Get Your Graphics On: OpenGL Advances with the Times

Alexander Wolfe, Science Writer

OpenGL, the decade-old mother of all graphics application programming interfaces (APIs), is getting two significant updates to bring it into the 21st century.

Accurately billed by its supporters as the premier environment for developing portable, interactive 2D and 3D graphics applications, OpenGL comes equipped with a broad set of rendering, texture mapping, special effects, and visualization functions.¹ Faced with increased competition on the desktop from Microsoft's DirectX and D3D offerings, however, OpenGL is getting a bit long in the tooth.

Accordingly, the OpenGL Architecture Review Board (the independent consortium chartered in 1992 to guide the future of the API, which was originally developed and is still owned by Silicon Graphics²) has pushed ahead on the desktop and workstation front to provide a programmable implementation dubbed the OpenGL Shading Language.

Meanwhile, a spin-off organization called Khronos³ has been formed to extend the OpenGL brand into the emerging world of downsized devices such as cellphones and handheld gaming devices. There, it's taking the form of a new API called OpenGL ES, for "embedded systems."

The recent Consumer Electronics Show (CES), held this past January in Las Vegas, constituted the public coming-out party for Khronos. The OpenGL ES 1.0 spec itself was actually disclosed in July 2003 at Siggraph in San Diego.⁴ But CES saw the release of the first products to execute apps taking advantage of the spec, thus moving it from vapor status to a real, 3D-embedded engine.

Initially, ES support is coming in the form of software engines. Texas Instruments⁵ and Symbian⁶ have developed the 3D Graphics Library plug-in based on OpenGL ES 1.0 for Symbian OS-based mobile phones equipped with TI's OMAP digital signal processors. In addition, Fathammer has integrated OpenGL ES into its X-Forge game software development kit (SDK),⁷ which is used to develop apps for Nokia's N-Gage handheld wireless game unit.⁸

In keeping with its embedded design goal, these ES

Enhancements to

THE OLD GRAPHICS

API WILL GIVE IT NEW LIFE.

engineers are keeping a tight footprint—on the order of 100 kilobytes resident on the target platform.

Such software will be followed later this year by faster, dedicated integrated circuits and cores (the latter being intellectual property ready to be transformed into working silicon) now under development at ATI, NeoMagic, Takumi, and 3Dlabs itself.

OpenGL ES's proponents see it moving rapidly into a vacuum in the embedded space. To date, Java-compliant 3D graphics have come in the form of the Java 3D API. Many developers, however, believe Java 3D is at too low a level. An initial indication that ES may see service in the Java arena comes via the news that Sun Microsystems and Silicon Graphics plan to cooperate in developing bindings from the mobile J2ME runtime environment⁹ into OpenGL ES.

For its part, Microsoft is developing a competitive offering in the form of D3D Mobile. That API is an embedded version of DirectX; however, it's expected that D3D Mobile will be very closely tied to Microsoft's embedded operating system offerings such as Windows CE and Smartphone. As such, it could be a good while before it migrates to Java-based devices of Symbian cellphones.

If this seems like a swipe at Microsoft, consider at the same time that Redmond may be in a better position when it comes to tools support. Microsoft can easily fold D3D support into Visual Studio and the .NET Framework. In contrast, the OpenGL Architectural Review Board won't be a one-stop shop in charge of building support for OpenGL ES into any of the popular IDEs (integrated development environments). Rather, such support will come in multiple plug-ins from third parties, such as the vendors offering the various ES software and hardware engines.

PROGRAMMABILITY COMES INTO VOGUE

While 2004 marks the infancy of good 3D graphics on compute-constrained handhelds, it will also see the move

to programmability on desktops, which as a category finally has enough graphics-engine heft to merit such a transition.

"In the last 18 months there's been a lot of activity on the desktop side toward making the [graphics] hardware programmable,"¹⁰ explains Neil Trevett, who has expertise in desktop graphics as a senior vice president at chip maker 3Dlabs, and also is a key contributor on the embedded side as chairman of the OpenGL ES working group. "A graphics chip is increasingly becoming like a CPU: It's multithreaded, it has virtual memory, it has intertask security, and it's programmable in a C-like language. The only difference is that graphics ICs are very parallel (as opposed to the main microprocessor), which is required so they can execute graphics more efficiently."

Enter the OpenGL Shading Language, also known as OpenGL 2.0 (see figure 1).¹¹ Due to hit the streets in a few months, the release will bring high-level shading programmability to OpenGL. It will expose a very C-like API and provide software developers with access to a wealth of graphics operations.

Standard OpenGL offers a fixed set of functionality. This limitation has been forced by the fact that lots of

compute power is required just to get polygons up on the screen. Now that graphics boards and ICs are powerful enough to off-load 3D processing from the main microprocessor (aka Pentium), however, OpenGL 2.0 is able to let programmers do things more flexibly without fear of bogging down the computer.

Accordingly, 2.0 will provide programmable alternatives to some of the fixed functions currently available. For example, programmers will have the ability to explicitly get into the nuts and bolts of things like vertex processing and fragment processing. This will allow them to revamp things according to their own needs or to tune performance for specific hardware configurations.

OpenGL 2.0 implements what's called a direct-compile model—that is, 2.0-compliant drivers will ship with built-in compilers. This means that a given graphics hardware/driver combination will take its OpenGL source code and compile it directly into machine code that runs on the graphics hardware. (The compilers themselves will be supplied by the individual graphics chip vendors.)

"Over the next year or two, I think you're going to see a whole range of applications that use your graphics board as a supercomputer,"¹² Trevett says enthusiastically.

OpenGL 2.0 Logical Diagram

Source: 3Dlabs and OpenGL Architectural Review Board

FIG 1

REALTIME RENDERING ON THE HORIZON

The turn toward programmability comes at a time when professional graphics developers are poised to make major advances. Consider that, currently, animated films are produced using large render farms—huge collections of

Because so much compute
power is required,
render farms usually run offline.

workstations that crunch polygons to create a movie's individual frames. Because so much compute power is required, render farms usually run offline, taking a day's work and turning it into frames overnight so it's ready when the animators return in the morning.

Recently, says Trevett, there's been interest from the photorealistic community in using newfound graphics programmability for doing movies such as "Toy Story" (2001) in realtime. "That's the dream everyone is going for,"¹³ he says. The goal is to take industrial-strength software—like the Renderman package that "Toy Story" producer Pixar uses—and accelerate it on graphics chips via programmability.

The result would be that far fewer workstations would be needed to render a movie, and the cost-per-frame of rendering would come down significantly.

Longer term, Trevett doesn't think that graphics software will stand still. "We haven't reached a stable state yet, either in the algorithms or the hardware," he says. "I think in 20 years we'll look back and be amazed at how primitive things were. The software community is still doing tremendously innovative work. This speaks to the fundamental market driver behind 3D graphics."¹⁴

Most immediately, software developers can access the OpenGL ES adopters' package at the Khronos Web site.¹⁵ An OpenGL ES code sample can be viewed on the ATI Technologies Web site.¹⁶ The sample is an OpenGL ES application written for Windows CE 4.2 that shows the initialization of OpenGL ES, the creation of some contexts and surfaces, and the use of some common profiling functions.

The specification for the OpenGL Shading Language is available on the OpenGL Web site.¹⁷ Some often-heated discussions of its features can be found in the OpenGL group's forums.¹⁸ Q

REFERENCES

1. OpenGL: see <http://www.opengl.org/>.
2. Silicon Graphics (SGI): see <http://www.sgi.com/>.
3. The Khronos Group: see <http://www.khronos.org/>.
4. Siggraph 2003: see <http://www.siggraph.org/s2003/>.
5. Texas Instruments: see <http://focus.ti.com/docs/pr/pressrelease.jhtml?preId=sc03095&prSection=overview>.
6. Symbian: see <http://www.symbian.com/news/2003/pr030429c.html>.
7. Fathammer: see <http://www.fathammer.com/>.
8. Nokia N-Gage: see <http://www.n-gage.com/R1/en/home.html>.
9. Wolfe, A. "Java is jumpin', this time for real," *ACM Queue* 1,10 (Feb. 2004), 17-19.
10. Neil Trevett quotes come from a phone conversation with the author January 2, 2004.
11. For presentations describing OpenGL 2.0, visit the 3Dlabs Web site: <http://www.3dlabs.com/support/developer/ogl2/presentations/index.htm>.
12. See reference 9.
13. See reference 9.
14. See reference 9.
15. To download the OpenGL ES adopters' package, visit the Khronos Website: http://www.khronos.org/opengles/become_opengles_adopter.html.
16. To view sample code, visit the ATI Web site: <http://www.ati.com/developer/indexsc.html#opengl>.
17. The specifications for the OpenGL Shading Language are available on the OpenGL Website: <http://www.opengl.org/documentation/oglsl/ShaderSpecV1.051.pdf>.
18. OpenGL group forums: http://www.opengl.org/discussion_boards/cgi_directory/forumdisplay.cgi?action=topics&number=7.

LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or www.acmqueue.com/forums

ALEXANDER WOLFE received his electrical engineering degree from Cooper Union in New York City. A science writer based in Forest Hills, New York, he has contributed to *IEEE Spectrum*, *EE Times*, *Embedded Systems Programming*, and *Byte.com*.

© 2004 ACM 1542-7730/04/0300 \$5.00