

Inhaltsverzeichnis

1. Einführung in OpenGL und LWJGL	1
1.1. Einführung in OpenGL	2
1.1.1. Leistungsfähigkeit des Rechners prüfen	2
1.1.2. OpenGL als Zustandsautomat	3
1.1.3. Objektrepräsentationen	4
1.1.4. Dreiecke unter der Lupe	4
1.1.5. Primitive in OpenGL	5
1.1.6. Vereinfachte Renderpipeline	5
1.2. Einführung in LWJGL	6
1.2.1. Vorbereitung von Eclipse und LWJGL	7
1.2.2. Die Klasse LWJGLBasisFenster	9
1.3. Beispielsammlung zu OpenGL und LWJGL	11
1.3.1. Punkte erzeugen, färben und platzieren	11
1.3.2. Bewegtes Dreieck	12
1.3.3. AWT/Swing und LWJGL zusammenbringen	13
1.3.4. Abfolgen von Transformationsketten	15
1.3.5. Drehender Würfel	16
1.3.6. Würfelprojektion auf eine Fläche	18
1.3.7. Wehende Fläche	18
1.4. Objekte erstellen, speichern und laden	20
1.4.1. Das Wavefront OBJ-Format	20
1.4.2. Objekte laden und visualisieren	22
1.5. Beschleunigung und Langlebigkeit durch Buffer-Objects	23
1.5.1. Bewegtes Dreieck mit Buffer-Objects	24
1.5.2. Drehender Würfel in Farbe mit Buffer-Objects	26
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	31
1.6.3. Viele Objekte verwalten	34
1.6.4. Strategisch mit Verhalten umgehen	35
1.6.5. Positionen und Geschwindigkeiten durch Vektoren ersetzen	39
1.7. Kollektive Intelligenz und Schwarmverhalten	39
1.7.1. Steuerungsverhalten statt individueller Wegeplanung	40
1.7.2. Beschleunigung ins Spiel bringen	40
1.7.3. Kräfte aus der Physik übernehmen	44
1.7.4. Steuerungskräfte Ziel folgen, Ziel erreichen oder fliehen	46
1.7.5. Die drei Kräfte der Flocking-Boids	47

1.8. Zusammenfassung und Ausblick	49
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. Wichtige Datentypen in GLSL	60
2.2.3. Modularisierung des Programmcodes	60
2.2.4. OpenGL- und GLSL-Versionen	61
2.2.5. Shadercode mit LWJGL	61
2.2.6. Gelbes Quad rotieren lassen	63
2.2.7. Farbe aus Tiefeninformationen erzeugen	65
2.3. Blur-Effekt mit Shader	65
2.3.1. Struktur für Blur-Effekt-Shader	66
2.3.2. Gaußverteilung statt Viereck	68
2.3.3. Shadercode für Visualisierung	69
2.4. Realisierung einer einfachen Wasserdynamik	70
2.4.1. Vereinfachte Schwingungsdynamik	70
2.4.2. Schwingungsdynamik in Java	71
2.4.3. Schwingungsdynamik mit einem Fragment-Shader	72
2.5. Refraktion und Schatten	74
2.5.1. Refraktionsvektor bestimmen	74
2.5.2. Shaderstruktur für Refraktion	76
2.5.3. Einfacher Schatteneffekt durch Gradienten	78
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	81
2.6.4. Umsetzungsidee zum Kaustikeffekt	82
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 der Wassersimulation	84
2.8. Wasserdynamik im Vertex-Shader	84
2.8.1. Verknüpfung von Geometrie und Textur	85
2.8.2. Wellendynamik und Geometrie verbinden	86
2.8.3. Visualisierung der Geometrie und Shaderkomposition	87
2.9. Zusammenfassung und Ausblick	87
2.10. Übungsaufgaben	88

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 Lichteinsatz	94
3.3. Licht transportiert Kontext	95
3.4. Lichtdesigner dürfen tricksen	95
3.5. Ziele des Beleuchtungsdesigns	95
3.5.1. Dinge glaubhaft machen	96
3.5.2. Dinge lesbar oder erkennbar machen	96
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	98
3.5.6. Materialeigenschaften verstärken	98
3.5.7. Emotionalität und Kontext transportieren	101
3.6. Beleuchtungskonzept Three-Point-Lighting	101
3.6.1. Einfluss des Key-Lights	101
3.6.2. Einfluss des Fill-Lights	101
3.6.3. Aufgabe des Back-Lights	101
3.6.4. Kombination von Key-, Fill- und Back-Light	104
3.7. Beleuchtungskonzept Four-Point-Lighting	104
3.8. Komposition einer Szene	104
3.8.1. Der goldene Schnitt	105
3.8.2. Zentrierter Bildaufbau und Symmetrie	105
3.8.3. Platzierung des Horizonts	105
3.8.4. Balance zwischen Vorder- und Hintergrund	105
3.8.5. Umrahmung/Framing	108
3.8.6. Punkte und kleine Farbakzente	108
3.8.7. Regel der Quote	108
3.8.8. Diagonalen und Dreiecke	108
3.8.9. Körperhaltung berücksichtigen	108
3.8.10. Links-nach-Rechts-Regel und Regel des Raumes	111
3.8.11. Gezielte Überdeckungen	111
3.9. Zusammenfassung und Ausblick	111
3.10. Übungsaufgaben	112
4. Vektoren als Positionen, Richtungen und Farben	113
4.1. Vektoren als Richtungen	114
4.1.1. Implementierungen von Vektor2D und Vektor3D	116
4.1.2. Vektorrichtung zum oder vom Licht	119
4.1.3. Skalare und Vektoren	120
4.1.4. Vektorlänge, Normierung und inneres Produkt	120
4.1.5. Die Kosinusformel	124

4.1.6.	Das Kreuzprodukt	128
4.2.	Vektoren als Positionen	129
4.2.1.	Euklidische Distanz	129
4.2.2.	Die Sinusformel	130
4.2.3.	Baryzentrische Interpolation	131
4.3.	Vektoren als Farben	133
4.4.	Zusammenfassung und Ausblick	134
4.5.	Übungsaufgaben	134
5.	Einführung in lokale Beleuchtungsmodelle	137
5.1.	Materialeigenschaften	138
5.2.	Elementare Lichtquellenmodelle	139
5.2.1.	Positionslicht	140
5.2.2.	Strahler	143
5.2.3.	Area-Lights	145
5.2.4.	Fortgeschrittene Positionslichter und Strahler	145
5.3.	Vereinfachtes Texture-Mapping	145
5.4.	Elementare Beleuchtungsmodelle	146
5.4.1.	Ambientes Licht	146
5.4.2.	Diffuses Licht	147
5.4.3.	Spiegelndes Licht	150
5.4.4.	Emissionslicht	155
5.5.	Phong-Beleuchtungsmodell	155
5.6.	OpenGL-Beleuchtungsmodell	157
5.7.	Zusammenfassung und Ausblick	158
5.8.	Übungsaufgaben	158
6.	Schattierungsmodelle und visuelle Wahrnehmung	159
6.1.	Flat-Shading	160
6.2.	Gouraud-Shading	160
6.3.	Machsche Streifen	162
6.4.	Kurze Geschichte der Neuronenmodelle	162
6.4.1.	McCulloch-Pitts-Zelle	162
6.4.2.	Perzeptron von Rosenblatt	163
6.4.3.	Retina mit sechs Rezeptoren	165
6.5.	Phong-Shading und Gegenüberstellung	165
6.6.	Zusammenfassung und Ausblick	165
6.7.	Übungsaufgaben	167
7.	Objekte mit Texturen bekleiden	169
7.1.	Texture-Mapping mit OpenGL	170
7.1.1.	Korrespondenzfunktionen	171
7.1.2.	Transformationsrichtungen	171
7.1.3.	Bildpyramide mit Mip-Mapping	173
7.2.	Komplexität durch Selbstähnlichkeit	173
7.2.1.	Das Sierpinski-Dreieck	173

7.2.2.	Texture-Blending	174
7.2.3.	Primitives und komplexes Texture-Mapping	176
7.3.	Prozedurales Texture-Mapping	179
7.3.1.	Kontinuierliche Zufallszahlen	180
7.3.2.	Perlin-Noise	180
7.3.3.	Einsatz von Perlin-Noise	181
7.4.	Zusammenfassung und Ausblick	181
7.5.	Übungsaufgaben	182
8.	Beleuchtungselemente in Texturen verstecken	183
8.1.	Idee der Normalenverbiegung	184
8.2.	Struktur über Höheninformationen	184
8.2.1.	Das Bump-Mapping	186
8.2.2.	Erweiterung zum Normal-Mapping	186
8.2.3.	Bump-Mapping versus Normal-Mapping	187
8.3.	Normalen aus Höhenkarte ermitteln	187
8.3.1.	Gradienten aus Höhenkarte bestimmen	188
8.3.2.	Beispiel im Shadercode	188
8.3.3.	Normale aus Gradienten	188
8.4.	Textur statt Struktur	188
8.4.1.	Normale in Textur speichern	189
8.4.2.	Wassershader um Normal-Map erweitern	189
8.4.3.	Normalen aus Textur lesen	189
8.5.	Automatische Generierung einer Normal-Map	190
8.6.	Zusammenfassung und Ausblick	190
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	197
9.3.1.	Zeilenweises Multiplizieren	198
9.3.2.	Spaltenweises Multiplizieren	199
9.3.3.	Vektoren-Kreuzprodukt als Matrixausdruck	199
9.3.4.	Erweiterung auf Matrix-Matrix-Multiplikation	200
9.3.5.	Falksches Schema	200
9.3.6.	Identitätsmatrix	201
9.3.7.	Übersicht zu Gesetzmäßigkeiten	202
9.4.	Lineare Gleichungssysteme	203
9.4.1.	Zeilenbild, Spaltenbild und Matrixgleichung	203
9.4.2.	Lösung eines Linearen Gleichungssystems	204
9.4.3.	Eliminationsverfahren - Echtes Versagen	205
9.4.4.	Eliminationsverfahren - Vorübergehendes Versagen	205

9.4.5.	Rang einer Matrix	205
9.5.	Matrizen invertieren	206
9.5.1.	Inverse Matrix	206
9.5.2.	Reguläre und singuläre Matrizen	206
9.5.3.	Determinante einer 2×2 -Matrix	206
9.5.4.	Determinante einer 3×3 -Matrix	207
9.5.5.	Geometrische Bedeutung der Determinante	208
9.5.6.	Matrizen direkt invertieren	208
9.6.	Zusammenfassung und Ausblick	208
9.7.	Übungsaufgaben	209
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	242
10.4.12.	Keyframe-Animation	243
10.5.	Zusammenfassung und Ausblick	243
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	246
11.1.2.	View-Space und Clip-Space	247

11.1.3. Kombination zur MVP-Matrix	249
11.2. Der Tangentialraum als neues Bezugssystem	249
11.2.1. Die Transformationsrichtung entscheiden	250
11.2.2. Herleitung des Tangentialraums	251
11.2.3. Transformation vom Tangentialraum in den Objektraum	251
11.2.4. Orthogonalisierung nach Gram-Schmidt	254
11.2.5. Transformation vom Objektraum in den Tangentialraum	258
11.3. Zusammenfassung und Ausblick	259
11.4. Übungsaufgaben	259
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	262
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	264
12.2.3. Einfluss unterschiedlicher Wellenlängen	265
12.2.4. Einfluss der Position	265
12.3. Interpretation von Licht	266
12.3.1. Welle-Teilchen-Dualismus	266
12.3.2. Strahlenoptik	266
12.3.3. Kurze Begriffseinführung in die Radiometrie	267
12.4. Energiemenge statt Richtung	268
12.4.1. Das kartesische Koordinatensystem	269
12.4.2. Das Polarkoordinatensystem	269
12.4.3. Transformation in Polarkoordinaten	271
12.4.4. Transformation in kartesische Koordinaten	274
12.4.5. Vereinfachung mit Einheitskugel	274
12.4.6. Alternative Schreibweise für die Reflexionsfunktion	275
12.5. Berechnung des Lichteinfalls	275
12.5.1. Der Raumwinkel	275
12.5.2. Einfallende und ausgehende Strahlungsintensitäten	276
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	279
12.6.3. Kontinuierliche versus diskrete Rendergleichung	281
12.6.4. BRDF in Echtzeit-Anwendungen	282
12.6.5. Eine BRDF maschinell messen	282
12.7. Synthetische BRDF-Modelle	282
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	287

12.9. Übungsaufgaben	288
13. Displacement-Mapping	289
13.1. Kritik am Normal-Mapping	289
13.1.1. Einführung in die Notation	290
13.1.2. Die Höhenkarte	291
13.2. Partielle Ableitungen	293
13.2.1. Funktionen mit einer Veränderlichen	293
13.2.2. Elementare Ableitungsregeln	293
13.2.3. Funktionen mit mehreren Veränderlichen	294
13.2.4. Partielle Ableitungen von Vektoren	295
13.3. Beleuchtung und Mesostruktur	296
13.3.1. Ableitung der Tangenten	296
13.3.2. Implementierung im Fragmentshader	297
13.4. Shaderstrategien des Displacement-Mapping	297
13.4.1. Per-Vertex-Displacement-Mapping	297
13.4.2. Per-Pixel-Displacement-Mapping	299
13.5. Nicht-iterative Per-Pixel-Verfahren	301
13.5.1. Bump-Mapping	301
13.5.2. Parallax-Mapping	301
13.5.3. Parallax-Mapping mit Offsetbegrenzung	303
13.5.4. Parallax-Mapping mit Normalenneigung	304
13.6. Iterative Per-Pixel-Verfahren	305
13.7. Zusammenfassung und Ausblick	305
13.8. Übungsaufgaben	306
14. Einführung in Echtzeitschatten	307
14.1. Relevanz der Schatten	308
14.2. Terminologie	308
14.2.1. Menge verdeckter Lichtstrahlen	310
14.2.2. Kern- und Halbschatten	310
14.3. Klassifizierung von Echtzeitschatten	310
14.3.1. Hard-Shadows	310
14.3.2. Soft-Shadows	310
14.3.3. Filtered-Hard-Shadows	311
14.3.4. Komplexität versus Realismus	311
14.4. Hard-Shadows durch Shadow-Volumes	312
14.4.1. Paritätstest - Binäre Variante	312
14.4.2. Paritätstest - Zählende Variante	313
14.4.3. Bewertung und Ausblick	314
14.5. Hard-Shadows mit Shadow-Mapping	314
14.5.1. Shadow-Mapping-Verfahren	314
14.5.2. Der Effekt Surface-Acne	315
14.5.3. Der Effekt der ausgefransten Schatten	316
14.5.4. Bewertung und Ausblick	316

14.6. Realisierung von Soft-Shadows	317
14.6.1. Lichtquellen und Herausforderungen	317
14.6.2. Einsatz mehrerer Lichtquellen	317
14.6.3. Bild-basierter Ansatz mit Multipositionslichtern	319
14.6.4. Mehrere verdeckende Objekte	319
14.6.5. Single-Sample-Soft-Shadows und Smoothies	319
14.7. Potential von Ambient-Light	320
14.7.1. Ambient-Occlusion	321
14.7.2. Screen-Space-Ambient-Occlusion	321
14.8. Zusammenfassung und Ausblick	323
14.9. Übungsaufgaben	323
15. Computergrafik, aber invertiert!	325
15.1. Computergrafik versus Computer-Vision	325
15.1.1. Einstieg in die Computer-Vision mit OpenCV	326
15.1.2. Kameraverzeichnung mit OpenCV korrigieren	327
15.2. 3D-Strukturen aus Bildserien ableiten	329
15.2.1. Idee von Structure-from-Motion	329
15.2.2. Markante Merkmale in Bildern finden	330
15.2.3. Modelle aus Punktmengen ableiten	331
15.2.4. Panoramabilderstellung mit RANSAC	335
15.2.5. Epipolargeometrie	337
15.2.6. Structure-from-Motion-Prozess im Überblick	341
15.3. 3D-Rekonstruktionsprozess am konkreten Beispiel	344
15.3.1. Open-Source und proprietäre Softwarepakete	344
15.3.2. Aufnahmestrategien der Fotos	344
15.3.3. 3D-Rekonstruktion mit VisualSFM	345
15.3.4. 3D-Rekonstruktion mit Colmap	347
15.3.5. 3D-Rekonstruktion mit AliceVision Meshroom	347
15.3.6. 3D-Rekonstruktion mit Agisoft Metashape	348
15.3.7. 3D-Rekonstruktion mit RealityCapture	350
15.4. Zusammenfassung und Ausblick	350
15.5. Übungsaufgaben	350
A. Lösungen ausgewählter Aufgaben	351
B. Sammlung visueller Experimente	367
Literaturverzeichnis	369
Stichwortverzeichnis	378