complexity could offer, approximately 100,000 polygons. Notice that much of the detail is simply missing from the image on the left, and much of the realism is lost, as compromises must be made to reduce the polygon counts. Notice how the wheel in the image on the left is octagonal, while the wheel in the image on the right is round. These subtle details make a tremendous difference in creating an immersive interactive experience.

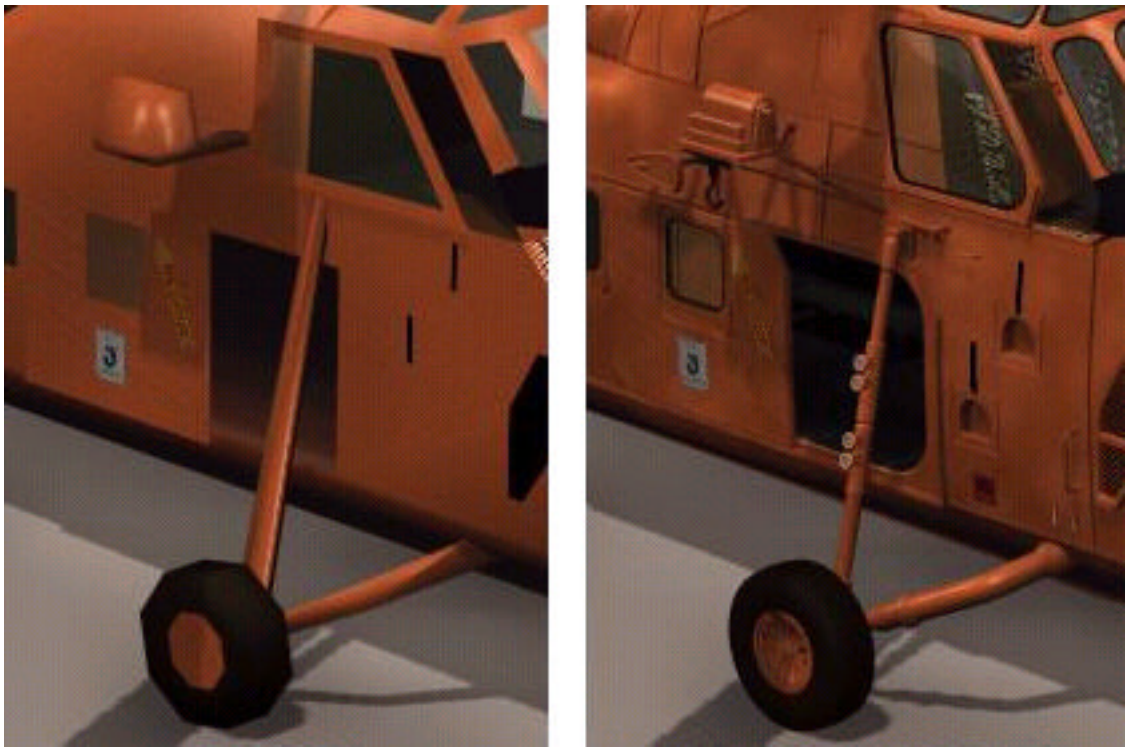

Image courtesy of Viewpoint.

Figure 5

Vertex Blending

Vertex blending is a sophisticated technique allowing game developers to create smooth joints between the multiple parts of a geometric model. Using vertex blending in combination with hardware like the GeForce 256 GPU will allow game developers new freedom in designing more lifelike and realistic characters and geometry. Vertex blending is particularly useful for characters and lifelike character animation. It allows content developers the ability to seamlessly join different pieces of geometry to form a continuous surface without seams or cracks. Imagine the ability to join the upper and lower arm of a humanoid character such that there were no seams or cracks at the joints. As Figure 6 illustrates below with simple geometry, it is possible to create a continuous surface from two separate pieces of geometry. On the left is what game developers are forced to deal with today, cracks and visible seams. On the right is the continuously smooth, crack-free surfaces that game developers will be able to implement using DirectX and the vertex blending features of the GeForce 256 GPU.

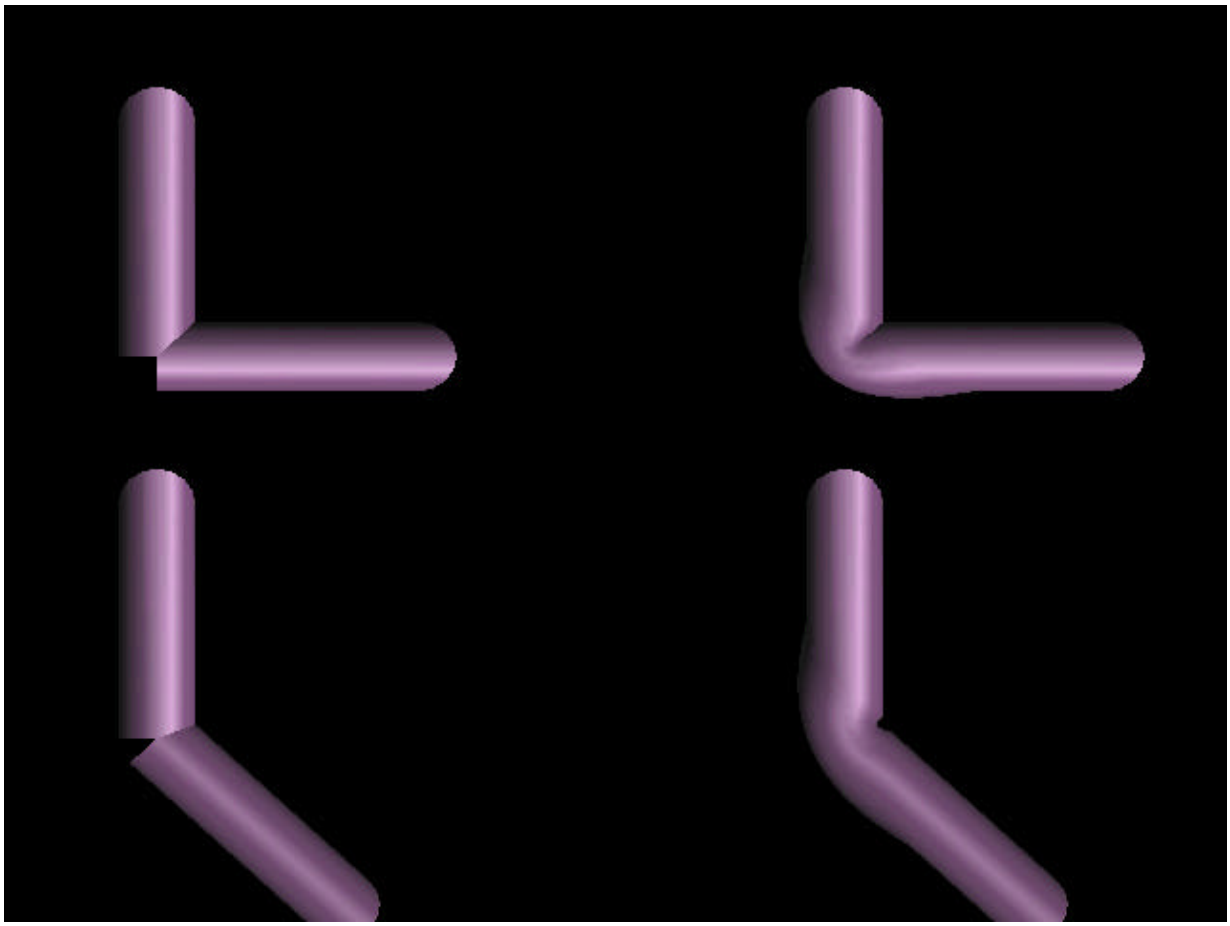

Figure 6

Pushing the Realism Envelope

NVIDIA and Microsoft together have created a new graphics platform that dramatically advances the state of the art for entertainment graphics. By enabling functionality such as hardware support for transformation and lighting, the largest graphics burden on the CPU has been moved to dedicated graphics hardware. This frees up valuable processing cycles to increase the realism and fidelity in functions like physics and artificial intelligence, which allows game developers to push the envelops of realism. Visual enhancements like cube environment mapping allow content developers to push the visual envelope while concurrently pushing the environmental realism envelope with the newfound CPU cycles that hardware transformation and lighting have provided.

By enabling this new functionality through an industry standard programming interface – DirectX – NVIDIA and Microsoft together in partnership with content developers are raising the bar for realism in visual content.

Notes

Microsoft Corporation graciously provided the “About DirectX” section of this white paper.