# Effects of Shader Technology: Current-Generation Game Consoles and Real-Time Graphics Applications

Matthew Christian

## A Quick History of Pixel and Vertex Shaders

Pixel and vertex shader technology built alongside new hardware has contributed to one of the largest effects on computer graphics since the start of computers. While we can implement high end graphics, different systems are still split on their implementations. Every system has its pros and cons regarding pixel shaders, vertex shaders, and the way they are currently implemented. Mixtures of hardware and software create a mess of uses on multiple video game consoles. And while each mixture has its plusses, each also has its shortcomings.

In 2001, the first consumer-level programmable graphics processing unit (GPU) was introduced to the computer user market by NVIDIA (NVIDIA Corporation, 1995-2007). Before this, games had to utilize hardware transformation and lighting which used light calculations that were built directly on the hardware itself. As a result, all games made during this period of time utilizing this hardware generally looked the same. During this same time period, movie studios such as Pixar Animation Studios created amazing visual effects using a custom built piece of software called RenderMan. RenderMan uses a 'shading language' which is specially created code that calculates the lighting and transforming of vertices and pixels in a digital scene and thus bypassing the use for the lighting built into the hardware (Kesson, 2002-2004).

The problem with RenderMan and its shading language is the time it takes to produce movie effects. Movies are created using the effects provided by RenderMan but are then compiled into a single movie file which, depending on the length of the film, could take days or even weeks before someone is able to view the effects produced by RenderMan. Programmable GPU's were created to provide this high

level of graphical detail in a very short (almost instantaneous) amount of time. With the help of DirectX 8.0, Microsoft's graphical application programming interface (API), programmer's finally had the power to apply their own lighting and transformation algorithm's to make their game completely unique looking. As time went on, shaders and the way they are managed has evolved to the point where there are multiple high level languages, most notably Microsoft's high-level shader language (HLSL), and are supported in other graphic API's such as OpenGL. Today, we've reached the point where graphics in new-age video game technology rely upon shaders to create real life quality graphics.

**Shader Implementation on the Xbox 360 and PlayStation 3**

Microsoft's Xbox 360, released in the first quarter of 2005, uses DirectX as its graphics API which runs through Microsoft's DirectX 9.0. In fact, some state it uses "DirectX 9 and then a little more" (Cross, 2005). DirectX was the first graphics API to allow programmers to use the programmable GPU which gives the Xbox and its graphics processor an advantage over the competition. Almost all personal computers (PC) running the Windows operating system utilize DirectX since its release. Thus, developing for the Xbox 360 will be easier for older PC developers to transfer to the Xbox system as well as allow Xbox developers to create equally impressive PC counterparts of their game. In late January 2007, Microsoft released the latest version of their popular Windows operating system, titled Windows Vista. The newest version of DirectX, DirectX 10, ships with the Vista operating system. DirectX 10 is a major improvement in the DirectX series of API's but at a cost. This recent version is only available for use with the Windows Vista platform and includes 2 separate versions of Direct3D (the portion of DirectX to produce 3D graphics) which are used to emulate DirectX 9.0 behaviors. DirectX API's have always been backwards compatible with previous versions but DirectX 10 was one of the only that wasn't widely available to DirectX 9.0 users. While Vista will take a few years to be widely accepted in PC's around the globe, games such as *Crysis* (Crytek, In Production) are already being built on DirectX 10 technology. There has been a lot of commotion due to the fact the team producing *Crysis* (Crytek, In Production) must write their game to support systems running DirectX 9.0 and DirectX 10, plus, since

shaders are based on graphics hardware, their game must support all different types of hardware configurations within each API, this could double the required amount of code and extend the development time of the game. Not only do the split versions of DirectX 9.0 and DirectX 10 on the PC create more problems during development, but the Xbox 360 only supports DirectX 9.0 (Klepek, 2006). Consoles are generally easier to develop for, considering the hardware is the same with every owner, unlike that of the PC hardware. Suppose you are developing a game for PC in DirectX 10, have all the different hardware areas covered, and want to bring a game to the Xbox 360, you'll need to spend even more development time bringing the game to DirectX 9.0 which could affect the graphics and feel of your game on the Xbox 360 console.

Sony's PlayStation 3 uses the PlayStation Graphics Library (PSGL) which is based on the Open Graphics Library for Embedded Systems (OpenGL ES) as its primary graphics API (Niizumi & Thorsen, 2005). OpenGL first began by a company named Silicon Graphics Inc. in 1992 but didn't support shaders until the release of OpenGL 2.0 in late 2004 (SIGGRAPH, 2004), four years after the inclusion of shaders in DirectX 8 which was released in late 2000. Since then, OpenGL has seen one update to the shader language and is currently being updated with a major update to OpenGL 3.0 which has yet to be released. In its 15 year span, OpenGL has had 3 large updates while DirectX sees updates to its development API every 2 months and updates to the consumer version every 1-4 years. Currently the lack of updates and support is the biggest hindrance to using the OpenGL API. While the support is lacking, OpenGL ES has an extremely streamlined design, allowing under-powered processors and hardware to perform the same computations as a high powered processor. OpenGL ES has been streamlined by the developers, removing much of the functionality from the original OpenGL API and providing a lightweight, yet powerful graphics API.

**Development Kits and Console Updates**

The current generation of gaming consoles has been developed with the internet and internet connections in mind. Often, updates are required for anyone using consoles connected to the internet and are applied to the console's operating system directly. These updates provide changes needed for certain games and even allow games to use specific hardware within a machine. Development kits are provided by the hardware manufacturers of game systems to console developers (for a price) allowing them to program games and applications for their hardware. This report has mentioned two different development kits, DirectX and OpenGL. As noted with OpenGL ES, the PlayStation 3 hardware, and PSGL, the company behind the hardware often builds a development kit between the hardware, the base graphics API, and the programmer. This way, the programmer doesn't write basic code in the base graphics API but writes specialized code defined in their custom development kit. Custom development kits also help regulate the games that are produced for the console by charging a high price and only selling the kit to game producing companies. Thus, the manufacturer of the console is assured unauthorized games are not produced for the system. Most, if not all game consoles, have their own development kit to create games specifically for that console. The use of a custom development kit plus the system software and updates via the internet provide a 'middle ground' where changes can be made to accommodate new features in the development kit and the system. As an example, if a new batch of games all use a custom function built to render a tree to the screen, the hardware manufacturer could apply updates to the system and the kits. Updates can only stretch so far in that if the developer of a game wants to use a feature unsupported by the hardware, even if the software was updated, they would have two choices, they could choose to use a lower version of the feature or use different hardware. Since the hardware within game consoles rarely ever changes they would be forced to use a lower version of the feature. Numerous years after a system is released and just prior to the release of its successor, the games produced usually have subpar graphics due to the fact the developers are making games for hardware and software that is not only out of date but only can be updated for so long.

The case arises where certain consumers may not use the system on the internet where the updates needed for games utilizing hardware can't be applied and games need to compensate for the fact consoles may not have the required software background. In any case, developers need to use caution when it comes to trying to improve the hardware and software on a system with updates. As the currently released consoles are the first to use the internet as a medium for changing a system, updating through the internet is a change to console trends that has just begun.

**Using Shaders on Game Consoles**

Programmable shaders are a relatively new technology in multiple ways and affect many components of a game system. Before programmable shaders were released in 2002, many games looked similar since they all used the same lighting algorithms built into the hardware. Custom vertex and pixel shaders finally allowed games to look unique. While shaders differ from company to company, they also differ in implementation depending on the hardware situation. In Microsoft's Xbox 360, the DirectX 9.0 graphics API provides a robust background but was introduced too early and missed the possibility of using a more powerful version with the release of DirectX 10. The PlayStation 3, released by Sony, uses PSGL (a modified version of OpenGL ES) which allows developers to use a lightweight graphics API, but the support and lack of shader background could hinder its usage. While both consoles implement graphics and shaders in different ways, both share a similarity since both use system software and development kits to provide expansion on their graphics capability software-wise. The drawback to both comes down to hardware, where new features may not be supported and aren't able to be used unless a lesser version exists. Programmable shaders are continuously evolving and new methods of using them and providing real-time, high quality graphics are always appearing. In order to improve the state of programmable shaders, their base systems including the hardware that runs them and the software that utilizes them is continuously expanded. Shaders are the connection between the software and the hardware, a change in one usually results in a change in the other. This change can require a consumer to purchase new hardware but provide some of the greatest visual graphical capabilities ever imagined.

References

Cross, J. (2005, July 27). *Microsoft Meltdown 2005: Vista and More--Extreme Tech Analysis: DirectX 10*. Retrieved October 9, 2007, from ExtremeTech: http://www.extremetech.com/article2/0,1697,1841223,00.asp

Crytek. (In Production). *Crysis*. [Microsoft Windows], United States of America: Electronic Arts.

Kesson, M. (2002-2004). *RenderMan - Shading Language Overview*. Retrieved October 9, 2007, from CG Reference & Tutorials: http://www.fundza.com/rman_shaders/overview/overview.html

Klepek, P. (2006, August 23). *Xbox 360 Not Upgradable to DirectX 10 news from 1UP.com*. Retrieved October 8, 2007, from 1UP.com: http://www.1up.com/do/newsStory?cId=3153097

Niizumi, H., & Thorsen, T. (2005, July 22). *PlayStation Meeting Report: PS3 hardware, dev kits, new games*. Retrieved October 8, 2007, from PlayStation 3 News at Gamespot: http://www.gamespot.com/news/2005/07/22/news_6129611.html

NVIDIA Corporation. (1995-2007). *Corporate Milestones*. Retrieved October 9, 2007, from NVIDIA.com: http://www.nvidia.com/page/corporate_timeline.html

SIGGRAPH. (2004, August 10). *SGI - Press Releases: OpenGL 2.0 Unleases the Power of Programmable Shaders*. Retrieved October 9, 2007, from SGI - Innovation for Results: http://www.sgi.com/company_info/newsroom/press_releases/2004/august/opengl.html