GDC 2000

Course: Advanced OpenGL Game Development

Brief Summary of All Things OpenGL and Linux

Brad Grantham, Senior Software Engineer, VA Linux Systems
grantham@valinux.com

Introduction

Linux support for 3D hardware acceleration has grown from a ghostly rumor of a few cards a year ago to support for almost all of the 3D architectures available for the PC. Quake III Arena, the most prominent Linux-compatible shooter game available, is comfortably playable on Linux on at least half a dozen cards including 3dfx Voodoo3, NVIDIA TNT2, ATI Rage Pro, and Matrox G400.

Support for hardware 3D acceleration is available through two APIs: OpenGL and Mesa. OpenGL, from Silicon Graphics, is implemented in at least five platforms that will be available by the end of March 2000: Metro-X's Extreme 3D, Xi Graphics' 3D Accelerated-X, NVIDIA's architectural driver, Hewlett-Packard's FX-6 Graphics, and Silicon Graphics' OpenGL on the Visual Workstation. Mesa, a free library which is compatible with OpenGL, along with Utah-GLX accelerates several cards as of March and will continue to mature in the months to come. Both of these APIs use GLX, an extension to the X Window System protocol, to provide OpenGL rendering in an X window.

Mesa

Mesa (http://mesa3d.sourceforge.net/), first released by Brian Paul in 1995, exports the OpenGL 1.2 functions and uses the same enumerants. This means it is both source and binary compatible with OpenGL 1.2. Mesa was originally (and still is primarily) a software transformation and rasterization engine. Its performance is low for large or textured polygons, but it still has functioned as the only OpenGL "implementation" available for free on several platforms, and many OpenGL programmers learned to use OpenGL by compiling programs with Mesa on Linux.

Although an official software-rendering OpenGL implementation has been available in the past from several vendors including Evans and Sutherland, Mesa has been widely accepted as a 3D API under Linux because if the benefits given to it by the open source development model, including fast release cycle times, a large and experienced development team (anyone who cares to contribute) fast bug turnarounds, and wide platform support.

Mesa is available as source code, both in gzipped tar files and through the CVS source control system. Several versions are also available as binaries (precompiled) as gzipped tar files and as RPM's (Red Hat's Package Manager format).

Access to GLX is provided by a series of functions which emulate the GLX API. The "GLX" functions in Mesa do not actually package up OpenGL commands in GLX protocol packets, but rather attempt to do a very good job of pretending. Utah-GLX, discussed below, does implement the GLX protocol. Mesa simply provides libGLU.so and libGL.so shared libraries, and an OpenGL application program may not know that it is not actually running on Mesa with emulated GLX.

Mesa also provides a free implementation of the GLU (OpenGL Utility) library through which applications can tessellate concave or self-intersecting polygons and NURBs into triangles, in addition to other functions which do not directly involve the OpenGL state machine.

Mesa is useful as a learning tool for OpenGL beginners, and provides a very rich feature set, including

- the OpenGL 1.2 API
- stencil buffers
- accumulation buffers
- the GL_ARB_multitexture extension for multiple textures
- extended blending functions through GL_EXT_blend_color and GL_EXT_blend_subtract
- ... many other features useful for testing and education.

Mesa is not a licensed OpenGL implementation. There is no central "ARB" to manage the certification process or extensions; Brian Paul fills that role, and he is not paid for individual Mesa binaries. You may find Mesa has bugs which you would not expect in an OpenGL implementation under Windows 98, but I have found the reverse is sometimes the case. Mesa passes most of the OpenGL conformance test suite and I have found it a very useful library for writing and testing my sample programs for the SIGGRAPH Course "Advanced Graphics Programming Techniques with OpenGL". It actually has helped me find incorrect use of OpenGL in some programs of my own, especially because I can read the source for Mesa hand in hand with the OpenGL specification, and this is why I believe it is a good learning tool.

Utah-GLX

The "Utah-GLX" project (http://utah-glx.sourceforge.net/) provides hardware accelerated Mesa on five architectures: the Riva series from NVIDIA, MGA-G200 and G400 from Matrox , ATI’s Rage Pro, Intel’s embeddded i810, and S3’s Virge DX. Utah-GLX accelerates the most common OpenGL functions, and offers a few extensions through Mesa.

Several prominent engineers have contributed to Utah-GLX, including Terrence Ripperda of NVIDIA, Keith Whitwell of Precision Insight, Stephen Crowley of the Debian project, and John Carmack of Doom and Quake fame from Id Software.

Utah-GLX provides "real" GLX functionality for Mesa, and the source package compiles into libGL.so, libGLU.so, and glx.so. "glx.so" is a loadable module for XFree86 3.3.x (where x >= 5) which provides Mesa inside the X server. The libGL.so from this package has two functions. If "direct rendering" is disabled or is unavailable because the client and server are on two separate machines, each OpenGL-compatible function call packages up the OpenGL command and parameters and sends it across either a network socket or a UNIX pipe to the X server, where glx.so decodes the command and parameters and calls Mesa to process the request. This means that any machine with GLX can render to a Linux machine with Utah-GLX in the server, and any Linux machine with Utah-GLX libGL.so can

render remotely to another GLX-capable system.

The glx.so also has hardware-accelerated Mesa support compiled in as driver modules for each of the architectures mentioned above. These drivers work in conjunction with the 2D support already in the X server. The most commonly-used OpenGL features are accelerated: singly-textured MIPmapped polygons, fast glClear, and fast glXSwapBuffers using 16-bit color buffers and 16-bit depth buffer. On some of the architectures which support it, multitexture, stencil buffers, 32-bit color, and 24-bit depth are accelerated as well. The most full-featured Utah-GLX platform at this time is the Matrox G400, which includes accelerated triangle setup, multitexture, and 32-bit rendering.

Finally, Utah-GLX supports "direct rendering". The libGL.so library can negotiate with glx.so when the client and server are on the same physical hardware to enable direct hardware access for the OpenGL client. This removes the overhead of packaging up rendering requests in the GLX protocol and allows Mesa performance to approach and, in some cases, exceed the performance of OpenGL under Windows.

Utah-GLX is available in the same ways as Mesa; through source code in archives or CVS, or through pre-compiled binaries. Daily snapshots are available through a link on the web page.

Development of Utah-GLX appears to be driven by the desire of many hackers to play Quake III Arena on their machines which would otherwise only run Quake III under Windows 98. This means that the performance is good but feature coverage may be spotty; some OpenGL state combinations may unexpectedly force software rendering even though the hardware supports the configuration, and some functions may not work at all, mostly because these functions are not used by Quake or other widely available applications. The team tries very hard to provide correct rendering for every state combination, though, and I have been very excited to run OpenGL programs accelerated on my laptop with the ATI Rage Pro LT chipset, including Quake III.

I expect Utah-GLX to continue to improve for the next few months, and support for one or two more chipsets may be added. However, development of hardware-accelerated Mesa under Linux and other free UNIX-like operating systems is steadily heading towards support of the Direct Rendering Infrastructure from Precision Insight in XFree86 4.0, discussed later.

Game developers may target Utah-GLX because of its support for several popular commodity cards and because its source code is fully free and available. Developers requiring special functionality may even contribute a new OpenGL extension to Mesa and GLX, if appropriate. Many of the newest Linux distributions, such as Red Hat 6.2, include a subset of the Utah-GLX hardware accelerated drivers, including the Matrox G400.

Xi Graphics 3D Accelerated-X

Xi Graphics, Inc. (http://www.xig.com/), sells an X server including OpenGL client libraries and accelerated rendering as part of the "3D Accelerated-X" package. They support at least eight different architectures at the time of writing in an "entertainment" version for commodity cards and a "professional" version which provides workstation features such as overlays and accelerated antialiased lines.

Although Accelerated-X is not Open Source, it is an OpenGL 1.1 implementation which supports accelerated textured polygons, fast clear, and fast swap, and provides acceleration for cards not likely to

be supported through Mesa either because of lack of availability or lack of chipset documentation. Many 3D companies provide their chipset specification to Xi Graphics in order to receive OpenGL support on Linux such as Evans and Sutherland, SiS, and Number Nine.

Developers looking for a fully compliant OpenGL implementation and a vendor to provide support may contact Xi Graphics for more information.

Although the "entertainment" package is an appropriate target for game developers wishing to ship a Linux version, it may also be useful to develop game content on a higher performance board supported by 3D Accelerated-X such as the 3Dlabs GVX1, because of it's high geometry performance and overlay support. A trickle of applications last year is growing into a flood of content development packages available under Linux, and many of these packages are ported using 3D Accelerated-X.

Metrolink's Extreme 3D

An early developer release is available of MetroLink's (http://www.metrolink.com/) "Extreme 3D" another X server providing OpenGL acceleration for the Evans and Sutherland RealImage architecture and 3Dlabs Permedia 2- and 3-based boards.

Like the Xi Graphics product, this X server is an OpenGL implementation and so may serve some users' needs better than Mesa, but because it is not available to the public, it is more interesting as a development aid for those content providers with 3Dlabs GMX boards.

NVIDIA's Riva Architectural Driver

NVIDIA is planning to release a binary-only OpenGL client library and XFree86 4.0 module by the middle of 2000. This driver shares the OpenGL implementation with NVIDIA's Windows 98 and NT OpenGL driver, and so will provide all the same OpenGL extensions and features and roughly the same performance as the Windows driver.

Game developers who have targeted some of the unique features of the GeForce 256 will find that their graphics subsystem will run just as well on Linux, minus the difference in operating system services between Linux and Windows and other porting concerns.

An open source version of a Mesa and GLX-based driver for the Riva 128 and TNT-series is available from NVIDIA's web site at http://www.nvidia.com/. This driver and X server provides 32-bit rendering including stencil buffers and geometry performance tweaks but does not support direct rendering and does not expose the full feature set of the TNT or any of the new features of the GeForce 256.

Hewlett-Packard FX-6

HP has committed to shipping OpenGL support for FX-6, a high-performance workstation-class OpenGL accelerator for HP Windows NT and UNIX machines. This implementation is based on the "Direct Rendering Infrastructure" from Precision Insight and boasts 32-bit rendering with up to 12 million polygons per second.

This is less a gaming platform and more a development station because of the cost of the machine and

graphics option.

Silicon Graphics Visual Workstation

SGI has demonstrated hardware-accelerated OpenGL on the 320 and 540 Visual Workstations, but has not committed to shipping this implementation.

Future Directions: Direct Rendering Infrastructure

Precision Insight, Inc., has designed and implemented a different form of direct rendering. Although similar to Utah-GLX in that there is an OpenGL client library which communicates with a server-side rendering module, it also provides a framework for locking and DMA through a kernel device, the "Direct Rendering Module". More information is currently available through http://www.precisioninsight.com/ and http://sourceforge.net/ under "Direct Rendering Infrastructure".

This architecture is already in place in prereleases of XFree86 4.0 including the most recent, XFree86 3.9.18. Support was demonstrated in a prototype implementation in Autumn 1999 for 3Dlabs Gamma-based OpenGL accelerator boards, and later for 3dfx Voodoo 3 and Banshee boards. Since then, the DRI has incorporated preliminary code for the Matrox G400 and for Intel’s i810 embedded graphics.

Most people believe this is the direction that commodity cards will take. The DRI provides a straightforward, complete, medium-performance architecture for Mesa hardware acceleration on most architectures.

Support - Libraries and Toolkits

The GLUT toolkit, popular among students and also among professionals for rapidly developing portable OpenGL applications, is available under Linux. GLUT provides functions for opening and manipulating windows, receiving mouse and keyboard input events, and drawing simple primitives. The source code is available from http://reality.sgi.com/opengl/glut3/glut3.html

PLIB (http://plib.sourceforge.net/), a portable multimedia library, is a collection of simple libraries providing support for audio, joystick input, simple GUI widgets, text, simple geometry, and scene graph primitives. The components of PLIB were originally collected as the basis for the game "Tux the Penguin: A Quest for Herring", but many new applications use PLIB to ease portability concerns including the open source FlightGear flight simulator and the open source polygonal model editor PrettyPoly.

SDL, the Simple DirectMedia Layer is another library which provides fast access to video and audio targeted at games. The Linux version of "Civilization: Call to Power" uses SDL to play MPEG movies and sound.

Driver support for many joystick and joystick-like input devices is included in recent releases of the Linux kernel. More information can be found at http://atrey.karlin.mff.cuni.cz/~vojtech/joystick/

Many other tools for gaming and interactive applications for Linux are available online. Some sites of interest include:

- Linux Games at http://www.linuxgames.com/
- Linux Game Development Center at http://sunsite.auc.dk/lgdc/
- Linux3D at http://www.linux3d.org/

Soothsaying (Predictions)

Support for OpenGL applications really has exploded in the last year. Applications developers from CAD to DCC to Vis Sim are eyeing the familiar UNIX environment that Linux provides coupled with the Intel/PC platform as a low-cost replacement for traditional UNIX workstations.

I expect to see nearly all PC chipsets supported by robust, high-performance, OpenGL-compatible solutions by the end of this year, either through vendor-specific drivers (a la the NVIDIA Architectural Driver), a third party vendor (e.g. Xi Graphics), or the efforts of the open source community in Utah-GLX and the DRI.

Linux is already being proven as a viable gaming platform by titles such as **Quake III/Arena**, **Tycoon II**, **Heretic II**, among others, which I can drive to Fry's (a California electronics department store) and buy today. I believe Linux will prove it's capabilities by the end of the year through the development of strong alternatives to the DirectX APIs that are already available under Linux.

Addendum

An HTML version of this whitepaper may be found (with any additions and corrections) and my slide set for the GDC 2000 course "Advanced OpenGL Game Development" may be found at http://www.plunk.org/opengl/.