

Introduction to
3D GAME PROGRAMMING
WITH DIRECTX[®] 12

Frank D. Luna



MERCURY LEARNING AND INFORMATION

Dulles, Virginia
Boston, Massachusetts
New Delhi

CONTENTS

<i>Dedication</i>	v
<i>Acknowledgments</i>	xxii
<i>Introduction</i>	xxv
Intended Audience	xxvi
Prerequisites	xxvi
Required Development Tools and Hardware	xxvii
Using the DirectX SDK Documentation and SDK Samples	xxvii
Clarity	xxix
Sample Programs and Online Supplements	xxix
Demo Project Setup in Visual Studio 2010	xxix
Download the Book's Source Code	xxx
Create a Win32 Project	xxx
Linking the DirectX Libraries	xxxi
Adding the Source Code and Building the Project	xxxii

PART I MATHEMATICAL PREREQUISITES

Chapter 1 Vector Algebra	3
1.1 Vectors	4
1.1.1 Vectors and Coordinate Systems	5
1.1.2 Left-Handed Versus Right-Handed Coordinate Systems	6
1.1.3 Basic Vector Operations	7
1.2 Length and Unit Vectors	9
1.3 The Dot Product	10
1.3.1 Orthogonalization	13
1.4 The Cross Product	14
1.4.1 Pseudo 2D Cross Product	16
1.4.2 Orthogonalization with the Cross Product	16

1.5	Points	17
1.6	DirectX Math Vectors	18
1.6.1	Vector Types	19
1.6.2	Loading and Storage Methods	21
1.6.3	Parameter Passing	21
1.6.4	Constant Vectors	23
1.6.5	Overloaded Operators	24
1.6.6	Miscellaneous	24
1.6.7	Setter Functions	25
1.6.8	Vector Functions	26
1.6.9	Floating-Point Error	30
1.7	Summary	31
1.8	Exercises	33
Chapter 2 Matrix Algebra		37
2.1	Definition	38
2.2	Matrix Multiplication	40
2.2.1	Definition	40
2.2.2	Vector-Matrix Multiplication	41
2.2.3	Associativity	42
2.3	The Transpose of a Matrix	42
2.4	The Identity Matrix	43
2.5	The Determinant of a Matrix	44
2.5.1	Matrix Minors	45
2.5.2	Definition	45
2.6	The Adjoint of a Matrix	47
2.7	The Inverse of a Matrix	47
2.8	DirectX Math Matrices	49
2.8.1	Matrix Types	50
2.8.2	Matrix Functions	52
2.8.3	DirectX Math Matrix Sample Program	53
2.9	Summary	54
2.10	Exercises	55
Chapter 3 Transformations		59
3.1	Linear Transformations	60
3.1.1	Definition	60
3.1.2	Matrix Representation	60
3.1.3	Scaling	61
3.1.4	Rotation	63

3.2	Affine Transformations	66
3.2.1	Homogeneous Coordinates	66
3.2.2	Definition and Matrix Representation	67
3.2.3	Translation	67
3.2.4	Affine Matrices for Scaling and Rotation	70
3.2.5	Geometric Interpretation of an Affine Transformation Matrix	70
3.3	Composition of Transformations	72
3.4	Change of Coordinate Transformations	73
3.4.1	Vectors	74
3.4.2	Points	74
3.4.3	Matrix Representation	75
3.4.4	Associativity and Change of Coordinate Matrices	76
3.4.5	Inverses and Change of Coordinate Matrices	77
3.5	Transformation Matrix versus Change of Coordinate Matrix	78
3.6	DirectX Math Transformation Functions	79
3.7	Summary	80
3.8	Exercises	82

PART II DIRECT 3D FOUNDATIONS

Chapter 4	Direct3D Initialization	89
4.1	Preliminaries	90
4.1.1	Direct3D 12 Overview	90
4.1.2	COM	90
4.1.3	Textures Formats	91
4.1.4	The Swap Chain and Page Flipping	93
4.1.5	Depth Buffering	93
4.1.6	Resources and Descriptors	96
4.1.7	Multisampling Theory	98
4.1.8	Multisampling in Direct3D	99
4.1.9	Feature Levels	101
4.1.10	DirectX Graphics Infrastructure	101
4.1.11	Checking Feature Support	105
4.1.12	Residency	107
4.2	CPU/GPU Interaction	107
4.2.1	The Command Queue and Command Lists	108
4.2.2	CPU/GPU Synchronization	112

4.2.3	Resource Transitions	114
4.2.4	Multithreading with Commands	116
4.3	Initializing Direct3D	116
4.3.1	Create the Device	117
4.3.2	Create the Fence and Descriptor Sizes	118
4.3.3	Check 4X MSAA Quality Support	119
4.3.4	Create Command Queue and Command List	119
4.3.5	Describe and Create the Swap Chain	120
4.3.6	Create the Descriptor Heaps	122
4.3.7	Create the Render Target View	123
4.3.8	Create the Depth/Stencil Buffer and View	125
4.3.9	Set the Viewport	129
4.3.10	Set the Scissor Rectangles	131
4.4	Timing and Animation	131
4.4.1	The Performance Timer	132
4.4.2	Game Timer Class	133
4.4.3	Time Elapsed Between Frames	134
4.4.4	Total Time	136
4.5	The Demo Application Framework	139
4.5.1	D3DApp	139
4.5.2	Non-Framework Methods	142
4.5.3	Framework Methods	143
4.5.4	Frame Statistics	145
4.5.5	The Message Handler	146
4.5.6	The “Init Direct3D” Demo	149
4.6	Debugging Direct3D Applications	153
4.7	Summary	155
Chapter 5 The Rendering Pipeline		159
5.1	The 3D Illusion	160
5.2	Model Representation	162
5.3	Basic Computer Color	163
5.3.1	Color Operations	164
5.3.2	128-Bit Color	165
5.3.3	32-Bit Color	165
5.4	Overview of the Rendering Pipeline	167
5.5	The Input Assembler Stage	168
5.5.1	Vertices	168
5.5.2	Primitive Topology	169
5.5.2.1	Point List	170

5.5.2.2	Line Strip	170
5.5.2.3	Line List	170
5.5.2.4	Triangle Strip	170
5.5.2.5	Triangle List	171
5.5.2.6	Primitives with Adjacency	171
5.5.2.7	Control Point Patch List	172
5.5.3	Indices	172
5.6	The Vertex Shader Stage	175
5.6.1	Local Space and World Space	175
5.6.2	View Space	179
5.6.3	Projection and Homogeneous Clip Space	182
5.6.3.1	Defining a Frustum	183
5.6.3.2	Projecting Vertices	184
5.6.3.3	Normalized Device Coordinates (NDC)	185
5.6.3.4	Writing the Projection Equations with a Matrix	186
5.6.3.5	Normalized Depth Value	187
5.6.3.6	XMMatrixPerspectiveFovLH	189
5.7	The Tessellation Stages	190
5.8	The Geometry Shader Stage	191
5.9	Clipping	191
5.10	The Rasterization Stage	193
5.10.1	Viewport Transform	193
5.10.2	Backface Culling	194
5.10.3	Vertex Attribute Interpolation	195
5.11	The Pixel Shader Stage	196
5.12	The Output Merger Stage	197
5.13	Summary	197
5.14	Exercises	198
Chapter 6	Drawing in Direct3D	203
6.1	Vertices and Input Layouts	203
6.2	Vertex Buffers	207
6.3	Indices and Index Buffers	212
6.4	Example Vertex Shader	216
6.4.1	Input Layout Description and Input Signature Linking	219
6.5	Example Pixel Shader	222
6.6	Constant Buffers	224
6.6.1	Creating Constant Buffers	224

6.6.2	Updating Constant Buffers	227
6.6.3	Upload Buffer Helper	227
6.6.4	Constant Buffer Descriptors	230
6.6.5	Root Signature and Descriptor Tables	232
6.7	Compiling Shaders	235
6.7.1	Offline Compilation	237
6.7.2	Generated Assembly	239
6.7.3	Using Visual Studio to Compile Shaders Offline	241
6.8	Rasterizer State	242
6.9	Pipeline State Object	243
6.10	Geometry Helper Structure	247
6.11	Box Demo	249
6.12	Summary	259
6.13	Exercises	260
Chapter 7 Drawing in Direct 3D Part II		265
7.1	Frame Resources	266
7.2	Render Items	269
7.3	Pass Constants	270
7.4	Shape Geometry	273
7.4.1	Generating a Cylinder Mesh	275
7.4.1.1	Cylinder Side Geometry	275
7.4.1.2	Cap Geometry	278
7.4.2	Generating a Sphere Mesh	279
7.4.3	Generating a Geosphere Mesh	279
7.5	Shapes Demo	281
7.5.1	Vertex and Index Buffers	282
7.5.2	Render Items	286
7.5.3	Frame Resources and Constant Buffer Views	288
7.5.4	Drawing the Scene	290
7.6	More on Root Signatures	293
7.6.1	Root Parameters	293
7.6.2	Descriptor Tables	295
7.6.3	Root Descriptors	297
7.6.4	Root Constants	298
7.6.5	A More Complicated Root Signature Example	299
7.6.6	Root Parameter Versioning	300
7.7	Land and Waves Demo	301
7.7.1	Generating the Grid Vertices	302

7.7.2	Generating the Grid Indices	304
7.7.3	Applying the Height Function	305
7.7.4	Root CBVs	307
7.7.5	Dynamic Vertex Buffers	309
7.8	Summary	311
7.9	Exercises	313
Chapter 8	Lighting	315
8.1	Light and Material Interaction	316
8.2	Normal Vectors	318
8.2.1	Computing Normal Vectors	319
8.2.2	Transforming Normal Vectors	321
8.3	Important Vectors in Lighting	323
8.4	Lambert's Cosine Law	323
8.5	Diffuse Lighting	325
8.6	Ambient Lighting	326
8.7	Specular Lighting	327
8.7.1	Fresnel Effect	328
8.7.2	Roughness	330
8.8	Lighting Model Recap	333
8.9	Implementing Materials	334
8.10	Parallel Lights	339
8.11	Point Lights	339
8.11.1	Attenuation	340
8.12	Spotlights	341
8.13	Lighting Implementation	342
8.13.1	Light Structure	342
8.13.2	Common Helper Functions	344
8.13.3	Implementing Directional Lights	346
8.13.4	Implementing Point Lights	346
8.13.5	Implementing Spotlights	347
8.13.6	Accumulating Multiple Lights	348
8.13.7	The Main HLSL File	349
8.14	Lighting Demo	351
8.14.1	Vertex Format	352
8.14.2	Normal Computation	352
8.14.3	Updating the Light Direction	354
8.14.4	Update to Root Signature	356
8.15	Summary	356
8.16	Exercises	357

Chapter 9	Texturing	359
9.1	Texture and Resource Recap	360
9.2	Texture Coordinates	362
9.3	Texture Data Sources	364
9.3.1	DDS Overview	365
9.3.2	Creating DDS Files	366
9.4	Creating and Enabling a Texture	367
9.4.1	Loading DDS Files	367
9.4.2	SRV Heap	368
9.4.3	Creating SRV Descriptors	368
9.4.4	Binding Textures to the Pipeline	370
9.5	Filters	373
9.5.1	Magnification	373
9.5.2	Minification	375
9.5.3	Anisotropic Filtering	376
9.6	Address Modes	376
9.7	Sampler Objects	378
9.7.1	Creating Samplers	379
9.7.2	Static Samplers	381
9.8	Sampling Textures in a Shader	384
9.9	Crate Demo	385
9.9.1	Specifying Texture Coordinates	385
9.9.2	Creating the Texture	386
9.9.3	Setting the Texture	386
9.9.4	Updated HLSL	387
9.10	Transforming Textures	389
9.11	Textured Hills and Waves Demo	390
9.11.1	Grid Texture Coordinate Generation	391
9.11.2	Texture Tiling	392
9.11.3	Texture Animation	393
9.12	Summary	393
9.13	Exercises	394
Chapter 10	Blending	397
10.1	The Blending Equation	398
10.2	Blend Operations	399
10.3	Blend Factors	400
10.4	Blend State	401
10.5	Examples	404
10.5.1	No Color Write	404

10.5.2	Adding/Subtracting	405
10.5.3	Multiplying	405
10.5.4	Transparency	406
10.5.5	Blending and the Depth Buffer	407
10.6	Alpha Channels	408
10.7	Clipping Pixels	408
10.8	Fog	410
10.9	Summary	416
10.10	Exercises	417
Chapter 11	Stenciling	419
11.1	Depth/Stencil Formats and Clearing	420
11.2	The Stencil Test	421
11.3	Describing the Depth/Stencil State	422
11.3.1	Depth Settings	423
11.3.2	Stencil Settings	423
11.3.3	Creating and Binding a Depth/Stencil State	425
11.4	Implementing Planar Mirrors	426
11.4.1	Mirror Overview	426
11.4.2	Defining the Mirror Depth/Stencil States	429
11.4.3	Drawing the Scene	430
11.4.4	Winding Order and Reflections	431
11.5	Implementing Planar Shadows	432
11.5.1	Parallel Light Shadows	433
11.5.2	Point Light Shadows	434
11.5.3	General Shadow Matrix	435
11.5.4	Using the Stencil Buffer to Prevent Double Blending	436
11.5.5	Shadow Code	437
11.6	Summary	438
11.7	Exercises	439
Chapter 12	The Geometry Shader	445
12.1	Programming Geometry Shaders	446
12.2	Tree Billboards Demo	451
12.2.1	Overview	451
12.2.2	Vertex Structure	453
12.2.3	The HLSL File	454
12.2.4	SV_PrimitiveID	458
12.3	Texture Arrays	459
12.3.1	Overview	459

12.3.2	Sampling a Texture Array	460
12.3.3	Loading Texture Arrays	461
12.3.4	Texture Subresources	462
12.4	Alpha-to-Coverage	463
12.5	Summary	464
12.6	Exercises	465
Chapter 13 The Compute Shader		469
13.1	Threads and Thread Groups	471
13.2	A Simple Compute Shader	472
13.2.1	Compute PSO	473
13.3	Data Input and Output Resources	474
13.3.1	Texture Inputs	474
13.3.2	Texture Outputs and Unordered Access Views (UAVs)	474
13.3.3	Indexing and Sampling Textures	477
13.3.4	Structured Buffer Resources	479
13.3.5	Copying CS Results to System Memory	482
13.4	Thread Identification System Values	485
13.5	Append and Consume Buffers	486
13.6	Shared Memory and Synchronization	487
13.7	Blur Demo	489
13.7.1	Blurring Theory	489
13.7.2	Render-to-Texture	493
13.7.3	Blur Implementation Overview	495
13.7.4	Compute Shader Program	501
13.8	Further Resources	505
13.9	Summary	506
13.10	Exercises	508
Chapter 14 The Tessellation Stages		513
14.1	Tessellation Primitive Types	514
14.1.1	Tessellation and the Vertex Shader	515
14.2	The Hull Shader	516
14.2.1	Constant Hull Shader	516
14.2.2	Control Point Hull Shader	518
14.3	The Tessellation Stage	520
14.3.1	Quad Patch Tessellation Examples	520
14.3.2	Triangle Patch Tessellation Examples	521
14.4	The Domain Shader	521
14.5	Tessellating a Quad	522

14.6	Cubic Bézier Quad Patches	526
14.6.1	Bézier Curves	527
14.6.2	Cubic Bézier Surfaces	529
14.6.3	Cubic Bézier Surface Evaluation Code	530
14.6.4	Defining the Patch Geometry	532
14.7	Summary	534
14.8	Exercises	536

PART III TOPICS

Chapter 15 Building a First Person Camera and Dynamic Indexing 539

15.1	View Transform Review	540
15.2	The Camera Class	541
15.3	Selected Method Implementations	543
15.3.1	XMVECTOR Return Variations	543
15.3.2	SetLens	543
15.3.3	Derived Frustum Info	544
15.3.4	Transforming the Camera	544
15.3.5	Building the View Matrix	546
15.4	Camera Demo Comments	547
15.5	Dynamic Indexing	548
15.6	Summary	555
15.7	Exercises	556

Chapter 16 Instancing and Frustum Culling 557

16.1	Hardware Instancing	557
16.1.1	Drawing Instanced Data	558
16.1.2	Instance Data	559
16.1.3	Creating the Instanced Buffer	564
16.2	Bounding Volumes and Frustums	566
16.2.1	DirectX Math Collision	566
16.2.2	Boxes	566
16.2.2.1	Rotations and Axis-Aligned Bounding Boxes	568
16.2.3	Spheres	570
16.2.4	Frustums	571
16.2.4.1	Constructing the Frustum Planes	571
16.2.4.2	Frustum/Sphere Intersection	574
16.2.4.3	Frustum/AABB Intersection	575

16.3	Frustum Culling	576
16.4	Summary	579
16.5	Exercises	580
Chapter 17 Picking		583
17.1	Screen to Projection Window Transform	585
17.2	World/Local Space Picking Ray	588
17.3	Ray/Mesh Intersection	589
	17.3.1 Ray/AABB Intersection	591
	17.3.2 Ray/Sphere Intersection	591
	17.3.3 Ray/Triangle Intersection	592
17.4	Demo Application	594
17.5	Summary	596
17.6	Exercises	596
Chapter 18 Cube Mapping		597
18.1	Cube Mapping	597
18.2	Environment Maps	599
	18.2.1 Loading and Using Cube Maps in Direct3D	601
18.3	Texturing a Sky	602
18.4	Modeling Reflections	606
18.5	Dynamic Cube Maps	609
	18.5.1 Dynamic Cube Map Helper Class	610
	18.5.2 Building the Cube Map Resource	611
	18.5.3 Extra Descriptor Heap Space	612
	18.5.4 Building the Descriptors	613
	18.5.5 Building the Depth Buffer	614
	18.5.6 Cube Map Viewport and Scissor Rectangle	615
	18.5.7 Setting up the Cube Map Camera	615
	18.5.8 Drawing into the Cube Map	617
18.6	Dynamic Cube Maps with the Geometry Shader	620
18.7	Summary	623
18.8	Exercises	624
Chapter 19 Normal Mapping		627
19.1	Motivation	628
19.2	Normal Maps	629
19.3	Texture/Tangent Space	631
19.4	Vertex Tangent Space	633

19.5	Transforming Between Tangent Space and Object Space	633
19.6	Normal Mapping Shader Code	635
19.7	Summary	639
19.8	Exercises	640
Chapter 20 Shadow Mapping		643
20.1	Rendering Scene Depth	644
20.2	Orthographic Projections	646
20.3	Projective Texture Coordinates	649
	20.3.1 Code Implementation	651
	20.3.2 Points Outside the Frustum	651
	20.3.3 Orthographic Projections	652
20.4	Shadow Mapping	653
	20.4.1 Algorithm Description	653
	20.4.2 Biasing and Aliasing	654
	20.4.3 PCF Filtering	657
	20.4.4 Building the Shadow Map	660
	20.4.5 The Shadow Factor	665
	20.4.6 The Shadow Map Test	667
	20.4.7 Rendering the Shadow Map	667
20.5	Large PCF Kernels	668
	20.5.1 The DDX and DDY Functions	669
	20.5.2 Solution to the Large PCF Kernel Problem	669
	20.5.3 An Alternative Solution to the Large PCF Kernel Problem	671
20.6	Summary	673
20.7	Exercises	674
Chapter 21 Ambient Occlusion		675
21.1	Ambient Occlusion via Ray Casting	676
21.2	Screen Space Ambient Occlusion	680
	21.2.1 Render Normals and Depth Pass	680
	21.2.2 Ambient Occlusion Pass	681
	21.2.2.1 Reconstruct View Space Position	682
	21.2.2.2 Generate Random Samples	683
	21.2.2.3 Generate the Potential Occluding Points	685
	21.2.2.4 Perform the Occlusion Test	685
	21.2.2.5 Finishing the Calculation	686
	21.2.2.6 Implementation	686

21.2.3	Blur Pass	690
21.2.4	Using the Ambient Occlusion Map	694
21.3	Summary	695
21.4	Exercises	695
Chapter 22 Quaternions		697
22.1	Review of the Complex Numbers	698
22.1.1	Definitions	698
22.1.2	Geometric Interpretation	699
22.1.3	Polar Representation and Rotations	700
22.2	Quaternion Algebra	701
22.2.1	Definition and Basic Operations	701
22.2.2	Special Products	702
22.2.3	Properties	703
22.2.4	Conversions	703
22.2.5	Conjugate and Norm	704
22.2.6	Inverses	705
22.2.7	Polar Representation	706
22.3	Unit Quaternions and Rotations	707
22.3.1	Rotation Operator	707
22.3.2	Quaternion Rotation Operator to Matrix	709
22.3.3	Matrix to Quaternion Rotation Operator	710
22.3.4	Composition	712
22.4	Quaternion Interpolation	713
22.5	DirectX Math Quaternion Functions	718
22.6	Rotation Demo	719
22.7	Summary	723
22.8	Exercises	724
Chapter 23 Character Animation		727
23.1	Frame Hierarchies	728
23.1.1	Mathematical Formulation	729
23.2	Skinned Meshes	731
23.2.1	Definitions	731
23.2.2	Reformulating the Bones To-Root Transform	732
23.2.3	The Offset Transform	732
23.2.4	Animating the Skeleton	733
23.2.5	Calculating the Final Transform	735
23.3	Vertex Blending	737

23.4	Loading Animation Data from File	740
23.4.1	Header	740
23.4.2	Materials	741
23.4.3	Subsets	741
23.4.4	Vertex Data and Triangles	742
23.4.5	Bone Offset Transforms	743
23.4.6	Hierarchy	743
23.4.7	Animation Data	743
23.4.8	M3DLoader	746
23.5	Character Animation Demo	747
23.6	Summary	750
23.7	Exercises	751
Appendix A: Introduction to Windows Programming		753
A.1	Overview	754
A.1.1	Resources	754
A.1.2	Events, the Message Queue, Messages, and the Message Loop	754
A.1.3	GUI	755
A.1.4	Unicode	757
A.2	Basic Windows Application	757
A.3	Explaining the Basic Windows Application	761
A.3.1	Includes, Global Variables, and Prototypes	761
A.3.2	WinMain	762
A.3.3	WNDCLASS and Registration	763
A.3.4	Creating and Displaying the Window	765
A.3.5	The Message Loop	767
A.3.6	The Window Procedure	768
A.3.7	The MessageBox Function	770
A.4	A Better Message Loop	770
A.5	Summary	771
A.6	Exercises	772
Appendix B: High Level Shader Language Reference		773
	Variable Types	773
	Scalar Types	773
	Vector Types	773
	Swizzles	774
	Matrix Types	775
	Arrays	776

Structures	776
The <code>typedef</code> Keyword	777
Variable Prefixes	777
Casting	777
Keywords and Operators	778
Keywords	778
Operators	778
Program Flow	780
Functions	781
User Defined Functions	781
Built-in Functions	782
Constant Buffer Packing	785
Appendix C: Some Analytic Geometry	789
C.1 Rays, Lines, and Segments	789
C.2 Parallelograms	790
C.3 Triangles	791
C.4 Planes	792
C.4.1 DirectX Math Planes	793
C.4.2 Point/Plane Spatial Relation	793
C.4.3 Construction	794
C.4.4 Normalizing a Plane	795
C.4.5 Transforming a Plane	795
C.4.6 Nearest Point on a Plane to a Given Point	795
C.4.7 Ray/Plane Intersection	796
C.4.8 Reflecting Vectors	797
C.4.9 Reflecting Points	797
C.4.10 Reflection Matrix	797
C.5 Exercises	799
Appendix D: Solutions to Selected Exercises	801
Appendix E: Bibliography and Further Reading	803
<i>Index</i>	809