

Heiko Katista

C++ **für Spieleprogrammierer**

2., erweiterte Auflage

Mit einem Geleitwort von Volker Wertich
(Phenomic Game Development)

HANSER



Inhalt

Grundlagen.....	1
1 Grundlagen.....	2
1.1 Einleitung.....	2
1.1.1 An wen richtet sich dieses Buch?.....	2
1.1.2 Welche Vorkenntnisse werden benötigt?.....	2
1.1.3 Wie arbeitet man am effektivsten mit diesem Buch?.....	2
1.1.4 Geduld, Motivation und gelegentliche Tiefschläge.....	3
1.1.5 Die beigelegte CD.....	5
1.1.6 Fragen zum Buch.....	5
1.2 Die Programmiersprache C++.....	5
1.2.1 Von Lochkarten zu C++.....	5
1.2.2 Objektorientiertes Programmieren.....	7
1.2.3 Der ANSI-Standard.....	7
1.2.4 Warum gerade C++?.....	8
1.3 Jetzt geht es los...unser erstes Programm.....	9
1.3.1 Kommentare im Quelltext.....	10
1.3.2 Die #include-Anweisung.....	11
1.3.3 Die main-Funktion.....	13
1.3.4 Der „Befehl“ cout und einige mögliche Escape-Zeichen.....	14
1.4 Die Entwicklungsumgebung.....	15
1.4.1 Erstellen eines neuen Arbeitsbereiches mit Microsoft Visual C++ 6.0.....	16
1.4.2 Erstellen eines neuen Arbeitsbereiches mit Microsoft Visual C++ .NET 2003.....	17
1.4.3 Das Programm mit Hilfe des Quelltexteditors eingeben.....	18
1.4.4 Laden der Programmbeispiele von CD.....	19
1.4.5 Das Programm kompilieren, linken und ausführen.....	19
1.4.6 Ausführen des Programms.....	22
1.4.7 Der Unterschied zwischen Debug und Release.....	23
1.5 Aufgabe.....	24

Variablen.....	27
2 Variablen.....	28
2.1 Was sind Variablen, und wozu dienen sie?.....	28
2.2 Datentyp, Variablenname und Wert.....	28
2.3 Deklarieren und definieren von Variablen.....	29
2.4 Rechnen mit Variablen.....	32
2.4.1 Weitere Rechenoperatoren.....	34
2.5 Die verschiedenen Datentypen.....	36
2.6 Namenskonventionen.....	38
2.7 Konstanten.....	39
2.7.1 Konstanten mit „const“ erzeugen.....	40
2.7.2 Konstanten mit „#define“ erzeugen.....	40
2.7.3 Konstanten mit „enum“ erzeugen.....	41
2.7.4 Welche der drei Möglichkeiten ist die Beste?.....	42
2.8 Mach mal Platz: Speicherbedarf der Datentypen.....	42
2.8.1 Überlauf von Variablen.....	44
2.9 Eingabe von Werten mit „ein“.....	45
2.10 Casting: Erzwungene Typenumwandlung.....	47
2.10.1 Casting im C-Stil.....	48
2.10.2 Casting mit C++.....	49
2.11 Fehler Quelltext.....	51
2.11.1 Was soll das Programm eigentlich tun?.....	51
2.11.2 Lösung zum Fehler Quelltext.....	52
Schleifen und Bedingungen.....	57
3 Schleifen und Bedingungen.....	58
3.1 Was sind Schleifen und Bedingungen, und wozu dienen sie?.....	58
3.2 Boolesche Operatoren (=, <, >, !=).....	58
3.3 Die if-Bedingung.....	60
3.4 Mittels „eise“ flexibler verzweigen.....	62
3.5 eise if und verschachtelte if-Bedingungen.....	64
3.6 Logische Operatoren.....	68
3.7 Verzweigen mit switch und case.....	70
3.8 Immer und immer wieder: for-Schleifen.....	73
3.8.1 Initialisierungsteil.....	75
3.8.2 Bedingungsteil.....	75
3.8.3 Aktionsteil.....	76
3.8.4 Zusammenfassung.....	76
3.9 Eine weitere Rechenoperation: Modulo.....	77
3.10 Aufgabenstellung.....	78
3.10.1 Wie geht man an die Aufgabe heran?.....	78
3.10.2 Lösungsvorschlag.....	79
3.11 Schleifen mit while und do-while.....	81
3.12 Verschachtelte Schleifen.....	84

3.13	Fehlerquelltext.....	85
3.13.1	Was soll das Programm eigentlich tun?.....	86
3.13.2	Lösung zum Fehlerquelltext.....	87
Funktionen.....		91
4	Funktionen.....	92
4.1	Was sind Funktionen, und wozu dienen sie?.....	92
4.2	Aufbau des Funktionskopfes.....	94
4.2.1	Rückgabetyyp.....	94
4.2.2	Funktionsname.....	94
4.2.3	Parameterliste.....	95
4.3	Aufrufen einer Funktion.....	95
4.3.1	Funktionsprototypen.....	95
4.4	Gültigkeitsbereiche.....	97
4.4.1	Lokale Variablen.....	98
4.4.2	Globale Variablen.....	98
4.4.3	Das wäre ja zu einfach gewesen: globale Variablen am Pranger.....	99
4.5	Verwenden der Funktionsparameter.....	99
4.5.1	Der Stack.....	101
4.6	inline-Funktionen.....	103
4.7	Wann setzt man Funktionen ein?.....	105
4.7.1	Und wann soll's inline sein?.....	106
4.8	Überladene Funktionen.....	107
4.9	Aufgabenstellung.....	111
4.9.1	Wie geht man an die Aufgabe heran?.....	111
4.9.2	Lösungsvorschlag.....	112
4.10	Der sinnvolle Aufbau des Quellcodes.....	113
4.11	Erstellen und Hinzufügen der neuen Dateien.....	114
4.12	Das Schlüsselwort „extern“.....	117
4.13	Ein kleines Spiel: Zahlenraten.....	118
4.13.1	Zufallszahlen und Bibliotheksdateien.....	123
4.13.2	Die Hauptfunktion (main).....	125
4.13.3	Die Funktion „WahleLevel“.....	126
4.13.4	Die Funktion „Spielen“.....	126
4.13.5	Was gibt es an diesem Listing zu kritisieren?.....	127
Arrays und Strukturen.....		129
5	Arrays und Strukturen.....	130
5.1	Was sind Arrays, und wozu dienen sie?.....	130
5.2	Ein Array erzeugen.....	130
5.3	Ein Array gleichzeitig deklarieren und definieren.....	132
5.4	Fehler beim Verwenden von Arrays.....	134
5.5	char-Arrays.....	134
5.6	Eingabe von Strings über die Tastatur.....	136

5.7	Mehrdimensionale Arrays.....	138
5.8	Arrays und Speicherbedarf.....	140
5.9	Was sind Strukturen, und wozu dienen sie?.....	140
5.10	Spielerverwaltung mit Strukturen und Arrays.....	143
5.11	Aufgabenstellung.....	146
5.11.1	Wie geht man an die Aufgabe heran?.....	147
5.11.2	Lösungsvorschlag.....	148
Zeiger und Referenzen.....		153
6	Zeiger und Referenzen.....	154
6.1	Was sind Zeiger, und wozu dienen sie?.....	154
6.1.1	Der Stack.....	155
6.1.2	Vom Flur in den Keller.....	156
6.2	Die Adresse einer Variablen.....	157
6.3	Die Adresse einer Variablen in einem Zeiger speichern.....	159
6.3.1	Schreibweisen bei der Deklaration.....	161
6.4	Variablen mittels Zeigern ändern.....	162
6.5	Schön und gut, aber wozu wird das gebraucht?.....	163
6.6	Noch einmal zurück zum Flur.....	166
6.7	Was sind Referenzen, und wozu dienen sie?.....	168
6.7.1	Mit Referenzen arbeiten.....	169
6.7.2	Regeln bei der Verwendung von Referenzen.....	170
6.8	Referenzen als Funktionsparameter.....	171
6.9	Warum Zeiger nehmen, wenn es Referenzen gibt?.....	173
6.10	Aufgabenstellung.....	175
6.10.1	Wie geht man an die Aufgabe heran?.....	176
6.10.2	Lösungsvorschlag.....	177
Klassen.....		181
7	Klassen..... *	182
7.1	Was sind Klassen, und wozu dienen sie?.....	182
7.2	Eine einfache Klasse erzeugen und verwenden.....	184
7.3	Ordnung muss sein.....	187
7.4	Jetzt wird es privat.....	189
7.4.1	Private Membervariablen.....	190
7.4.2	Private Membervariablen und Performance.....	192
7.4.3	Private Memberfunktionen.....	193
7.5	Konstruktoren und Destruktoren.....	194
7.5.1	Der Konstruktor.....	194
7.5.2	Konstruktoren mit Parameterliste.....	196
7.5.3	Überladene Konstruktoren.....	198
7.6	Der Destruktor.....	200
7.7	Speicherreservierung.....	202
7.7.1	NewundDelete.....	203

7.7.2	Ein sinnvollerer Beispiel.....	204
7.7.3	Friss mich, ich bin Dein Speicher.....	207
7.8	Aufgabenstellung.....	209
7.8.1	Wie geht man an die Aufgabe heran?.....	210
7.8.2	Lösungsvorschlag.....	211
7.9	Vererbung.....	213
7.9.1	Überschreiben von Memberfunktionen.....	219
7.9.2	Virtuelle Memberfunktionen.....	222
7.9.3	Vererbung und Performance.....	226
7.10	Statische Membervariablen.....	227
Fortgeschrittene Themen.....		231
8	Fortgeschrittene Themen.....	232
8.1	Über dieses Kapitel.....	232
8.2	printfundsprintf.....	232
8.2.1	printf.....	232
8.2.2	sprintf.....	234
8.3	Templates.....	236
8.3.1	Template-Funktionen.....	237
8.3.2	Template-Klassen.....	239
8.4	Singletons.....	243
8.4.1	Eine Klasse für Singleton.....	243
8.4.2	Einsatz von Singleton.....	245
8.5	Dateien: Ein- und Ausgabe.....	248
8.5.1	Werte in eine Datei schreiben und auslesen.....	248
8.5.2	So viel Zeit muss sein: Fehlerabfrage.....	250
8.5.3	Instanzen von Klassen in Dateien schreiben.....	252
8.5.4	Weitere Flags und ihre Bedeutung.....	254
8.6	Eine nützliche Logfile-Klasse.....	255
8.6.1	Die Header-Datei der Logfile-Klasse.....	255
8.6.2	Die Implementierung der Logfile-Klasse.....	258
8.6.3	Anwendung der Logfile-Klasse.....	264
8.7	Try, Catch und Assert.....	266
8.7.1	Das Makro „assert“.....	266
8.7.2	Fang mich, wenn Du kannst: try und catch.....	269
8.8	Der Debugger.....	271
8.8.1	Das Programm im Einzelschrittmodus durchlaufen.....	271
8.8.2	Haltepunkte und Funktionsaufrufe.....	274
8.9	SAFEDELETE - Ein nützliches Makro.....	277
DieSTL.....		279
9	DieSTL.....	280
9.1	STL - was ist das?.....	280
9.1.1	Vektoren.....	280

9.1.2	Verkettete Listen.....	287
9.1.3	Strings.....	297
9.1.4	Maps und Multimaps.....	303
Grundlagen der Windows-Programmierung.....		313
10	Grundlagen der Windows-Programmierung.....	314
10.1	Raus aus der Konsole, rein ins Fenster.....	314
10.1.1	Anlegen eines Win32-Projektes.....	314
10.1.2	Ein Windows-Grundgerüst.....	315
10.1.3	Die WinMain-Funktion.....	318
10.1.4	Die Callback-Funktion.....	326
10.1.5	Zusammenfassung.....	328
10.1.6	Ein kurzer Abstecher: Funktionszeiger.....	329
10.2	Aufgabenstellung.....	332
10.2.1	Wie geht man an die Aufgabe heran?.....	332
10.2.2	Lösungsvorschlag.....	334
10.3	Ein bisschen mehr Interaktion.....	336
10.3.1	Statischer Text.....	336
10.3.2	Buttons und Editboxen.....	338
10.3.3	Messageboxen.....	339
10.4	Alles noch einmal zusammen.....	341
Sonst noch was?.....		351
11	Sonst noch was?.....	352
11.1	Um was geht es in diesem Kapitel?.....	352
11.2	Standardwerte für Funktionsparameter.....	353
11.3	Memberinitialisierung im Konstruktor.....	355
11.4	Der this-Zeiger.....	360
11.5	Der Kopierkonstruktor.....	365
11.6	Überladen von Operatoren.....*	370
11.7	Mehrfachvererbung.....	375
11.8	Friend-Klassen.....	380
Ein Spiel mit der SDL.....		387
12	Ein Spiel mit der SDL.....	388
12.1	Die SDL - was ist das?.....	388
12.2	Erstellen und Einrichten des Projektes.....	389
12.2.1	Projekt mit Microsoft Visual C++6.0 anlegen.....	390
12.2.2	Projekt mit Microsoft Visual C++ .net 2003 anlegen.....	390
12.2.3	Die restlichen Quellcode-Dateien.....	391
12.3	Projektübersicht.....	392
12.3.1	Wieso plötzlich Englisch? Und wo sind die Zeilennummern?.....	393

12.3.2	Übersicht der Klassen.....	393
12.4	Die Implementierung des Spiels.....	395
12.4.1	Die main-Funktion des Spiels.....	395
12.4.2	Zeit ist wichtig: die Klasse CTimer.....	397
12.4.3	Die Klasse CFramework.....	399
12.4.4	Bunte Bilder: die Klasse CSprite.....	407
12.4.5	Feuer frei: die Klasse CShot.....	416
12.4.6	Die Klasse CAsteroid.....	419
12.4.7	Die Hauptfigur: die Klasse CPlayer.....	422
12.4.8	Die Klasse CGame.....	429
12.5	Erweiterungsmöglichkeiten.....	438
Der Einstieg in die Szene.....		441
13	Der Einstieg in die Szene.....	442
13.1	Wie geht's nun weiter?.....	442
13.2	Die Szene...was ist das eigentlich?.....	443
13.3	Welche Möglichkeiten gibt es?.....	444
13.4	Foren benutzen.....	445
13.4.1	Ich sag Dir, wer ich bin.....	445
13.4.2	Richtig posten.....	445
13.4.3	FAQs und die Suchfunktion.....	448
13.4.4	Die Kunst zu lesen.....	448
13.4.5	Selbst Initiative ergreifen und anderen helfen.....	449
13.5	Weiterbildung mit Tutorials.....	449
13.6	Anlegen einer Linksammlung.....	450
13.7	Copy& Paste.....	451
13.8	Die Sprache neben C++: Englisch.....	452
13.9	Auf dem Weg zum eigenen Spiel.....	452
13.9.1	Mein erstes Spiel: ein 3D-online-Rollenspiel für 500 Leute.....	452
13.9.2	Teammitgliedersuchen.....	453
13.9.3	Das fertige Spiel bekannt machen.....*	454
13.9.4	Besuchen von Events zum Erfahrungsaustausch.....	454
13.10	return 0;	
Index.....		457