

Contents

Figures	xv
Tables	xxi
Listings	xxiii
Foreword	xxxi
Preface	xxxiii
The Architecture of the Book	xxxiv
What's New in This Edition	xxxvi
How to Build the Examples	xxxvii
Errata	xxxviii
Note from the Publisher	xxxviii
Acknowledgments	xxxix
About the Author	xli
I Foundations	1
1 Introduction	3
OpenGL and the Graphics Pipeline	4
The Origins and Evolution of OpenGL	6
Core Profile OpenGL	8
Primitives, Pipelines, and Pixels	10
Summary	11
2 Our First OpenGL Program	13
Creating a Simple Application	14
Using Shaders	17

Drawing Our First Triangle	24
Summary	26
3 Following the Pipeline	27
Passing Data to the Vertex Shader	28
Vertex Attributes	30
Passing Data from Stage to Stage	31
Interface Blocks	33
Tessellation	33
Tessellation Control Shaders	35
The Tessellation Engine	35
Tessellation Evaluation Shaders	37
Geometry Shaders	39
Primitive Assembly, Clipping, and Rasterization	39
Clipping	40
Viewport Transformation	41
Culling	43
Rasterization	43
Fragment Shaders	47
Framebuffer Operations	47
Pixel Operations	48
Compute Shaders	49
Using Extensions in OpenGL	50
Enhancing OpenGL with Extensions	54
Summary	54
4 Math for 3D Graphics	55
Is This the Dreaded Math Chapter	56
A Crash Course in 3D Graphics Math	57
Vectors, or Which Way Is Which?	60
Common Vector Operators	64
Matrices	66
Matrix Construction and Operators	69
Understanding Transformations	70
Coordinate Spaces in OpenGL	72
Coordinate Transformations	80
Concatenating Transformations	81
Quaternions	83
The Model-View Transform	86
Projection Transformations	89
Interpolation, Lines, Curves, and Splines	89

Curves	91
Splines	94
Summary	97
5 Data	99
Buffers	100
Creating Buffers and Allocating Memory	100
Filling and Copying Data in Buffers	107
Feeding Vertex Shaders from Buffers	109
Uniforms	117
Default Block Uniforms	117
Uniform Blocks	121
Using Uniforms to Transform Geometry	135
Shader Storage Blocks	140
Synchronizing Access to Memory	145
Atomic Counters	147
Synchronizing Access to Atomic Counters	151
Textures	152
Creating and Initializing Textures	152
Texture Targets and Types	154
Reading from Textures in Shaders	156
Loading Textures from Files	160
Controlling How Texture Data Is Read	164
Array Textures	177
Writing to Textures in Shaders	182
Synchronizing Access to Images	194
Texture Compression	195
Texture Views	199
Summary	203
6 Shaders and Programs	205
Language Overview	206
Data Types	206
Built-In Functions	213
Compiling, Linking, and Examining Programs	219
Getting Information from the Compiler	219
Getting Information from the Linker	223
Separate Programs	225
Shader Subroutines	231
Program Binaries	235
Summary	238

II In Depth

239

7 Vertex Processing and Drawing Commands	241
Vertex Processing	242
Vertex Shader Inputs	242
Vertex Shader Outputs	247
Drawing Commands	249
Indexed Drawing Commands	249
Instancing	255
Indirect Draws	269
Storing Transformed Vertices	278
Using Transform Feedback	279
Starting, Pausing, and Stopping Transform Feedback	284
Ending the Pipeline with Transform Feedback	286
Transform Feedback Example: Physical Simulation	286
Clipping	296
User-Defined Clipping	300
Summary	303
8 Primitive Processing	305
Tessellation	306
Tessellation Primitive Modes	307
Tessellation Subdivision Modes	316
Passing Data between Tessellation Shaders	319
Communication between Shader Invocations	322
Tessellation Example: Terrain Rendering	323
Tessellation Example: Cubic Bézier Patches	329
Geometry Shaders	333
The Pass-Through Geometry Shader	334
Using Geometry Shaders in an Application	336
Discarding Geometry in the Geometry Shader	341
Modifying Geometry in the Geometry Shader	344
Generating Geometry in the Geometry Shader	346
Changing the Primitive Type in the Geometry Shader	349
Multiple Streams of Storage	352
New Primitive Types Introduced by the Geometry Shader	353
Multiple Viewport Transformations	361
Summary	364
9 Fragment Processing and the Framebuffer	365
Fragment Shaders	366
Interpolation and Storage Qualifiers	366

Per-Fragment Tests	369
Scissor Testing	369
Stencil Testing	372
Depth Testing	376
Early Testing	380
Color Output	382
Blending	382
Logical Operations	387
Color Masking	388
Off-Screen Rendering	390
Multiple Framebuffer Attachments	395
Layered Rendering	397
Framebuffer Completeness	403
Rendering in Stereo	407
Antialiasing	412
Antialiasing by Filtering	413
Multi-Sample Antialiasing	415
Multi-Sample Textures	417
Sample Rate Shading	421
Centroid Sampling	424
Advanced Framebuffer Formats	428
Rendering with No Attachments	428
Floating-Point Framebuffers	430
Integer Framebuffers	444
The sRGB Color Space	446
Point Sprites	448
Texturing Points	449
Rendering a Star Field	450
Point Parameters	453
Shaped Points	454
Rotating Points	456
Getting at Your Image	458
Reading from a Framebuffer	458
Copying Data between Framebuffers	461
Reading Back Texture Data	464
Summary	466
10 Compute Shaders	467
Using Compute Shaders	468
Executing Compute Shaders	469
Compute Shader Communication	474
Examples	479

Compute Shader Parallel Prefix Sum	479
Compute Shader Flocking	492
Summary	502
11 Advanced Data Management	503
Eliminating Binding	504
Sparsely Populated Textures	516
Texture Compression	516
The RGTC Compression Scheme	519
Generating Compressed Data	525
Packed Data Formats	527
High-Quality Texture Filtering	531
Summary	533
12 Controlling and Monitoring the Pipeline	533
Queries	534
Occlusion Queries	535
Timer Queries	545
Transform Feedback Queries	549
Pipeline State Queries	555
Synchronization in OpenGL	556
Draining the Pipeline	557
Synchronization and Fences	562
Summary	565
III In Practice	565
13 Rendering Techniques	567
Lighting Models	568
The Phong Lighting Model	568
Blinn-Phong Lighting	577
Rim Lighting	579
Normal Mapping	582
Environment Mapping	587
Material Properties	597
Casting Shadows	599
Atmospheric Effects	605
Non-Photo-Realistic Rendering	610
Cell Shading: Texels as Light	610

Alternative Rendering Methods	613
Deferred Shading	613
Screen-Space Techniques	613
Rendering without Triangles	624
Two-Dimensional Graphics	631
Distance Field Textures	647
Bitmap Fonts	647
Summary	656
14 High-Performance OpenGL	661
Optimizing CPU Performance	661
Multi-Threading in OpenGL	662
Packet Buffers	668
Low-Overhead OpenGL	677
Indirect Rendering	678
GPU Work Generation	683
Zero Copy	691
Performance Analysis Tools	699
Windows Performance Toolkit and GPUView	699
GPU PerfStudio	703
Tuning Your Application for Speed	706
Summary	726
15 Debugging and Stability	729
Debugging Your Applications	730
Debug Contexts	730
Security and Robustness	737
Graphics Reset	737
Range-Checked Reads	740
Summary	742
A Further Reading	743
B The SBM File Format	749
C The SuperBible Tools	759
Glossary	797
Index	805