Contents

Figures ..... **xxi**

Tables ..... **xxvii**

Examples ..... **xxxi**

About This Guide ..... **xxxix**
What This Guide Contains xxxix
What's New in This Edition xli
What You Should Know Before Reading This Guide xli
How to Obtain the Sample Code xlii
Errata xliii
Style Conventions xliv
About the OpenGL Series xliv

Acknowledgments ..... **xlv**

1. **Introduction to OpenGL** 1
What Is OpenGL? 2
Your First Look at an OpenGL Program 4
OpenGL Syntax 8
OpenGL's Rendering Pipeline 10
Preparing to Send Data to OpenGL 11
Sending Data to OpenGL 11
Vertex Shading 11
Tessellation Shading 12
Geometry Shading 12
Primitive Assembly 12
Clipping 12
Rasterization 12

Fragment Shading 13
Per-Fragment Operations 13
Our First Program: A Detailed Discussion 14
Entering **main()** 14
OpenGL Initialization 16
Our First OpenGL Drawing 28

2. Shader Fundamentals 33
Shaders and OpenGL 34
OpenGL's Programmable Pipeline 35
An Overview of the OpenGL Shading Language 37
Creating Shaders with GLSL 37
Storage Qualifiers 46
Statements 49
Computational Invariance 55
Shader Preprocessor 57
Compiler Control 59
Global Shader-Compilation Option 60
Interface Blocks 61
Uniform Blocks 61
Specifying Uniform Blocks in Shaders 62
Accessing Uniform Blocks from Your Application 64
Buffer Blocks 71
In/Out Blocks, Locations, and Components 71
Compiling Shaders 72
Shader Subroutines 79
GLSL Subroutine Setup 80
Selecting Shader Subroutines 81
Separate Shader Objects 84
SPIR-V 86
Reasons to Choose SPIR-V 86
Using SPIR-V with OpenGL 88
Using GLSL to Generate SPIR-V for OpenGL 90
Glslang 92
What's Inside SPIR-V? 92

3. Drawing with OpenGL 95
OpenGL Graphics Primitives 96
Points 97

Lines, Strips, and Loops 98
Triangles, Strips, and Fans 99
Data in OpenGL Buffers 102
Creating and Allocating Buffers 102
Getting Data into and out of Buffers 104
Accessing the Content of Buffers 110
Discarding Buffer Data 116
Vertex Specification 117
VertexAttribPointer in Depth 117
Static Vertex-Attribute Specification 121
OpenGL Drawing Commands 124
Restarting Primitives 134
Instanced Rendering 137

4. Color, Pixels, and Fragments 151
Basic Color Theory 152
Buffers and Their Uses 154
Clearing Buffers 156
Masking Buffers 158
Color and OpenGL 158
Color Representation and OpenGL 159
Smoothly Interpolating Data 160
Testing and Operating on Fragments 163
Scissor Test 164
Multisample Fragment Operations 165
Stencil Test 166
Stencil Examples 168
Depth Test 170
Blending 173
Logical Operations 178
Occlusion Query 179
Conditional Rendering 183
Multisampling 185
Sample Shading 187
Per-Primitive Antialiasing 188
Antialiasing Lines 190
Antialiasing Polygons 190
Reading and Copying Pixel Data 191
Copying Pixel Rectangles 193

5. **Viewing Transformations, Culling, Clipping, and Feedback** **197**
Viewing 198
Viewing Model 199
Camera Model 199
Orthographic Viewing Model 203
User Transformations 203
Matrix Multiply Refresher 205
Homogeneous Coordinates 207
Linear Transformations and Matrices 210
Transforming Normals 222
OpenGL Matrices 223
OpenGL Transformations 226
Advanced: User Culling and Clipping 228
Controlling OpenGL Transformations 230
Transform Feedback 231
Transform Feedback Objects 231
Transform Feedback Buffers 233
Configuring Transform Feedback Varyings 236
Starting and Stopping Transform Feedback 244
Transform Feedback Example—Particle System 247

6. **Textures and Framebuffers** **255**
Introduction to Texturing 257
Basic Texture Types 258
Creating and Initializing Textures 260
Proxy Textures 264
Specifying Texture Data 265
Explicitly Setting Texture Data 266
Loading Textures from Buffers 268
Loading Images from Files 270
Retrieving Texture Data 274
Texture Data Layout 275
Texture Formats 279
Internal Formats 280
External Formats 283
Compressed Textures 285
Sampler Objects 288
Sampler Parameters 289

Using Textures 291
Texture Coordinates 293
Arranging Texture Data 297
Using Multiple Textures 299
Complex Texture Types 302
3D Textures 302
Array Textures 304
Cube-Map Textures 305
Shadow Samplers 313
Depth-Stencil Textures 314
Buffer Textures 314
Texture Views 317
Filtering 321
Linear Filtering 322
Using and Generating Mipmaps 324
Calculating the Mipmap Level 329
Mipmap Level-of-Detail Control 330
Advanced Texture Lookup Functions 331
Explicit Level of Detail 331
Explicit Gradient Specification 332
Texture Fetch with Offsets 332
Projective Texturing 333
Texture Queries in Shaders 334
Gathering Texels 336
Combining Special Functions 336
Bindless Textures 337
Texture Handles 338
Texture Residency 339
Sampling Bindless Textures 340
Sparse Textures 341
Sparse Texture Commitment 342
Sparse Texture Pages 343
Point Sprites 344
Textured Point Sprites 344
Controlling the Appearance of Points 347
Framebuffer Objects 348
Rendering to Texture Maps 351
Discarding Rendered Data 353

Shared Tessellated Edges and Cracking 518
Displacement Mapping 520

10. Geometry Shaders 521
Creating a Geometry Shader 523
Geometry Shader Inputs and Outputs 526
Geometry Shader Inputs 526
Special Geometry Shader Primitives 530
Geometry Shader Outputs 535
Producing Primitives 537
Culling Geometry 537
Geometry Amplification 539
Advanced Transform Feedback 544
Multiple Output Streams 545
Primitive Queries 550
Using Transform Feedback Results 551
Geometry Shader Instancing 561
Multiple Viewports and Layered Rendering 562
Viewport Index 563
Layered Rendering 568
Chapter Summary 572
Geometry Shader Redux 572
Geometry Shader Best Practices 574

11. Memory 577
Using Textures for Generic Data Storage 578
Binding Textures to Image Units 582
Reading and Writing to Images 585
Shader Storage Buffer Objects 589
Writing Structured Data 590
Atomic Operations and Synchronization 591
Atomic Operations on Images 591
Atomic Operations on Buffers 600
Sync Objects 602
Image Qualifiers and Barriers 606
High-Performance Atomic Counters 617
Example: Order-Independent Transparency 621
Principles of Operation 622
Initialization 623

Rendering 625
Sorting and Blending 629
Results 633

12. Compute Shaders 635
Overview 636
Workgroups and Dispatch 637
Knowing Where You Are 642
Communication and Synchronization 644
Communication 645
Synchronization 646
Examples 648
Physical Simulation 648
Image Processing 654
Chapter Summary 659
Compute Shader Redux 660
Compute Shader Best Practices 661

A. Support Libraries 663
Basics of GLFW: The OpenGL Utility Framework 664
Initializing and Creating a Window 664
Handling User Input 667
Controlling the Window 671
Shutting Down Cleanly 675
GL3W: OpenGL Glue 676

B. OpenGL ES and WebGL 679
OpenGL ES 680
WebGL 682
Setting Up WebGL Within an HTML5 Page 682
Initializing Shaders in WebGL 684
Initializing Vertex Data in WebGL 688
Using Texture Maps in WebGL 689

C. Built-in GLSL Variables and Functions 693
Built-in Variables 693
Built-in Variable Declarations 694
Built-in Variable Descriptions 696
Built-in Constants 705

Built-in Functions 707
Angle and Trigonometry Functions 708
Exponential Functions 711
Common Functions 712
Floating-Point Pack and Unpack Functions 718
Geometric Functions 720
Matrix Functions 722
Vector Relational Functions 724
Integer Functions 725
Texture Functions 728
Atomic-Counter Functions 742
Atomic Memory Functions 743
Image Functions 745
Fragment Processing Functions 749
Geometry Shader Functions 752
Shader Invocation Control Functions 753
Shader Memory Control Functions 754

D. State Variables 757
The Query Commands 757
OpenGL State Variables 765
Current Values and Associated Data 766
Vertex Array Object State 767
Vertex Array Data 769
Buffer Object State 770
Transformation State 771
Coloring State 772
Rasterization State 772
Multisampling 774
Textures 775
Pixel Operations 785
Framebuffer Controls 788
Framebuffer State 789
Renderbuffer State 793
Pixel State 795
Shader Object State 798
Shader Program Pipeline Object State 799
Shader Program Object State 800
Program Interface State 812

Program Object Resource State 813
Vertex and Geometry Shader State 816
Query Object State 816
Image State 817
Transform Feedback State 818
Atomic Counter State 819
Shader Storage Buffer State 820
Sync Object State 821
Hints 822
Compute Dispatch State 822
Implementation-Dependent Values 823
Tessellation Shader Implementation-Dependent Limits 829
Geometry Shader Implementation-Dependent Limits 832
Fragment Shader Implementation-Dependent Limits 834
Implementation-Dependent Compute Shader Limits 836
Implementation-Dependent Shader Limits 838
Implementation-Dependent Debug Output State 843
Implementation-Dependent Values 844
Internal Format-Dependent Values 846
Implementation-Dependent Transform Feedback Limits 846
Framebuffer-Dependent Values 847
Miscellaneous 847

E. Homogeneous Coordinates and Transformation Matrices 849
Homogeneous Coordinates 849
Transforming Vertices 850
Transforming Normals 850
Transformation Matrices 851
Translation 851
Scaling 852
Rotation 852
Perspective Projection 853
Orthographic Projection 853

F. Floating-Point Formats for Textures, Framebuffers, and Renderbuffers 855
Reduced-Precision Floating-Point Values 855
16-Bit Floating-Point Values 856
10- and 11-Bit Unsigned Floating-Point Values 858

G. Debugging and Profiling OpenGL 863
Creating a Debug Context 863
Debug Output 866
Debug Messages 867
Filtering Messages 870
Application-Generated Messages 872
Debug Groups 874
Naming Objects 875
Profiling 877
Profiling Tools 877
In-Application Profiling 879

H. Buffer Object Layouts 883
Using Standard Layout Qualifiers 883
The `std140` Layout Rules 884
The `std430` Layout Rules 885

Glossary 887

Index 917