

CARL HANSER VERLAG

David Scherfgen

**3D-Spieleprogrammierung**  
Modernes Game Design mit DirectX 9 und C++

3-446-22152-2

[www.hanser.de](http://www.hanser.de)



# Inhalt

Einleitung .....	1
<b>1</b> Einleitung .....	<b>2</b>
1.1 Ein paar Dinge im Voraus .....	2
1.1.1 Was Sie erwartet .....	2
1.1.2 Was Sie nicht erwartet .....	2
1.1.3 Voraussetzungen .....	3
1.1.4 Die Schriftformate in diesem Buch .....	3
1.1.5 Zu den Versionen von Visual C++ .....	3
1.1.6 Die Ungarische Notation .....	3
1.2 Einführung in die Spieleprogrammierung .....	4
1.2.1 Der kleine Unterschied .....	4
1.2.2 Was macht ein Spiel eigentlich? .....	5
1.2.3 Eingliederung in die Windows-Architektur .....	7
1.2.4 Das Problem mit der Zeit .....	8
1.2.5 Die verschiedenen Seiten eines Spiels .....	12
1.2.6 Rückblick .....	13
1.3 DirectX und C++ .....	14
1.3.1 Was ist DirectX? .....	14
1.3.2 Die perfekte Kombination mit C++ .....	16
1.3.3 Das COM – Grundlage von DirectX .....	17
1.3.4 Rückblick .....	20
1.4 Wir bauen uns eine eigene Engine! .....	20
1.4.1 Was versteht man unter einer Engine? .....	20
1.4.2 Verschiedene Entwicklungsansätze .....	21
1.4.3 Konkrete Planung .....	22
1.4.4 Installation der Engine und Einrichten eines Projekts .....	24
1.4.5 Vorgabefunktionen, Klassen und Makros vorgestellt .....	27
1.4.6 Rückblick .....	37
1.5 Tipps zum Debuggen .....	37
1.6 Ausblick .....	41
<b>3D-Grafik .....</b>	<b>43</b>
<b>2</b> 3D-Grafik .....	<b>44</b>
2.1 Was Sie in diesem Kapitel erwartet .....	44
2.2 Einführung in die 3D-Grafik .....	44

2.2.1	Ein neues Koordinatensystem .....	44
2.2.2	Theorie der 3D-Grafik .....	46
2.2.3	Vektoren .....	48
2.2.4	Matrizen .....	59
2.2.5	Ebenen .....	74
2.2.6	Das RGB-Farbsystem .....	78
2.2.7	Rückblick .....	81
2.3	Direct3D-Grundlagen .....	81
2.3.1	Was ist Direct3D? .....	82
2.3.2	Die Transformationspipeline .....	82
2.3.3	Der Rasterizer .....	84
2.3.4	Die wichtigsten Schnittstellen .....	84
2.3.5	Ressourcen .....	85
2.3.6	Oberflächen .....	85
2.3.7	Direct3D im C++-Programm ansprechen .....	90
2.3.8	Rückblick .....	90
2.4	Initialisierung von Direct3D .....	91
2.4.1	Erstellen der <i>IDirect3D9</i> -Schnittstelle .....	91
2.4.2	Adapterinformationen .....	91
2.4.3	Caps – die Fähigkeiten eines Geräts .....	96
2.4.4	Erstellen des Fensters .....	97
2.4.5	Erstellen der <i>IDirect3DDevice9</i> -Schnittstelle .....	99
2.4.6	Direct3D herunterfahren .....	111
2.4.7	Beispielprogramm: eine komplette Direct3D-Anwendung .....	111
2.4.8	Rückblick .....	113
2.5	Das erste Dreieck .....	114
2.5.1	Vertizes .....	114
2.5.2	Erste Render-States .....	116
2.5.3	Setup der Transformationspipeline .....	120
2.5.4	Der Zeichenvorgang .....	122
2.5.5	Rückblick .....	128
2.6	Texturen .....	129
2.6.1	Was Texturen sind .....	129
2.6.2	Grundlegende Dinge .....	132
2.6.3	Der bilineare Filter .....	134
2.6.4	MIP-Mapping – schnell und schön .....	136
2.6.5	Weitere Spielereien .....	138
2.6.6	Texturen mit D3DX laden .....	139
2.6.7	Texturinformationen abfragen .....	142
2.6.8	Das Beispielprogramm .....	143
2.6.9	Kachelfähige Texturen erzeugen .....	147
2.6.10	Rückblick .....	149
2.7	Vertex- und Index-Buffer .....	150
2.7.1	Zweck von Vertex- und Index-Buffern .....	150
2.7.2	Der Vertex-Buffer im Detail .....	151
2.7.3	Der Index-Buffer im Detail .....	155
2.7.4	Das Beispielprogramm .....	158

2.7.5	Rückblick.....	163
2.8	Nebel.....	164
2.8.1	Die Theorie.....	164
2.8.2	Nebel mit Direct3D.....	167
2.8.3	Das Beispielprogramm.....	168
2.8.4	Rückblick.....	169
2.9	Beleuchtung.....	170
2.9.1	Ein einfaches Beleuchtungssystem.....	170
2.9.2	Die Praxis – Beleuchtung mit Direct3D.....	177
2.9.3	Das Beispielprogramm.....	185
2.9.4	Rückblick.....	187
2.10	Alpha-Blending.....	188
2.10.1	Die Theorie.....	188
2.10.2	Alpha-Blending mit Direct3D.....	190
2.10.3	Das Beispielprogramm.....	193
2.10.4	Rückblick.....	194
2.11	Multi-Texturing.....	195
2.11.1	Der theoretische Teil.....	195
2.11.2	Multi-Texturing anwenden.....	200
2.11.3	Mehr über Texturkoordinaten.....	203
2.11.4	Das Beispielprogramm.....	205
2.11.5	Rückblick.....	207
2.12	Exotische Texturformen.....	208
2.12.1	Volumentexturen.....	208
2.12.2	Umgebungstexturen.....	213
2.12.3	Rückblick.....	223
2.13	Der Stencil-Buffer.....	223
2.13.1	Was war das noch gleich?.....	223
2.13.2	Die Details.....	224
2.13.3	Das Beispielprogramm.....	228
2.13.4	Rückblick.....	230
2.14	D3DX-Effekte.....	231
2.14.1	Probleme beim Verwalten von Modellen.....	231
2.14.2	„Effekte“.....	231
2.14.3	Laden eines Effekts.....	235
2.14.4	Mit Effekten rendern.....	235
2.14.5	Variablen von außen setzen und abfragen.....	237
2.14.6	Das Beispielprogramm.....	240
2.14.7	<i>EffectEdit</i> .....	242
2.14.8	Rückblick.....	242
2.15	Transformierte Vertizes für 2D-Grafik.....	242
2.15.1	Wozu denn noch 2D?.....	242
2.15.2	Die Transformation umgehen.....	243
2.15.3	Ein anderes Vertexformat.....	244
2.15.4	DirectDraw imitieren.....	245
2.15.5	Eine andere Methode für 2D-Grafik.....	247
2.15.6	Rückblick.....	247

2.16	Ausblick .....	247
	3D-Grafik mit der TriBase-Engine .....	249
3	3D-Grafik mit der TriBase-Engine .....	250
3.1	Was Sie in diesem Kapitel erwartet .....	250
3.2	Direct3D mit der TriBase-Engine .....	250
3.2.1	Was uns das Leben leichter machen kann .....	250
3.2.2	Die Klasse <i>tbDirect3D</i> .....	252
3.2.3	Der Texturmanager – <i>tbTextureManager</i> .....	260
3.2.4	<i>tbVertexBuffer</i> und <i>tbIndexBuffer</i> .....	274
3.2.5	Die Effektklasse <i>tbEffect</i> .....	286
3.2.6	Ein allumfassendes Beispielprogramm .....	290
3.2.7	Rückblick .....	299
3.2.8	Ausblick .....	300
3.3	Modelldateien .....	300
3.3.1	Die Vorarbeit .....	300
3.3.2	Der Konverter .....	306
3.3.3	Eine Modellklasse .....	308
3.3.4	Das Beispielprogramm .....	320
3.3.5	Rückblick .....	323
3.4	Texte zeichnen .....	324
3.4.1	Speicherung der Zeichen .....	324
3.4.2	Das Format der Textur .....	325
3.4.3	Transformierte Vertices für Texte .....	325
3.4.4	Der Weg von TrueType zur Bitmap-Font .....	326
3.4.5	Inhalt der TBF-Dateien .....	327
3.4.6	Programmierung einer Schriftartklasse .....	328
3.4.7	Das Beispielprogramm .....	338
3.4.8	Rückblick .....	340
3.5	Ausblick .....	340
	Eingabe .....	341
4	Eingabe .....	342
4.1	Was uns in diesem Kapitel erwartet .....	342
4.2	DirectInput kurz vorgestellt .....	342
4.2.1	Was kann DirectInput besser als Windows? .....	342
4.2.2	Geräte und Geräteklassen .....	343
4.2.3	GUIDs .....	343
4.2.4	Achsen und Knöpfe .....	343
4.2.5	Die Funktionsweise von DirectInput .....	344
4.2.6	Ein paar Worte zum Debuggen .....	345
4.3	Der Startschuss fällt .....	345
4.3.1	Erstellen des <i>IDirectInput8</i> -Objekts .....	346
4.3.2	Eingabegeräte abzählen .....	347
4.3.3	Rückblick .....	349
4.4	Initialisierung eines Geräts und Datenabfrage .....	349

4.4.1	Keine Angst vor <i>CreateDevice</i> !	349
4.4.2	Vorbereitungen treffen	350
4.4.3	Auf verlorene Eingabe achten!	351
4.4.4	Hinterlassen Sie Ihren Platz	352
4.4.5	Einmal Daten, bitte!	352
4.4.6	Rückblick	352
4.5	Die Tastatur	353
4.5.1	Das Datenformat der Tastatur	353
4.5.2	Tastencodes	353
4.5.3	Das Beispielprogramm	355
4.5.4	Begrenzungen der Tastatur	357
4.5.5	Rückblick	357
4.6	Die Maus	357
4.6.1	Das Datenformat der Maus	358
4.6.2	Relative Achsen	358
4.6.3	Die Mausknöpfe	359
4.6.4	Der exklusive Modus	359
4.6.5	Das Beispielprogramm	359
4.6.6	Rückblick	360
4.7	Der Joystick	361
4.7.1	Achsen, Knöpfe, POVs und Schieberegler	361
4.7.2	Das Joystick-Datenformat	362
4.7.3	Das Beispielprogramm	363
4.7.4	Rückblick	364
4.8	Objekte abzählen und kalibrieren	365
4.8.1	Objekte abzählen	365
4.8.2	Eigenschaften festlegen	366
4.8.3	Achsenmodus	367
4.8.4	Achsenkalibrierung	367
4.8.5	Die tote Zone	368
4.8.6	Sättigung	369
4.8.7	Das Beispielprogramm	369
4.8.8	Gepufferte Daten und direkte Daten	369
4.8.9	Rückblick	370
4.9	Eine Eingabeklasse für die Engine	370
4.9.1	Probleme mit <i>DirectInput</i>	370
4.9.2	Das Prinzip der analogen Knöpfe	370
4.9.3	Die <i>tbDirectInput</i> -Klasse	373
4.9.4	Das Beispielprogramm	391
4.9.5	Rückblick	393
4.10	Ausblick	393
	Sound und Musik	395
5	Sound und Musik	396
5.1	DirectSound kurz vorgestellt	396
5.1.1	Was kann DirectSound besser als Windows?	396

5.1.2	Soundpuffer und Mixer.....	397
5.1.3	Die Schnittstellen.....	398
5.2	Initialisierung von DirectSound.....	398
5.2.1	Formale Dinge.....	398
5.2.2	Abzählen der DirectSound-Geräte.....	398
5.2.3	Erstellung der <i>IDirectSound8</i> -Schnittstelle.....	399
5.2.4	Die Kooperationsebene wählen.....	400
5.2.5	Rückblick.....	400
5.3	Erstellen von Soundpuffern.....	400
5.3.1	Eigenschaften der Soundpuffer.....	401
5.3.2	Das Format eines Soundpuffers.....	403
5.3.3	Anfordern der <i>IDirectSoundBuffer8</i> -Schnittstelle.....	404
5.3.4	Der primäre Soundpuffer.....	405
5.3.5	Rückblick.....	406
5.4	Füllen eines sekundären Soundpuffers.....	406
5.4.1	Eine kleine Einführung in die Akustik.....	406
5.4.2	Wir sperren den Soundpuffer.....	411
5.4.3	Entsperren.....	413
5.4.4	Hinein mit den Daten!.....	413
5.4.5	Rückblick.....	415
5.5	Kontrolle eines Sounds.....	416
5.5.1	Die <i>Play</i> -Methode.....	416
5.5.2	Festlegen der Lautstärke.....	418
5.5.3	Festlegen der Balance.....	418
5.5.4	Festlegen der Abspielfrequenz.....	419
5.5.5	Das Beispielprogramm.....	419
5.5.6	Rückblick.....	420
5.6	WAV-Dateien laden.....	420
5.6.1	Der RIFF-Header.....	421
5.6.2	Die WAV-Chunks.....	421
5.6.3	Die Funktion <i>LoadWAVFile</i> .....	422
5.7	3D-Sound.....	424
5.7.1	Theorie des 3D-Sounds.....	425
5.7.2	Die <i>IDirectSound3DBuffer8</i> -Schnittstelle.....	425
5.7.3	Die <i>IDirectSound3DListener8</i> -Schnittstelle.....	427
5.7.4	Das Beispielprogramm.....	429
5.7.5	Rückblick.....	430
5.8	Echtzeiteffekte.....	430
5.8.1	Effekte – vorberechnet und in Echtzeit.....	430
5.8.2	Verschiedene Effektschnittstellen.....	431
5.8.3	Vorwarnung erforderlich!.....	432
5.8.4	Effekte mit <i>SetFX</i> anfordern.....	432
5.8.5	Die Effektschnittstelle abfragen.....	433
5.8.6	Effektparameter am Beispiel des Echos.....	434
5.8.7	Experimentieren ist angesagt!.....	435
5.8.8	Rückblick.....	436
5.9	Ergänzende Informationen.....	436

5.9.1	Die verschiedenen Schnittstellen .....	436
5.9.2	Klonen von Sounds.....	437
5.9.3	Status eines Soundpuffers.....	438
5.10	Die Klasse <i>tbDirectSound</i> .....	439
5.10.1	Erweiterung des Konfigurationsdialogs.....	439
5.10.2	Was <i>tbDirectSound</i> können soll.....	440
5.10.3	Die Klassendefinition .....	440
5.10.4	Die Initialisierungsmethode <i>Init</i> .....	441
5.10.5	Der Umgang mit dem Hörer .....	443
5.11	Die <i>tbSound</i> -Klasse .....	444
5.11.1	Fähigkeiten der Klasse.....	445
5.11.2	Das Prinzip der Soundpufferliste .....	445
5.11.3	Die Klassendefinition .....	445
5.11.4	Laden des Sounds .....	447
5.11.5	Freier Soundpuffer gesucht! .....	449
5.11.6	Die <i>SetPosition</i> -Methode.....	450
5.11.7	Einen Sound abspielen.....	451
5.11.8	Die restlichen Methoden .....	451
5.12	Musik ins Spiel bringen.....	452
5.12.1	Was unterscheidet Musik von Soundeffekten? .....	452
5.12.2	DirectShow-Grundlagen .....	452
5.12.3	Kontrolle über den Filtergraphen.....	455
5.12.4	Die Klasse <i>tbMusic</i> .....	456
5.12.5	Das Beispielprogramm .....	459
5.13	Ausblick .....	460
	Theorie der Spieleprogrammierung .....	461
6	Theorie der Spieleprogrammierung .....	462
6.1	Was Sie in diesem Kapitel erwartet .....	462
6.2	Warum Planung wichtig ist .....	462
6.3	Am Anfang steht die Idee.....	462
6.3.1	Inspiration.....	463
6.3.2	Auf Ideen vorbereitet sein.....	464
6.3.3	Aussortieren.....	464
6.3.4	Storydesign .....	465
6.3.5	Entwicklung eines Ablaufschemas .....	466
6.4	Suche nach Teammitgliedern .....	466
6.5	Vermitteln des Spiels und gemeinsame Analyse .....	467
6.5.1	Die Absichten klarmachen .....	467
6.5.2	Machbarkeitsanalyse.....	467
6.5.3	Tipps.....	468
6.6	Ausarbeitung der Details .....	469
6.7	Einteilung in Module.....	469
6.8	Level-Design und Atmosphäre.....	469
6.8.1	Abenteuer-, Action- und Rollenspiele.....	470
6.8.2	Puzzlespiele .....	470



6.8.3	Simulatoren.....	470
6.8.4	Wann eine Aufgabe zu schwer ist.....	471
6.8.5	Tipps für das Level-Design.....	472
6.8.6	Allgemeine Tipps für eine bessere Spielatmosphäre .....	473
6.9	Tipps zum Programmieren .....	475
6.9.1	Planung und Standard .....	475
6.9.2	Implementierung neuer Features.....	476
6.9.3	Die Liebe zum Detail.....	477
6.10	Testen Ihres Spiels.....	478
6.10.1	Testen während des Entwicklungsprozesses.....	478
6.10.2	Testen des fertigen Spiels .....	478
6.11	Ausblick .....	479
	<b>Das erste Spiel .....</b>	<b>481</b>
7	<b>Das erste Spiel.....</b>	<b>482</b>
7.1	Was Sie in diesem Kapitel erwartet.....	482
7.2	Planung.....	482
7.2.1	Das Spielprinzip und der Name des Spiels .....	482
7.2.2	Die Darstellung.....	483
7.2.3	Die Spielzustände .....	483
7.2.4	Das Spielgerüst .....	484
7.3	Die Grundklasse <i>CBreakanoid</i> .....	485
7.3.1	Variablen .....	485
7.3.2	Methoden .....	486
7.3.3	Die <i>WinMain</i> -Funktion für Breakanoid .....	491
7.4	Das Titelbild.....	492
7.4.1	Planung des Titelbilds.....	492
7.4.2	Die Schriftarten.....	493
7.4.3	Initialisieren, Laden und Entladen des Titelbilds .....	493
7.4.4	Rendern des Titelbilds .....	494
7.4.5	Bewegung des Titelbilds.....	496
7.5	Das Hauptmenü .....	497
7.5.1	Planung des Hauptmenüs.....	497
7.5.2	Laden, Entladen, Betreten und Verlassen .....	498
7.5.3	Rendern.....	498
7.5.4	Bewegen des Hauptmenüs .....	500
7.5.5	Sound für das Hauptmenü! .....	501
7.6	Das Spiel .....	503
7.6.1	Planung des Spiels .....	503
7.6.2	Schritt 1: die <i>CGame</i> -Klasse.....	504
7.6.3	Schritt 2: Anzeigen des Levelmodells.....	506
7.6.4	Schritt 3: Her mit dem Schläger! .....	508
7.6.5	Schritt 4: Ein Levelsystem .....	512
7.6.6	Schritt 5: Bälle hinzufügen .....	512
7.6.7	Schritt 6: Die Blöcke.....	519
7.6.8	Schritt 7: Versuche .....	525

7.6.9	Schritt 8: Punkte .....	528
7.6.10	Schritt 9: Sound für das Spiel .....	529
7.6.11	Schritt 10: Hier spielt die Musik! .....	530
7.7	Minimieren im Vollbildmodus .....	530
7.7.1	Das Problem .....	530
7.7.2	Die Lösung .....	531
7.8	Erweiterungsvorschläge .....	533
7.9	Ausblick .....	533
	<b>Das zweite Spiel .....</b>	<b>535</b>
8	<b>Das zweite Spiel .....</b>	<b>536</b>
8.1	Was Sie in diesem Kapitel erwartet .....	536
8.2	Planung .....	536
8.2.1	Das Spielprinzip und der Name des Spiels .....	536
8.2.2	Die Spielzustände .....	537
8.2.3	Die Schiffe .....	537
8.2.4	Die Waffensysteme .....	538
8.2.5	Speicherung der Informationen .....	539
8.2.6	Die Schiffssysteme .....	540
8.3	Schiffs- und Waffentypen .....	541
8.3.1	Die Struktur <i>SShipType</i> .....	541
8.3.2	Die Struktur <i>SWeaponType</i> .....	542
8.3.3	Laden aus der INI-Datei .....	543
8.4	Die Klasse <i>tObject</i> .....	545
8.4.1	Unser bisheriger Ansatz .....	545
8.4.2	Das neue Prinzip .....	545
8.4.3	Position und Skalierung .....	546
8.4.4	Ein Fall für die Matrix! .....	546
8.4.5	Relativ zu absolut – und zurück .....	547
8.4.6	Die Physik kommt hinzu .....	548
8.4.7	Implementierung von <i>tObject</i> .....	548
8.5	Der Umgang mit Schiffen .....	555
8.5.1	Die <i>CShip</i> -Klasse .....	555
8.5.2	Integrierung in <i>CGame</i> .....	556
8.5.3	Bewegen der Schiffe .....	558
8.5.4	Kontrolle eines Schiffs .....	562
8.5.5	Rendern der Schiffe .....	564
8.5.6	Aufschalten von Zielen .....	566
8.6	Alle Waffen abfeuern! .....	568
8.6.1	Die <i>CProjectile</i> -Klasse .....	568
8.6.2	Feuern .....	569
8.6.3	Bewegen .....	571
8.6.4	Rendern .....	573
8.7	Sprites .....	573
8.7.1	Was sind Sprites? .....	573
8.7.2	Missbrauch der Kameraachsen .....	574

8.7.3	Die TriBase-Sprite-Engine.....	574
8.7.4	Zurück zum Spiel: Rendern der Laser .....	581
8.8	Kollisionserkennung.....	584
8.8.1	Rückblick: Umgebungskugel und Umgebungsquader .....	584
8.8.2	Das Prinzip der Kollisionserkennung .....	584
8.8.3	Kugel – Kugel.....	585
8.8.4	Linie – Kugel.....	586
8.8.5	Linie – Dreieck .....	591
8.8.6	Dreieck – Dreieck .....	596
8.8.7	Linie – Quader .....	602
8.8.8	Quader – Quader.....	607
8.8.9	Wie wir mit Modellen umgehen .....	609
8.8.10	Vorberechnungen.....	611
8.8.11	Linien und Modelle.....	621
8.8.12	Kollision zwischen zwei Modellen .....	624
8.8.13	Hardcore-Kollisionserkennung .....	629
8.8.14	Volltreffer!.....	629
8.8.15	Zusammenstoß zweier Schiffe.....	632
8.9	Auto-Aiming .....	635
8.9.1	Definition.....	635
8.9.2	Der mathematische Hintergrund .....	636
8.9.3	Die neue <i>CShip::Fire</i> -Methode .....	637
8.10	„Künstliche Intelligenz“ .....	638
8.10.1	Das Verhalten eines Schiffs.....	638
8.10.2	Schritt 1: Kurs auf das Ziel nehmen.....	639
8.10.3	Schritt 2: Feuern .....	641
8.10.4	Schritt 3: Ausweichmanöver bei Treffer .....	642
8.10.5	Schritt 4: Ausweichmanöver bei drohender Kollision .....	643
8.10.6	Schritt 5: Wechseln des Ziels.....	643
8.11	Partikel .....	644
8.11.1	Was Partikel sind.....	644
8.11.2	Das Partikelsystem der TriBase-Engine.....	644
8.11.3	Antriebs- und Raketenflammen .....	646
8.11.4	Explosionen .....	647
8.11.5	Aufleuchten des Schutzschilds .....	651
8.11.6	Trümmer .....	652
8.12	Weitere optische Verfeinerungen .....	653
8.12.1	Sky-Box, Licht und Nebel .....	653
8.12.2	Ein „Sternenfeld“ .....	657
8.12.3	Glühen von Projektilen .....	658
8.13	Die Kamera .....	659
8.14	Das Cockpit.....	661
8.14.1	Das Cockpitmodell .....	662
8.14.2	Die Anzeigen .....	663
8.14.3	Das HUD .....	666
8.14.4	Radar.....	667
8.15	Der Sound.....	673

8.15.1	Schüsse .....	673
8.15.2	Antriebsounds .....	675
8.15.3	Der Hörer .....	676
8.16	Die Benutzeroberfläche .....	676
8.16.1	Die TriBase-Benutzeroberfläche.....	677
8.16.2	Erstellung des Hauptmenüs.....	687
8.17	Optimierungen und der Feinschliff.....	693
8.17.1	Sichtbarkeit eines Objekts.....	693
8.17.2	Render-Modell und Kollisionsmodell.....	697
8.17.3	Musik.....	698
8.17.4	Wackelndes Cockpit .....	698
8.17.5	Screenshots schießen .....	698
8.18	Erweiterungsvorschläge .....	699
8.19	Ausblick .....	700
	Fortgeschrittene Techniken.....	701
9	Fortgeschrittene Techniken .....	702
9.1	Was Sie in diesem Kapitel erwartet.....	702
9.2	Schatten mit dem Stencil-Buffer .....	702
9.2.1	Schatten in der 3D-Grafik.....	702
9.2.2	Ansätze .....	702
9.2.3	Das Prinzip .....	703
9.2.4	Die Klasse <i>tbShadowVolume</i> .....	708
9.2.5	Das Beispielprogramm .....	717
9.3	Videos abspielen.....	717
9.3.1	Zielsetzung.....	717
9.3.2	Schreiben eines eigenen Filters.....	718
9.3.3	Verwenden des Filters .....	728
9.3.4	Das richtige Timing .....	730
9.4	Stereo-3D-Grafik.....	731
9.4.1	Das räumliche Sehen .....	731
9.4.2	Trennen der Bilder.....	733
9.4.3	Implementierung.....	735
9.4.4	Beispielprogramme .....	735
9.5	Raumaufteilung .....	735
9.5.1	Rekursives Rendern .....	736
9.5.2	PVS und Portale.....	737
9.5.3	Light-Mapping.....	738
9.5.4	Die TriBase-Klasse <i>tbOctree</i> .....	739
9.6	Terrain-Rendering .....	740
9.6.1	Repräsentierung eines Terrains.....	740
9.6.2	Unterteilung des Terrains.....	740
9.6.3	Erzeugen der Dreiecke.....	741
9.6.4	Terrain-Rückruffunktion.....	743
9.6.5	Geo-MIP-Mapping .....	743
9.6.6	Texturierung .....	743

9.6.7	Beleuchtung .....	745
9.6.8	Das TriBase-Tool <i>TerrainEditor</i> .....	745
9.7	Was sind Vertex- und Pixel-Shader? .....	747
9.7.1	Was ein Shader ist .....	747
9.7.2	Einsatzgebiete .....	747
9.7.3	Die fixe und die programmierbare Rendering-Pipeline .....	748
9.8	PlugIns schreiben und laden .....	748
9.8.1	DLL-Dateien explizit laden .....	748
9.8.2	Adresse einer Funktion abfragen .....	749
9.8.3	DLL-Dateien erzeugen .....	750
9.8.4	Die Kommunikation zwischen Anwendung und PlugIn .....	751
9.8.5	Das Beispielprogramm .....	752
9.9	Ausblick .....	752
	Internetseiten und CD-ROM .....	753
10	Internetseiten und CD-ROM .....	754
10.1	Interessante Internetseiten .....	754
10.2	Die Programme auf der Begleit-CD-ROM .....	755
10.2.1	AC3D v3.6 Demo .....	755
10.2.2	gmax v1.2 .....	756
10.2.3	MilkShape 3D v1.65 Demo .....	757
10.2.4	Paint Shop Pro 7AE Demo .....	758
10.2.5	POV-Ray v3.5 .....	759
10.2.6	Sound Forge 6 Demo .....	760
10.2.7	Terragen .....	761
10.2.8	Texture Maker v2.5 Demo .....	762
10.2.9	Visual Assist 6/.NET Demo .....	763
10.2.10	3Dografe .....	763
10.3	Das Ende .....	764
	Index .....	765