

OpenGL and the Linux desktop

LIFE IN 3D

New technologies will change the way you view the objects on your Linux desktop. **BY JOE CASAD**

The X graphics system has been at the heart of the Unix GUI desktop since 1984. Of course, its horizons expanded slowly. In the early years, no one knew they even needed a graphics subsystem, and if anyone did venture out to try X with one of the few applications that supported it, they needed to be ready for late nights of tinkering.

But the X system kept getting better, and the X protocol, with its surrounding technologies, served a key role in the evolution of Linux. When graphic desktop systems came into vogue with the appearance of MacOS and a Mac clone known as Windows, Linux could build its own equivalent graphic desktop upon the foundation of X. Today's Linux graphic desktops are every bit as powerful and far more flexible than the Apple and Windows equivalents, but the simple fact is, developers and hardware vendors – for all platforms – have gotten a bit restless with the ordinary, garden variety X.

One of the reasons for this restlessness is that X is primarily designed for 2D graphics, and the real world is 3D. Of course, when X first appeared, 2D graphics techniques were very amazing and wonderful compared to the punch card or the command prompt, but users today want more, and the hardware of today is fully capable of giving them more. Because of the emphasis on fast action and realism, vendors want to squeeze the best possible performance from the hardware, and the

antiquated design of the X system is not, on its own, flexible enough to support this kind of performance.

In the Linux world, wherever you find a very big and intriguing question, you can bet that someone out there is working on the answer. A solution to the worries of the restless graphics developers is converging now around the emergence of two very important developments:

- Extensions to X that offer increased flexibility and *shortcuts* around some of the more sluggish parts of the X system
- The growth and widespread acceptance of the OpenGL graphics programming interface

OpenGL desktop integration is already underway in the Windows and Mac worlds, and two systems known as Xgl and AIGLX now offer alternative visions for how to merge OpenGL with the Linux desktop.

OpenGL

OpenGL is a popular-graphics library

specification originally created by Silicon Graphics. The OpenGL specification provides a standard for vendors to create graphics programming interfaces. The purpose of OpenGL is to provide a single interface that gives graphics programs a standard way of talking to graphics hardware. One of the reasons why OpenGL is necessary is that graphics hardware vendors are busy with their own experiments to deliver richer graphics and better performance to their customers. One of the techniques used by graphics hardware vendors is known as 3D acceleration. 3D acceleration lets the graphics hardware assume a larger share of the responsibility for the details of generating 3D images. OpenGL lets programmers write code that anticipates and capitalizes on the benefits of 3D acceleration.

OpenGL with X

One free (but unofficial) implementation of OpenGL for X-based platforms is Mesa [3]. In theory, Mesa and other

COVER STORY

Xgl and Compiz	24
AIGLX	28
X11R7.....	32



Figure 1: Projecting a movie on the edge of a cube may not be practical, but it displays some of the power that OpenGL brings to the Linux desktop.

alternative OpenGL APIs let Linux programs access OpenGL functions. However, the X system was not designed for futuristic techniques like OpenGL. In the early attempts to integrate OpenGL, 3D-based data simply had to be translated into ordinary calls to the X server, which required additional translation steps that took the edge off performance. Since those early attempts, programmers have been looking for ways to work with X but circumvent some of its limitations.

One popular technique for integrating accelerated 3D graphics with X makes use of the X Composite extension. The Composite extension essentially draws windows into memory, rather than directly onto the screen. This reduces some translation and allows for fuller and more efficient expression of the potential of OpenGL. It also provides for more efficient use of special effects, such as magnification, translucence, and other nifty manipulations.

Now the Desktop

Game programmers and other graphics specialists have pioneered several impressive techniques for extending and expanding X to accommodate OpenGL, and eventually, developers began to ask whether the desktop itself could take on OpenGL capabilities.

In Linux and other X-based systems, the task of generating and organizing the window environment is performed by a

window manager. The call for integrating OpenGL with the desktop began to crystallize around the task of creating a *compositing window manager*, or a *composite manager*, a window manager that makes use of the X Composite extension. Xgl and AIGLX bring the glories of OpenGL to the Linux desktop by supporting the integration of a composite manager. As you'll learn in this issue, Compiz is a composite manager designed for Xgl. AIGLX developers have experimented with the Luminocity composite manager, and Red Hat is currently working on implementing compositing features into their Metacity window manager.

Strategies

The appearance of Xgl and AIGLX have followed a familiar pattern for open source software: controversy, followed by standoff, followed by a remarkable level of cooperation. Novell announced they were working on a technology called Xgl that would deliver impressive OpenGL tricks to a working X server, but the whole project went silent for several months while Novell worked behind the scenes. This level of secrecy was upsetting to some in the community. Red Hat then stepped up to say they were developing their own OpenGL desktop technology, which they called AIGLX. Red Hat undoubtedly had their own worries about accepting a specification they

didn't participate in creating. But they also expressed concerns about parts of Xgl, and they argued that their approach would have some technical benefits.

Since then, the community has vigorously debated the technical benefits of these two solutions, and they have also debated which was truer to the open source spirit. As of now, both Xgl and AIGLX are available to developers. The two projects actually share some of their code and are proceeding in parallel with much interest from the community.

In the next few years, Xgl and AIGLX may redefine the way we think about the Linux desktop. Features like wobbly windows and the famous desk-

top cube may seem like gimmicks at a glance, but they have a subtle effect on the way the user perceives the user space, making the objects on the virtual desktop seem more like objects on a real desktop. Developers have only begun to explore the benefits of features such as translucent windows and the many other effects that are only now making their way into the public eye.

For an inside look at these desktop technologies of tomorrow, we invited Novell's Mathias Hopf and Red Hat's Kevin E. Martin to talk about Xgl and AIGLX. Read on for more on these innovative technologies. And to round out this month's cover story on the Future of Linux Graphics, we bring you a report on the latest release of X11, the dominant version of the X system, which offers a new modular design and many other important improvements. We hope you enjoy this month's cover story. ■

INFO

- [1] X.org Foundation: <http://www.x.org/>
- [2] An introduction to the X window system: http://en.wikipedia.org/wiki/X_Window
- [3] Mesa project: <http://www.mesa3d.org/>
- [4] "Accelerated X Flame Wars!...Maybe Not," by Edward Macnaghten, *Free Software Magazine*, 2/27/06: http://www.freesoftwaremagazine.com/articles/accelerated_x