

***Introduction to 3D
Game Programming
with DirectX[®] 10***

Frank D. Luna

Wordware Publishing, Inc.

Contents

Acknowledgments	xvii
Introduction	xix

Part I — Mathematical Prerequisites

Chapter 1 Vector Algebra	3
1.1 Vectors	3
1.1.1 Vectors and Coordinate Systems	4
1.1.2 Left-Handed Versus Right-Handed Coordinate Systems.	6
1.1.3 Basic Vector Operations.	6
1.2 Length and Unit Vectors	8
1.3 The Dot Product	9
1.4 The Cross Product	12
1.5 Points	13
1.6 D3DX Vectors.	14
1.7 Summary	18
1.8 Exercises	19
Chapter 2 Matrix Algebra	23
2.1 Definition	23
2.2 Matrix Multiplication	25
2.2.1 Definition	25
2.2.2 Vector-Matrix Multiplication.	27
2.2.3 Associativity	27
2.3 The Transpose of a Matrix	28
2.4 The Identity Matrix	28
2.5 The Inverse of a Matrix.	29
2.6 D3DX Matrices	30
2.7 Summary	34
2.8 Exercises	35
Chapter 3 Transformations	37
3.1 Basic Transformations	38
3.1.1 Scaling	39
3.1.2 Rotation	40
3.1.3 Translation	43
3.2 D3DX Transformation Functions	44
3.3 Composition of Transformations	45
3.4 Change of Coordinate Transformations.	46
3.4.1 Vectors	47

- 3.4.2 Points 48
- 3.4.3 Matrix Representation 49
- 3.4.4 Associativity and Change of Coordinate Matrices 50
- 3.4.5 Inverses and Change of Coordinate Matrices 51
- 3.5 Summary 52
- 3.6 Exercises 53

Part II — Direct3D Foundations

- Chapter 4 Direct3D Initialization. 57**
 - 4.1 Preliminaries 58
 - 4.1.1 Direct3D Overview 58
 - 4.1.2 COM 58
 - 4.1.3 Textures and Data Resource Formats 59
 - 4.1.4 The Swap Chain and Page Flipping 60
 - 4.1.5 Depth Buffering 61
 - 4.1.6 Texture Resource Views 64
 - 4.1.7 Multisampling 65
 - 4.2 Initializing Direct3D. 66
 - 4.2.1 Describe the Swap Chain 67
 - 4.2.2 Create the Device and Swap Chain 68
 - 4.2.3 Create the Render Target View 70
 - 4.2.4 Create the Depth/Stencil Buffer and View 71
 - 4.2.5 Bind the Views to the Output Merger Stage 73
 - 4.2.6 Set the Viewport 73
 - 4.3 Timing and Animation. 75
 - 4.3.1 The Performance Timer 75
 - 4.3.2 Game Timer Class 76
 - 4.3.3 Time Elapsed between Frames 77
 - 4.3.4 Game Time. 79
 - 4.4 Text Output 82
 - 4.5 The Demo Application Framework. 83
 - 4.5.1 D3DApp 84
 - 4.5.2 Non-Framework Methods 85
 - 4.5.3 Framework Methods. 86
 - 4.5.4 Frame Statistics 87
 - 4.5.5 The Message Handler 89
 - 4.5.6 Going Full Screen 91
 - 4.5.7 The Init Direct3D Demo. 91
 - 4.6 Debugging Direct3D Applications 93
 - 4.7 Summary 94
 - 4.8 Exercises 96
- Chapter 5 The Rendering Pipeline 99**
 - 5.1 The 3D Illusion 100
 - 5.2 Model Representation 102
 - 5.3 Basic Computer Color 104

5.3.1 Color Operations	105
5.3.2 128-Bit Color	106
5.3.3 32-Bit Color.	107
5.4 Overview of the Rendering Pipeline.	108
5.5 The Input Assembler Stage	109
5.5.1 Vertex Layouts	109
5.5.2 Vertex Buffers	114
5.5.3 Primitive Topology	118
5.5.3.1 Point List	119
5.5.3.2 Line Strip	119
5.5.3.3 Line List	119
5.5.3.4 Triangle Strip	119
5.5.3.5 Triangle List	120
5.5.3.6 Primitives with Adjacency	120
5.5.4 Indices and Index Buffers	121
5.6 The Vertex Shader Stage	126
5.6.1 Local Space and World Space.	126
5.6.2 View Space	128
5.6.3 Projection and Homogeneous Clip Space	131
5.6.3.1 Defining a Frustum.	132
5.6.3.2 Projecting Vertices	134
5.6.3.3 Normalized Device Coordinates (NDC).	134
5.6.3.4 Writing the Projection Equations with a Matrix	135
5.6.3.5 Normalized Depth Value	136
5.6.3.6 D3DXMatrixPerspectiveFovLH.	138
5.6.4 Example Vertex Shader	139
5.6.5 Constant Buffers	141
5.7 The Geometry Shader Stage	143
5.8 The Clipping Stage.	143
5.9 The Rasterization Stage	145
5.9.1 Viewport Transform	145
5.9.2 Backface Culling	146
5.9.3 Vertex Attribute Interpolation	147
5.10 The Pixel Shader Stage.	148
5.11 The Output Merger Stage	150
5.12 Render States.	150
5.13 Effects	153
5.13.1 Effect Files	153
5.13.2 Creating an Effect	155
5.13.3 Interfacing with Effects from the C++ Application	156
5.13.4 Using Effects to Draw.	157
5.14 Colored Cube Demo	158
5.15 Peaks and Valleys Demo	166
5.16 Dynamic Vertex Buffers	169
5.17 Summary	171
5.18 Exercises	173

- Chapter 6 Lighting 177**
 - 6.1 Light and Material Interaction 178
 - 6.2 Normal Vectors 180
 - 6.2.1 Computing Normal Vectors 181
 - 6.2.2 Transforming Normal Vectors 183
 - 6.3 Lambert’s Cosine Law. 185
 - 6.4 Diffuse Lighting 186
 - 6.5 Ambient Lighting 187
 - 6.6 Specular Lighting 188
 - 6.7 Brief Recap. 191
 - 6.8 Parallel Lights 193
 - 6.9 Point Lights 194
 - 6.9.1 Attenuation 194
 - 6.9.2 Range 195
 - 6.10 Spotlights 195
 - 6.11 Implementation. 197
 - 6.11.1 Lighting Structures 197
 - 6.11.2 Implementing Parallel Lights 198
 - 6.11.3 Implementing Point Lights 198
 - 6.11.4 Implementing Spotlights 199
 - 6.12 Lighting Demo 200
 - 6.12.1 Effect File 200
 - 6.12.2 Structure Packing 202
 - 6.12.3 C++ Application Code 204
 - 6.12.4 Normal Computation 206
 - 6.13 Summary 207
 - 6.14 Exercises 208

- Chapter 7 Texturing 211**
 - 7.1 Texture and Resource Recap 212
 - 7.2 Texture Coordinates 214
 - 7.3 Creating and Enabling a Texture. 216
 - 7.4 Filters 218
 - 7.4.1 Magnification 218
 - 7.4.2 Minification 220
 - 7.4.2.1 Creating Mipmaps 221
 - 7.4.3 Anisotropic Filtering 221
 - 7.5 Sampling Textures 222
 - 7.6 Textures as Materials 223
 - 7.7 Crate Demo 225
 - 7.7.1 Specifying Texture Coordinates 225
 - 7.7.2 Creating the Texture 226
 - 7.7.3 Setting the Texture. 226
 - 7.7.4 Texture Effect 227
 - 7.8 Address Modes 229
 - 7.9 Transforming Textures. 232
 - 7.10 Land Tex Demo. 233
 - 7.10.1 Grid Texture Coordinate Generation 234

7.10.2 Texture Tiling	235
7.10.3 Texture Animation	235
7.11 Compressed Texture Formats	236
7.12 Summary	238
7.13 Exercises	239
Chapter 8 Blending	241
8.1 The Blending Equation	242
8.2 Blend Operations	243
8.3 Blend Factors	243
8.4 Blend State.	244
8.5 Examples.	247
8.5.1 No Color Write	247
8.5.2 Adding/Subtracting	248
8.5.3 Multiplying	248
8.5.4 Transparency	249
8.5.5 Blending and the Depth Buffer	250
8.6 Alpha Channels	251
8.7 Transparent Water Demo	253
8.8 Clipping Pixels	255
8.9 Fog	256
8.10 Summary	260
8.11 Exercises	261
Chapter 9 Stenciling	263
9.1 Depth/Stencil Formats and Clearing.	264
9.2 The Stencil Test	265
9.3 The Depth/Stencil State Block	266
9.3.1 Depth Settings	266
9.3.2 Stencil Settings.	267
9.3.3 Creating and Binding a Depth/Stencil State	269
9.3.4 Depth/Stencil States in Effect Files	269
9.4 Mirror Demo.	270
9.4.1 Overview	271
9.4.2 Defining the Depth/Stencil States	272
9.4.3 Blending the Reflection.	273
9.4.4 Drawing the Scene	274
9.5 Summary.	276
9.6 Exercises.	277
Chapter 10 The Geometry Shader.	281
10.1 Programming Geometry Shaders	282
10.2 Tree Billboards Demo	286
10.2.1 Overview	286
10.2.2 Vertex Structure	288
10.2.3 Effect File	289
10.2.4 SV_PrimitiveID	292
10.3 Texture Arrays	293

- 10.3.1 Overview 293
- 10.3.2 Sampling a Texture Array 294
- 10.3.3 Loading Texture Arrays 295
- 10.3.4 Texture Subresources 297
- 10.4 Summary 298
- 10.5 Exercises 300

Part III — Direct3D Topics

- Chapter 11 Cube Mapping 305**
 - 11.1 Cube Mapping 305
 - 11.2 Environment Maps 307
 - 11.2.1 Loading and Using Cube Maps in Direct3D 310
 - 11.3 Texturing a Sky 311
 - 11.4 Modeling Reflections 313
 - 11.5 New Application Code 316
 - 11.5.1 Texture Manager 316
 - 11.5.2 Global Effects and Input Layouts 316
 - 11.5.3 New 3D Object Classes 317
 - 11.6 Summary 319
 - 11.7 Exercises 319
- Chapter 12 Normal Mapping 321**
 - 12.1 Motivation 321
 - 12.2 Normal Maps 323
 - 12.3 Texture/Tangent Space 325
 - 12.4 Vertex Tangent Space 327
 - 12.5 Transforming between Tangent Space and Object Space 328
 - 12.6 Shader Code 329
 - 12.7 Summary 332
 - 12.8 Exercises 333
- Chapter 13 Shadow Mapping 335**
 - 13.1 Render to Texture 335
 - 13.1.1 Constructor/Destructor/Accessors 337
 - 13.1.2 DrawableTex2D::init 337
 - 13.1.3 DrawableTex2D::buildDepthMap 338
 - 13.1.4 DrawableTex2D::buildColorMap 339
 - 13.1.5 DrawableTex2D::begin 340
 - 13.1.6 DrawableTex2D::end 341
 - 13.2 Orthographic Projections 342
 - 13.3 Projective Texture Coordinates 344
 - 13.3.1 Code Implementation 346
 - 13.3.2 Points Outside the Frustum 346
 - 13.3.3 Orthographic Projections 347
 - 13.4 Shadow Mapping 348
 - 13.4.1 Algorithm Description 348

13.4.2 Building the Shadow Map	349
13.4.3 Restore Rendering to the Back Buffer	351
13.4.4 The Shadow Factor	352
13.4.5 The Shadow Map Test.	353
13.4.6 Filtering and the Shadow Map Test.	356
13.4.7 Rendering the Shadow Map.	358
13.5 Summary	359
13.6 Exercises	360
Chapter 14 Meshes	361
14.1 ID3DX10MeshBuffer.	361
14.2 Geometry Info	362
14.3 Subsets and the Attribute Buffer.	365
14.4 Drawing.	367
14.5 Adjacency Information	368
14.6 Optimizing	369
14.7 The Attribute Table.	370
14.8 Cloning	372
14.9 Commit to Device	374
14.10 Creating a Mesh.	374
14.11 Mesh Viewer Demo.	376
14.12 Summary	377
14.13 Exercises	378
Chapter 15 Picking	379
15.1 Screen to Projection Window Transform.	381
15.2 World/Local Space Picking Ray	384
15.3 Ray/Mesh Intersection	385
15.4 Bounding Volumes	387
15.4.1 Computing an Axis-Aligned Bounding Box.	388
15.4.2 Rotations and Axis-Aligned Bounding Boxes.	389
15.4.3 Computing a Bounding Sphere	391
15.4.4 Scaling and Bounding Spheres	391
15.5 Demo Application.	391
15.6 Summary	392
15.7 Exercises	392
Chapter 16 Terrain Rendering	395
16.1 Heightmaps.	396
16.1.1 Creating a Heightmap.	397
16.1.2 Loading a RAW File	398
16.1.3 Smoothing.	400
16.2 Texturing	402
16.3 Lighting	405
16.4 Terrain FX	406
16.5 Terrain Height	408
16.6 Summary	411
16.7 Exercises	412

Chapter 17 Particle Systems and Stream Output 413

- 17.1 Particle Representation. 413
- 17.2 Particle Motion 415
- 17.3 Randomness 417
- 17.4 Blending and Particle Systems 419
- 17.5 Stream Output 421
 - 17.5.1 Creating a Geometry Shader for Stream Output 421
 - 17.5.2 Stream Output Only 422
 - 17.5.3 Creating a Vertex Buffer for Stream Output 423
 - 17.5.4 Binding to the SO Stage. 424
 - 17.5.5 Unbinding from the Stream Output Stage 424
 - 17.5.6 Auto Draw 425
 - 17.5.7 Ping-Ponging Vertex Buffers 425
- 17.6 GPU-Based Particle System 425
 - 17.6.1 Particle Effects 426
 - 17.6.2 The Particle System Class 426
 - 17.6.3 Emitter Particles 428
 - 17.6.4 The Initialization Vertex Buffer 428
 - 17.6.5 The Update/Draw Method 429
- 17.7 Fire 431
- 17.8 Rain 436
- 17.9 Summary 442
- 17.10 Exercises 443

Appendices

Appendix A Introduction to Windows Programming 445

- A.1 Overview 446
 - A.1.1 Resources 446
 - A.1.2 Events, the Message Queue, Messages, and the Message Loop . . 447
 - A.1.3 GUI. 448
 - A.1.4 Unicode. 449
- A.2 Basic Windows Application 450
- A.3 Explaining the Basic Windows Application 454
 - A.3.1 Includes, Global Variables, and Prototypes 454
 - A.3.2 WinMain 454
 - A.3.3 WNDCLASS and Registration 455
 - A.3.4 Creating and Displaying the Window 457
 - A.3.5 The Message Loop. 459
 - A.3.6 The Window Procedure 460
 - A.3.7 The MessageBox Function 462
- A.4 A Better Message Loop 462
- A.5 Summary 463
- A.6 Exercises 464

Appendix B High-Level Shading Language Reference	465
B.1 Variable Types	465
B.1.1 Scalar Types	465
B.1.2 Vector Types	465
B.1.2.1 Swizzles	466
B.1.3 Matrix Types	467
B.1.4 Arrays	468
B.1.5 Structures	468
B.1.6 The typedef Keyword	468
B.1.7 Variable Prefixes	469
B.1.8 Casting	469
B.2 Keywords and Operators	470
B.2.1 Keywords.	470
B.2.2 Operators.	470
B.3 Program Flow	472
B.4 Functions	473
B.4.1 User-Defined Functions	473
B.4.2 Built-in Functions	475
Appendix C Some Analytic Geometry	479
C.1 Rays, Lines, and Segments	479
C.2 Parallelograms.	480
C.3 Triangles.	481
C.4 Planes	481
C.4.1 D3DXPlane.	483
C.4.2 Point/Plane Spatial Relation	483
C.4.3 Construction	484
C.4.4 Normalizing a Plane	485
C.4.5 Transforming a Plane.	486
C.4.6 Finding the Nearest Point on a Plane to a Given Point	486
C.4.7 Ray/Plane Intersection.	487
C.4.8 Reflecting Vectors	487
C.4.9 Reflecting Points	488
C.4.10 Reflection Matrix	488
C.5 Exercises	490
 Bibliography and Further Reading.	 491
 Index.	 495