About This Guide

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 14 chapters, one more than the ideal number. 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 onto 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− and two−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 efficiently generating curves or surfaces.

Chapter 13, **"Selection and Feedback,"** explains how you can use OpenGL's selection mechanism to select an object on the screen. It also 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.

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 MacIntosh, IBM OS/2, and Microsoft Windows NT and Windows 95 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, **"Color Plates,"** contains the color plates that appear in the printed version of this guide.

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

What's New in This Edition

To the question, "What's new in this edition?" the wiseacre answer is "About 100 pages." The more informative answer follows.

Detailed information about the following new features of OpenGL Version 1.1 has been added.

- Vertex arrays
- Texturing enhancements, including texture objects (including residency and prioritization), internal texture image format, texture subimages, texture proxies, and copying textures from frame buffer data
- Polygon offset
- Logical operation in RGBA mode

Program examples have been converted to Mark Kilgard's GLUT, which stands for Graphics Library Utility Toolkit. GLUT is an increasingly popular windowing toolkit, which is well−documented and has been ported to different window systems.

More detail about some topics that were in the first edition, especially coverage of the OpenGL Utility (GLU) Library.

- An entire chapter on GLU tessellators and quadrics
- A section (in Chapter 3) on the use of *gluProject()* and *gluUnProject()*, which mimics or reverses the operations of the geometric processing pipeline (This has been the subject of frequent discussions on the Internet newsgroup on OpenGL, *comp.graphics.api.opengl*.)
- Expanded coverage (and more diagrams) about images
- Changes to GLU NURBS properties
- Error handling and vendor−specific extensions to OpenGL
- Appendix C expanded to include OpenGL interfaces to several window/operating systems

The first edition's appendix on the OpenGL Utility Library was removed, and its information has been integrated into other chapters.

A much larger and more informative index

Bug fixes and minor topic reordering. Moving the display list chapter is the most noticeable change.

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 (Reading, MA: 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: A User's Guide for Artists and Designers* by Andrew S. Glassner (New York: Design Press, 1989)—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.

Once you begin programming with OpenGL, you might want to obtain the *OpenGL Reference Manual* by the OpenGL Architecture Review Board (Reading, MA: Addison−Wesley Developers Press, 1996), which is designed as a companion volume to this guide. The *Reference Manual* provides a technical view of how OpenGL operates on data that describes a geometric object or an image to produce an image on the screen. It also contains full descriptions of each set of related OpenGL commands—the parameters used by the commands, the default values for those parameters, and what the commands accomplish. Many OpenGL implementations have this same material on−line, in the form of*man* pages or other help documents, and it's probably more up−to−date. There is also a http version on the World Wide Web; consult Silicon Graphics OpenGL Web Site (http://www.sgi.com/Technology/openGL) for the latest pointer.

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 (Reading, MA: Addison−Wesley Developers Press, 1996). The section "OpenGL−Related Libraries" in Chapter 1 and Appendix D gives 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, grab this file:

```
ftp://sgigate.sgi.com/pub/opengl/opengl1_1.tar.Z
```

The files you receive are compressed *tar* archives. To uncompress and extract the files, type

```
uncompress opengl1_1.tar
```

```
tar xf opengl1_1.tar
```

For Mark Kilgard's source code for an X Window System version of GLUT, you need to know what the most current version is. The filename will be *glut−i.j.tar.Z*, where *i* is the major revision number and *j* is the minor revision number of the most recent version. Check the directory for the right numbers, then grab this file:

```
ftp://sgigate.sgi.com/pub/opengl/xjournal/GLUT/glut−i.j.tar.Z
```

This file must also be uncompressed and extracted by using the *tar* command. The sample programs and GLUT library are created as subdirectories from wherever you are in the file directory structure.

Other ports of GLUT (for example, for Microsoft Windows NT) are springing up. A good place to start searching for the latest developments in GLUT and for OpenGL, in general, is Silicon Graphics' OpenGL Web Site:

```
http://www.sgi.com/Technology/openGL
```

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.

Errata

Although this book is ideal and perfec in every conceivable way, there is a a pointer to an errata list from the Silicon Graphics OpenGL Web Site:

http://www.sgi.com/Technology/openGL

The authors are quite certain there will be a little note there to reassure the reader of the pristeen quality of this book.

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 the first occurrence 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 choices 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}(TYPEx1, TYPEy1, TYPEx2, TYPEy2);