Contents

Foreword xix

Preface xxiii

1. The Fixed-Function Graphics Pipeline 1

The Traditional View 2

The Vertex Operation 2
The Fragment Processing Part of the Pipeline 6
State in the Graphics Pipeline 7

How the Traditional View Is Implemented 8
Vertex Processing 9
Rendering Processing 10
Homogeneous Coordinates in the Fixed-Function Pipeline 14
Vertex Arrays 17
Conclusions 20
Exercises 21

2. **OpenGL Shader Evolution** **25**
History of Shaders 27
OpenGL Shader History 30
OpenGL 2.0/GLSL 1.10 30
OpenGL 3.x/GLSL 3.30 31
OpenGL 4.0/GLSL 4.00 32
OpenGL 4.x/GLSL 4.x0 33
What's Behind These Developments? 34
OpenGL ES 34
How Can You Respond to These Changes? 35
Our Approach in this Book 36
Variable Name Convention 36
Exercises 37

3. **Fundamental Shader Concepts** **39**
Shaders in the Graphics Pipeline 39
Vertex Shaders 42
Fragment Shaders 47
Tessellation Shaders 50
Geometry Shaders 53

The GLSL Shading Language 54

Passing Data from Your Application into Shaders 59

Defining Attribute Variables in Your Application 59
Defining Uniform Variables in Your Application 62
A Convenient Way to Transition to the Newer Versions of GLSL 64

Exercises 67

4. **Using glman** 69

Using glman 71

Loading a GLIB File 72
Editing GLIB and Shader Source Files 72

GLIB Scene Creation 72

Window and Viewing 73
Transformations 73
Defining Geometry 73
Specifying Textures 76
Specifying Shaders 77
Miscellaneous 78
Specifying Uniform Variables 79
Examples of GLIB Files 81

More on Textures and Noise 82

Using Textures 82
Using Noise 84

Functions in the glman Interface Window 86

Generating and Displaying a Hardcopy of Your Scene 86
Global Scene Transformation 86
Eye Transformation 87
Object Picking and Transformation 87
Texture Transformation 88
Monitoring the Frame Rate 88
Miscellaneous 89

Exercises 90

5. The GLSL Shader Language 91

Factors that Shape Shader Languages 92
- Graphics Card Capabilities 93

General GLSL Language Concepts 95
- Shared Namespace 95
- Extended Function and Operator Capabilities 96
- New Functions 97
- New Variable Types 97
- New Function Parameter Types 98

Language Details 98
- Omitted Language Features 98
- New Matrix and Vector Types 99
- Name Sets 100
- Vector Constructors 101
- Functions Extended to Matrices and Vectors 102
- Operations Extended to Matrices and Vectors 105
- New Functions 106
- Swizzle 112
- New Function Parameter Types 112
- Const 113

Compatibility Mode 114
- Defining Compatibility Mode 114
- OpenGL 2.1 Built-in Data Types 114

Summary 120

Exercises 120

6. Lighting 123

The ADS Lighting Model 124
- The ADS Lighting Model Function 125

Types of Lights 127
- Positional Lights 128
- Directional Lights 128
- Spot Lights 129

Setting Up Lighting for Shading 131
Flat Shading 132
Smooth (Gouraud) Shading 133
Phong Shading 134
Anisotropic Shading 135
Exercises 137

7. Vertex Shaders 139
Vertex Shaders in the Graphics Pipeline 140
Input to Vertex Shaders 140
Output from Vertex Shaders 142
Fixed-Function Processing After the Vertex Shader 145
The Relation of Vertex Shaders to Tessellation Shaders 146
The Relation of Vertex Shaders to Geometry Shaders 146
Replacing Fixed-Function Graphics with Vertex Shaders 146
Standard Vertex Processing 147
Going Beyond the Fixed-Function Pipeline with Vertex Shaders 148
Vertex Modification 148
Issues in Vertex Shaders 151
Creating Normals 152
Summary 153
Exercises 154

8. Fragment Shaders and Surface Appearance 157
Basic Function of a Fragment Shader 158
Inputs to Fragment Shaders 158
Particularly Important "In" Variables for the Fragment Shader 161
Coordinate Systems 162

Fragment Shader Processing 163
Outputs from Fragment Shaders 163
Replacing Fixed-Function Processing with Fragment Shaders 163
Shading 164
Traditional Texture Mapping 165
False Coloring 166
What Follows a Fragment Shader? 168
Additional Shader Effects 169
Discarding Pixels 169
Phong Shading 169
Shading with Analytic Normals 170
Anisotropic Shading 172
Data-Driven Coloring 173
Images Using Other Data 175
Exercises 177

9. Surface Textures in the Fragment Shader 179

Texture Coordinates 180
Traditional Texture Mapping 180
GLSL Texture Mapping 182
The Texture Context 184
Texture Environments in the Fixed-Function World 185
Texture Sampling Parameters 186
Samplers 186
Procedural Textures 187
Bump Mapping 193
Cube Maps 200
Render to Texture 205
Render to Texture for Multipass Rendering in glman 209
Exercises 212

10. Noise 213

Fundamental Noise Concepts 214

Three Types of Noise: Value, Gradient, and Value+Gradient 214
Cubic and Quintic Interpolation 215
Noise Equations 216

Other Noise Concepts 220

Fractional Brownian Motion (FBM, 1/f, Octaves) 220
Noise in Two and Three Dimensions 221
Using Noise with glman 223
Using Noise with the Built-In GLSL Functions 225
Turbulence 225

Some Examples of Noise in Different Environments 228

Marble Shader 230
Cloud Shader 231
Wood Shader 233

Advanced Noise Topics 235

Using Noisegraph 235

Exercises 237

11. Image Manipulation with Shaders 239

Basic Concepts 240

Single-Image Manipulation 241

Luminance 241
CMYK Conversions 243
Hue Shifting 246
Image Filtering 248
Image Blurring 249
Chromakey Images 251
Stereo Anaglyphs 252
3D TV 256
Edge Detection 259
Embossing 260

Toon Shader 262
Artistic Effects 264
Image Flipping, Rotation, and Warping 265

The Image Blending Process 270

Blending an Image with a Constant Base Image 271

Color Negative 272
Brightness 273
Contrast 274

Blending an Image with a Version of Itself 275

Saturation 275
Sharpness 276

Blending Two Different Images 277

Other Combinations 278
Image Transitions 281

Notes 286

Exercises 287

12. Geometry Shader Concepts and Examples 291

What Does the Geometry Shader Do? 292

New Adjacency Primitives 294
Layouts for Input and Output Variables 295
New OpenGL API Functions 296
New GLSL Variables and Variable Types 299
Communication between a Vertex or Tessellation Shader and a Geometry Shader 299

Normals in Geometry Shaders 301

Examples 301

Bézier Curves 301
Shrinking Triangles 303
Sphere Subdivision 305
3D Object Silhouettes 309

Exercises 312

13. Tessellation Shaders 315

What Are Tessellation Shaders? 315
- Tessellation Shaders or Geometry Shaders? 317

Tessellation Shader Concepts 318
- Issues in Setting Tessellation Levels 323

Examples 323
- Isolines 324
- Bézier Surface 327
- Sphere Subdivision 334
- Whole Sphere Subdivision while Adapting to Screen Coverage 341
- PN Triangles 344

Summary 350

Exercises 351

14. The GLSL API 353

Shaders in the OpenGL Programming Process 353
- Handling OpenGL Extensions 355

How Is a GLSL Shader Program Created? 355

Creating and Compiling Shader Objects 357
- The CheckGLErrors Function 359

Creating, Attaching, Linking, and Activating Shader Programs 360
- Creating a Shader Program and Attaching Shader Objects 361
- Linking Shader Programs 361
- Activating a Shader Program 362

Passing Data into Shaders 364
- Defining Uniform Variables in Your Application 364
- Uniform Variables in Compatibility Mode 367
- Defining Attribute Variables in Your Application 368
- Attribute Variables in Compatibility Mode 370
- A C++ Class to Handle Shader Program Creation 371

Exercises 372

15. Using Shaders for Scientific Visualization 375

Image-Based Visualization Techniques 376
- Image Negative 376
- Image Edge Detection 377
- Toon Rendering 377

Hyperbolic Geometry 378

3D Scalar Data Visualization 381
- Point Clouds 383
- Cutting Planes 387
- Volume Probe 390
- Direct Volume Rendering 392

More on Transfer Functions 398

Passing in Data Values with Your Geometry 403

Terrain Bump-Mapping 405

Flow Visualization 408
- 2D Line Integral Convolution 408
- 3D Line Integral Convolution 411
- Extruding Objects for Streamlines 413

Geometry Visualization 416
- Silhouettes 416
- Hedgehog Plots 417

Exercises 420

16. Serious Fun 425

Light Interference 426
- Diffraction Gratings 427
- Oil Slicks 431

Lens Effects 433

Bathroom Glass 438

Atmospheric Effects 440

Rainbows 441

The Glory 445

Fun with One 448

Using the glman Timer Function 449

Disco Ball 449

Fog, with and without Noise 452

Morphing 3D Geometry 453

Algorithmic Art 456

Connett Circles 456

Making Information Visible Through Motion 459

An Explosion Shader 461

Exercises 462

Appendices

A. GLSLProgram C++ Class 465

B. Matrix4 C++ Class 469

C. Vec3 C++ Class 473

D. Vertex Array Class 477

References 483

Index 487