Contents

Preface xiii

Part I - Rendering Pipeline

1 Introduction 1
- 1.1 Computer Graphics Production 1
- 1.2 Graphics API 5

2 Mathematics: Basics 7
- 2.1 Matrices and Vectors 7
- 2.2 Coordinate System and Basis 9
- 2.3 Dot Product 10
- 2.4 Cross Product 11
- 2.5 Line, Ray, and Linear Interpolation 13

3 Modeling 17
- 3.1 Polygon Mesh 17
  - 3.1.1 Polygon Mesh Creation* 19
  - 3.1.2 Polygon Mesh Representation 23
- 3.2 Surface Normals 24
  - 3.2.1 Triangle Normals 24
  - 3.2.2 Vertex Normals 25
- 3.3 Polygon Mesh Export and Import 26

4 Spaces and Transforms 31
- 4.1 2D Transforms and Matrix Representations 31
  - 4.1.1 Scaling 31
  - 4.1.2 Rotation 32
  - 4.1.3 Translation and Homogeneous Coordinates 33
  - 4.1.4 Composition of 2D Transforms 35
- 4.2 Affine Transform 38
- 4.3 3D Transforms and Matrix Representations 39
  - 4.3.1 Scaling 39
  - 4.3.2 Rotation 40
  - 4.3.3 Translation and Homogeneous Coordinates 41

4.4 Application: World Transform 42
4.5 Rotation and Object-space Basis 46
4.6 Inverse Transforms 48

5 Vertex Processing 53
5.1 World Transform Revisited 54
5.2 View Transform 57
5.2.1 Camera Space 57
5.2.2 View Matrix for Space Change 58
5.3 Right-hand System versus Left-hand System 60
5.4 Projection Transform 62
5.4.1 View Frustum 62
5.4.2 Projection Matrix and Clip Space 63
5.4.3 Derivation of Projection Matrix* 67

6 OpenGL ES and Shader 75
6.1 OpenGL ES and Shading Language 75
6.2 Vertex Shader 75
6.3 OpenGL ES for Shaders 77
6.4 Attributes and Uniforms 78
6.4.1 Attributes and Buffer Objects 79
6.4.2 Uniforms 82
6.5 Drawcalls 83

7 Rasterizer 87
7.1 Clipping 87
7.2 Perspective Division 88
7.3 Back-face Culling 89
7.3.1 Concept 90
7.3.2 Implementation 90
7.4 Viewport Transform 92
7.5 Scan Conversion 94

8 Image Texturing 103
8.1 Texture Coordinates 103
8.2 Surface Parameterization 106
8.3 Texture Definition in GL 108
8.4 Texture Wrapping 109
8.5 Texture Filtering 111
8.5.1 Magnification 113
8.5.2 Minification 113
8.6 Mipmapping 114
8.6.1 Mipmap Construction 114
8.6.2 Mipmap Filtering 115
8.7 Texture Filtering in GL 118

8.8 Mipmappig Examples in GL* 118
8.9 Fragment Shader for Texturing 121

9 Lighting 127
9.1 Phong Lighting Model 127
9.1.1 Diffuse Reflection 128
9.1.2 Specular Reflection 130
9.1.3 Ambient Reflection 132
9.1.4 Emissive Light 132
9.2 Shaders for Phong Lighting 133

10 Output Merger 139
10.1 Z-buffering 139
10.2 Alpha Blending 142

Part II - Advanced Topics

11 Euler Transforms and Quaternions 149
11.1 Euler Transforms 149
11.1.1 World-space Euler Transforms 149
11.1.2 Object-space Euler Transforms* 150
11.2 Euler Transforms for Keyframe Animation 153
11.2.1 2D Keyframe Animation 153
11.2.2 3D Keyframe Animation 154
11.2.3 Interpolation of Euler Angles 156
11.3 Quaternions 157
11.3.1 Quaternion Representation 158
11.3.2 Rotations Using Quaternions 158
11.3.3 Interpolation of Quaternions 162
11.3.4 Conversion between Quaternion and Rotation Matrix 163

12 Screen-space Object Manipulation 167
12.1 Picking an Object 167
12.1.1 Screen-space Ray 168
12.1.2 Camera-space Ray 168
12.1.3 Object-space Ray 170
12.1.4 Intersection between Ray and Bounding Volume . . . 171
12.1.5 Intersection between Ray and Triangle 175
12.2 Rotating an Object 178

13 Character Animation 185
13.1 Skeleton and Space Change 185
13.1.1 Skeleton 185
13.1.2 Space Change between Bones 186

13.1.3 Character Space to Bone Space 189
13.2 Forward Kinematics 192
13.3 Skinning 195
13.3.1 Vertex Blending 196
13.3.2 Integration with Keyframe Animation* 198
13.4 Inverse Kinematics 200
13.4.1 Analytic Solution 201
13.4.2 Cyclic Coordinate Descent 202

14 Normal Mapping 209
14.1 Height Map 210
14.2 Normal Map 213
14.3 Shaders for Normal Mapping 214
14.4 Tangent-space Normal Mapping 218
14.4.1 Tangent-space Normals 218
14.4.2 Shaders for Tangent-space Normal Mapping 219
14.4.3 Computing Tangent Spaces* 222
14.5 Authoring Normal Maps 223

15 Shadow Mapping 231
15.1 Two-pass Algorithm 231
15.2 Shadow Map Filtering 234
15.3 GL Program and Shaders for Shadow Mapping* 236
15.3.1 First-pass Shaders 236
15.3.2 Render-to-Texture and Framebuffer Object 238
15.3.3 Second-pass Shaders 239
15.4 Hard Shadow versus Soft Shadow 242

16 Texturing toward Global Illumination 249
16.1 Global Illumination 250
16.1.1 Ray Tracing 250
16.1.2 Radiosity* 254
16.2 Light Mapping 258
16.3 Environment Mapping 259
16.3.1 Cube Mapping 261
16.3.2 GL Program and Shaders for Cube Mapping 262
16.4 Ambient Occlusion 264

17 Parametric Curves and Surfaces 271
17.1 Parametric Curves 271
17.1.1 Bézier Curves 271
17.1.2 Hermite Curve and Catmull-Rom Spline 275
17.1.3 Application: Camera Path 277
17.2 Parametric Surfaces 279
17.2.1 Bilinear Patch 280

17.2.2 Biquadratic Bézier Patch 282
17.2.3 Bicubic Bézier Patch 286
17.2.4 Bézier Triangle 288

18 Surface Tessellation 295
18.1 Displacement Mapping 295
18.1.1 GPU Tessellation 295
18.1.2 Shaders and Tessellator 297
18.2 PN-triangles* 303
18.2.1 Computing Control Points 303
18.2.2 Computing Control Normals 308
18.2.3 GPU Tessellation 310
18.2.4 Shaders and Tessellator 311

References 319

Index 321

Preface

OpenGL ES is the standard graphics API for mobile and embedded systems. Virtually every pixel on a smartphone's screen is generated by OpenGL ES. However, there exists no textbook on OpenGL ES which has a balance between theory and practicality. This book is written to answer that need and presents the must-know in real-time graphics with OpenGL ES. This book suits the advanced undergraduate and beginner graduate courses in computer graphics.

Another primary group of readers that this book may benefit includes mobile 3D app developers, who have experiences in OpenGL ES and shader programming but lack theoretical background in 3D graphics. A few excellent programming manuals on OpenGL ES can be found in bookstores, but they do not provide a sufficient level of mathematical background for developers. Assuming that the readers have a minimal understanding of vectors and matrices, this book provides an opportunity to combine their knowledge with the background theory of computer graphics.

This book is built upon the author's previous work *3D Graphics for Game Programming* published in 2011. Reusing roughly half of the contents from that book, several new topics and a considerable number of OpenGL ES and shader programs have been added. As OpenGL ES is a subset of OpenGL, this book is also suitable for beginner OpenGL programmers.

The organization and presentation of this book have been carefully designed so as to enable the readers to easily understand the key aspects of real-time graphics and OpenGL ES. Over the chapters, numerous 3D illustrations are provided to help the readers effortlessly grasp the complicated topics. An important organizational feature of this book is that "non-core" details are presented in separate notes (in shaded boxes) and in optional sections (marked by asterisks). They can be safely skipped without incurring any difficulty in understanding the subsequent topics of the book.

If the optional parts are excluded, the entire contents of this book can be covered in a 16-week semester for graduate classes. For undergraduate classes, however, this feat will be difficult. According to the author's experience, teaching Chapters 1 through 14 is a feasible goal.

The sample programs presented in this book are available on GitHub: https://github.com/medialab-ku/openGLESbook. The site also provides links to the full-length lecture notes as PowerPoint files and additional materials including video clips.