The Industry's Foundation for High Performance Graphics

from games to virtual reality, mobile phones to supercomputers

OpenGL Overview

OpenGL - The Industry's Foundation for High Performance Graphics

- About OpenGL
- The OpenGL Programming Visualization Programming Pipeline
- Available Everywhere
- Architected for Flexibility and Differentiation: Extensions
- API Hierarchy
- The Foundation for Advanced APIs
- Governance
- Continued Innovation through Extensions
- Licensing
- OpenGL Applications & Games

Most Widely Adopted Graphics Standard

OpenGL is the premier environment for developing portable, interactive 2D and 3D graphics applications. Since its introduction in 1992, OpenGL has become the industry's most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms. OpenGL fosters innovation and speeds application development by incorporating a broad set of rendering, texture mapping, special effects, and other powerful visualization functions. Developers can leverage the power of OpenGL across all popular desktop and workstation platforms, ensuring wide application deployment.

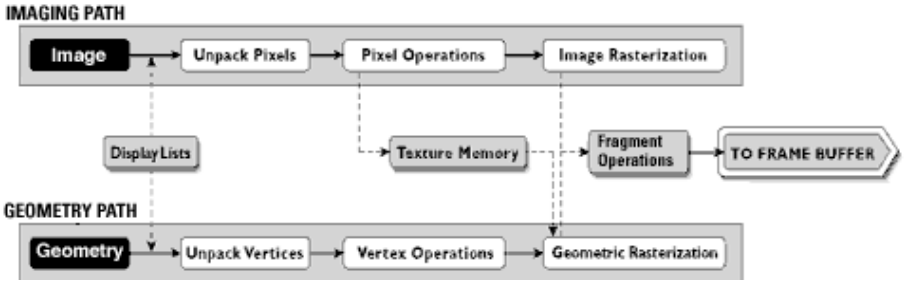
High Visual Quality and Performance

Any visual computing application requiring maximum performance-from 3D animation to CAD to visual simulation-can exploit high-quality, high-performance OpenGL capabilities. These capabilities allow developers in diverse markets such as broadcasting, CAD/CAM/CAE, entertainment, medical imaging, and virtual reality to produce and display incredibly compelling 2D and 3D graphics.

Developer-Driven Advantages

- **Industry standard**
  An independent consortium, the OpenGL Architecture Review Board, guides the OpenGL specification. With broad industry support, OpenGL is the only truly open, vendor-neutral, multiplatform graphics standard.
- **Stable**
  OpenGL implementations have been available for more than seven years on a wide variety of platforms. Additions to the specification are well controlled, and proposed updates are announced in time for developers to adopt changes. Backward compatibility requirements ensure that existing applications do not become obsolete.
- **Reliable and portable**
  All OpenGL applications produce consistent visual display results on any OpenGL API-compliant hardware, regardless of operating system or windowing system.
- **Evolving**
  Because of its thorough and forward-looking design, OpenGL allows new hardware innovations to be accessible through the API via the OpenGL extension mechanism. In this way, innovations appear in the API in a timely fashion, letting application developers and hardware vendors incorporate new features into their normal product release cycles.
- **Scalable**
  OpenGL API-based applications can run on systems ranging from consumer electronics to PCs, workstations, and supercomputers. As a result, applications can scale to any class of machine that the developer chooses to target.
- **Easy to use**
  OpenGL is well structured with an intuitive design and logical commands. Efficient OpenGL routines typically result in applications with fewer lines of code than those that make up programs generated using other graphics libraries or packages. In addition, OpenGL drivers encapsulate information about the underlying hardware, freeing the application developer from having to design for specific hardware features.
- **Well-documented**
  Numerous books have been published about OpenGL, and a great deal of sample code is readily available, making information about OpenGL inexpensive and easy to obtain.

The OpenGL Visualization Programming Pipeline



OpenGL operates on image data as well as geometric primitives.

Simplifies Software Development, Speeds Time-to-Market
OpenGL routines simplify the development of graphics software—from rendering a simple geometric point, line, or filled polygon to the creation of the most complex lighted and texture-mapped NURBS curved surface. OpenGL gives software developers access to geometric and image primitives, display lists, modeling transformations, lighting and texturing, anti-aliasing, blending, and many other features.

Every conforming OpenGL implementation includes the full complement of OpenGL functions. The well-specified OpenGL standard has language bindings for C, C++, Fortran, Ada, and Java. All licensed OpenGL implementations come from a single specification and language binding document and are required to pass a set of conformance tests. Applications utilizing OpenGL functions are easily portable across a wide array of platforms for maximized programmer productivity and shorter time-to-market.

All elements of the OpenGL state—even the contents of the texture memory and the frame buffer—can be obtained by an OpenGL application. OpenGL also supports visualization applications with 2D images treated as types of primitives that can be manipulated just like 3D geometric objects. As shown in the OpenGL visualization programming pipeline diagram above, images and vertices defining geometric primitives are passed through the OpenGL pipeline to the frame buffer.

Available Everywhere
Supported on all UNIX® workstations, and shipped standard with every Windows 95/98/2000/NT and MacOS PC, no other graphics API operates on a wider range of hardware platforms and software environments. OpenGL runs on every major operating system including Mac OS, OS/2, UNIX, Windows 95/98, Windows 2000, Windows NT, Linux, OPENStep, and BeOS; it also works with every major windowing system, including Win32, MacOS, Presentation Manager, and X-Window System. OpenGL is callable from Ada, C, C++, Fortran, Python, Perl and Java and offers complete independence from network protocols and topologies.

Architected for Flexibility and Differentiation: Extensions
Although the OpenGL specification defines a particular graphics processing pipeline, platform vendors have the freedom to tailor a particular OpenGL implementation to meet unique system cost and performance objectives. Individual calls can be executed on dedicated hardware, run as software routines on the standard system CPU, or implemented as a combination of both dedicated hardware and software routines. This implementation flexibility means that OpenGL hardware acceleration can range from simple rendering to full geometry and is widely available on everything from low-cost PCs to high-end workstations and supercomputers. Application developers are assured consistent display results regardless of the platform implementation of the OpenGL environment.

Using the OpenGL extension mechanism, hardware developers can differentiate their products by developing extensions that allow software developers to access additional performance and technological innovations.

Many OpenGL extensions, as well as extensions to related APIs like GLU, GLX, and WGL, have been defined by vendors and groups of vendors. The OpenGL Extension Registry is maintained by SGI and contains specifications for all known extensions, written as modifications to the appropriate specification documents. The registry also defines naming conventions, guidelines for creating new extensions and writing suitable extension specifications, and other related documentation.

API Hierarchy

- OpenGL applications use the window system's window, input, and event mechanism
- GLU supports quadrics, NURBS, complex polygons, matrix utilities, and more

This diagram demonstrates the relationship between OpenGL GLU and windowing APIs.

The Foundation for Advanced APIs

Leading software developers use OpenGL, with its robust rendering libraries, as the 2D/3D graphics foundation

for higher-level APIs. Developers leverage the capabilities of OpenGL to deliver highly differentiated, yet widely supported vertical market solutions. For example, Open Inventor provides a cross-platform user interface and flexible scene graph that makes it easy to create OpenGL applications. IRIS Performer < leverages OpenGL functionality and delivers additional features tailored for the demanding high frame rate markets such as visual simulation and virtual sets OpenGL Optimizer is a toolkit for real-time interaction, modification, and rendering of complex surface-based models such as those found in CAD/CAM and special effects creation. OpenGL Volumizer is a high-level immediate mode volume rendering API for the energy, medical and sciences markets. OpenGL Shader provides a common interface to support realistic visual effects, bump mapping, multiple textures, environment maps, volume shading and an unlimited array of new effects using hardware acceleration on standard OpenGL graphics cards.

Governance

The OpenGL Architecture Review Board (ARB), was an independent consortium formed in 1992, that governed the future of OpenGL, proposing and approving changes to the specification, new releases, and conformance testing. In Sept 2006, the ARB became the OpenGL Working Group under the Khronos Group consortium for open standard APIs.

The OpenGL Performance Characterization Committee, another independent organization, creates and maintains OpenGL benchmarks and publishes the results of those benchmarks on its Web site: www.specbench.org/gpc/opc.static/index.html.

Continued Innovation

The OpenGL standard is constantly evolving. Formal revisions occur at periodic intervals, and extensions allowing application developers to access the latest hardware advances through OpenGL are continuously being developed. As extensions become widely accepted, they are considered for inclusion into the core OpenGL standard. This process allows OpenGL to evolve in a controlled yet innovative manner.

Licensing

ARB-approved OpenGL specifications and source code are available to licensed hardware platform vendors. End users, independent software vendors, and others writing code based on the OpenGL API are free from licensing requirements. See SGI's Licensing web site for more information.

OpenGL Applications & Games

OpenGL is the pervasive standard for 3D consumer and professional applications across all major OS platforms. A partial list for Window, Linux and MacOS are available in the products section.

Column Header

About OpenGL

- Contact Us
- OpenGL logo
- Advertise on OpenGL
- Jobs

Column Footer

- About OpenGL
- Privacy Policy