About This Guide

The OpenGL graphics system is a software interface to graphics hardware. (The GL stands for Graphics Library.) It allows you to create interactive programs that produce color images of moving three-dimensional objects. With OpenGL, you can control computer-graphics technology to produce realistic pictures or ones that depart from reality in imaginative ways. This guide explains how to program with the OpenGL graphics system to deliver the visual effect you want.

What This Guide Contains

This guide has 15 chapters. The first five chapters present basic information that you need to understand to be able to draw a properly colored and lit three-dimensional object on the screen.

- Chapter 1, "**Introduction to OpenGL**," provides a glimpse into the kinds of things OpenGL can do. It also presents a simple OpenGL program and explains essential programming details you need to know for subsequent chapters.
- Chapter 2, "**State Management and Drawing Geometric Objects**," explains how to create a three-dimensional geometric description of an object that is eventually drawn on the screen.
- Chapter 3, "**Viewing**," describes how such three-dimensional models are transformed before being drawn on a two-dimensional screen. You can control these transformations to show a particular view of a model.
- Chapter 4, "**Color**," describes how to specify the color and shading method used to draw an object.

- Chapter 5, "**Lighting**," explains how to control the lighting conditions surrounding an object and how that object responds to light (that is, how it reflects or absorbs light). Lighting is an important topic, since objects usually don't look three-dimensional until they're lit.

The remaining chapters explain how to optimize or add sophisticated features to your three-dimensional scene. You might choose not to take advantage of many of these features until you're more comfortable with OpenGL. Particularly advanced topics are noted in the text where they occur.

- Chapter 6, "**Blending, Antialiasing, Fog, and Polygon Offset**," describes techniques essential to creating a realistic scene—alpha blending (to create transparent objects), antialiasing (to eliminate jagged edges), atmospheric effects (to simulate fog or smog), and polygon offset (to remove visual artifacts when highlighting the edges of filled polygons).
- Chapter 7, "**Display Lists**," discusses how to store a series of OpenGL commands for execution at a later time. You'll want to use this feature to increase the performance of your OpenGL program.
- Chapter 8, "**Drawing Pixels, Bitmaps, Fonts, and Images**," discusses how to work with sets of two-dimensional data as bitmaps or images. One typical use for bitmaps is describing characters in fonts.
- Chapter 9, "**Texture Mapping**," explains how to map one-, two-, and three-dimensional images called *textures* onto three-dimensional objects. Many marvelous effects can be achieved through texture mapping.
- Chapter 10, "**The Framebuffer**," describes all the possible buffers that can exist in an OpenGL implementation and how you can control them. You can use the buffers for such effects as hidden-surface elimination, stenciling, masking, motion blur, and depth-of-field focusing.
- Chapter 11, "**Tessellators and Quadrics**," shows how to use the tessellation and quadrics routines in the GLU (OpenGL Utility Library).
- Chapter 12, "**Evaluators and NURBS**," gives an introduction to advanced techniques for efficient generation of curves or surfaces.
- Chapter 13, "**Selection and Feedback**," explains how you can use OpenGL's selection mechanism to select an object on the screen. Additionally, the chapter explains the feedback mechanism, which allows you to collect the drawing information OpenGL produces, rather than having it be used to draw on the screen.

- Chapter 14, "**Now That You Know,**" describes how to use OpenGL in several clever and unexpected ways to produce interesting results. These techniques are drawn from years of experience with both OpenGL and the technological precursor to OpenGL, the Silicon Graphics IRIS Graphics Library.
- Chapter 15, "**The OpenGL Shading Language,**" discusses the changes that occurred starting with OpenGL Version 2.0. This includes an introduction to the OpenGL Shading Language, also commonly called the "GLSL," which allows you to take control of portions of OpenGL's processing for *vertices* and *fragments*. This functionality can greatly enhance the image quality and computational power of OpenGL.

In addition, there are several appendices that you will likely find useful:

- Appendix A, "**Order of Operations,**" gives a technical overview of the operations OpenGL performs, briefly describing them in the order in which they occur as an application executes.
- Appendix B, "**State Variables,**" lists the state variables that OpenGL maintains and describes how to obtain their values.
- Appendix C, "**OpenGL and Window Systems,**" briefly describes the routines available in window-system-specific libraries, which are extended to support OpenGL rendering. Window system interfaces to the X Window System, Apple's Mac/OS, IBM OS/2, and Microsoft Windows are discussed here.
- Appendix D, "**Basics of GLUT: The OpenGL Utility Toolkit,**" discusses the library that handles window system operations. GLUT is portable and it makes code examples shorter and more comprehensible.
- Appendix E, "**Calculating Normal Vectors,**" tells you how to calculate normal vectors for different types of geometric objects.
- Appendix F, "**Homogeneous Coordinates and Transformation Matrices,**" explains some of the mathematics behind matrix transformations.
- Appendix G, "**Programming Tips,**" lists some programming tips based on the intentions of the designers of OpenGL that you might find useful.
- Appendix H, "**OpenGL Invariance,**" describes when and where an OpenGL implementation must generate the exact pixel values described in the OpenGL specification.

- Appendix I, **"Built-In OpenGL Shading Language Variables and Functions,"** lists all of the built-in variables and functions available in the OpenGL Shading Language.

Finally, an extensive Glossary defines the key terms used in this guide.

What's New in This Edition

The sixth edition of the *OpenGL Programming Guide* includes new and updated material covering OpenGL Version 2.1:

- Coverage of the following new core capabilities has been added:
  - Storage of pixel rectangles in buffer objects.
  - Support for sRGB formatted textures. The sRGB color space roughly corresponds to a gamma-corrected RGB space (using a gamma value of 2.2).
  - Specification of nonsquare matrices as uniform variables in OpenGL shading language shader programs.
  - Expanded discussion of the OpenGL shading language, including additions supporting OpenGL Version 2.1 functionality.
- Bug fixes and other clarifications

What You Should Know Before Reading This Guide

This guide assumes only that you know how to program in the C language and that you have some background in mathematics (geometry, trigonometry, linear algebra, calculus, and differential geometry). Even if you have little or no experience with computer graphics technology, you should be able to follow most of the discussions in this book. Of course, computer graphics is a huge subject, so you may want to enrich your learning experience with supplemental reading:

- *Computer Graphics: Principles and Practice* by James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes (Addison-Wesley, 1990)—This book is an encyclopedic treatment of the subject of computer graphics. It includes a wealth of information but is probably best read after you have some experience with the subject.

- *3D Computer Graphics* by Andrew S. Glassner (The Lyons Press, 1994)—This book is a nontechnical, gentle introduction to computer graphics. It focuses on the visual effects that can be achieved, rather than on the techniques needed to achieve them.

Another great place for all sorts of general information is the Official OpenGL Web Site. This Web site contains software, sample programs, documentation, FAQs, discussion boards, and news. It is always a good place to start any search for answers to your OpenGL questions:

`http://www.opengl.org/`

Additionally, full documentation of all the procedures that compose OpenGL Version 2.1 are documented at the Official OpenGL Web site. These Web pages replace the *OpenGL Reference Manual* that was published by the OpenGL Architecture Review Board and Addison-Wesley.

OpenGL is really a hardware-independent specification of a programming interface, and you use a particular implementation of it on a particular kind of hardware. This guide explains how to program with any OpenGL implementation. However, since implementations may vary slightly—in performance and in providing additional, optional features, for example—you might want to investigate whether supplementary documentation is available for the particular implementation you're using. In addition, you might have OpenGL-related utilities, toolkits, programming and debugging support, widgets, sample programs, and demos available to you with your system.

How to Obtain the Sample Code

This guide contains many sample programs to illustrate the use of particular OpenGL programming techniques. These programs make use of Mark Kilgard's OpenGL Utility Toolkit (GLUT). GLUT is documented in *OpenGL Programming for the X Window System* by Mark Kilgard (Addison-Wesley, 1996). The section "OpenGL-Related Libraries" in Chapter 1 and Appendix D give more information about using GLUT. If you have access to the Internet, you can obtain the source code for both the sample programs and GLUT for free via anonymous ftp (file-transfer protocol).

For the source code examples found in this book, please visit

`http://www.opengl-redbook.com/code/`

For Mark Kilgard's source code for GLUT (for Microsoft Windows or the X Window System), check this Web page to find out what current version of GLUT is available and where to download the source code from:

`http://www.opengl.org/resources/libraries/glut/`

Many implementations of OpenGL might also include the code samples as part of the system. This source code is probably the best source for your implementation, because it might have been optimized for your system. Read your machine-specific OpenGL documentation to see where the code samples can be found.

Nate Robins' OpenGL Tutors

Nate Robins has written a suite of tutorial programs that demonstrate basic OpenGL programming concepts by allowing the user to modify the parameters of a function and interactively see their effects. Topics covered include transformations, lighting, fog, and texturing. These highly recommended tutorials are portable and require the aforementioned GLUT. To get the source code for these tutorials, see this Web site:

`http://www.xmission.com/~nate/tutors.html`

Errata

Undoubtedly this book has errors. An error list is maintained at the following Web site:

`http://www.opengl-redbook.com/errata/`

If you find any bugs, please use this Web site to report them.

Style Conventions

These style conventions are used in this guide:

- **Bold**—Command and routine names and matrices
- *Italics*—Variables, arguments, parameter names, spatial dimensions, matrix components, and first occurrences of key terms
- Regular—Enumerated types and defined constants

Code examples are set off from the text in a monospace font, and command summaries are shaded with gray boxes.

In a command summary, braces are used to identify options among data types. In the following example, **glCommand** has four possible suffixes: s, i, f, and d, which stand for the data types GLshort, GLint, GLfloat, and GLdouble. In the function prototype for **glCommand**, *TYPE* is a wildcard that represents the data type indicated by the suffix.

void **glCommand**{sifd}(*TYPE x1*, *TYPE y1*, *TYPE x2*, *TYPE y2*);