

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Was Sie in diesem Buch erwartet . . . . .	2
1.2	Übersicht: Java auf dem Handy . . . . .	3
1.3	Wie alles begann . . . . .	4
1.4	Die Entwicklungsumgebung im Einsatz . . . . .	6
1.5	Website zum Buch . . . . .	10
1.6	Danke . . . . .	10
<b>2</b>	<b>Das erste Spiel</b>	<b>11</b>
2.1	Klassiker: Spiele mit Ball und Schläger . . . . .	12
2.2	Der grundsätzliche Aufbau eines Spiels . . . . .	13
	2.2.1 Konstruktionsmuster: Modell – Präsentation – Steuerung . . . . .	13
	2.2.2 Die Spielschleife . . . . .	14
	2.2.3 Spieldefinition: Ping . . . . .	15
2.3	Die Grafik – Präsentation . . . . .	16
	2.3.1 Das Paket javax.microedition.lcdui . . . . .	16
	2.3.2 Die Klasse GameCanvas . . . . .	17
	2.3.3 Die Klasse Graphics . . . . .	21
2.4	Der Rahmen . . . . .	27
	2.4.1 Das MIDlet . . . . .	27
	2.4.2 Thread . . . . .	30
2.5	Die Steuerung . . . . .	33
	2.5.1 Animation des Balls . . . . .	33
	2.5.2 Eingabe des Spieler . . . . .	35
	2.5.3 Kollisionsverarbeitung . . . . .	36
2.6	Verbesserungen . . . . .	37

<b>3</b>	<b>Game API</b>	<b>39</b>
3.1	Klassiker: Plattformspiele	40
3.2	Das Spiel: FreddyTheFish	41
3.2.1	Game-Design-Dokument	41
3.2.2	Technisches Design	42
3.2.3	Besseres Timing	43
3.3	Übersicht Game API	45
3.4	Sprite	46
3.4.1	Sprite erzeugen	46
3.4.2	Sprite bewegen	48
3.4.3	Die Klasse FishActor	50
3.4.4	Kollisionsverarbeitung	52
3.5	Hintergrund – TiledLayer	54
3.6	LayerManager – Organisation der Ausgabe	57
3.7	FreddyTheFish fertiggestellt	59
3.7.1	Kameraführung	59
3.7.2	Spielstatus	61
3.8	Verbesserungen	64
<b>4</b>	<b>Die Benutzeroberfläche</b>	<b>65</b>
4.1	Grundsätzliches zu Oberflächen	66
4.1.1	Übersicht Dialoge	66
4.1.2	Die Benutzeroberflächen-Komponente	68
4.2	Dialoge mit LCDUI	70
4.2.1	Start-Dialog - Klasse Canvas	72
4.2.2	Hauptmenü - Klasse List	74
4.2.3	Hilfe – Klasse TextBox	78
4.2.4	Highscore-Verwaltung	80
4.2.5	Sonstige Dialoge	88
4.3	Alles zusammengefügt	89
4.3.1	Die Klasse UiManager – Oberflächenmanagement	89
4.3.2	Die Klasse Game – Applikationsmanagement	91
4.3.3	Die Klasse GameDialog – Spielmanagement	94
4.4	Dialoge mit Scalable Vector Graphics (SVG)	95
4.4.1	Grundsätzliches zu SVG	95
4.4.2	Start-Dialog	97
4.4.3	Menü-Dialog	99
4.5	Verbesserungen	100

<b>5</b>	<b>Audio</b>	<b>101</b>
5.1	Klassiker: Spiele zum Thema Musik . . . . .	102
5.2	Grundlagen: Vom Pieps zur Symphonie . . . . .	102
5.2.1	Audio für Spiele . . . . .	103
5.2.2	MIDI . . . . .	105
5.2.3	WAVE und MP3 . . . . .	108
5.2.4	Audio-APIs in der Übersicht . . . . .	110
5.3	MMAPI – Mobile Media API . . . . .	110
5.3.1	Aufbau . . . . .	110
5.3.2	Wiedergabe – Player . . . . .	113
5.3.3	Steuern mit Controls . . . . .	114
5.4	Integration von Audio in Spiele . . . . .	117
5.4.1	Architektur . . . . .	118
5.4.2	Einsatzbereiche . . . . .	121
5.4.3	Ausblick . . . . .	124
5.5	Verbesserungen . . . . .	127
<b>6</b>	<b>Kommunikation</b>	<b>129</b>
6.1	Grundlagen . . . . .	130
6.1.1	Netzwerk-APIs . . . . .	132
6.1.2	SNAP Mobile . . . . .	135
6.2	Aufbau eines Mehrbenutzerspiels . . . . .	141
6.2.1	Architektur . . . . .	142
6.2.2	Spielstart . . . . .	144
6.2.3	Spielablauf . . . . .	156
6.2.4	Spielende . . . . .	158
6.3	Ortsbezogene Spiele – Geogaming . . . . .	161
<b>7</b>	<b>Darstellung in 3D</b>	<b>167</b>
7.1	Klassiker: Murmelspiele . . . . .	168
7.2	Das Spiel: Murmel . . . . .	169
7.3	3D-Grafik mit Java ME . . . . .	171
7.4	M3G-API . . . . .	171
7.5	Koordinatensystem . . . . .	172
7.6	Aufbau von 3D-Figuren – Mesh . . . . .	173
7.6.1	Zerlegen in Dreiecke – Triangulation . . . . .	174
7.6.2	Einfaches Speichermodell . . . . .	175
7.6.3	Indexliste . . . . .	176
7.6.4	Zusammenfassen von Dreiecken . . . . .	177

7.6.5	Gleitkommazahlen in ganze Zahlen umwandeln . . . . .	178
7.6.6	Die Klasse Mesh . . . . .	180
7.6.7	Culling – Rückseitenentfernung . . . . .	184
7.7	Ausgabe der 3D-Grafik . . . . .	185
7.7.1	Die Klasse Graphics3D . . . . .	185
7.7.2	Szenegraph – die Klasse World . . . . .	188
7.8	Bewegung in 3D . . . . .	190
7.8.1	Translation . . . . .	190
7.8.2	Skalierung . . . . .	191
7.8.3	Rotation . . . . .	192
7.8.4	Homogenisierung . . . . .	195
7.8.5	Klasse Transformation . . . . .	198
7.8.6	Klasse Transformable . . . . .	200
7.8.7	Die Grafik-Pipeline . . . . .	202
7.9	Texturen . . . . .	203
7.10	Licht und Material . . . . .	207
7.10.1	Normalvektor . . . . .	208
7.10.2	Licht . . . . .	208
7.10.3	Material . . . . .	210
7.11	Eine Szene mit 3D-Modeler erstellen . . . . .	213
7.11.1	Export aus Blender . . . . .	214
7.11.2	M3G-Dateien einlesen . . . . .	216
<b>8</b>	<b>Steuerung in 3D</b>	<b>217</b>
8.1	Programmaufbau und Szenegraph . . . . .	218
8.2	Steuerung durch den Spieler . . . . .	219
8.3	Kameraführung . . . . .	221
8.4	Spielephysik . . . . .	224
8.4.1	Physikalisches Modell . . . . .	224
8.4.2	Position – Geschwindigkeit – Beschleunigung . . . . .	227
8.4.3	Angewandte Physik . . . . .	228
8.5	Kollisionsverarbeitung . . . . .	230
8.5.1	Phase 1: Aufteilung des Raumes . . . . .	231
8.5.2	Phase 2: Kollision Kugel mit Quader . . . . .	234
8.6	Verbesserungen . . . . .	236
8.7	Schluss . . . . .	236
	<b>Weiterführende Literatur</b>	<b>237</b>
	<b>Index</b>	<b>241</b>