

The Direct3D Graphics Pipeline

Richard Atwater Thomson

August 13, 2006

Contents

Preface xxvii

I Preliminaries 1

1 Introduction 3

1.1	Overview	3
1.2	Display Technology	6
1.3	Gamma Correction	8
1.4	Color	10
1.5	Basic Mathematics	12
1.6	Homogeneous Coordinates	14
1.7	Vectors	14
1.8	Coordinate Systems	15
1.9	Matrices	17
1.10	Aliasing	18
1.11	Style Conventions	20
1.12	COM Objects	22
1.13	Code Techniques	27
1.14	Previous Versions of Direct3D	37
1.14.1	DirectX 1-2	37
1.14.2	DirectX 3	38
1.14.3	DirectX 5	38
1.14.4	DirectX 6	38
1.14.5	DirectX 7	38
1.14.6	DirectX 8.0	39
1.14.7	DirectX 8.1	39
1.14.8	DirectX 9.0c (June 2006)	39
1.15	Further Reading	40

2 Direct3D 43

2.1	Overview	43
2.2	A Minimal Direct3D Application	44
2.3	Direct3D Architecture	50
2.4	Direct3D HRESULTs	51

2.5	Windowed and Exclusive Mode	52
2.6	Device Types	52
2.7	Resources	53
2.8	IDirect3D9	57
2.9	Selecting a Device	59
2.10	Determining Available Resource Memory	61
2.11	Device Capabilities	61
2.12	Identifying a Particular Device	61
2.13	Creating the Device	63
2.14	Multiple Monitors	66
2.15	Adapter Group Devices	67
3	Direct3D Devices	69
3.1	Overview	69
3.2	IDirect3DDevice9	70
3.3	Capabilities	76
3.4	Resources	82
3.5	IDirect3DResource9	82
3.6	Destroying a Device	86
3.7	Miscellaneous Properties	86
3.8	Device Queries	87
3.8.1	Resource Manager Statistics Queries	90
3.8.2	Vertex Statistics Queries	91
3.8.3	Vertex Cache Queries	91
3.8.4	PIX Related Queries	92
3.9	Device States	93
3.9.1	Miscellaneous Render States	101
3.9.2	Setting Groups of State	101
3.10	Device State Blocks	104
3.11	Pure Devices	109
4	2D Applications	111
4.1	Overview	111
4.2	Pixel Surfaces	112
4.3	Accessing Surface Pixel Data	114
4.4	Using GDI On A Surface	119
4.5	Swap Chains	119
4.6	Presentation	122
4.7	Lost Devices and Reset	125
4.8	Video Scan Out	125
4.8.1	Cursor	126
4.8.2	Gamma Ramp	127
4.9	2D Pixel Copies	128
4.9.1	Pixel Copies Within Device Memory	129
4.9.2	Copies From System Memory To Device Memory	131
4.9.3	Copies From Device Memory To System Memory	133

4.10	Filling Rectangles	133
4.11	Window Messages	133
4.12	rt_2DApp Sample Application	135
II Geometry Processing		157
5	Modeling	159
5.1	Overview	159
5.2	Modeling Scenes	160
5.3	Visibility	161
5.4	Render Targets	163
5.5	Primitive Types	164
5.6	Vertex Data	167
5.7	Flexible Vertex Formats	167
5.8	Vertex Declarations	172
	5.8.1 Fixed-Function Declarations	177
5.9	Vertex Buffers	178
5.10	Indexed Primitives	183
	5.10.1 Index Buffers	184
5.11	The Vertex Shader	185
5.12	Drawing Primitives	186
5.13	Vertex Data Streams	188
5.14	Capabilities for Vertex Assembly	190
5.15	Enhanced Primitives	191
	5.15.1 Point Sprites	191
	5.15.2 <i>N</i> -Patches	192
5.16	Higher Order Surfaces	193
	5.16.1 Triangular Patches	195
	5.16.2 Rectangular Patches	196
5.17	Object Approximations	197
5.18	Further Reading	198
6	Vertex Transformations	199
6.1	Overview	199
6.2	Vertex Processing	200
6.3	Transformation Matrices	201
6.4	Order of Transformations	203
6.5	World Transformation	205
6.6	Transformation Hierarchy	206
6.7	Vertex Blending	207
	6.7.1 Basic Vertex Blending	212
	6.7.2 Indexed Vertex Blending	213
	6.7.3 Vertex Tweening	214
	6.7.4 Blending Within Device Limitations	215
6.8	Vertex Fog	216

6.9	Face Culling	218
6.10	Clipping	220
	6.10.1 User Clip Planes	221
	6.10.2 View Frustum Clipping	222
	6.10.3 Guard Band Clipping	223
6.11	Screen Space and the Viewport	224
6.12	<code>rt.VertexBlend</code> Sample Application	225
6.13	Further Reading	237
7	Viewing and Projection	239
7.1	Overview	239
7.2	Viewing Through a Virtual Camera	240
7.3	Planar Geometric Projections	242
	7.3.1 Parallel Projections	242
	7.3.2 Perspective Projections	244
7.4	Projection Transformation	244
	7.4.1 Projection Matrices	245
7.5	Further Reading	246
7.6	<code>rt.Views</code> Sample Application	246
8	Lighting and Materials	255
8.1	Overview	255
8.2	Transformed Vertices	256
8.3	Lighting Calculations	257
8.4	Surface Material Properties	260
8.5	Light Sources	262
	8.5.1 Directional Lights	264
	8.5.2 Point Lights	264
	8.5.3 Spot Lights	264
	8.5.4 Light Attenuation	265
8.6	The Illumination Model	266
	8.6.1 Transforming to Camera Space	267
	8.6.2 Ambient Reflection	267
	8.6.3 Diffuse Reflection	267
	8.6.4 Specular Reflection	268
	8.6.5 Emission	269
	8.6.6 Summary	269
8.7	<code>rt.Lighting</code> Sample Application	270
9	Vertex Shaders	301
9.1	Overview	301
9.2	Vertex Shader Architecture	302
	9.2.1 Input Registers	303
	9.2.2 Constant Registers and the Address Register	304
	9.2.3 Output Registers	305
	9.2.4 Temporary Registers	305

9.2.5	The Loop Counter Register	305
9.2.6	The Predicate Register	306
9.2.7	Sampler Registers	306
9.2.8	Register Modifiers	306
9.3	Vertex Shader 1.1 Architecture	307
9.4	Vertex Shader 2.0 Architecture	308
9.5	Vertex Shader 2.x Architecture	309
9.6	Vertex Shader 3.0 Architecture	310
9.7	Shader Instruction Syntax	310
9.8	Execution Model	312
9.9	Software Vertex Processing	312
9.10	Vertex Shader 1.1 Instructions	313
9.10.1	Declaration Instructions	315
9.10.2	Basic Arithmetic Instructions	316
9.10.3	Matrix Instructions	318
9.10.4	Comparison Instructions	319
9.10.5	Lighting Instructions	320
9.11	Vertex Shader 2.0 Instructions	321
9.11.1	Version 2.0 Flow Control Nesting Limits	325
9.12	Vertex Shader 2.x Instructions	326
9.12.1	Version 2.x Flow Control Nesting Limits	329
9.13	Vertex Shader 3.0 Instructions	330
9.13.1	Version 3.0 Flow Control Nesting Limits	332
9.14	Manipulating Shaders	332
9.15	Drawing Multiple Instances	334
9.16	Shader Design	335
9.17	Common Computations	336
9.17.1	Constant Generation	337
9.17.2	Fractional Part	337
9.17.3	Absolute Value	337
9.17.4	Division	338
9.17.5	Square Root	338
9.17.6	Conditional Selection	338
9.17.7	Clamping To An Interval	339
9.17.8	Floor and Ceiling	339
9.17.9	Vector Cross Product	340
9.17.10	Vector Normalization	340
9.17.11	Transposed Matrix Multiplication	340
9.17.12	Signum Function	341
9.17.13	Minimum and Maximum Vector Component	341
9.17.14	Trigonometric Functions	342
9.17.15	Exponential and Logarithmic Functions	344
9.18	Fixed-Function Processing	346
9.18.1	Register Layout	346
9.18.2	Constant Register Layout	346
9.18.3	Coordinate Transformation	348

9.18.4	Vertex Blending	348
9.18.5	Position Output	349
9.18.6	Normalize Eye Normal	349
9.18.7	Non-Homogeneous Eye Position	349
9.18.8	Eye Space Vectors	349
9.18.9	Fog Output	350
9.18.10	Point Parameters	350
9.18.11	Lighting	350
9.18.12	Texture Coordinate Generation	352
9.19	Beyond Fixed-Function Processing	353
9.20	<code>rt.VertexShader</code> Sample Application	354
9.21	Vertex Shader SDK Samples	354
9.21.1	<code>BumpSelfShadow</code>	354
9.21.2	<code>BumpWaves</code>	355
9.21.3	<code>DolphinVS</code>	355
9.21.4	<code>SkinnedMesh</code>	356
9.21.5	<code>SphereMap</code>	357
9.21.6	<code>VertexBlend</code>	357
9.21.7	<code>VertexShader</code>	357
 III Pixel Processing		377
10 Rasterization and Shading		379
10.1	Overview	379
10.2	Shading and Filling	380
10.3	Scanline Rendering	382
10.4	Source Pixel Generation	383
10.5	Source Pixel Data	385
10.6	<code>rt.Rasterize</code> Sample Application	386
 11 Basic Texturing		403
11.1	Overview	403
11.2	Direct3D Multitexture Architecture	404
11.3	Texture Resources	407
11.3.1	<code>IDirect3DBaseTexture9</code>	409
11.3.2	<code>IDirect3DTexture9</code>	411
11.3.3	<code>IDirect3DCubeTexture9</code>	414
11.3.4	<code>IDirect3DVolume9</code>	416
11.3.5	<code>IDirect3DVolumeTexture9</code>	420
11.4	Texture Formats	422
11.4.1	Palette Based Textures	422
11.4.2	Compressed Texture Formats	423
11.5	Texture Coordinate Processing	427
11.5.1	Texture Coordinate Generation	430
11.5.2	Texture Coordinate Transformation	430

11.5.3	Texture Addressing	432
11.5.4	Texture Wrapping	434
11.6	Texture Sampling	435
11.7	Texture Stage Processing	439
11.8	Examples	445
11.8.1	Single Stage Light Maps	446
11.8.2	Decals	447
11.9	Specular Addition	448
11.10	Textured Primitives	449
11.11	Further Reading	451
11.12	Sample Applications	451
11.12.1	Book Samples	452
11.12.2	SDK Samples	452
11.12.3	rt_Texture Sample Application	453
12	Advanced Texturing	469
12.1	Overview	469
12.2	Multiple Texture Stages	469
12.3	Environment Mapping	470
12.3.1	Cubic Environment Mapping	470
12.3.2	Spherical Environment Mapping	470
12.4	Bump Mapping Techniques	470
12.4.1	Dot3 Bump Mapping	470
12.4.2	Emboss Mapping	470
12.4.3	Environment Mapped Bump Mapping	470
12.4.4	Tangent Basis Bump Mapping	470
12.5	Texturing Samples	470
13	Pixel Shaders	471
13.1	Overview	471
13.2	Pixel Shader Architecture	472
13.3	Pixel Shader Architecture Versions	472
13.3.1	Pixel Shader 1.0	472
13.3.2	Pixel Shader 1.1	482
13.3.3	Pixel Shader 1.2	482
13.3.4	Pixel Shader 1.3	482
13.3.5	Pixel Shader 1.4	482
13.3.6	Pixel Shader 2.0	483
13.3.7	Pixel Shader 2.x	483
13.3.8	Pixel Shader 3.0	483
13.4	Instruction Set	483
13.4.1	Declaration Instructions	483
13.4.2	Basic Arithmetic Instructions	484
13.4.3	Comparison Instructions	484
13.4.4	Basic Texture Addressing	485
13.4.5	Bump Mapping Instructions	487

13.5	PS 2.0 Instructions	487
13.6	PS 2.x Instructions	488
13.7	PS 3.0 Instructions	488
13.8	Manipulating Pixel Shaders	489
13.9	Texture Stage States	490
13.10	Examples	490
13.11	Debugging a Pixel Shader	490
14	The Frame Buffer	491
14.1	Overview	491
14.2	Fog Blending	492
14.3	Alpha Test	492
14.4	The Z Buffer and Visibility	494
14.5	Stencil Test	497
14.6	Alpha Blending	501
14.7	Dithering	505
14.8	Multisampling	506
14.9	Writing to the Render Target	508
14.10	rt.FrameBuffer Sample Application	508
IV	D3DX Library	573
15	D3DX Utility Library	575
15.1	Overview	575
15.2	Headers and Libraries	576
15.3	Data Types	576
15.4	Helper Objects	576
15.5	Meshes	577
15.6	Resources	577
15.7	Resource Image Information	578
15.8	Resource Requirements	579
15.9	Format Conversion	580
15.10	Creating Texture Resources	582
15.10.1	Creating Uninitialized Textures	582
15.10.2	Creating Textures From Files	583
15.10.3	Creating Textures From Files In Memory	585
15.10.4	Creating Textures From Resources	586
15.11	Loading Resources	588
15.12	Saving Resources	591
15.13	Filling Textures	591
15.14	Filtering Textures	592
15.15	Creating Normal Maps	593
15.16	Miscellaneous	594
15.16.1	Macros	594
15.16.2	Fresnel's Formulas	594

15.17	Error Handling	595
15.18	ID3DXBuffer	596
15.19	Vertex Declarations	597
15.20	Intersection Testing	597
15.21	Shader Assembly	598
15.22	Further Reading	599
16	D3DX Concrete Types	601
16.1	Overview	601
16.2	Colors	601
16.3	Vectors	604
16.4	Matrices	614
16.4.1	Arithmetic Functions	617
16.4.2	Identity, Translation, and Scaling Matrices	618
16.4.3	Rotation Matrices	618
16.4.4	Composite Transformation Matrices	619
16.4.5	Viewing Matrices	619
16.4.6	Orthographic Projection Matrices	620
16.4.7	Perspective Projection Matrices	621
16.4.8	Shadow and Reflection Matrices	622
16.4.9	Matrix Helper Classes	622
16.5	Planes	623
16.6	Quaternions	625
17	D3DX Helper Objects	631
17.1	Overview	631
17.2	Matrix Stacks	631
17.3	Fonts	634
17.4	Sprites	638
17.5	Surface Rendering	640
17.6	Environment Map Rendering	642
18	High Level Shaders and Effects	645
18.1	Overview	645
18.2	Visual Effects	645
18.2.1	Text Syntax	645
18.2.2	Effect Objects	652
18.3	Sample Applications	657
19	D3DX Mesh Objects	671
19.1	Overview	671
19.2	Use of <code>const</code> Qualifiers	672
19.3	Triangle Adjacencies	674
19.4	Vertex and Triangle Remapping	675
19.5	Base Mesh	676
19.6	Mesh	683

19.6.1	Creating Meshes From Scratch	685
19.6.2	Primitive Construction	687
19.6.3	Reducing Mesh Data	689
19.6.4	Adding Mesh Data	693
19.6.5	Mesh Intersection Tests	694
19.6.6	Mesh Conversion	695
19.6.7	Meshes in X Files	698
19.7	Progressive Mesh	700
19.8	Simplification Mesh	704
19.9	Skin Mesh	707
19.9.1	Skin Meshes in X Files	713
19.10	rt_D3DXSphere Sample Application	714
19.11	SDK Samples	714
19.12	Further Reading	715
20	D3DX Animation Objects	717
20.1	Overview	717
V	The Application	719
21	X Files	721
21.1	Overview	721
21.2	X File Structure	722
21.2.1	X File Header	722
21.2.2	Base Data Types	723
21.2.3	Templates	723
21.2.4	Data Nodes	723
21.2.5	Identifiers	724
21.3	Text Encoding	724
21.3.1	Reserved Words	725
21.3.2	Templates	725
21.3.3	Data Nodes	726
21.3.4	Nested Data	728
21.4	Binary Encoding	729
21.4.1	Token Records	730
21.5	Predefined Templates	731
21.5.1	Animation	732
21.5.2	Animation Key	732
21.5.3	Animation Options	733
21.5.4	Animation Set	734
21.5.5	Boolean	734
21.5.6	Boolean 2D	735
21.5.7	Color RGB	735
21.5.8	Color RGBA	735
21.5.9	Coords 2D	736

21.5.10 External Visual	736
21.5.11 Float Keys	736
21.5.12 Frame	737
21.5.13 Frame Transform Matrix	737
21.5.14 FVF Data	737
21.5.15 Guid	738
21.5.16 Header	739
21.5.17 Indexed Color	739
21.5.18 Inline Data	740
21.5.19 Material	740
21.5.20 Material Wrap	741
21.5.21 Matrix 4x4	741
21.5.22 Mesh	741
21.5.23 Mesh Face	742
21.5.24 Mesh Face Wraps	743
21.5.25 Mesh Material List	743
21.5.26 Mesh Normals	744
21.5.27 Mesh Texture Coords	744
21.5.28 Mesh Vertex Colors	745
21.5.29 Patch	745
21.5.30 Patch Mesh	747
21.5.31 Progressive Mesh Attribute Range	747
21.5.32 Progressive Mesh Info	747
21.5.33 Progressive Mesh Vertex Split Record	748
21.5.34 Progressive Mesh	748
21.5.35 Property Bag	749
21.5.36 Right Handed	749
21.5.37 Skin Weights	749
21.5.38 String Property	750
21.5.39 Texture Filename	750
21.5.40 Timed Float Keys	751
21.5.41 Url	751
21.5.42 Vector	751
21.5.43 Vertex Duplication Indices	752
21.5.44 X Skin Mesh Header	752
21.6 Headers, Libraries and Interfaces	753
21.7 IDirectXFile	753
21.8 IDirectXFileEnumObject	756
21.9 IDirectXFileObject	756
21.10 IDirectXFileData	757
21.11 IDirectXFileBinary	760
21.12 IDirectXFileDataReference	761
21.13 IDirectXFileSaveObject	762
21.14 Reading Scene Data	763
21.15 Writing Scene Data	764
21.16 Exporting and Conversion	764

21.17	Example X Files	765
21.18	rt_MakeScene Sample Program	767
22	Debugging	789
22.1	Overview	789
22.2	Version Control Systems	790
22.3	C++ Debugging	790
22.3.1	Dynamic Memory Allocation	790
22.3.2	Accidental Assignments	791
22.3.3	String Handling	792
22.3.4	Casts	792
22.3.5	C++ Exceptions	792
22.4	Debugging Windows Applications	793
22.4.1	ANSI vs. Unicode	794
22.4.2	Security Environment	794
22.4.3	Special Directories	794
22.4.4	Output Debug Stream	796
22.4.5	Structured Exception Handling	797
22.4.6	Stack Traces	799
22.4.7	Mini Dumps	799
22.4.8	Debugging Buffer Overruns and Underruns	800
22.4.9	Fast User Switching in XP	800
22.5	Debugging With Visual C++ 6	801
22.5.1	Decoding Direct3D HRESULTs	801
22.5.2	Highlighting Direct3D Identifiers	801
22.5.3	Expanding Direct3D Structures	801
22.5.4	Limiting Step-Into	802
22.5.5	Remote Debugging	803
22.5.6	Multiple Monitor Debugging	803
22.6	Debugging With Direct3D	804
22.6.1	Debug and Retail Runtimes	804
22.6.2	Checking HRESULTs	805
22.6.3	Reference Rasterizer	805
22.6.4	DirectX Error Routines	805
22.6.5	Reproducing Problems	806
22.6.6	Exclusive Mode Problems	806
22.6.7	Driver Problems	806
22.6.8	Shader Problems	807
22.6.9	Texturing Problems	808
22.6.10	Processor Specific Graphics Processing	808
22.6.11	Interfering Applications	809
22.7	DirectX Diagnostic Tool	809
22.8	DirectX Control Panel Applet	810
22.9	Getting Help	812
22.9.1	Asking Smart Questions	813
22.9.2	IHV Developer Relations	814

22.10 Filing Bug Reports	814
23 Pipeline Performance	817
23.1 Overview	817
23.2 Pipeline Bottlenecks	819
23.3 Optimization Techniques	819
23.3.1 Reordering Data	819
23.3.2 Reducing Data	820
23.3.3 Redesigning	821
23.4 Strategies	821
23.4.1 Vertex Processing	822
23.4.2 Pixel Processing	822
23.4.3 Frame Buffer Processing	822
23.5 Image Compositing	822
23.6 Vertex Buffers	822
23.7 Texture Map Size	822
23.8 Scalable Shading	822
23.9 Model Simplification	822
23.10 Alpha Test	823
23.11 Stencil Test	823
23.12 Z Buffer Test	823
23.13 Guard Band clip	823
24 Installation and Setup	825
24.1 Overview	825
24.2 Determining the DirectX Version	826
24.3 DirectSetup	827
24.3.1 Determining the DirectX Version	827
24.3.2 Installing the DirectX Runtime	827
24.3.3 Using a Callback Procedure	829
24.3.4 Registering With DirectXPlay Lobby	832
24.3.5 Preparing the Redistributables	833
24.4 Windows Installer	833
24.4.1 Using a Launch Sequencer	834
24.4.2 Requiring the Proper Runtime Version	834
24.4.3 Using a Custom Action to Invoke DirectSetup	835
24.5 Logo Requirements	835
24.6 AutoRun Enabled Installations	836
24.7 Testing Installation Programs	836
24.8 Sample Program	836
VI Appendices	839
A A Guided Tour of the SDK	841
A.1 Overview	841

A.2	A Quick Start	841
A.3	Installing the SDK	842
A.4	Organization	842
A.5	Documentation	845
A.6	Include Files	845
A.7	Libraries	846
A.8	Tutorials	846
A.9	Tools	847
A.10	SDK Samples	847
B	D3DFrame	849
B.1	Overview	849
B.2	AppWizard	850
B.3	Files	850
B.4	CD3DApplication	850
B.5	CD3DScreensaver	854
B.6	CD3DFont	854
B.7	CD3DFile	855
B.8	CD3DFrame	856
B.9	CD3DMesh	857
B.10	CD3DScreensaver	858
B.11	CD3DArcBall	860
B.12	CD3DCamera	861
B.13	D3DUtil	862
B.14	DXUtil	863
C	Shader Tokens	867
C.1	Overview	867
C.2	Instruction Tokens	867
D	Sample Code	873
D.1	Overview	873
D.2	C++ Sample Code	873
	D.2.1 rt Library Code	875
D.3	Visual Basic Sample Code	876
	Glossary	877
	Index	877

Source Code Listings

1.1	dshow.cpp: A simple AVI file player.	24
1.2	<rt/hr.h>: mapping HRESULTs to exceptions.	29
1.3	dshow2.cpp: An improved simple AVI player.	36
2.1	minimal.cpp: A minimal well-behaved Direct3D application.	44
3.1	<rt/states.h>: Setting aggregate state	101
4.1	<rt/surface.h>: A surface lock helper class.	116
4.2	rt_2DApp.cpp: A simple 2D application using StretchRect.	136
5.1	<rt/vertexbuf.h>: A vertex buffer lock helper class.	182
6.1	rt_VertexBlend.cpp: Vertex blending.	225
7.1	rt_Views.cpp: Drawing multiple views	246
8.1	rt_Lighting.cpp: Lighting and material parameters.	270
9.1	BumpShader.vsh: BumpSelfShadow shader 1.	359
9.2	BumpShader2.vsh: BumpSelfShadow shader 2.	360
9.3	BumpShader3.vsh: BumpSelfShadow shader 3.	361
9.4	BumpShader4.vsh: BumpSelfShadow shader 4.	362
9.5	BumpWaves vertex shader.	363
9.6	DolphinTween.vsh: DolphinVS shader 1.	364
9.7	DolphinTween2.vsh: DolphinVS shader 2.	366
9.8	SkinnedMesh4.vsh: SkinnedMesh shader 4.	368
9.9	SphereMap shader	371
9.10	Blend.vsh: VertexBlend shader	372
9.11	Ripple.vsh: VertexShader shader	374
9.12	FogShader.vsh: VolumeFog shader 1	376
10.1	rt_Rasterize.cpp: Rasterization.	386
11.1	rt_Texture.cpp: Texturing.	453
14.1	rt_FrameBuffer.cpp: Frame buffer processing.	510
18.1	water.fx: Water sample effect file.	658
18.2	cubemap.fx: CubeMap sample effect file.	667
21.1	rt_MakeScene.cpp: X file output.	769
21.2	frame.h: X file output.	772
21.3	frame.cpp: X file output.	776

Figures

1.1	Subsystems of a typical 2D graphics adapter.	7
1.2	Left-handed, right-handed and screen coordinate systems	16
1.3	Illustration of Aliasing.	18
1.4	Antialiasing During Resampling.	20
2.1	Relationship of Direct3D to other Windows subsystems.	51
2.2	Direct3D device architecture	51
3.1	Resource interface hierarchy.	82
3.2	Query object state transitions.	88
4.1	Semantics of D3DSWAPEFFECT on a swap chain	123
5.1	Illustration of a cycle of occlusion with four rectangles.	162
5.2	Illustration of primitive types	165
5.3	Modeling with primitive types	166
5.4	Relationship between arguments to <code>DrawIndexedPrimitive</code>	190
5.5	Point sprite structure in screen space.	192
5.6	Bézier triangular patch vertex order.	195
6.1	Order of transformations	204
6.2	Illustration of a composite transformation.	205
6.3	Jointed linkage transformation hierarchy	207
6.4	Visualization of vertex blending.	208
6.5	Sample blend weight functions.	209
6.6	The effect of blend weight functions on vertex blending.	210
6.7	The effect of the world transformation on vertex blending.	211
6.8	Fog factor functions.	217
6.9	Computing a triangle's plane normal	218
6.10	Relationship of triangles to the line of sight in culling.	219
6.11	Illustration of clipping	221
7.1	Viewing	240
7.2	Taxonomy of planar geometric projections.	242
8.1	Lighting and materials control flow	258
8.2	Computing vertex surface normals for triangles.	259

8.3	Light source geometry.	264
8.4	Lambert’s law of diffuse reflection.	268
8.5	The geometry of specular reflection.	269
9.1	Vertex shader architecture	302
9.2	Shader assembly syntax	311
10.1	Scanline rendering of a triangle	382
10.2	Rasterization of a triangle	384
11.1	Multitexture cascade.	405
11.2	Organization of mipmapped textures.	408
11.3	Cube texture organization.	415
11.4	Volume organization.	417
11.5	Organization of mipmapped volume textures.	421
11.6	Texel organization within a block for compressed textures.	423
11.7	Sample block of compressed texels.	424
11.8	Texture address space.	428
11.9	Illustration of the reflection vector.	430
11.10	Texture coordinate addressing modes	432
11.11	Texture coordinate wrapping	435
11.12	Texture minification filtering	437
11.13	Texture stage processing	439
13.1	Pixel shader architecture overview.	472
13.2	Pixel shader instruction execution.	473
18.1	Structure of an effect file	646
18.2	Effect file syntax summary	646
19.1	D3DX Mesh Interface Hierarchy.	672
19.2	Triangle Adjacency	674
19.3	Bowtie arrangement of vertices in a triangle mesh	689
19.4	Mesh simplification by collapsing edges.	691
21.1	X file grammar	724
21.2	Template grammar	725
21.3	Data node grammar	727
21.4	Order of indices for Bézier patches.	746
21.5	.x File Interface Hierarchy	753
22.1	DirectX Diagnostic Tool	810
22.2	DirectX Control Panel Applet	811
23.1	The performance optimization process.	817
A.1	SDK Installation Wizard	843

A.2 DirectX 9.0c SDK Directory Structure. 844
A.3 HTML Help Viewer displaying the DirectX SDK documentation. 845
B.1 D3DFrame Class Hierarchy 849
D.1 Samples Directory Structure. 874

Tables

1.1	Vector operations	15
1.2	Matrix operations	17
1.3	Macros for HRESULT return values.	24
3.1	Device Methods by Pipeline Section	76
3.2	Device Capabilities by Pipeline Section	81
3.3	Implicit Query Beginnings	89
3.4	Query Data Types	89
3.5	Device States by Pipeline Section	94
3.6	Device State Data Types	100
3.7	Methods captured by BeginStateBlock	106
3.8	Device Properties by State Block Type	109
3.9	Device property support on a pure HAL device.	110
4.1	Presentation intervals supported in windowed and exclusive mode.	124
4.2	Combinations of source and destination surfaces supported with UpdateSurface.	132
5.1	Flexible Vertex Format	170
5.2	Vertex usage for the fixed-function pipeline.	178
5.3	Relationship of primitive count to vertex count	187
5.4	Vertex count for triangular Bézier patches.	196
5.5	Rectangular patch vertex grid sizes.	197
8.1	Material parameter defaults.	260
8.2	The D3DLIGHT9 members used by each light type.	263
8.3	Light attenuation coefficient examples.	265
8.4	Symbols used for D3DLIGHT9 and D3DMATERIAL9.	266
9.1	Vertex shader version instruction counts.	303
9.2	Vertex shader version register counts.	304
9.3	Vertex shader register modifiers	306
9.4	Vertex Shader 1.1 Instruction Summary	314
9.5	Vertex Shader 2.0 Instruction Summary	322

9.6	<code>sincos</code> results based on write mask. ? indicates a component that has an undefined value after the instruction executes. * indicates a component that is not changed by the instruction.	323
9.7	Vertex shader version nesting limits.	326
9.8	Vertex Shader 2.x Instruction Summary	326
9.9	Vertex shader instructions supporting predication	327
9.10	Vertex Shader 3.0 Instruction Summary	330
9.11	Register layout for fixed-function emulation.	346
9.13	BumpSelfShadow vertex shader constants.	355
9.14	Input registers used by the DolphinVS sample	356
11.1	Summary of texture operations.	442
13.1	Pixel shader architecture versions.	473
13.2	Pixel shader instruction availability.	474
13.3	Pixel shader instruction set summary.	475
13.4	Pixel shader 1.0 register restrictions.	476
13.5	Pixel shader 1.1 register restrictions.	477
13.6	Pixel shader 1.2 register restrictions.	478
13.7	Pixel shader 1.3 register restrictions.	479
13.8	Pixel shader 1.4 register restrictions in phase 1.	480
13.9	Pixel shader 1.4 register restrictions in phase 2.	481
13.10	Pixel shader 2.0 instructions in assembly syntax.	487
13.11	Pixel shader 2.x instructions in assembly syntax.	488
13.12	Pixel shader 3.0 instructions in assembly syntax.	488
13.13	Texture stage states honored by pixel shaders.	490
14.1	Summary of stencil operations	499
14.2	Blend factors described by <code>D3DBLEND</code>	503
14.3	Blending functions described by <code>D3DBLENDOP</code>	503
14.4	Blend factors for common compositing operators.	504
14.5	Command-line switches supported by <code>rt_FrameBuffer</code>	510
16.1	<code>D3DXCOLOR</code> class members.	603
16.2	<code>D3DXCOLOR</code> global functions.	604
16.3	<code>D3DX</code> vector class members	605
16.4	<code>D3DX</code> vector class function summary.	608
16.5	Vector function arithmetic operation summary	609
16.6	<code>D3DXMATRIX</code> class members	615
16.7	<code>D3DXQUATERNION</code> class members.	627
18.1	Effect value syntax	648
19.1	Vertex data for a flat shaded cube.	681
19.2	Face data for a flat shaded cube.	681
21.1	X File Header	722

21.2	Meaning of terminal X file grammar tokens	724
21.3	X file tokens	729
21.4	X file record tokens	730
21.5	Animation key types	733
22.1	Common CSIDL values.	795
22.2	Format specifiers for debugger structure expansion formatting.	802
22.3	Registry keys to disable CPU specific software processing.	809
22.4	Community resources	813
22.5	IHV Contact Information	815
24.1	<code>GetDXVersion</code> return values.	826
24.2	<code>DirectXSetupGetVersion</code> return values.	828
A.1	Direct3D related include files.	846
A.2	Summary of SDK libraries for Direct3D applications.	847
A.3	Summary of SDK tools.	847

COM Interfaces

1.1	IUnknown	23
2.1	IDirect3D9	57
3.1	IDirect3DDevice9	71
3.2	IDirect3DResource9	83
3.3	IDirect3DQuery9	87
3.4	IDirect3DStateBlock9	105
4.1	IDirect3DSurface9	112
4.2	IDirect3DSwapChain9	120
5.1	IDirect3DVertexDeclaration9	176
5.2	IDirect3DVertexBuffer9	179
5.3	IDirect3DIndexBuffer9	184
5.4	IDirect3DVertexShader9	186
11.1	IDirect3DBaseTexture9	410
11.2	IDirect3DTexture9	413
11.3	IDirect3DCubeTexture9	415
11.4	IDirect3DVolume9	417
11.5	IDirect3DVolumeTexture9	420
15.1	ID3DXBuffer	597
17.1	ID3DXMatrixStack	632
17.2	ID3DXFont	634
17.3	ID3DXSprite	639
17.4	ID3DXRenderToSurface	640
17.5	ID3DXRenderToEnvMap	642
18.1	ID3DXEffect	652
19.1	ID3DXBaseMesh	676
19.2	ID3DXMesh	683
19.3	ID3DXPMesh	701
19.4	ID3DXSPMesh	705
19.5	ID3DXSkinInfo	708
21.1	IDirectXFile	754

21.2	IDirectXFileEnumObject	756
21.3	IDirectXFileObject	757
21.4	IDirectXFileData	757
21.5	IDirectXFileBinary	760
21.6	IDirectXFileDataReference	761
21.7	IDirectXFileSaveObject	762

Preface

Who Should Read This Book

This book is for C++ programmers who want to understand how to create images rendered from scenes using Microsoft's Direct3D application programming interface.

The reader is assumed to be familiar with basic Win32 programming: windows, messages, keyboard and mouse input, device contexts, GDI, and so-on. If these areas are new to the reader, they are encouraged to start first with a basic text on Windows. The author recommends *Win32 Programming* by Brent E. Rector and Joseph M. Newcomer.

Direct3D is provided by the system as a set of interfaces provided by COM objects. No existing knowledge of COM is assumed and the basics of COM as used with Direct3D are reviewed.

COM objects can be used from a variety of languages. This book will use C++ for its programming examples and the reader should be comfortable with C++ concepts such as classes, inheritance, abstract interfaces, and generic programming with template classes such as those in the ANSI C++ standard library. The author believes in the KISS (Keep It Simple, Stupid) maxim – C++ features are used only when they provide a clear benefit.

Three dimensional scenes are described to the computer with the assistance of some mathematical concepts. Readers should be comfortable with the mathematics of geometry, trigonometry, vectors and matrices. A summary review of the mathematics common in computer graphics is given for reference.

What's Not In This Book

This book provides a complete explanation of the Direct3D graphics pipeline provided in release 9.0c of Microsoft's DirectX technology. A sophisticated 3D multimedia application may use many other technologies such as DirectSound and DirectMusic for sound and music, DirectInput for input device processing, DirectPlay or TCP/IP for networking, physically-based models and dynamics simulations for realistic animation, persistent storage for scenes, interactive modelling of scenes, and Win32 for GUI presentation.

In this book we will touch on input processing with the core Win32 messages, but other areas of DirectX and application processing will not be covered. References are provided for other books that cover these areas when they arise in our discussion of the pipeline.

Thanks

List of people to thank.

This book is dedicated to:

Sylvia Mead Atwater

1901-1990

who shared her joy of knowledge,

Anne and Bill Thomson

for making all things possible,

and

Pete Townshend

for over 20 years of musical ecstasy.

