

Contents

Figures	xiii
Tables	xv
Listings	xvii
About This Book	xxi
About the Sample Code	xxiv
Errata	xxv
Acknowledgments	xxvii
About the Author	xxix
 1 Overview of Vulkan	 1
Introduction	2
Instances, Devices, and Queues	3
The Vulkan Instance	4
Vulkan Physical Devices	7
Physical Device Memory	10
Device Queues	12
Creating a Logical Device	14
Object Types and Function Conventions	18
Managing Memory	19
Multithreading in Vulkan	20
Mathematical Concepts	21
Vectors and Matrices	21
Coordinate Systems	22

Enhancing Vulkan	23
Layers	23
Extensions	27
Shutting Down Cleanly	31
Summary	33
Memory and Resources	35
Host Memory Management	36
Resources	41
Buffers	42
Formats and Support	45
Images	49
Resource Views	63
Destroying Resources	72
Device Memory Management	73
Allocating Device Memory	75
Host Access to Device Memory	78
Binding Memory to Resources	81
Sparse Resources	85
Summary	93
Queues and Commands	95
Device Queues	96
Creating Command Buffers	97
Recording Commands	101
Recycling Command Buffers	105
Submission of Commands	107
Summary	109
Moving Data	111
Managing Resource State	112
Pipeline Barriers	113
Global Memory Barriers	116
Buffer Memory Barriers	119
Image Memory Barriers	120
Clearing and Filling Buffers	123
Clearing and Filling Images	125
Copying Image Data	128
Copying Compressed Image Data	132
Stretching Images	133
Summary	135

5 Presentation	137
Presentation Extension	138
Presentation Surfaces	138
Presentation on Microsoft Windows	139
Presentation on Xlib-Based Platforms	140
Presentation with Xcb	142
Swap Chains	143
Full-Screen Surfaces	152
Performing Presentation	159
Cleaning Up	162
Summary	163
6 Shaders and Pipelines	165
An Overview of GLSL	166
An Overview of SPIR-V	169
Representation of SPIR-V	169
Handling SPIR-V to Vulkan	173
Pipelines	175
Compute Pipelines	175
Creating Pipelines	177
Specialization Constants	178
Accelerating Pipeline Creation	181
Binding Pipelines	186
Executing Work	186
Resource Access in Shaders	188
Descriptor Sets	189
Binding Resources to Descriptor Sets	199
Binding Descriptor Sets	207
Uniform, Texel, and Storage Buffers	208
Push Constants	212
Sampled Images	215
Summary	224
7 Graphics Pipelines	225
The Logical Graphics Pipeline	226
Renderpasses	230
The Framebuffer	237
Creating a Simple Graphics Pipeline	240
Graphics Shader Stages	242
Vertex Input State	245
Input Assembly	251
Tessellation State	255

Viewport State	256
Rasterization State	257
Multisample State	260
Depth and Stencil State	261
Color Blend State	262
Dynamic State	264
Summary	266
8 Drawing	267
Getting Ready to Draw	268
Vertex Data	271
Indexed Draws	272
Index-Only Rendering	276
Reset Indices	278
Instancing	280
Indirect Draws	281
Summary	286
9 Geometry Processing	287
Tessellation	288
Tessellation Configuration	288
Tessellation Variables	295
Tessellation Example: Displacement Mapping	304
Geometry Shaders	309
Cutting Primitives	315
Geometry Shader Instancing	316
Programmable Point Size	319
Line Width and Rasterization	321
User Clipping and Culling	324
The Viewport Transformation	331
Summary	335
10 Fragment Processing	337
Scissor Testing	338
Depth and Stencil Operations	340
Depth Testing	341
Stencil Testing	346
Early Fragment Tests	348
Multisample Rendering	350
Sample Rate Shading	352
Multisample Resolves	355
Logic Operations	357

Fragment Shader Outputs	358
Color Blending	362
Summary	366
11 Synchronization	367
Fences	368
Events	376
Semaphores	381
Summary	384
12 Getting Data Back	387
Queries	388
Executing Queries	390
Timing Queries	397
Reading Data with the Host	398
Summary	400
13 Multipass Rendering	401
Input Attachments	402
Attachment Contents	408
Attachment Initialization	408
Render Areas	411
Preserving Attachment Content	412
Secondary Command Buffers	420
Summary	423
Appendix: Vulkan Functions	425
Glossary	429
Index	443
Figure 7.1 The full Vulkan graphics pipeline	227
Figure 7.2 Strip, fan, and triangle topologies	252
Figure 7.3 Triangles with adjacency topology	254
Figure 7.4 Triangle strip with adjacency topology	254
Figure 8.1 Index Data Flow	274
Figure 8.2 The Effect of Primitive Restart on Triangle Strips	279
Figure 8.3 Many Instances, One Buffer	281
Figure 9.1 Quad Tessellation	290
Figure 9.2 Triangle Tessellation	290