OpenGL® Programming Guide

Seventh Edition

The Official Guide to Learning OpenGL®, Versions 3.0 and 3.1

Dave Shreiner

The Khronos OpenGL ARB Working Group

OpenGL®
Programming Guide
Seventh Edition

The Official Guide to
Learning OpenGL®, Versions 3.0 and 3.1

Dave Shreiner
The Khronos OpenGL ARB Working Group

Addison-Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact:

International Sales
international@pearsoned.com

Visit us on the Web: informit.com/aw

Library of Congress Cataloging-in-Publication Data

Shreiner, Dave.
OpenGL programming guide : the official guide to learning OpenGL, versions 3.0 and 3.1 / Dave Shreiner; the Khronos OpenGL ARB Working Group — 7th ed.
p. cm.
Includes index.
ISBN 978-0-321-55262-4 (pbk. : alk. paper)
1. Computer graphics. 2. OpenGL. I. Title.
T385.O635 2009
006.6'6—dc22

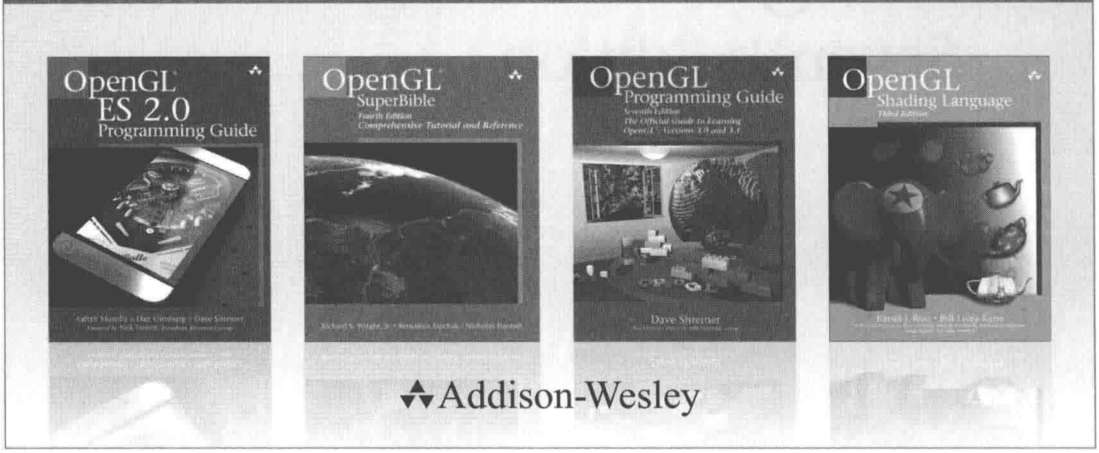
2009018793

Pearson Education, Inc.
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax (617) 671-3447

ISBN 13: 978-0-321-55262-4
ISBN 10: 0-321-55262-8
Text printed in the United States on recycled paper at Edwards Brothers in Ann Arbor, Michigan.
Fourth printing, November 2011

OpenGL®
Programming Guide
Seventh Edition

OpenGL® Series



Visit **informit.com/opengl** for a complete list of available products

The OpenGL graphics system is a software interface to graphics hardware. ("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.

The **OpenGL Series** from Addison-Wesley Professional comprises tutorial and reference books that help programmers gain a practical understanding of OpenGL standards, along with the insight needed to unlock OpenGL's full potential.

PEARSON

Addison-Wesley Cisco Press EXAM/CRAM IBM Press. QUE® PRENTICE HALL SAMS | Safari Books Online

For my family—Felicity, Max, Sarah, and Scout.
—JLN

For my family—Ellyn, Ricky, and Lucy.
—TRD

To Tom Doeppner and Andy van Dam, who started me along this path.
—MW

For my family—Vicki, Bonnie, Bob, Phantom, Squiggles, Tuxedo, and Toby.
—DRS

In memory of Phil Karlton, Celeste Fowler, and Ben Cheatham.

About This Guide

The OpenGL graphics system is a software interface to graphics hardware. "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 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.

There are also several appendices that you will likely find useful:

- Appendix A, "**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 B, "**State Variables,**" lists the state variables that OpenGL maintains and describes how to obtain their values.
- Appendix C, "**Homogeneous Coordinates and Transformation Matrices,**" explains some of the mathematics behind matrix transformations.
- Appendix D, "**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, and Microsoft Windows are discussed here.

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

In addition, the appendices listed below are available at the following Web site:

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

- Appendix E, "**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 F, "**Programming Tips,**" lists some programming tips based on the intentions of the designers of OpenGL that you might find useful.
- Appendix G, "**OpenGL Invariance,**" describes when and where an OpenGL implementation must generate the exact pixel values described in the OpenGL specification.

- Appendix H, "**Calculating Normal Vectors,**" tells you how to calculate normal vectors for different types of geometric objects.
- Appendix I, "**Built-In OpenGL Shading Language Variables and Functions,**" describes the built-in variables and functions available in the OpenGL Shading Language.
- Appendix J, "**Floating-Point Formats for Textures, Framebuffers, and Renderbuffers,**" documents the various floating-point and shared-exponent pixel and texel formats.
- Appendix K, "**RGTC Compressed Texture Format,**" describes the texture format for storing one- and two-component compressed textures.
- Appendix L, "**std140 Uniform Buffer Layout,**" documents the standard memory layout of uniform-variable buffers for GLSL 1.40.

What's New in This Edition

This seventh edition of the *OpenGL Programming Guide* includes new and updated material covering OpenGL Versions 3.0 and 3.1. With those versions, OpenGL—which is celebrating its eighteenth birthday the year of this writing—has undergone a drastic departure from its previous revisions. Version 3.0 added a number of new features as well as a *depreciation model,* which sets the way for antiquated features to be removed from the library. Note that only new features were added to Version 3.0, making it completely source and binary backward compatible with previous versions. However, a number of features were marked as *deprecated,* indicating that they may potentially be removed from future versions of the API.

Updates related to OpenGL Version 3.0 that are discussed in this edition include the following items:

- New features in OpenGL:
  - An update to the OpenGL Shading Language, creating version 1.30 of GLSL
  - Conditional rendering
  - Finer-grained access to mapping buffer objects' memory for update and reading
  - Floating-point pixel formats for framebuffers in addition to texture map formats (which were added in OpenGL Version 2.1)

- Framebuffer and *renderbuffer* objects
- Compact floating-point representations for reducing the memory storage usage for small dynamic-range data
- Improved support for multisample buffer interactions when copying data
- Non-normalized integer values in texture maps and renderbuffers whose values retain their original representation, as compared to OpenGL's normal operation of mapping those values into the range [0,1]
- One- and two-dimensional texture array support
- Additional packed-pixel formats allowing access to the new renderbuffer support
- Separate blending and writemask control for multiple rendering targets
- Texture compression format
- Single- and double-component internal formats for textures
- Transform feedback
- Vertex-array objects
- sRGB framebuffer format

- An in-depth discussion of the deprecation model
- Bug fixes and updated token names

And for OpenGL Version 3.1:

- Identification of features removed due to deprecation in Version 3.0
- New features:
    - An update to the OpenGL Shading Language, creating version 1.40 of GLSL
    - Instanced rendering
    - Efficient server-side copies of data between buffers
    - Rendering of multiple similar primitives within a single draw call using a special (user-specified) token to indicate when to restart a primitive
    - Texture buffer objects

  - Texture rectangles
  - Uniform buffer objects
  - Signed normalized texel formats

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 an ever-expanding 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 Versions 3.0 and 3.1 will be 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, the provider of your particular implementation might have OpenGL-related utilities, toolkits, programming and debugging support, widgets, sample programs, and demos available at its Web site.

How to Obtain the Sample Code

This guide contains many sample programs to illustrate the use of particular OpenGL programming techniques. As the audience for this guide has a wide range of experience—from novice to seasoned veteran—with both computer graphics and OpenGL, the examples published in these pages usually present the simplest approach to a particular rendering situation, demonstrated using the OpenGL Version 3.0 interface. This is done mainly to make the presentation straightforward and obtainable to those readers just starting with OpenGL. For those of you with extensive experience looking for implementations using the latest features of the API, we first thank you for your patience with those following in your footsteps, and ask that you please visit our Web site:

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

There, you will find the source code for all examples in this text, implementations using the latest features, and additional discussion describing the modifications required in moving from one version of OpenGL to another.

All of the programs contained within this book use the OpenGL Utility Toolkit (GLUT), originally authored by Mark Kilgard. For this edition, we use the open-source version of the GLUT interface from the folks developing the freeglut project. They have enhanced Mark's original work (which is thoroughly documented in his book, *OpenGL Programming for the X Window System* (Addison-Wesley, 1996)). You can find their open-source project page at the following address:

`http://freeglut.sourceforge.net/`

You can obtain code and binaries of their implementation at this site.

The section "OpenGL-Related Libraries" in Chapter 1 and Appendix A give more information about using GLUT. Additional resources to help accelerate your learning and programming of OpenGL and GLUT can be found at the OpenGL Web site's resource pages:

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

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 those code samples can be found.

Errata

Unfortunately, it is likely this book will have errors. Additionally, OpenGL is updated during the publication of this guide: Errors are corrected and clarifications are made to the specification, and new specifications are released. We keep a list of bugs and updates at our Web site, `http://www.opengl-redbook.com/`, where we also offer facilities for reporting any new bugs you might find. If you find an error, please accept our apologies, and our thanks in advance for reporting it. We'll get it corrected as soon as possible.

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*);

Distinguishing Deprecated Features

As mentioned, this edition of the *OpenGL Programming Guide* details Versions 3.0 and 3.1. OpenGL Version 3.0 is entirely backward compatible with all of the versions made available to this point. However, Version 3.1 employed the deprecation model to remove a number of older features that were less compatible with modern graphics systems. While numerous features were removed from the "core" of OpenGL, to ease the transition between versions, the OpenGL ARB released the GL_ARB_compatibility extension. If your implementation supports this extension, it will be able to use all of the removed functionality. To easily identify features that were removed from OpenGL in Version 3.1, but are still supported by the compatibility extension, an informational table listing the affected functions or tokens will be shown in the margin of this book next to where the command or feature is introduced in its gray box.

Compatibility Extension
glBegin
GL_POLYGON

While only features from OpenGL were deprecated and removed, some of those features affect libraries, such as the OpenGL Utility Library, commonly called GLU. Those functions that are affected by the changes in OpenGL Version 3.1 are also listed in a table in the margin.

Acknowledgments

The Seventh Edition

OpenGL Versions 3.0 and 3.1, which this guide covers, mark a new era in the evolution of OpenGL. Once again, the members of the OpenGL ARB Working Group, as part of the Khronos Group, have worked tirelessly to provide new versions that leverage the latest developments in graphics technology. Barthold Lichtenbelt, Bill Licea-Kane, Jeremy Sandmel, and Jon Leech, all of whom lead the technical sub-groups of the OpenGL ARB Working group deserve our thanks. Additionally, without the tireless efforts of Neil Trevett, President of the Khronos Group, who has carried the torch on open-standard media APIs.

The staff at Addison-Wesley once again worked miracles in producing this edition. Debra Williams Cauley, Anna Popick, John Fuller, Molly Sharp, and Jill Hobbs helped with advice and recommendations in making this manuscript better. A thorough technical review was provided by Sean Carmody and Bob Kuehne. Their help is greatly appreciated.

The Sixth Edition

As with the seven preceding versions of OpenGL, the guidance of the OpenGL Architecture Review Board was paramount in its evolution and development. Without the ARB's guidance and devotion, OpenGL would surely languish, and once again we express our gratitude for their efforts.

Once again, the staff of Addison-Wesley provided the support and encouragement to have this edition come to fruition. Debra Williams Cauley, Tyrrell Albaugh, and John Fuller once again worked miracles in producing this manuscript. Thanks once again for an effort second to none.

The Fifth Edition

OpenGL continued its evolutionary track under the careful guidance of the OpenGL Architecture Review Board and its working groups. The small committees that help unify the various business and technical differences among the ARB's membership deserve our thanks and gratitude. They continue to push OpenGL's success to new levels.

As always, the ever-patient and helpful staff at Addison-Wesley were indispensable. Once again, Mary O'Brien, perhaps OpenGL's most devoted non-programming (at least to our knowledge) proponent, continues to encourage us to update the programming guide for the community. Tyrrell Albaugh and John Fuller worked tirelessly in preparing the manuscript for production. Thanks to you all.

The Fourth Edition

OpenGL continued its evolution and success with the aid of many individuals. The OpenGL Architecture Review Board, along with its many participants, help to mold OpenGL. Their contributions were much appreciated.

Numerous example programs were written by Stace Peterson. Helpful discussions and clarifications were provided by Maryann Simmons, Patrick Brown, Alan Commike, Brad Grantham, Bob Kuehne, Jon Leech, Benjamin Lipchak, Marc Olano, and Vicki Shreiner.

Once again, the editorial and production staff at Addison-Wesley were extremely helpful. Thanks to Mary O'Brien, John Fuller, and Brenda Mulligan.

The Third Edition

The third edition of this book required the support of many individuals.

Special thanks are due to the reviewers who volunteered and trudged through the now seven hundred pages of technical material that constitute the third edition: Bill Armstrong, Bob Beretta, David Blythe, Dan Brokenshire, Norman Chin, Steve Cunningham, Angus Dorbie, Laurence Feldman, Celeste Fowler, Jeffery Galinovsky, Brad Grantham, Eric Haines, David Ishimoto, Mark Kilgard, Dale Kirkland, Jon Leech, Seth Livingston, Chikai Ohazama, Bimal Poddar, Mike Schmit, John Stauffer, R. Scott Thompson, David Yu, and Hansong Zhang. Their careful diligence has greatly improved the quality of this book.

An immeasurable debt of gratitude goes to Laura Cooper, Dany Galgani, and Dan Young for their production support, and to Mary O'Brien, Elizabeth Spainhour, Chanda Leary, and John Fuller of Addison-Wesley. Additionally, Miriam Geller, Shawn Hopwood, Stacy Maller, and David Story were instrumental in the coordination and marketing of this effort.

The First and Second Editions

Thanks to the long list of pioneers and past contributors to the success of OpenGL and of this book.

Thanks to the chief architects of OpenGL: Mark Segal and Kurt Akeley. Special recognition goes to the pioneers who heavily contributed to the initial design and functionality of OpenGL: Allen Akin, David Blythe, Jim Bushnell, Dick Coulter, John Dennis, Raymond Drewry, Fred Fisher, Celeste Fowler, Chris Frazier, Momi Furuya, Bill Glazier, Kipp Hickman, Paul Ho, Rick Hodgson, Simon Hui, Lesley Kalmin, Phil Karlton, On Lee, Randi Rost, Kevin P. Smith, Murali Sundaresan, Pierre Tardif, Linas Vepstas, Chuck Whitmer, Jim Winget, and Wei Yen.

The impetus for the second edition began with Paula Womack and Tom McReynolds of Silicon Graphics, who recognized the need for a revision and also contributed some of the new material. John Schimpf, OpenGL Product Manager at Silicon Graphics, was instrumental in getting the revision off and running.

Many thanks go to the people who contributed to the success of the first and second editions of this book: Cindy Ahuna, Kurt Akeley, Bill Armstrong, Otto Berkes, Andy Bigos, Drew Bliss, Patrick Brown, Brian Cabral, Norman Chin, Bill Clifford, Jim Cobb, Dick Coulter, Kathleen Danielson, Suzy Deffeyes, Craig Dunwoody, Fred Fisher, Chris Frazier, Ken Garnett, Kathy Gochenour, Michael Gold, Mike Heck, Paul Ho, Deanna Hohn, Brian Hook, Kevin Hunter, Phil Huxley, Renate Kempf, Mark Kilgard, Dale Kirkland, David Koller, Kevin LeFebvre, Hock San Lee, Zicheng Liu, Rob Mace, Kay Maitz, Tim Misner, Jeremy Morris, Dave Orton, Bimal Poddar, Susan Riley, Randi Rost, Mark Segal, Igor Sinyak, Bill Sweeney, Pierre Tardif, Andy Vesper, Henri Warren, Paula Womack, Gilman Wong, Steve Wright, and David Yu.

The color plates received a major overhaul for this edition. The sequence of plates based on the cover image (Plates 1 through 9) was created by Thad Beier, Seth Katz, and Mason Woo. Plates 10 through 20, 22, and 23 are snapshots of programs created by Mason Woo. Plate 21 was created by Paul Haeberli. Plate 24 was created by Cyril Kardassevitch of the Institue de