

Inhaltsverzeichnis

1. Einführung in OpenGL und LWJGL	1
1.1. Einführung in OpenGL	2
1.1.1. Leistungsfähigkeit des Rechners	3
1.1.2. OpenGL als Zustandsautomat	3
1.1.3. Objektrepräsentationen	4
1.1.4. Dreiecke unter der Lupe	5
1.1.5. Vereinfachte Renderpipeline	7
1.2. Einführung in LWJGL	8
1.2.1. Vorbereitung von Eclipse und LWJGL	8
1.2.2. Die Klasse LWJGLBasisFenster	9
1.3. Beispielsammlung zu OpenGL und LWJGL	11
1.3.1. Bewegtes Dreieck	11
1.3.2. AWT/Swing und LWJGL zusammenbringen	13
1.3.3. Abfolgen von Transformationsketten	15
1.3.4. Drehender Würfel	15
1.3.5. Würfelprojektion auf eine Fläche	17
1.3.6. Wehende Fläche	18
1.4. Objekte erstellen, speichern und laden	19
1.4.1. Das Wavefront OBJ-Format	20
1.4.2. Objekte laden und visualisieren	22
1.5. Beschleunigung und Langlebigkeit durch Buffer-Objects	22
1.5.1. Bewegtes Dreieck mit Buffer-Objects	23
1.5.2. Drehender Würfel in Farbe mit Buffer-Objects	25
1.6. Eine kleine Spielumgebung schaffen	29
1.6.1. Der perfekte Flummi in Minimallösung	29
1.6.2. Ebenen von statischen und dynamischen Objekten	30
1.6.3. Viele Objekte verwalten	33
1.6.4. Strategisch mit Verhalten umgehen	35
1.7. Kollektive Intelligenz und Schwarmverhalten	38
1.7.1. Steuerungsverhalten statt individueller Wegeplanung	38
1.7.2. Beschleunigung ins Spiel bringen	39
1.7.3. Kräfte aus der Physik übernehmen	43
1.7.4. Steuerungskräfte Ziel folgen, Ziel erreichen oder fliehen	45
1.7.5. Die drei Kräfte der Flocking-Boids	46
1.8. Zusammenfassung und Ausblick	50
1.9. Übungsaufgaben	50

2. Eigene Shader in GLSL entwickeln	53
2.1. Konvolution und Blur-Effekt in Java	54
2.1.1. Glättung durch Konvolution	54
2.1.2. Dämpfung durch Maus-Interaktion	55
2.1.3. Pseudocode für gedämpften Mittelwertfilter	56
2.1.4. Programmcode der ersten Projektstufe	56
2.2. Einführung in Vertex- und Fragment-Shader	57
2.2.1. Vektoren und Matrizen in GLSL	58
2.2.2. Modularisierung des Programmcodes	59
2.2.3. Shadercode mit LWJGL	60
2.2.4. Gelbes Quad rotieren lassen	61
2.2.5. Farbe aus Tiefeninformationen erzeugen	64
2.3. Blur-Effekt mit Shader	65
2.3.1. Struktur für Blur-Effekt-Shader	65
2.3.2. Gaußverteilung statt Viereck	67
2.3.3. Shadercode für Visualisierung	68
2.4. Realisierung einer einfachen Wasserdynamik	69
2.4.1. Vereinfachte Schwingungsdynamik	69
2.4.2. Schwingungsdynamik in Java	70
2.4.3. Schwingungsdynamik mit einem Fragment-Shader	71
2.5. Refraktion und Schatten	73
2.5.1. Refraktionsvektor bestimmen	73
2.5.2. Shaderstruktur für Refraktion	76
2.5.3. Einfacher Schatteneffekt durch Gradienten	77
2.6. Reflexion und Kaustik	79
2.6.1. Umsetzungsidee zum Reflexionseffekt	79
2.6.2. Reflexionsvektor berechnen	79
2.6.3. Shadercode zur Reflexion	80
2.6.4. Umsetzungsidee zum Kaustikeffekt	81
2.6.5. Shadercode zur Kaustik	82
2.7. Kombination zu experimenteller Wassersimulation	83
2.7.1. Kontinuierliches Wolkenbild	83
2.7.2. Shadercode für Effektkombination	83
2.8. Wasserdynamik im Vertex-Shader	84
2.8.1. Verknüpfung von Geometrie und Textur	84
2.8.2. Wellendynamik und Geometrie verbinden	85
2.8.3. Visualisierung der Geometrie und Shaderkomposition	86
2.9. Zusammenfassung und Ausblick	86
2.10. Übungsaufgaben	87
3. Grundlagen des Beleuchtungsdesigns	89
3.1. Wahrnehmung von sichtbarem Licht	90
3.1.1. Entstehung eines Bildes im Auge	90
3.1.2. Entstehung eines Bildes in der Kamera	92
3.1.3. Bedeutung der Farben	92

3.1.4.	Farbschwächen und -blindheit	93
3.2.	Motivierter Light Einsatz	95
3.3.	Licht transportiert Kontext	95
3.4.	Lichtdesigner dürfen tricksen	95
3.5.	Ziele des Beleuchtungsdesigns	96
3.5.1.	Dinge glaubhaft machen	96
3.5.2.	Dinge lesbar oder erkennbar machen	97
3.5.3.	Szenen durch Schatten und weitere Effekte bereichern	97
3.5.4.	Erhaltung der Kontinuität	98
3.5.5.	Fokus des Betrachters lenken	101
3.5.6.	Materialeigenschaften verstärken	102
3.5.7.	Emotionalität und Kontext transportieren	102
3.6.	Beleuchtungskonzept Three-Point-Lighting	102
3.6.1.	Einfluss des Key-Lights	103
3.6.2.	Einfluss des Fill-Lights	103
3.6.3.	Aufgabe des Back-Lights	103
3.6.4.	Kombination von Key-, Fill- und Back-Light	105
3.7.	Beleuchtungskonzept Four-Point-Lighting	105
3.8.	Komposition einer Szene	106
3.8.1.	Der goldene Schnitt	106
3.8.2.	Körperhaltung berücksichtigen	107
3.8.3.	Balance zwischen Vorder- und Hintergrund	107
3.8.4.	Gezielte Überdeckungen	109
3.9.	Zusammenfassung und Ausblick	109
3.10.	Übungsaufgaben	110
4.	Vektoren als Positionen, Richtungen und Farben	111
4.1.	Vektoren als Richtungen	112
4.1.1.	Implementierung von Vektor3D	114
4.1.2.	Vektorrichtung zum oder vom Licht	116
4.1.3.	Skalare und Vektoren	118
4.1.4.	Vektorlänge, Normierung und inneres Produkt	118
4.1.5.	Die Kosinusformel	121
4.1.6.	Das Kreuzprodukt	126
4.2.	Vektoren als Positionen	127
4.2.1.	Hilfreiche Funktionen	127
4.2.2.	Die Sinusformel	128
4.3.	Vektoren als Farben	129
4.4.	Zusammenfassung und Ausblick	131
4.5.	Übungsaufgaben	131
5.	Einführung in lokale Beleuchtungsmodelle	133
5.1.	Materialeigenschaften	134
5.2.	Elementare Lichtquellenmodelle	135
5.2.1.	Positionslicht	136
5.2.2.	Strahler	139

5.2.3.	Area-Lights	141
5.2.4.	Fortgeschrittene Positionslichter und Strahler	142
5.3.	Vereinfachtes Texture-Mapping	142
5.4.	Elementare Beleuchtungsmodelle	143
5.4.1.	Ambientes Licht	143
5.4.2.	Diffuses Licht	144
5.4.3.	Spiegelndes Licht	147
5.4.4.	Emissionslicht	152
5.5.	Phong-Beleuchtungsmodell	152
5.6.	OpenGL-Beleuchtungsmodell	154
5.7.	Zusammenfassung und Ausblick	155
5.8.	Übungsaufgaben	155
6.	Schattierungsmodelle und visuelle Wahrnehmung	157
6.1.	Flat-Shading	158
6.2.	Gouraud-Shading	158
6.3.	Machsche Streifen	159
6.4.	Kurze Geschichte der Neuronenmodelle	161
6.4.1.	McCulloch-Pitts-Zelle	161
6.4.2.	Perzeptron von Rosenblatt	162
6.4.3.	Retina mit sechs Rezeptoren	163
6.5.	Phong-Shading	164
6.6.	Zusammenfassung und Ausblick	164
6.7.	Übungsaufgaben	166
7.	Objekte mit Texturen bekleiden	167
7.1.	Texture-Mapping mit OpenGL	168
7.1.1.	Korrespondenzfunktionen	169
7.1.2.	Transformationsrichtungen	169
7.1.3.	Bildpyramide mit Mip-Mapping	171
7.2.	Komplexität durch Selbstähnlichkeit	172
7.2.1.	Das Sierpinski-Dreieck	172
7.2.2.	Texture-Blending	174
7.2.3.	Primitives und komplexes Texture-Mapping	174
7.3.	Prozedurales Texture-Mapping	178
7.3.1.	Kontinuierliche Zufallszahlen	179
7.3.2.	Perlin-Noise	179
7.3.3.	Einsatz von Perlin-Noise	180
7.4.	Zusammenfassung und Ausblick	180
7.5.	Übungsaufgaben	181
8.	Beleuchtungselemente in Texturen verstecken	183
8.1.	Idee der Normalenverbiegung	184
8.2.	Struktur über Höheninformationen	184
8.2.1.	Das Bump-Mapping	185
8.2.2.	Erweiterung zum Normal-Mapping	187

8.2.3.	Bump-Mapping versus Normal-Mapping	187
8.3.	Normalen aus Höhenkarte ermitteln	188
8.3.1.	Gradienten aus Höhenkarte bestimmen	188
8.3.2.	Beispiel im Shadercode	188
8.3.3.	Normale aus Gradienten	189
8.4.	Textur statt Struktur	189
8.4.1.	Normale in Textur speichern	189
8.4.2.	Normalen aus Textur lesen	190
8.5.	Automatische Generierung einer Normal-Map	191
8.6.	Zusammenfassung und Ausblick	191
8.7.	Übungsaufgaben	191
9.	Grundlegender Umgang mit Matrizen	193
9.1.	Nützliche Javabibliotheken	194
9.2.	Matrixdarstellung und einfache Operationen	194
9.2.1.	Transponierte Vektoren und Matrizen	195
9.2.2.	Addition und Subtraktion	196
9.2.3.	Multiplikation mit Skalaren	197
9.3.	Multiplikation von Matrizen	198
9.3.1.	Zeilenweises Multiplizieren	198
9.3.2.	Spaltenweises Multiplizieren	199
9.3.3.	Vektoren-Kreuzprodukt als Matrixausdruck	200
9.3.4.	Erweiterung auf Matrix-Matrix-Multiplikation	200
9.3.5.	Falksches Schema	201
9.3.6.	Identitätsmatrix	203
9.3.7.	Übersicht zu Gesetzmäßigkeiten	203
9.4.	Lineare Gleichungssysteme	204
9.4.1.	Zeilenbild, Spaltenbild und Matrixgleichung	204
9.4.2.	Lösung eines Linearen Gleichungssystems	205
9.4.3.	Eliminationsverfahren - Echtes Versagen	206
9.4.4.	Eliminationsverfahren - Vorübergehendes Versagen	206
9.4.5.	Rang einer Matrix	206
9.5.	Matrizen invertieren und Pseudoinverse	206
9.5.1.	Inverse Matrix	207
9.5.2.	Reguläre und singuläre Matrizen	207
9.5.3.	Determinante einer 2×2 -Matrix	207
9.5.4.	Determinante einer 3×3 -Matrix	208
9.5.5.	Geometrische Bedeutung der Determinante	209
9.5.6.	Matrizen direkt invertieren	209
9.6.	Zusammenfassung und Ausblick	210
9.7.	Übungsaufgaben	210
10.	Transformationen als Vollzeitjob	211
10.1.	Transformationen in 2D und 3D	212
10.1.1.	Koordinatentransformationen im 3D-Raum	212
10.1.2.	Vektor von Raum A nach B	212

10.1.3. Vektor von Raum B nach A	213
10.1.4. Rotationsbeispiel im 2D-Raum	214
10.1.5. Transformation für Rotation im 2D-Raum	218
10.2. Homogene Koordinaten	219
10.2.1. Homogene Koordinaten in 2D	219
10.2.2. Homogene Koordinaten in 3D	220
10.2.3. Translation im 3D-Raum	220
10.2.4. Skalierung im 3D-Raum	221
10.2.5. Rotation im 3D-Raum	221
10.2.6. Transformationen kombinieren	222
10.3. Rotation mit Euler-Winkel	226
10.4. Rotationen mit Quaternionen	227
10.4.1. Von reellen zu komplexen Zahlen	227
10.4.2. Addition und Multiplikation komplexer Zahlen	228
10.4.3. Komplexe Zahlen in Matrixform	230
10.4.4. Von komplexen Zahlen zu Quaternionen	231
10.4.5. Addition, Subtraktion und Multiplikation von Quaternionen	232
10.4.6. Konjugierte, Länge und Inverse einer Quaternion	233
10.4.7. Übersicht zu Gesetzmäßigkeiten	234
10.4.8. Rotation um eine Achse	234
10.4.9. Rotation in Matrixform	238
10.4.10. Zwischen Rotationen interpolieren	239
10.4.11. Rotation um mehrere Achsen	243
10.4.12. Keyframe-Animation	243
10.5. Zusammenfassung und Ausblick	244
10.6. Übungsaufgaben	244
11. Die Koordinatensysteme der Renderpipeline	245
11.1. Von der Objektfabrik zur Bildschirmausgabe	246
11.1.1. World-Space und View-Space	247
11.1.2. View-Space und Clip-Space	247
11.1.3. Kombination zur MVP-Matrix	249
11.2. Der Tangentialraum als neues Bezugssystem	250
11.2.1. Die Transformationsrichtung entscheiden	250
11.2.2. Herleitung des Tangentialraums	251
11.2.3. Transformation vom Tangentialraum in den Objektraum	252
11.2.4. Orthogonalisierung nach Gram-Schmidt	255
11.2.5. Transformation vom Objektraum in den Tangentialraum	259
11.3. Zusammenfassung und Ausblick	260
11.4. Übungsaufgaben	260
12. Die Suche nach dem heiligen Gral	261
12.1. Interaktionen von Licht und Oberflächen	262
12.1.1. Reflexion, Transmission und Absorption	262
12.1.2. Reflexionsanteil ist von Interesse	263

12.2. Die Suche nach der allgemeinen Reflexionsfunktion	263
12.2.1. Grundsätzliche Form der Funktion	264
12.2.2. Einfluss von Licht- und Blickrichtung	265
12.2.3. Einfluss unterschiedlicher Wellenlängen	265
12.2.4. Einfluss der Position	266
12.3. Interpretation von Licht	266
12.3.1. Welle-Teilchen-Dualismus	266
12.3.2. Strahlenoptik	267
12.3.3. Kurze Begriffseinführung in die Radiometrie	268
12.4. Energiemenge statt Richtung	268
12.4.1. Das kartesische Koordinatensystem	269
12.4.2. Das Polarkoordinatensystem	270
12.4.3. Transformation in Polarkoordinaten	271
12.4.4. Transformation in kartesische Koordinaten	274
12.4.5. Vereinfachung mit Einheitskugel	275
12.4.6. Alternative Schreibweise für die Reflexionsfunktion	275
12.5. Berechnung des Lichteinfalls	276
12.5.1. Der Raumwinkel	276
12.5.2. Einfallende und ausgehende Strahlungsintensitäten	277
12.5.3. Die Rendergleichung	277
12.6. Bi-Directional Reflectance Distribution Function	278
12.6.1. Isotropische und anisotropische BRDF	279
12.6.2. Eigenschaften einer physikalischen BRDF	280
12.6.3. Kontinuierliche versus diskrete Rendergleichung	281
12.6.4. BRDF in Echtzeit-Anwendungen	282
12.6.5. Eine BRDF maschinell messen	283
12.7. Synthetische BRDF-Modelle	283
12.7.1. BRDF eines perfekten Spiegels	284
12.7.2. Phong-Beleuchtungsmodell als BRDF	284
12.7.3. BRDF für Mikrofacetten-Modelle	285
12.8. Zusammenfassung und Ausblick	289
12.9. Übungsaufgaben	289
13. Displacement-Mapping	291
13.1. Kritik am Normal-Mapping	291
13.1.1. Einführung in die Notation	292
13.1.2. Die Höhenkarte	293
13.2. Partielle Ableitungen	295
13.2.1. Funktionen mit einer Veränderlichen	295
13.2.2. Übersicht zu elementaren Ableitungsregeln	295
13.2.3. Erste und höhere Ableitungen	296
13.2.4. Funktionen mit mehreren Veränderlichen	296
13.2.5. Partielle Ableitungen von Vektoren	297
13.3. Beleuchtung und Mesostruktur	298
13.3.1. Ableitung der Tangenten	299

13.3.2. Implementierung im Fragmentshader	299
13.4. Shaderstrategien des Displacement-Mapping	300
13.4.1. Per-Vertex-Displacement-Mapping	300
13.4.2. Per-Pixel-Displacement-Mapping	301
13.5. Nicht-iterative Per-Pixel-Verfahren	304
13.5.1. Bump-Mapping	304
13.5.2. Parallax-Mapping	305
13.5.3. Parallax-Mapping mit Offsetbegrenzung	307
13.5.4. Parallax-Mapping mit Normalenneigung	308
13.6. Iterative Per-Pixel-Verfahren	309
13.7. Zusammenfassung und Ausblick	309
13.8. Übungsaufgaben	310
14. Einführung in Echtzeitschatten	311
14.1. Relevanz der Schatten	312
14.2. Terminologie	312
14.2.1. Menge verdeckter Lichtstrahlen	314
14.2.2. Kern- und Halbschatten	314
14.3. Klassifizierung von Echtzeitschatten	315
14.3.1. Hard-Shadows	315
14.3.2. Soft-Shadows	315
14.3.3. Filtered-Hard-Shadows	316
14.3.4. Komplexität versus Realismus	316
14.4. Hard-Shadows durch Shadow-Volumes	316
14.4.1. Paritätstest - Binäre Variante	318
14.4.2. Paritätstest - Zählende Variante	318
14.4.3. Bewertung und Ausblick	319
14.5. Hard-Shadows mit Shadow-Mapping	319
14.5.1. Shadow-Mapping-Verfahren	319
14.5.2. Der Effekt Surface-Acne	320
14.5.3. Der Effekt der ausgefransten Schatten	321
14.5.4. Bewertung und Ausblick	321
14.6. Realisierung von Soft-Shadows	322
14.6.1. Lichtquellen und Herausforderungen	322
14.6.2. Einsatz mehrerer Lichtquellen	322
14.6.3. Bild-basierter Ansatz mit Multipositionslichtern	324
14.6.4. Mehrere verdeckende Objekte	324
14.6.5. Single-Sample-Soft-Shadows und Smoothies	325
14.7. Potential von Ambient-Light	326
14.7.1. Ambient-Occlusion	326
14.7.2. Screen-Space-Ambient-Occlusion	327
14.8. Zusammenfassung und Ausblick	328
14.9. Übungsaufgaben	328

15. Computergrafik, aber invertiert!	331
15.1. Computergrafik versus Computer-Vision	332
15.1.1. Einstieg in die Computer-Vision mit OpenCV	332
15.1.2. Kameraverzeichnung mit OpenCV korrigieren	333
15.2. 3D-Strukturen aus Bildserien ableiten	336
15.2.1. Idee von Structure-from-Motion	336
15.2.2. Markante Merkmale in Bildern finden	336
15.2.3. Modelle aus Punktmengen ableiten	339
15.2.4. Panoramabilderstellung mit RANSAC	342
15.2.5. Epipolargeometrie	345
15.2.6. Structure-from-Motion-Prozess im Überblick	349
15.3. 3D-Rekonstruktionsprozess am konkreten Beispiel	351
15.3.1. Open-Source und proprietäre Softwarepakete	352
15.3.2. Aufnahmestrategie der Fotos	352
15.3.3. 3D-Rekonstruktion mit VisualSFM	353
15.3.4. 3D-Rekonstruktion mit Colmap	355
15.3.5. 3D-Rekonstruktion mit AliceVision Meshroom	356
15.3.6. 3D-Rekonstruktion mit Agisoft Metashape	356
15.3.7. 3D-Rekonstruktion mit RealityCapture	358
15.4. Zusammenfassung und Ausblick	359
15.5. Übungsaufgaben	359
A. Lösungen ausgewählter Aufgaben	361
B. Sammlung visueller Experimente	377
Literaturverzeichnis	379
Stichwortverzeichnis	388