

Introduction

Welcome to the fourth edition of the *OpenGL SuperBible*. For more than ten years, we have striven to provide the world's best introduction to not only OpenGL, but 3D graphics programming in general. This book is both a comprehensive reference of the entire OpenGL API and a tutorial that will teach you how to use this powerful API to create stunning 3D visualizations, games, and other graphics of all kinds. Starting with basic 3D terminology and concepts, we take you through basic primitive assembly, transformations, lighting, texturing, and eventually bring you into the full power of the programmable graphics pipeline with the OpenGL Shading Language.

Regardless of whether you are programming on Windows, Mac OS X, Linux, or a hand-held gaming device, this book is a great place to start learning OpenGL, and how to make the most of it on your specific platform. The majority of the book is highly portable C++ code hosted by the GLUT or FreeGLUT toolkit. You will also find OS-specific chapters that show how to wire OpenGL into your native window systems. Throughout the book, we try to make few assumptions about how much previous knowledge the reader has of 3D graphics programming topics. This yields a tutorial that is accessible by both the beginning programmer and the experienced programmer beginning OpenGL.

What's New in This Edition

Readers of the previous editions will notice right away that the reference material has been reorganized. Instead of attempting to place individual functions with chapters that use them, we now have Appendix C, which contains the complete OpenGL API reference for the GL function. This is a much more appropriate and useful organizational structure for this material. These reference pages are also now based on the "official" OpenGL man pages, which means there will be no more incomplete or missing function calls. Detailed function entries will also be more concise and complete.

The Mac OS X and Linux chapters in this edition have been totally rewritten from the ground up. Sometimes a revision is not sufficient, and the best thing to do is just start over. We think readers will like these two newly rewritten chapters, which will be useful to anyone needing an introduction to the specifics of getting OpenGL up and running on their particular platform. Also, on the platform topic, the Windows chapter has been updated and pruned of some older and obsolete topics. Of note is the fact that OpenGL's widely rumored demise on Vista has, in fact, NOT occurred.

We have also added two completely new chapters. In this edition, we bring you full coverage of the latest OpenGL ES specification. We also provide a very exciting chapter on advanced OpenGL buffer usage, including off screen rendering, floating point buffers and

textures, and pixel buffer objects. Throughout all the chapters, coverage has been touched up to include OpenGL 2.1 functionality, and to focus more on current OpenGL programming techniques. (Chapter 11, for example, deals with geometry submission and was modified heavily for this purpose.)

Finally, you'll find a Color insert with a gallery of images for which black and white just does not do adequate justice. A book on graphics programming is certainly more useful with color images. Some techniques, for example, are impossible to demonstrate on the printed page without the use of color. Other images are provided because the black-and-white versions simply do not convey the same information about how a particular image should look.

How This Book Is Organized

The *OpenGL SuperBible* is divided into three parts: The Old Testament, The New Testament, and the Apocrypha. Each section covers a particular personality of OpenGL—namely, the fixed pipeline, programmable hardware, and finally some platform-specific bindings. We certainly would not equate our humble work with anyone's sacred texts. However, the informed reader will certainly see how strong and irresistible this metaphor actually is.

Part I: The Old Testament

You'll learn how to construct a program that uses OpenGL, how to set up your 3D-rendering environment, and how to create basic objects and light and shade them. Then we'll delve deeper into using OpenGL and some of its advanced features and different special effects. These chapters are a good way to introduce yourself to 3D graphics programming with OpenGL and provide the conceptual foundation on which the more advanced capabilities later in the book are based.

Chapter 1—Introduction to 3D Graphics and OpenGL. This introductory chapter is for newcomers to 3D graphics. It introduces fundamental concepts and some common vocabulary.

Chapter 2—Using OpenGL. In this chapter, we provide you with a working knowledge of what OpenGL is, where it came from, and where it is going. You will write your first program using OpenGL, find out what headers and libraries you need to use, learn how to set up your environment, and discover how some common conventions can help you remember OpenGL function calls. We also introduce the OpenGL state machine and error-handling mechanism.

Chapter 3—Drawing in Space: Geometric Primitives and Buffers. Here, we present the building blocks of 3D graphics programming. You'll basically find out how to tell a computer to create a three-dimensional object with OpenGL. You'll also learn the basics of hidden surface removal and ways to use the stencil buffer.

Chapter 4—Geometric Transformations: The Pipeline. Now that you’re creating three-dimensional shapes in a virtual world, how do you move them around? How do you move yourself around? These are the things you’ll learn here.

Chapter 5—Color, Materials, and Lighting: The Basics. In this chapter, you’ll take your three-dimensional “outlines” and give them color. You’ll learn how to apply material effects and lights to your graphics to make them look real.

Chapter 6—More on Colors and Materials. Now it’s time to learn about blending objects with the background to make transparent (see-through) objects. You’ll also learn some special effects with fog and the accumulation buffer.

Chapter 7—Imaging with OpenGL. This chapter is all about manipulating image data within OpenGL. This information includes reading a TGA file and displaying it in an OpenGL window. You’ll also learn some powerful OpenGL image-processing capabilities.

Chapter 8—Texture Mapping: The Basics. Texture mapping is one of the most useful features of any 3D graphics toolkit. You’ll learn how to wrap images onto polygons and how to load and manage multiple textures at once.

Chapter 9—Texture Mapping: Beyond the Basics. In this chapter, you’ll learn how to generate texture coordinates automatically, use advanced filtering modes, and use built-in hardware support for texture compression. You’ll also learn about OpenGL’s support for point sprites.

Chapter 10—Curves and Surfaces. The simple triangle is a powerful building block. This chapter gives you some tools for manipulating the mighty triangle. You’ll learn about some of OpenGL’s built-in quadric surface generation functions and ways to use automatic tessellation to break complex shapes into smaller, more digestible pieces. You’ll also explore the utility functions that evaluate Bézier and NURBS curves and surfaces. You can use these functions to create complex shapes with an amazingly small amount of code.

Chapter 11—It’s All About the Pipeline: Faster Geometry Throughput. For this chapter, we introduce OpenGL display lists, vertex arrays, and vertex buffer objects for improving performance and organizing your models. You’ll also learn how to create a detailed analysis showing how to best represent large, complex models.

Chapter 12—Interactive Graphics. This chapter explains two OpenGL features: selection and feedback. These groups of functions make it possible for the user to interact with objects in the scene. You can also get rendering details about any single object in the scene.

Chapter 13—Occlusion Queries: Why Do More Work Than You Need To? Here, you’ll learn about the OpenGL occlusion query mechanism. This feature effectively lets you perform an inexpensive test-render of objects in your scene to find out whether they will be hidden behind other objects, in which case you can save time by not drawing the actual full-detail version.

Chapter 14—Depth Textures and Shadows. This chapter covers OpenGL’s depth textures and shadow comparisons. You’ll learn how to introduce real-time shadow effects to your scene, regardless of the geometry’s complexity.

Part II: The New Testament

In the second part of the book, you’ll find chapters on the new features in OpenGL supporting programmable hardware, in particular the OpenGL Shading Language (GLSL). These chapters don’t represent just the newest OpenGL features, they cover the fundamental shift that has occurred in graphics programming—a shift that is fundamentally different, yet complementary, and descended from the old fixed-pipeline-based hardware.

Chapter 15—Programmable Pipeline: This Isn’t Your Father’s OpenGL. Out with the old, in with the new. This chapter revisits the conventional fixed-functionality pipeline before introducing the new programmable vertex and fragment pipeline stages.

Programmability via the OpenGL Shading Language allows you to customize your rendering in ways never before possible.

Chapter 16—Vertex Shading: Do-It-Yourself Transform, Lighting, and Texgen. This chapter illustrates the usage of vertex shaders by surveying a handful of examples, including lighting, fog, squash and stretch, and skinning.

Chapter 17—Fragment Shading: Empower Your Pixel Processing. You learn by example—with a variety of fragment shaders. Examples include per-pixel lighting, color conversion, image processing, bump mapping, and procedural texturing. Some of these examples also use vertex shaders; these examples are representative of real-world usage, where you often find vertex and fragment shaders paired together.

Chapter 18—Advanced Buffers. Here, we discuss some of the latest and most exciting features in OpenGL, including offscreen accelerated rendering, faster ways to copy pixel data asynchronously, and floating-point color data for textures and color buffers.

Part III: The Apocrypha

Where do we put material that does not belong in the OpenGL canon? The Apocrypha! The third and last part of the book is less about OpenGL than about how different operating systems interface with and make use of OpenGL. Here we wander outside the “official” OpenGL specification to see how OpenGL is supported and interfaced with on Windows, Mac OS X, Linux, and hand-held devices.

Chapter 19—Wiggle: OpenGL on Windows. Here, you’ll learn how to write real Windows (message-based) programs that use OpenGL. You’ll learn about Microsoft’s “wiggle” functions that glue OpenGL rendering code to Windows device contexts. You’ll also learn how to respond to Windows messages for clean, well-behaved OpenGL applications. Yes, we also talk about OpenGL on Vista.

Chapter 20—OpenGL on Mac OS X. In this chapter, you'll learn how to use OpenGL in native Mac OS X applications. Sample programs show you how to start working with GLUT, Carbon, or Cocoa using the Xcode development environment.

Chapter 21—GLX: OpenGL on Linux. This chapter discusses GLX, the OpenGL extension used to support OpenGL applications through the X Window System on UNIX and Linux. You'll learn how to create and manage OpenGL contexts as well as how to create OpenGL drawing areas.

Chapter 22—OpenGL ES: OpenGL on the Small. This chapter is all about how OpenGL is pared down to fit on hand-held and embedded devices. We cover what's gone, what's new, and how to get going even with an emulated environment.

Conventions Used in This Book

The following typographic conventions are used in this book:

- Code lines, commands, statements, variables, and any text you type or see onscreen appear in a computer typeface.
- Placeholders in syntax descriptions appear in an *italic computer* typeface. You should replace the placeholder with the actual filename, parameter, or whatever element it represents.
- *Italics* highlight technical terms when they first appear in the text and are being defined.

About the Companion Web Site

This is the first time this book has shipped without a CD-ROM. Welcome to the age of the Internet! Instead, all our source code is available online at our support Web site:

www.opengl.org/superbible

Here you'll find the source code to all the sample programs in the book, as well as prebuilt projects for Developers Studio (Windows), and Xcode (Mac OS X). For Linux users we'll have make files for command-line building of the projects as well. We even plan to post a few tutorials, so check back from time to time, even after you've downloaded all the source code.