

Inhaltsverzeichnis

	Vorwort	21
	Einleitung	25
	Inhalt dieses Buches	25
	Was steht in diesem Buch?	25
	Was steht nicht in diesem Buch?	26
	Konzeptionelle Vorgehensweise	27
	Verwendete Werkzeuge	29
	Hardware-Voraussetzungen	30
	Was ist ZFX?	31
	Historie von ZFX	32
	Entwickler-Community ZFX	32
	Entwickler-Event zfxCON	33
Teil 1	Einführung in die Thematik	35
Kapitel 1	3D-Engines und Spieleprogrammierung	37
1.1	Modewort 3D-Engine	37
1.2	Eine Art von Magie	41
	Nostalgie und Neuzeit	41
	Der Zauberer von Oz	43
	... und die Verantwortung eines Gottes	44
1.3	Entstehungsprozess eines Spiels	45
	Spielerprogrammierung (Game-Programming)	47
	Spieler-Design (Game-Design)	48
	Spiel-Entwurf (Game-Proposal)	50

	Publisher, Veröffentlichung und Erlös	55
	Verkaufszahlen.	57
	Alternativen	58
1.4	Sage mir, mit wem du gehst	59
1.5	DirectX und OpenGL, Versionspolitik	61
1.6	Warum schwer, wenn's auch einfach geht?	62
1.7	Auf los geht's los	63
Kapitel 2	Design der ZFXEngine	65
2.1	Anforderungen an die Engine	65
2.2	API-Unabhängigkeit durch Interface-Definitionen	67
2.3	Struktur der ZFXEngine	69
2.4	Komponenten der ZFXEngine	70
	ZFXRenderDevice Interface	70
	ZFXInputDevice Interface.	71
	ZFXNetworkDevice Interface	71
	ZFXAudioDevice Interface	71
	ZFX3D Bibliothek	72
	ZFXGeneral Bibliothek.	72
2.5	Ein Blick zurück, zwei Schritt nach vorn	72
Kapitel 3	Rahmenanwendung der ZFXEngine	73
3.1	Begriffsbestimmung Interface	73
3.2	Unser Interface	75
3.3	Der Arbeitsbereich für unsere Implementierung	77
	ZFXRenderer, eine statische Bibliothek als Manager	79
	ZFXD3D, eine dynamische Bibliothek als Render-Device	80
	ZFXRenderDevice, ein Interface als abstrakte Klasse.	82
3.4	Implementierung der statischen Bibliothek	84
3.5	Implementierung der dynamischen Bibliothek	89
	Exportierte Funktionen	91
	Komfort durch einen Dialog	93

	Initialisierung, Enumeration und Shutdown	99
	Zwischen Child-Windows wechseln	106
	Render-Funktionen	107
3.6	Testlauf der Implementierung	109
3.7	Ein Blick zurück, zwei Schritt nach vorn	115
Kapitel 4	Schnelle 3D-Mathematik	117
4.1	Schnell, schneller, am schnellsten	117
	Grundlagen der Assembler-Programmierung	119
	Einführung in SIMD	122
	Wie sag ich's meinem Compiler?	133
	Identifikation einer CPU.	134
	Unterstützung für SSE zur Laufzeit überprüfen	140
4.2	Rechnen mit Vektoren	141
	Grundlegende (arithmetische) Operationen	144
	Komplexere Operationen mit SSE-Unterstützung	147
4.3	Rechnen mit Matrizen	153
	Grundlegende Operationen	155
4.4	Rechnen mit Strahlen	158
	Grundlegende Operationen	159
	Kollision mit Dreiecken	161
	Kollision mit Ebenen	162
	Kollision mit Bounding-Boxen	164
4.5	Rechnen mit Ebenen	169
	Grundlegende Operationen	171
	Kollision mit Dreiecken	172
	Kollision zwischen Ebenen	173
	Kollision mit Bounding-Boxen	174
4.6	Rechnen mit AABB und OBB	176
	Grundlegende Operationen und Culling	178
	Kollision mit Dreiecken	182
	Kollision zweier orientierter Boxen	185

Inhaltsverzeichnis

	Ebenen einer AABB	185
	Strahl in AABB	186
4.7	Rechnen mit Polygonen	186
	Grundlegende Operationen	187
	Punkte für das Polygon festlegen	189
	Clippen eines Polygons	192
	Culling mit Bounding-Boxen	199
	Kollision mit Strahlen	201
4.8	Rechnen mit Quaternionen	203
	Einführung in den 4D-Raum	203
	Grundlegende Operationen	206
	Multiplikation zweier Quaternionen	207
	Konstruktion aus Euler-Winkeln	208
	Rotationsmatrix zu einem Quaternion	209
4.9	Ein Blick zurück, zwei Schritt nach vorn	210
Teil 2	Rendern von Grafik	213
Kapitel 5	Materialien, Texturen und Transparenz	215
5.1	Mittleres Management	215
5.2	Eine Klasse für Skins	219
	Texturen	219
	Licht und Material.	220
	Grundlegende Strukturen.	224
	Interface-Definition für einen Skin-Manager	230
	Skin-Manager des Direct3D-Renders	233
	Farben und Materialien vergleichen	237
	Skins austeilen.	238
5.3	Skins und Materialien aufnehmen	239
5.4	Texturen aufnehmen	241
	Texturen zu den Skins hinzufügen	242
	Grafikdateien als Texturen laden	245

5.5	Transparenz der Texturen einstellen	249
	Color-Keys über Alpha Channels	252
	Allgemeine Transparenz über Alpha Channels	254
5.6	Ein Blick zurück, zwei Schritt nach vorn	256
Kapitel 6	Das Render-Interface der ZFXEngine	259
6.1	Simplizität versus Flexibilität	259
6.2	Projekteinstellungen	260
6.3	Sicht und Projektion	262
	Multiple Stages	264
	Viewports, Viewmatrizen und das Frustum	266
	Orthogonale Projektion	273
	Perspektivische Projektion	275
	Aktivieren von Sicht und Projektion	278
	Koordinatenumrechnung 2D zu 3D und zurück	282
	Resümee: Sicht und Projektion	286
	Festlegen der Welttransformation	287
6.4	Vertex-Strukturen	288
6.5	Shader-Support	290
	Notwendige Vorbereitungen	291
	Vertex-Shader	295
	Pixel-Shader	301
6.6	Aktivierung von Renderstates	302
6.7	Effizientes Rendern von grafischen Primitiven	307
	Grundlagen zu Hardware und Performance	307
	Caching beim Rendern	310
	Statische vs. dynamische Vertex- und Index-Buffer	311
	Interface-Definition für einen Vertex-Cache-Manager	313
	Vertex-Cache-Objekt	315
	Vertex-Cache-Manager	328
6.8	Rendern von Text, Punkten und Linien	344
	Fonts anlegen und Text rendern	344

6.9	Punktlisten rendern	348
	Linienlisten rendern	350
6.10	Darstellung einer Szene	352
6.11	Demo-Applikation zur Anwendung der DLL	355
	Multiple 3D-Child-Windows mit multiplen Viewports	355
	Einfacher Geometrie-Loader	358
6.12	Ein Blick zurück, zwei Schritt nach vorn	359
Kapitel 7	3D-Pipeline und Shader	361
7.1	Grundlagen von Shadern	361
	3D-Pipeline	362
	CPU-lastig versus GPU-lastig	364
	Vertex-Manipulation über Vertex-Shader	366
	Pixel-Manipulation über Pixel-Shader	368
7.2	Shader-Techniken und Beispiele	369
	Demo 1: Basistransformationen	370
	Demo 2: Single-Pass-Multitexturing	378
	Demo 3: Directional Lighting per Pixel	380
	Demo 4: Per-Pixel-Omni-Lights	383
	Demo 5: Graustufenfilter	393
	Demo 6: Bump-Mapping	394
7.3	Ein Blick zurück, zwei Schritt nach vorn	410
Kapitel 8	Skeletale Animation von Charakteren	411
8.1	Eine Revolution?	411
	Der Siegeszug	413
8.2	Das Modellformat CBF	418
	Was ist ein Chunk?	419
	Einen Chunk auslesen (GetNextChunk)	421
	Unsere Hauptmethode	421
	Den Kopf einlesen (ReadHeader)	422
	Einlesen der Vertices (ReadVertices)	424
	Triangle-Information einlesen (ReadFaces)	427

	Das Netz (ReadMesh)	428
	Auf das Material kommt es an (ReadMaterial)	429
	Die Joints, bitte (ReadJoints)	431
	Der Hauptjoint (ReadJoint_Main)	434
	Die Rotation (ReadJoint_KeyFrame_Rot)	434
	Die Position (ReadJoint_KeyFrame_Pos)	436
	Sei animiert (ReadAnimations)	437
	Passt es? (SetScaling)	439
8.3	Verarbeitung der Daten im Speicher	442
	Vorbereitung der Daten (Prepare)	442
	Skeletale Animation (SetupBones)	447
	Bewegung im Modell (Animation)	452
	Vorbereitung ist alles (AnimationPrepare)	454
	Meine Position (AnimationVertices)	459
8.4	Updaten und Nutzen des Modells	462
	Aktueller Stand (Update)	462
	Präsentationstermin (Render)	463
	Bis auf die Knochen runter (RenderBones)	464
	Voll normal (RenderNormals)	466
8.5	Ein Blick zurück, zwei Schritt nach vorn	468
Teil 3	Hilfsmodule für die Engine	469
Kapitel 9	Eingabe-Interface der ZFXEngine	471
9.1	Kurz und schmerzlos	471
	Altbekanntes Interface-Design	472
	Interface-Definition für eine Eingabe-Klasse	473
9.2	Basisklasse für DirectInput-Devices	474
	Erstellen und Freigeben des Objekts	475
	Inbetriebnahme	476
	Abfrage des Inputs	478
9.3	Ran an die Tasten	481
	ZFXKeyboard-Klasse	482
	Initialisierung und Freigabe	482

Inhaltsverzeichnis

	Update	483
	Abfrage des Inputs	484
9.4	Die Rattenfänger von Redmond	485
	ZFXMouse-Klasse	485
	Initialisierung und Freigabe	486
	Update	487
9.5	Kein Spaß ohne Joystick	490
	ZFXJoystick-Klasse	490
	Initialisierung und Freigabe	491
	Update	493
9.6	Implementierung des Interfaces	494
	Initialisierung und Freigabe	495
	Update	497
	Abfrage der Daten	498
9.7	Demo-Applikation zur Anwendung der DLL	499
9.8	Ein Blick zurück, zwei Schritt nach vorn	502
Kapitel 10	Audio-Interface der ZFXEngine	505
10.1	Kurz und schmerzlos	505
	Altbekanntes Interface-Design	506
	Interface-Definition für eine Audio-Klasse	507
10.2	Implementierung des Interfaces	508
	ZFXAudio Klasse	509
	Initialisierung und Freigabe	512
	Das Laden und Abspielen von Sounds	515
	Zuhörer und Quelle für 3D-Sound	520
10.3	Demo-Applikation zur Anwendung der DLL	522
10.4	Ein Blick zurück, zwei Schritt nach vorn	522
Kapitel 11	Netzwerk-Interface der ZFXEngine	523
11.1	Netzwerk-Spiele	523
	Session-basiert	524

	Persistente Welten	526
	LAG	527
11.2	Netzwerk-Architektur	528
	Peer-to-Peer	528
	Client-Server	528
11.3	Netzwerk-Technik	530
	Das OSI-Modell	530
	Protokolle	532
	APIs	533
11.4	Implementierung der Netzwerk-Bibliothek	535
	Altbekanntes Interface-Design	535
	Server versus Clients	536
	Pakete schnüren	536
	Warteschlangen	538
	Socket-Objekte	542
	Interface-Definition für eine Netzwerk-Klasse	557
	Windows Sockets-Kapselung	559
11.5	Demo-Applikation zur Anwendung der DLL	573
	Chatten über das Netzwerk	574
	Dateien versenden	579
11.6	Ein Blick zurück, zwei Schritt nach vorn	588
Kapitel 12	Timing und Bewegung in der ZFXEngine	589
12.1	Hilfsbibliothek ZFXGeneral	589
12.2	Verschiedene Kamera-Modi	590
	Freie Kamera	590
	1st-Person-Kamera	590
	3rd-Person Kamera	591
	Fixe Kamera	592
12.3	Bewegung durch ZFXMovementController	592
	Wozu einen Movement-Controller?	592
	Implementierung der Basisklasse	593

Inhaltsverzeichnis

	Ableitung einer freien Kamera	596
	Ableitung einer 1st-Person-Kamera	598
12.4	Demo-Applikation zur Anwendung der Bibliothek	602
12.5	Ein Blick zurück, zwei Schritt nach vorn	602
Kapitel 13	Scene-Management	605
13.1	Sinn des Scene-Managements	605
13.2	Scene-Management-Techniken	607
	Keine Lösung ist auch eine Lösung	608
	Kontinuierliche und diskrete Detail-Level	609
	Quadtrees	611
	Octrees	621
	Binary Space Partitioning Trees	623
	Portal-Engines	634
	Potential Visibility Set	643
13.3	Implementierung eines BSP-Baums	645
	Klassendeklaration	647
	Erzeugen und Freigeben einer Instanz	650
	Rekursives Erstellen des Baums	650
	Auswahl des besten Splitters	654
	Durchlaufen des Baums	655
	Kollisionsabfragen	658
13.4	Implementierung eines Octrees	660
	Klassen-Deklaration	661
	Erzeugen und Freigeben einer Instanz	663
	Initialisieren eines Child-Nodes	664
	Rekursives Erstellen des Baums	665
	Polygonliste auf einen Node beschneiden	667
	Kollisionsabfragen im Octree	670
	Höhe des Spielers im Octree	672
	Durchlaufen des Baums	675

13.5	Demo-Applikation BSP-Tree und Octree	678
	Rendern von ZFXPolygon-Instanzen	679
	Laden der Level-Daten	681
	Berechnung eines Frames	684
	Sehenswertes in der Demo	687
13.6	Ein Blick zurück, zwei Schritt nach vorn	688
Teil 4	Schwarze Magie der Spieleprogrammierung	691
Kapitel 14	Computer Aided Design (CAD)	693
14.1	Anwendungen für CAD-Tools	693
	Ingenieur, Architekt, Spiele-Entwickler	693
	Level-Editing-Tools	694
14.2	Low-Polygon-Editor PanBox Edit	695
	Fähigkeiten des Tools	696
	WinAPI-Rahmenanwendung	698
14.3	Klassen-Design des Tools	698
	Genereller Aufbau eines Levels	699
	Die Grundlage allen Seins: CLevelObject	701
	Auf unterster Ebene: CPolygon	705
	Komplexe Modelle: CPolymesh	725
	Aufbruch in eine neue Welt: CPortal	744
	Es werde Licht: CLight	751
	Interaktive Objekte: CEntity	756
	Ich mach den Fisch: CSpawnPoint	761
	Lokales Management: CSector	762
	Alle zusammen: CLevel	789
	In der engeren Wahl: CSelectionBuffer	793
14.4	Ausgewählte Aspekte des GUI	805
	Klassendeklaration	806
	Wichtige Attribute	809
	Update-Methode	811
	Ein Polygon erstellen	812

14.5 Ein Blick zurück, zwei Schritt nach vorn 814

Kapitel 15 Pandoras Legacy 815

15.1 Der Deathmatch-Ego-Shooter Pandoras Legacy 815

Einfaches Game-Design 816

Altbekannte Klassen neu aufgelegt 817

15.2 Im Schatten unser selbst 818

Verschiedene Verfahren zum Schattenwurf 819

Zur Theorie der Shadow-Volumes 820

Implementierung von Shadow-Volumes 827

Erstellen des Shadow-Volumes 831

15.3 Einen Level laden 840

Benötigte Hilfsdatenstrukturen 841

Lade-Methode in CGameLevel 842

Lade-Methode in CGameSector 843

Verbindungen zwischen den Portalen herstellen 853

15.4 Einen Level rendern 855

Rendern der Geometrie 856

Die Schatten im Level rendern 863

15.5 Integration von Characters 865

CGameCharacter, die Basisklasse für Spieler und NPCs 866

Netzwerknachrichten von und für Characters 868

15.6 CGame, die Klasse für das Spiel 870

Update des Spiels 870

Aufgaben für das Netzwerk 872

15.7 Ein Blick zurück, zwei Schritt nach vorn 874

Kapitel 16 Scripting und Definition von NPC-Verhalten 877

16.1 Kontrolle von NPCs 877

16.2 ZBL/0 ZFX Bot Language – die Scriptsprache für ZFX-Bots 878

16.3 ZBL-API – Integrieren der ZBL/0-Virtual Machine in eigene Projekte 879

Mehr über das ZBL-API-Interface 879

	Mehr über die Virtual Machine	883
16.4	ZBL/0 Toolkit – die ZBL-Entwicklungsumgebung	892
16.5	ZBL/0-Bot-Design – Entwickeln von Bots mit der ZFX-Bot-Language	894
	Die Scriptsprache ZBL/0 (Syntax)	894
	EZBL/0-Standardbefehle und Anweisungen	898
	Die Entwicklung eines Game-Bots mit ZBL/0	901
16.6	Ein Blick zurück, zwei Schritt nach vorn	910
Teil 5	Anhang	913
Anhang A	Die Funktionen von ZBL/0	915
A.1	House-Keeping Funktionen zur generellen Botsteuerung	915
A.2	Modifikatoren zur Problemspezifikation	917
A.3	Game-Bot Kontrollfunktionen	918
A.4	Game-Bot Sensor-Funktionen	921
A.5	Andere Funktionen in ZBL/0	923
Anhang B	CD-ROM und begleitendes Material zu diesem Buch	925
Anhang C	Internetseiten rund um die Spieleentwicklung	927
Anhang D	Epilog	929
	Stichwortverzeichnis	931