Advancement in computer graphics

Technology, Computer

Since the dawn period of video graphics, graphics developers had to depend on complete knowledge of the hardware they were working with. This knowledge came from lots of experimentation and electronics know-how. (Rickitt, Richard. 2000.)

Games, such as Pong or Asteroids, graphic designers had to have a good knowledge of what every chip on the game motherboard could provide them.

They were required to be fully aware of the amount of memory they might require like colors and sounds etc. some times they had to get through with the manufacturers to get assistance in designing chips that cater to their requirement.

Sound and graphics were primitive at best, but they worked. Since these early games were in fact a pioneering attempt, they also had to generate new user controls to work with. (Ritter, R. (2002)

Early DOS games were dependent on knowledge of machine and assembly languages to enter the OS’s lower levels. With the passage of time, more intelligible languages were introduces, such as BASIC and LOGO; though, they didn't have the potential of assembly language.

Next innovation in language world was Pascal and C. With C, game developers could design advanced code routines devoid of having to work with tons of assembly code, therefore making graphics easier. Very soon C language became the preferred language of graphic designers. (Timo Aila, 2003)

With these language changes, one difficulty remained. To develop high-quality games, developers still required a thorough knowledge of the

hardware they were running with. There was no need to know every chip's capabilities, however they had to know things such as video cards, sound cards, and input devices. They also had to cope with memory constraints, which were turning out to be a hindrance to graphic designs.

Along Came Windows

Advancement in computer operating system it turned out to be very easy to maintain specific standards in the computer industry. These standards consisted hardware interfaces, video memory standards, sound standards, and CPU memory specifications. Yet, there was still a lot of conflict within the industry, and lots of hardware turned out to be unsatisfactory. Consequently, numerous games developed in Windows faced a lot of problems.

Microsoft understood this problem early on and emphasized its efforts on improving itsenvironmentand working to improve performance for the parts of the operating system that cater to game and multimedia requirements.

These efforts were obvious in Windows 95, one of the first operating systems that produced reliable hardware-software interaction a reality. Since all hardware companies had to meet a comparable standard and Microsoft controlled how its operating system responded to the hardware.

In the DOS days, you had to build your own drivers for every type of video card, but with Windows, you had to develop to only one standard. In spite of spending time producing and understanding numerous hardware specifications, designers relied on Microsoft to do all that for them and wrap up it for their use in the OS.

There was one flaw with all of this. Windows was initially designed for homes and offices not for professional multimedia development. It lacked little necessary items that had become trademark in the multimedia market mostly graphics speed, DOS was better then windows.

It was the same hardware, but there were many more layers between the developers' code and the hardware they wanted to control. The effect was that a lot of developers deserted the idea of Windows game development and depended on the Windows capability to disburse to DOS.

After Microsoft understand that game developers agreed with its idea but reluctant to give up the performance they had become used to in DOS, Microsoft started exploring ways to facilitate developers to reach that hardware layer. Some of the initial attempts involved WinG, WinToon, and OpenGL.

Wings become popular at the time of Windows 3. 1 and Windows for WorkGroups. WinG was the first try at a complete graphics API (Application Programming Interface) library to eradicate the limitations of the Windows API.

WinG rendered high-performance graphics system that could be accessed through Windows game development world to that community who worked completely in DOS.

WinToon worked as first API system for Windows 95 that supplied the potential of easy animation playback. API also provided better performance as compared to existing Windows 95 APIs. Thus it aided to improve the

multimedia playback capabilities of animation programs for instance kids' games and educational programs.

OpenGL is a powerful set of APIs that let the developers to cut through the Windows red tape to integrate some of the hardware procedures of 2D and 3D graphics. Unexpectedly, OpenGL was not the product of Microsoft; instead it was introduced by Silicon Graphics, Inc.

These technological advancements were an enormous boost, but eventually turn out to be short. As the game market boosted and the requirement for more games on the Windows 95 platform rose, Microsoft developers went back to work.

What they came up with was the Windows 95 Game Software Developer's Kit (SDK), which introduced the first version of DirectX. DirectX was actually restricted for the computer game industry but has evolved to embrace other areas, for instance, improvement in the arcade business and handheld Windows CE devices.

As DirectX has improved, Microsoft rendered network support, in addiction with force feedback support, and 3D graphics capabilities. When ever improvements are required, the DirectX team has gone on to improve the performance of this amazing software library. (John Owens. 1983.)

Video graphics games fueling hardware growth and hardware in turn fueling games will continue on. Thus the future of video games will most likely be tied to the future of graphics hardware and tactile hardware. With the introduction of the DirectX 8. 0 API and same type functionality in OpenGL, GPUs increased programmable shading to their functions.

The entire pixel could now be achieved by a short program that could involve additional image textures as inputs, and every geometric vertex could similarly be processed by a short program before it was shown on the screen. nVidia introduced chip capable of programmable shading, the GeForce 3 (widely known as NV20).

By October 2002, with the introduction of the ATI Radeon 9700 (also known as R300), the world's first Direct3D 9. 0 accelerator, pixel and vertex shades could implement looping and lengthy floating point math, and in general were soon becoming as flexible as CPUs, and orders of magnitude faster for image-array operations. (Dan McCabe 1998)

Nowadays, parallel Graphics Processing Unit have started computable graphic inroads against the operating system, and a subfield of study, dubbed GPGPU for General Purpose Computing on GPU has paved its way into areas as diverse as oil exploration, scientific image processing, and even stock options pricing determination.

There is increased pressure on GPU manufacturers to update hardware design, generally emphasizing on adding more flexibility to the programming model

Bibliography

1. Bartle, Richard A, 2003, Designing Virtual Worlds , 1st Edition, New Riders

   Dan McCabe and John Brothers. DirectX6 Texture Map Compression. Game Developer Magazine, August 1998

2. John Owens. EEC 277: Graphics Architecture. Technical report, 2005 Lance Williams. Pyramidal Parametrics. In Computer Graphics (Proceedings of ACM SIGGRAPH 83), pages 1–11. ACM, July 1983.
3. Ritter, R. (2002). The Oxford Style Manual. Oxford University Press. Rickitt, Richard. Special Effects: The History and Technique. New York: Billboard Books, 2000.
4. Timo Aila, Ville Miettinen, and Petri Nordlund. Delay Streams for Graphics Hardware. ACM Transactions on Graphics,, 2003