

Inhalt

1	Grundlagen.....	1
1.1	Einleitung.....	1
1.1.1	An wen richtet sich dieses Buch?	1
1.1.2	Welche Vorkenntnisse werden benötigt?	1
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	4
1.1.6	Fragen zum Buch.....	4
1.2	Die Programmiersprache C++.....	4
1.2.1	Von Lochkarten zu C++	4
1.2.2	Objektorientiertes Programmieren.....	6
1.2.3	Der ANSI-Standard	6
1.2.4	Warum gerade C++?	7
1.3	Jetzt geht es los ... unser erstes Programm	8
1.3.1	Kommentare im Quelltext	9
1.3.2	Die #include-Anweisung	10
1.3.3	Die main-Funktion	12
1.3.4	„cout“ und einige mögliche Escape-Zeichen.....	13
1.4	Die Entwicklungsumgebung	14
1.4.1	Erstellen eines neuen Arbeitsbereiches.....	14
1.4.2	Das Programm mithilfe des Quelltexteditors eingeben	16
1.4.3	Laden der Programmbeispiele von CD.....	17
1.4.4	Das Programm kompilieren und linken	18
1.4.5	Ausführen des Programms.....	20
1.4.6	Der Unterschied zwischen Debug und Release	21
1.5	Aufgabe.....	22
2	Variablen.....	23
2.1	Was sind Variablen, und wozu dienen sie?	23
2.2	Datentyp, Variablenname und Wert	24
2.3	Deklarieren und definieren von Variablen	24

2.4	Rechnen mit Variablen	28
2.4.1	Weitere Rechenoperatoren	29
2.5	Die verschiedenen Datentypen.....	31
2.6	Namenskonventionen.....	33
2.7	Konstanten	34
2.7.1	Konstanten mit „const“ erzeugen	35
2.7.2	Konstanten mit „#define“ erzeugen.....	35
2.7.3	Konstanten mit „enum“ erzeugen.....	36
2.7.4	Welche der drei Möglichkeiten ist die beste?	37
2.8	Mach mal Platz: Speicherbedarf der Datentypen	38
2.8.1	Überlauf von Variablen	39
2.9	Eingabe von Werten mit „cin“	41
2.10	Casting: Erzwungene Typenumwandlung.....	42
2.10.1	Casting im C-Stil	43
2.10.2	Casting mit C++	45
2.11	Fehlerquelltext	46
2.11.1	Was soll das Programm eigentlich tun?	47
2.11.2	Lösung zum Fehlerquelltext.....	48
3	Schleifen und Bedingungen	51
3.1	Was sind Schleifen und Bedingungen, und wozu dienen sie?.....	51
3.2	Boolesche Operatoren ($=$, $<$, $>$, \neq).....	52
3.3	Die if-Bedingung	53
3.4	Mittels „else“ flexibler verzweigen.....	55
3.5	else if und verschachtelte if-Bedingungen	57
3.6	Logische Operatoren.....	61
3.7	Verzweigen mit switch und case.....	63
3.8	Immer und immer wieder: for-Schleifen.....	67
3.8.1	Initialisierungsteil.....	68
3.8.2	Bedingungsteil.....	69
3.8.3	Aktionsteil.....	69
3.8.4	Zusammenfassung	70
3.9	Eine weitere Rechenoperation: Modulo	70
3.10	Aufgabenstellung	71
3.10.1	Wie geht man an die Aufgabe heran?	72
3.10.2	Lösungsvorschlag	72
3.11	Schleifen mit while und do-while	74
3.12	Verschachtelte Schleifen	78
3.13	Fehlerquelltext	79
3.13.1	Was soll das Programm eigentlich tun?	79
3.13.2	Lösung zum Fehlerquelltext.....	80
4	Funktionen	83
4.1	Was sind Funktionen, und wozu dienen sie?	83
4.2	Aufbau des Funktionskopfes.....	85
4.2.1	Rückgabetyp	85
4.2.2	Funktionsname	85
4.2.3	Parameterliste	86

4.3	Aufrufen einer Funktion.....	86
4.3.1	Funktionsprototypen.....	87
4.4	Gültigkeitsbereiche	88
4.4.1	Lokale Variablen	89
4.4.2	Globale Variablen.....	90
4.4.3	Das wäre ja zu einfach gewesen: globale Variablen am Pranger.....	90
4.5	Verwenden der Funktionsparameter.....	91
4.5.1	Der Stack.....	93
4.6	inline-Funktionen	95
4.7	Wann setzt man Funktionen ein?	97
4.7.1	Und wann soll's inline sein?.....	97
4.8	Überladene Funktionen	98
4.9	Aufgabenstellung	102
4.9.1	Wie geht man an die Aufgabe heran?.....	103
4.9.2	Lösungsvorschlag.....	103
4.10	Der sinnvolle Aufbau des Quellcodes	105
4.11	Erstellen und Hinzufügen der neuen Dateien.....	106
4.12	Das Schlüsselwort „extern“	109
4.13	Ein kleines Spiel: Zahlenraten.....	110
4.13.1	Zufallszahlen und Bibliotheken.....	115
4.13.2	Die Hauptfunktion (main)	116
4.13.3	Die Funktion „WaehleLevel“	117
4.13.4	Die Funktion „Spielen“	118
4.13.5	Was gibt es an diesem Listing zu kritisieren?.....	119
5	Arrays und Strukturen.....	121
5.1	Was sind Arrays, und wozu dienen sie?.....	121
5.2	Ein Array erzeugen	121
5.3	Ein Array gleichzeitig deklarieren und definieren.....	123
5.4	Fehler beim Verwenden von Arrays.....	125
5.5	char-Arrays	126
5.6	Eingabe von Strings über die Tastatur.....	128
5.7	Mehrdimensionale Arrays	129
5.8	Arrays und Speicherbedarf.....	131
5.9	Was sind Strukturen, und wozu dienen sie?	132
5.10	Spielerverwaltung mit Strukturen und Arrays.....	134
5.11	Aufgabenstellung	137
5.11.1	Wie geht man an die Aufgabe heran?.....	138
5.11.2	Lösungsvorschlag	139
6	Zeiger und Referenzen	145
6.1	Was sind Zeiger, und wozu dienen sie?	145
6.1.1	Der Stack.....	147
6.1.2	Vom Flur in den Keller.....	148
6.2	Die Adresse einer Variablen	149
6.3	Die Adresse einer Variablen in einem Zeiger speichern	150
6.3.1	Schreibweisen bei der Deklaration	152
6.4	Variablen mittels Zeigern ändern	153

6.5	Schön und gut, aber wozu wird das gebraucht?	154
6.6	Noch einmal zurück zum Flur.....	157
6.7	Was sind Referenzen, und wozu dienen sie?	159
6.7.1	Mit Referenzen arbeiten	160
6.7.2	Regeln bei der Verwendung von Referenzen.....	161
6.8	Referenzen als Funktionsparameter	162
6.9	Warum Zeiger nehmen, wenn es Referenzen gibt?.....	164
6.10	Aufgabenstellung	166
6.10.1	Wie geht man an die Aufgabe heran?.....	167
6.10.2	Lösungsvorschlag.....	168
7	Klassen	173
7.1	Was sind Klassen, und wozu dienen sie?	173
7.2	Eine einfache Klasse erzeugen und verwenden.....	175
7.3	Ordnung muss sein.....	178
7.4	Jetzt wird es privat	181
7.4.1	Private Membervariablen	181
7.4.2	Private Membervariablen und Performance	183
7.4.3	Private Memberfunktionen.....	184
7.5	Konstruktoren und Destruktoren.....	185
7.5.1	Der Konstruktor	185
7.5.2	Konstruktoren mit Parameterliste.....	187
7.5.3	Überladene Konstruktoren	189
7.6	Der Destruktor	191
7.7	Speicherreservierung.....	193
7.7.1	New und Delete.....	194
7.7.2	Ein sinnvollerer Beispiel	196
7.7.3	Friss mich, ich bin Dein Speicher.....	199
7.8	Aufgabenstellung	201
7.8.1	Wie geht man an die Aufgabe heran?.....	202
7.8.2	Lösungsvorschlag.....	202
7.9	Vererbung	205
7.9.1	Überschreiben von Memberfunktionen	210
7.9.2	Virtuelle Memberfunktionen.....	213
7.9.3	Vererbung und Performance	218
7.10	Statische Membervariablen.....	218
8	Fortgeschrittene Themen	223
8.1	Über dieses Kapitel	223
8.2	printf und sprintf	223
8.2.1	printf.....	224
8.2.2	sprintf	226
8.3	Templates.....	228
8.3.1	Template-Funktionen	228
8.3.2	Template-Klassen.....	231
8.4	Singletons	234
8.4.1	Eine Klasse für Singletons	235
8.4.2	Einsatz von Singletons	237

8.5	Dateien: Ein- und Ausgabe	239
8.5.1	Werte in eine Datei schreiben und auslesen	239
8.5.2	So viel Zeit muss sein: Fehlerabfrage.....	242
8.5.3	Instanzen von Klassen in Dateien schreiben.....	243
8.5.4	Weitere Flags und ihre Bedeutung	246
8.6	Eine nützliche Logfile-Klasse	246
8.6.1	Die Header-Datei der Logfile-Klasse	247
8.6.2	Die Implementierung der Logfile-Klasse	250
8.6.3	Anwendung der Logfile-Klasse.....	256
8.7	Try, Catch und Assert	257
8.7.1	Das Makro „assert“	258
8.7.2	Fang mich, wenn Du kannst: try und catch.....	261
8.8	Der Debugger.....	263
8.8.1	Das Programm im Einzelschrittmodus durchlaufen	263
8.8.2	Haltepunkte und Funktionsaufrufe	266
8.9	SAFE_DELETE – ein nützliches Makro.....	269
9	Die STL	271
9.1	STL – was ist das?	271
9.1.1	Vektoren.....	272
9.1.2	Verkettete Listen	278
9.1.3	Strings	288
9.1.4	Maps und Multimaps.....	294
10	Grundlagen der Windows-Programmierung.....	303
10.1	Raus aus der Konsole, rein ins Fenster.....	303
10.1.1	Anlegen eines Win32-Projektes	304
10.1.2	Ein Windows-Grundgerüst.....	304
10.1.3	Die WinMain-Funktion	308
10.1.4	Die Callback-Funktion	315
10.1.5	Zusammenfassung	317
10.1.6	Ein kurzer Abstecher: Funktionszeiger.....	318
10.2	Aufgabenstellung	321
10.2.1	Wie geht man an die Aufgabe heran?.....	322
10.2.2	Lösungsvorschlag.....	323
10.3	Ein bisschen mehr Interaktion.....	325
10.3.1	Statischer Text.....	325
10.3.2	Buttons und Editboxen	327
10.3.3	Messageboxen	328
10.4	Alles noch einmal zusammen.....	330
11	Sonst noch was?	339
11.1	Um was geht es in diesem Kapitel?.....	339
11.2	Standardwerte für Funktionsparameter	340
11.3	Memberinitialisierung im Konstruktor.....	343
11.4	Der this-Zeiger	347
11.5	Der Kopierkonstruktor	352
11.6	Überladen von Operatoren	358

Inhalt

11.7	Mehrfachvererbung	363
11.8	Friend-Klassen	368
12	Ein Spiel mit der SDL	373
12.1	Die SDL – was ist das?	373
12.2	Erstellen und Einrichten des Projektes	374
12.2.1	Das Projekt anlegen und einrichten	375
12.2.2	Die restlichen Quellcodedateien	376
12.3	Projektübersicht	376
12.3.1	Warum plötzlich Englisch? Und wo sind die Zeilennummern?	377
12.3.2	Übersicht der Klassen	378
12.4	Die Implementierung des Spiels	379
12.4.1	Die main-Funktion des Spiels	380
12.4.2	Zeit ist wichtig: Die Klasse CTimer	381
12.4.3	Die Klasse CFramework	384
12.4.4	Bunte Bilder: Die Klasse CSprite	391
12.4.5	Feuer frei: die Klasse CShot	400
12.4.6	Die Klasse CAsteroid	404
12.4.7	Die Hauptfigur: Die Klasse CPlayer	406
12.4.8	Die Klasse CGame	414
12.5	Erweiterungsmöglichkeiten	422
13	Der Einstieg in die Szene	425
13.1	Wie geht's nun weiter?	425
13.2	Die Szene ... was ist das eigentlich?	426
13.3	Welche Möglichkeiten gibt es?	427
13.4	Foren benutzen.....	428
13.4.1	Ich sag Dir, wer ich bin	428
13.4.2	Richtig posten	429
13.4.3	FAQs und die Suchfunktion	431
13.4.4	Die Kunst zu lesen	432
13.4.5	Selbst Initiative ergreifen und anderen helfen	432
13.5	Weiterbildung mit Tutorials	433
13.6	Anlegen einer Linkssammlung	434
13.7	Copy & Paste	434
13.8	Die Sprache neben C++: Englisch	435
13.9	Auf dem Weg zum eigenen Spiel	436
13.9.1	Mein erstes Spiel: ein 3D-Online-Rollenspiel für 500 Leute	436
13.9.2	Teammitglieder suchen	437
13.9.3	Das fertige Spiel bekannt machen	437
13.9.4	Besuchen von Events zum Erfahrungsaustausch	438
13.10	return 0;	438
Register.....	441	