

DAS GROSSE BUCH DER COMPUTERSPIELE

Unter den großen Namen der Heimcomputer-Literatur ist Tim Hartnell einer der größten, dank seiner mehr als 30 weltweit in Riesenaufgaben verbreiteten Lehr- und Spiel-Bücher ist der junge Engländer völlig zurecht als „Chef-Guru der internationalen Computer Freak-Szene“ (SILICON TIMES) berühmt geworden und es gibt wohl kaum einen Computer-User, der dem Können und Witz von Tim Hartnell nicht seine wichtigsten Lehrerehrungen verdankt.

Mit DAS GROSSE BUCH DER COMPUTER-SPIELE präsentiert Tim Hartnell nun sein opus magnum, das Hauptwerk, das die Summe seiner eigenen Erfahrungen und der Computer-Erfahrungen seiner Leser zieht. Das Buch enthält mehr als 40 Spiele, alle Programme sind Microsoft BASIC kompatibel und laufen auf folgenden Maschinen: COMMODORE 64, APPLE II, IBM PC, ZX SPECTRUM, ACORN B, VC 20, ORIC, DRAGON 32, TANDY COLOR, TRS-80, T/S 2000, VZ 200 und TEXET.

Tim Hartnell: „Ich habe die verschiedensten Spiele in dieses Buch aufgenommen, um den verschiedensten Geschmäckern gerecht zu werden. Die Spiele umfassen klassische Brettspiele wie Schach und Reversi, Abenteuer-Programme wie 'Festung der Zwergefürsten' und 'Herzog von Drachenfurcht', Simulationen wie 'Herrin von Xenophobia' sowie Denk-Spiele, Weltall-Spiele, Spaß-Spiele, Würfel-Spiele und vieles andere mehr. Ich habe das Buch absichtlich so geschrieben, daß es 'offene Enden' hat, die Programme sind bewußt so gehalten, daß sie verbesserungs- und ausbaufähig sind. Jeder User kann die Programme auf seine Weise fortschreiben, und so wird jeder Leser zum Mitautor dieses Buches – und das ist für mich eine sehr angenehme Vorstellung!“

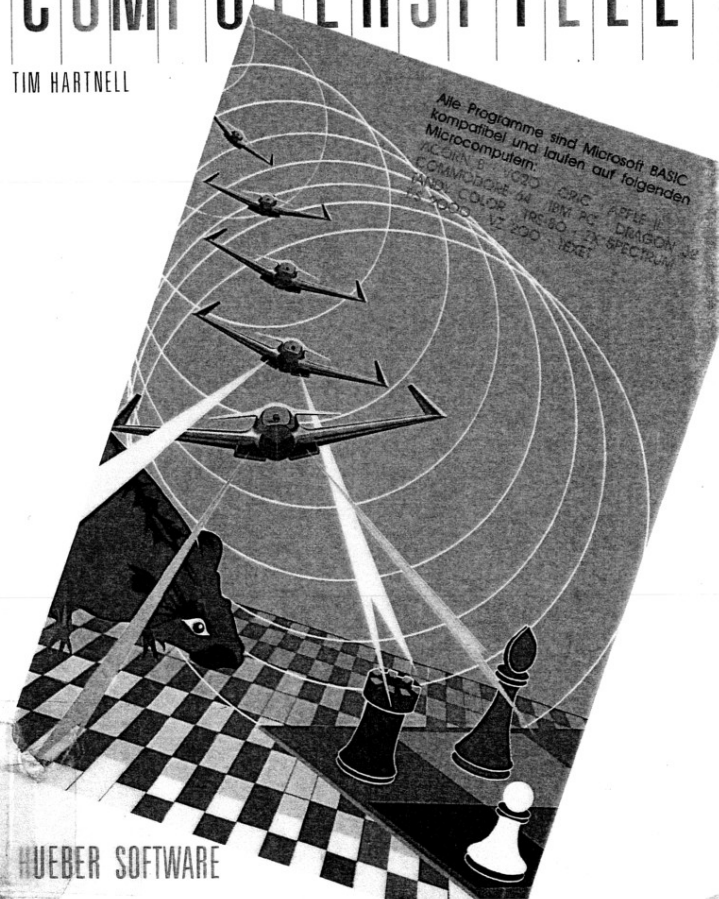
ISBN 3-19-008210-3

8210

HUEBER SOFTWARE

DAS GROSSE BUCH DER COMPUTERSPIELE

TIM HARTNELL





TIM HARTNELL

DAS GROSSE BUCH DER COMPUTERSPIELE

Letzter Rückgabetermin — nach Überschreiten dieses Termins
wird eine Versäumnisgebühr erhoben.

28. FEB. 1992 23	25. Mai 1998 23 ⁰⁰
10. NOV. 1993 23	15. Okt. 1998 23
15. FEB. 1994 23	2. März 1999 23 ⁰⁰
17. FEB. 1994 23	
12. MAI 1994 23	
13. OKT. 1994 23	
21. FEB. 1995 23	
18. JAN. 1996 23	
04. OKT. 1996 23	
27. April 1998 23	

Titel der englischen Originalausgabe:
Giant Book of Computer Games

Übersetzung: Thomas Müller
Umschlagfoto: Computer Grafik Mayer, München
Umschlagentwurf: Agentur Cooperation, München



CIP-Kurztitelaufnahme der Deutschen Bibliothek

Hartnell, Tim:
Das große Buch der Computer-Spiele / Tim Hartnell.
[Übers.: Thomas Müller]. – 1. Aufl. – München. [i.e. Ismaning]:
Hueber, 1984. (Hueber-Software)
Einheitssacht.: Giant book of computer games «dt.»
ISBN 3-19-008210-3

Alle Rechte, auch die des Nachdruckes, der Wiedergabe in jeder Form und der Übersetzung in andere Sprachen, behalten sich Urheber und Verleger vor. Es ist ohne schriftliche Genehmigung des Verlages nicht erlaubt, das Buch oder Teile daraus auf fotomechanischem Weg (Fotokopie, Mikrokopie) zu vervielfältigen oder unter Verwendung elektronischer bzw. mechanischer Systeme zu speichern, systematisch auszuwerten oder zu verbreiten (mit Ausnahme der in den §§ 53, 54 URG ausdrücklich genannten Sonderfälle).

1. Auflage^{3 21}

© der Originalausgabe by Tim Hartnell 1983; erschienen bei Interface Publications 1983

© der deutschsprachigen Ausgabe by Max Hueber Verlag, München 1984
Satz: Typo Dreitausend, München

Druck: Ludwig Auer, Donauwörth · Printed in Germany
ISBN 3-19-008210-3

Inhalt

Einführung	5
Wie man Programme zum Laufen bringt	5
Brettspiele	9
Wie der Computer spielt	10
Schach	12
Gomoku	33
Vier-in-einer-Reihe	37
Dame	43
Shogun	50
Awari	54
Springeraufmarsch	59
Reversi/Othello	62
Abenteuer	69
Festung der Zwergefürsten	74
Herzog von Drachenfurcht	79
Simulationen	89
Herrin von Xenophobia	90
Ländliche Geschäfte	94
Geschäftsführer	98
Würfelspiele	107
Chemin de Computer	108
Sieben/Elf	114
Keine Ursache	116
Eins-und-Zwanzig	120
Schlangenaugen	124
Drunter und Drüber	132
Künstliche Intelligenz	137
Eliza	139
Elektronengehirn	147
Memory	154
Dichter	159
Nur zum Spaß	163
Dazwischen	163
Nimgrab	166

Caddy	170
Aufmarsch von Puzzles	176
Roboterminfeld	192
Drei in einer Reihe	200
Las Vegas	205
Spaß mit Ihrem Drucker	211
Celestia	211
Plakat	217
Weitallspiele	223
Mondlandung I und II	223
Hyperkrieg	227
Gehirnspiele	235
Fastermind	236
Idaho-Sterne	239
Henker	245
Kubist	250
Murmelerde Marmor	253
Zyklotron	256
Vertauschereien	262
Flipper	267
Suche nach Daryl	272
Erstellen Ihrer eigenen Spiele – Spiele erfinden	285
Anzupassende Spiele	286
Glossar	293
Konversionen	301

Einführung

Es gab einmal eine Zeit, da konnte man noch nicht so einfach Computerspiele schreiben und mit ihnen spielen. In den dunklen Zeiten der Vergangenheit kostete es unter Umständen bis zu 300 DM pro Stunde, um ein Spiel auf einem time-sharing System zu spielen. Die einzige Ausgabemöglichkeit, die verfügbar war, war zumeist ein lärmender, alter Drucker. Man stelle sich die Auswirkungen dieser Umstände auf das jetzige Arbeiten vor!

Glücklicherweise brauchen wir nicht unter solchen mittelalterlichen Bedingungen zu leiden. Sie haben einen Computer, mit dem Sie stundenlang arbeiten können, um Programme zu entwickeln und damit zu spielen.

In diesem Buch finden Sie eine Sammlung meiner liebsten Computerspiele. Ich hoffe, daß Sie einige dieser Programme verwenden werden und daß sie auch zu Ihren Lieblingsspielen werden. Sie werden sicherlich viel Gefallen an den Spielen finden.

Ich habe versucht, die verschiedensten Spiele in dieses Buch aufzunehmen, um den verschiedenen Geschmäckern gerecht zu werden. Die Spiele umfassen klassische Brettspiele, wie z.B. **Schach** und **Reversi/Othello**, weiter unterhaltsame Programme, wie z.B. **Die Herrscherin von Xenophobia**, herausfordernde Spiele wie das Adventure-Spiel **Die Festung des Zwergenfürsten**. Das Buch beinhaltet fast 40 Spiele, um sicherzugehen, daß Ihr Computer garantiert beschäftigt wird. Ich habe dieses Buch absichtlich so geschrieben, daß es ein „offenes“ Ende hat. Im Gegensatz zu vielen Sammlungen, die auf dem Markt sind, habe ich die Programme bewußt so gehalten, daß Sie verbesserungsfähig sind. Deshalb habe ich Programmunterbrechungen eingebaut, damit man sieht, was jeder einzelne Teil des Programmes macht. In der Einleitung zu jedem Teil wird kurz die Philosophie hinter diesem kurz beschrieben. Das soll es in Zukunft selbst ermöglichen, solche Programme zu schreiben.

Am Ende des Buches finden Sie Vorschläge für Spiele, die Sie selbst verwirklichen können.

Wie man Programme zum Laufen bringt:

Ich habe die Programme absichtlich in der allgemeinsten Basic-Version geschrieben, die ich finden konnte. PEEK und POKE kommen ebenso wenig vor wie Graphiken, Beeps, Töne usw. Ich nahm an, daß Ihr Computer READ und DATA hat, und daß Ihr Bildschirm 32 – 40 Zeichen

pro Zeile hat (der VC-20 hat nicht so viele Zeichen, doch Sie werden sicherlich in der Lage sein, die Programme leicht anzupassen, da die meisten Programme nicht mehr als 12 Zeichen pro Zeile erfordern). Standard String-handling Befehle, wie z.B. Left\$ - Right\$ etc., werden verwendet. Diese müssen dem String-spaltenden System des Spectrum und des TS 2000 angepaßt werden. Ich habe versucht, so wenige String-handling Befehle wie möglich zu verwenden, und die meisten Programme sollten ohne weitere Änderungen funktionieren.

Möglicherweise muß die Bildschirmausgabe etwas adaptiert werden, um die Effizienz des Computers voll auszunützen. Ich nehme an, daß Sie versuchen werden, je nach Gerät Farben und Töne einzubauen. Viele Spiele bauen nach jedem Zug die Spielfläche neu auf, und um den Bildschirm zu löschen, habe ich einen CLS-Befehl eingebaut. Manche Computer brauchen einen Befehl des Typs "CLR", und ich setze voraus, daß Sie den entsprechenden Befehl selbst einbauen. Viel wichtiger jedoch ist, daß der Effekt des Spiels erheblich gesteigert werden kann, nachdem der Bildschirm zu Spielbeginn gelöscht wurde. Man verwende dazu den "HOME" Befehl. Dadurch überschreibt sich das Programm nach jeder Runde selbst. Im Spiel **Roboter-Minenfeld** wirken die Roboter so, als ob sie leben. Dasselbe gilt für Brettspiele. Die Wirkung erhöht sich, und man hat auch mehr Freude an den Spielen. Falls Ihr System eine Möglichkeit haben sollte, den Cursor in die linke, obere Ecke zu positionieren, ohne jedoch den Bildschirm zu löschen (wie z.B. PRINT AT 0,0 etc.), dann vergessen Sie nicht, diesen Befehl anstelle des CLS zu geben. Der Befehl müßte am Anfang des Löschteiles zu finden sein.

Diese Programme wurden auf einem BBC Micro Model B geschrieben. Wenn ich auf diesem Computer Zufallszahlen generieren möchte (z.B. im Bereich von 1 bis 10), dann verwendete ich folgenden Befehl: $A = \text{INT}(\text{RND}(1) * 10) + 1$. Dieses Format wird im ganzen Buch verwendet. Sollte Ihr Computer diesen Befehl nicht verarbeiten, dann substituieren Sie folgendermaßen:

$A = \text{RND}(10)$ oder $A = \text{INT}(\text{RND}(0) * 10) + 1$ oder $A = \text{INT}(\text{RND} * 10) \text{ INT}(\text{RND}(1) * 10) + 1$. (Sehr oft geht diesen Befehlen ein LET voraus). Sie sollten nun in der Lage sein, den richtigen Befehl anzuwenden, falls jedoch Zweifel bestehen, schauen Sie das entsprechende Kapitel im Handbuch an.

Auf dem Computer, auf dem diese Programme entstanden sind, erscheinen bei jeder Programmdurchführung immer dieselben Zufallszahlen. Da zumeist unterschiedliche Zufallszahlen in den einzelnen Programmen notwendig sind, habe ich den Zufallsgenerator manchmal nach der einen oder anderen Methode verbessert.

Dazu verwendete ich INKEY\$ und baute eine Schleife auf, wobei ich gleichzeitig eine Variable addierte und nahm, wenn eine Taste gedrückt wurde, die Variable als Basis für den Zufallsgenerator (wie bei RANDOMIZE N).

Sollte Ihr Computer verschiedene Zufallszahlen erzeugen, dann können Sie diese Befehle vergessen. Sonst wenden Sie am besten RANDOMIZE an, falls Ihr Computer das akzeptiert. Im Zweifelsfalle schlagen Sie im Handbuch nach.

Obwohl ein Großteil der Ausgabe unter Hochkomma Kleinbuchstaben sind, erwartet der Computer die meisten Eingaben in Form von Großbuchstaben. Eine Ausnahme bildet das Schachspiel, wo zur Unterscheidung der Figuren der Gegner Kleinbuchstaben notwendig sind. Falls Ihr Computer keine Kleinbuchstaben haben sollte, dann schreiben Sie die Buchstaben in einem PRINT-Statement einfach groß. Da Kleinbuchstaben besser wirken, habe ich sie in diesem Programm verwendet.

Wenn Sie Programme eintippen, werden Sie merken, daß Sie die notwendigen Adjustierungen wie von selbst vornehmen. Um die Implementierung Ihrer Programme zu beschleunigen, habe ich die obigen Anmerkungen zusätzlich geschrieben. Es könnte jedoch sein, daß Sie sie gar nicht brauchen.

Der Großteil der Programme benötigt ca. 8 K. Das Schachspiel braucht 8 K, die kleineren Spiele 4 K. Sollten Sie das Programm wegen eines zu kleinen Speichers nicht laden können, dann entfernen Sie die REM-Statements (die meisten GOSUB- und GOTO-Befehle sprechen nicht das führende REM an, sondern die eigentliche Routine, die dem REM-Statement folgt) und kürzen Sie die PRINT-Statements. Es ist schwer vorhersehbar, wieviel Speicherplatz Ihr Computer brauchen wird, da jeder Computer einen anderen Aufbau hat.

Um die Länge der Programme zu maximieren, wurden Instruktionen ausgelassen (dadurch wird der Zeitfaktor minimiert). Falls noch Speicherplatz zur Verfügung steht, läßt sich ein Instruktionsteil zusätzlich einbauen. In das Initialisierungsunterprogramm läßt sich leicht eine Zusammenfassung der Bedienungsanleitung implementieren.

Soweit möglich habe ich die strukturierte Programmierung verwendet. Die Erklärung dafür finden Sie in der Einleitung zum Schachspiel. Nachdem ich Artikel und Bücher über den "Top-Down"-Entwurf gelesen hatte, stellte ich fest, daß das Einwickeln von Computerprogrammen einfacher und präziser ging. Weiter ist das Programm leichter zum Laufen zu bringen. Und was das Wichtigste ist: Das Implementieren wird beschleunigt.

Viele Programme beinhalten eine Verzögerungsschleife. Da die eigentliche Verzögerung von dem jeweiligen System abhängt, sollten die Schleifen angepaßt werden, um die Bildschirmanzeige zu optimieren. Die Verzögerungsschleifen befinden sich sehr oft am Ende des Programms.

Zum Abschluß noch ein Wort zu den Variablen. In einigen Fällen wurden ganze Namen als Variablenamen verwendet (z.B. RESULTAT). Signifikant sind jedoch nur die ersten zwei Buchstaben des Namens. Sollte Ihr System nur Variablenamen mit zwei Buchstaben akzeptieren (z.B. RESULT), dann tippen Sie nur die ersten zwei Buchstaben ein.

Zwecks übersichtlicher Programmdokumentation empfiehlt es sich, die Namen auszusprechen.

Ich finde, wir sollten jetzt zu den Spielen übergehen.

Viel Vergnügen!

Tim Hartnell, London, Mai 1983

Brettspiele

Einige Spiele hätte es ohne Computer nicht gegeben.

Man denke nur an die vielen verschiedenen Space-Intruders- und Pac-person-Spiele. Ohne den Computer wären sie nie entstanden. Viele Spiele sind um tausend Jahre älter als der Computer. Und viele Brettspiele wurden computerisiert. Computerspieler haben sich mit ihnen ebenso angefreundet, wie die Spieler der Vorcomputerära.

In diesem Teil des Buches gibt es eine große Anzahl Brettspiele:

*Schach	*Shogun
*Gomoku	*Awari
*Vier-in-einer-Reihe	*Springeraufmarsch
*Dame	*Reversi (Othello)

In diesem Teil des Buches möchte ich die Art und Weise, wie ein Brettspiel leicht programmiert werden kann, besprechen. Abgesehen vom Spiel **Awari** sind die nachfolgenden Absätze auf alle anderen Spiele angewandt worden. Die Methodik zieht sich wie ein roter Faden durch alle Spiele und kann für weitere Programmentwicklungen in diesem Bereich verwendet werden.

Betrachten Sie das Diagramm auf der nächsten Seite. Es zeigt ein Muster, das für den Computer leicht zu behandeln ist.

Man kann jedes einzelne Element entlang der linken Seite ansprechen, wie z.B. 3. Weiter kann man die Elemente der oberen Reihe ebenso ansprechen, wie z.B. 4. In diesem Fall treffen sich die beiden Linien auf Quadrat 34. Möchten Sie eine Figur ziehen (z.B. von Quadrat 55 zu Quadrat 66), dann versteht der Computer sehr wohl, welchen Zug Sie durchführen wollen. Der Satz braucht, um vom Computer verstanden zu werden, nicht mehr geändert zu werden.

Das ist das erste „Geheimnis“, wie man Brettspiele auf dem Computer schreibt. Dieses System hat noch einen Vorteil gegenüber Systemen, bei denen die Felder nur von 1 bis 64 numeriert sind. Zieht man eine Figur, so gibt es keine Unterschiede zwischen den Quadraten, egal in welcher Richtung man sich bewegt. Ich werde versuchen, dieses einigermaßen kryptische Statement zu erklären. Zieht man eine Figur gerade und dann nach rechts – wie beim Damespiel – z.B. von 24 nach 35 oder von 53 nach 64 oder von 71 nach 82, dann ist die Differenz immer 11. Zieht man schräg nach links, dann ist die Differenz immer 9 (von 26 nach 35, von 66 nach 75...). Diese Tatsachen ermöglichen es dem Computer, ein Brettspiel leicht durchzuführen.

	1	2	3	4	5	6	7	8
8	81	82	83	84	85	86	87	88
7	71	72	73	74	75	76	77	78
6	61	62	63	64	65	66	67	68
5	51	52	53	54	55	56	57	58
4	41	42	43	44	45	46	47	48
3	31	32	33	34	35	36	37	38
2	21	22	23	24	25	26	27	28
1	11	12	13	14	15	16	17	18

Wie ein Computer spielt:

Stellen Sie sich vor, der Computer hätte eine Figur in einem Damespiel auf Quadrat 24. Er könnte so programmiert sein, daß er jedes Quadrat der Spielfläche prüft und jedesmal, wenn er eine eigene Figur findet, 11 dazuzählt, um herauszufinden, ob sich eine gegnerische Figur auf dem Quadrat befindet (siehe vorhergehendes Beispiel: 24+11=35). Danach könnte der Computer feststellen, ob das Quadrat 46 leer ist. Trifft das zu, dann könnte der Computer die gegnerische Figur auf Quadrat 35 schlagen.

Das ist kurz zusammengefaßt die Wirkungsweise der meisten Spiele (Schach, Reversi...).

Wenn man ein Schachspiel auf einem 8x8 Brett schreibt, könnte man die Züge des Rössels so definieren, daß man die numerischen Distanzen eingibt: 21 12 -8 -19 -21 -12 19 8

Legen Sie nun eine Münze auf Feld 55 und rechnen Sie sich die mathematischen Relationen zwischen dem Beginnfeld und dem Zielfeld aus. Die Differenzen sollten den obigen entsprechen. Ich habe mich bewußt an die strukturierte Programmierung und an den "Top-Down"-Entwurf gehalten, um die Übersichtlichkeit zu erhöhen. Viele Programme begannen mit folgendem Code:

```

10 REM Name des Spiels
20 GOSUB 9000: REM INITIALISIEREN
30 GOSUB 8000: REM SPIELBRETT DRUCKEN
40 GOSUB 1000: REM COMPUTERZUEGE
50 GOSUB 8000: REM SPIELBRETT DRUCKEN
60 GOSUB 7000: REM SPIELZUEGE
70 prüfen, ob Mensch oder Computer gewonnen hat, wenn ja, Spiel
   beenden
80 GOTO 30

```

In fast allen Fällen habe ich diese Struktur geschrieben, bevor ich noch die geringste Ahnung hatte, wie die Aufgaben der einzelnen Subroutinen gestaltet werden sollten. Alle Programme wurden zuerst komplett auf Papier niedergeschrieben, noch bevor der Computer eingeschaltet wurde. So konnte jedes Spiel zunächst ausprobiert werden, sozusagen im „Trockenlauf“, ohne daß am Computer gearbeitet wurde. Dadurch konnten die größten Fehler noch vor dem eigentlichen Programmieren gefunden und eliminiert werden.

Wenn Sie Programme schreiben, rate ich Ihnen, es ähnlich zu machen. Den gleichen Rat habe ich selbst in einigen Büchern und Zeitungen gelesen und ihn natürlich ignoriert. Als ich jedoch einen zweiwöchigen Urlaub in Wales machte und kilometerweit kein Computergeschäft zu finden war, verspürte ich den Wunsch, ein Schachprogramm zu schreiben. Dabei erkannte ich den Vorteil der oben geschilderten Methode. Das Programm in diesem Buch basiert auf jenem Originalprogramm, das in einem gemieteten Haus an der walisischen Küste auf Papier geschrieben wurde und das während kilometerlanger Spaziergänge mit meinem Hund an der Küste entstand.

Unter den vielen großen Vorteilen, die das Programmieren auf Papier mit sich bringt, ist die Bereitwilligkeit des Programmierers, ganze, nicht funktionierende Programmteile zu entfernen, eine der hervorstechendsten. Denn wenn das Programm einmal in den Computer eingegeben ist, kostet es größere Überwindung, ganze Programmabschnitte zu löschen, als einfach ein Stück Papier zu zerreißen. Man unterliegt vielmehr der Versuchung, die Programmzeilen so zu abzuändern, daß sie funktionieren.

Das Arbeiten auf Papier vermeidet also das Entstehen von unnötigen Programmteilen, die nur Ballast darstellen. Arbeitet man mit einer strukturierten Vorlage, so wie ich es beschrieben habe, kann man sehr leicht erkennen, welche Aufgaben die einzelnen Programmteile haben. Ein Beispiel: Bei der oben erwähnten Struktur weiß ich, daß das Schach-

brett in der mit Zeile 8000 beginnenden Subroutine gezeichnet wird. Sollte diese Zeichnung nun nicht meinen Vorstellungen entsprechen, weiß ich sofort, an welchem Teil des Programms ich noch arbeiten muß. Die Vorteile des strukturierten Programmes machen sich nicht nur in den Beginnstadien des Programmierens bemerkbar, sondern machen auch nachträgliche Verbesserungen wesentlich einfacher. Ein Beispiel: Sie können Ihr selbstgeschriebenes und auch bereits funktionierendes Dameprogramm durch kleine nachträgliche Änderungen, die Sie nur in den sich mit den Zügen des Computers beschäftigenden Programmteil einbauen, viel effizienter machen. Ein strukturiertes Programm macht es einfach, die gewünschten Programmstellen zu finden, und erspart ein langwieriges Durchwühlen des gesamten Programmes. Außerdem sind auch die Funktionen der einzelnen Programmblöcke viel leichter ersichtlich.

Die von mir beschriebenen Methoden werden im ganzen Buch verwendet. Es wäre bestimmt sehr lehrreich, wenn Sie sich einige Programme ansehen und die Aufgaben der einzelnen Programmteile herausfinden würden. Sie werden bemerken, daß ich in vielen Programmen Zeilen mit Sternchen als REM-(Erklärungs- oder Trennungszeilen) Bemerkungen verwende. Die dadurch entstehenden Abschnitte sollen Ihnen helfen, dem Programm zu folgen.

Schach

Schach stellt eine der größten Herausforderungen für einen Programmierer dar. Sowohl die Analyse der menschlichen Spielzüge wie auch die klare und logische Darstellung dieser, welche für eine Verwendung in einem Programm unbedingt nötig ist, stellen eine interessante Aufgabe dar.

Man kann nicht behaupten, daß mein Schachprogramm ein sehr erfolgreicher Versuch ist, diese Herausforderung zu meistern. Obwohl es Schach auf einer guten Basis spielt, spielt es nicht gerade hervorragend und stellt keine allzugroße Herausforderung dar. Warum ist es aber dann mit praktisch unschlagbaren Spielen in diesem Buch?

Ich habe dieses Schachprogramm in mein Buch aufgenommen, weil Schach für jeden Programmierer faszinierend ist. Sehr wenige Schachprogramme sind je veröffentlicht worden (ich weiß nur von einem in "Creative Computing" vom Dezember 1981 unter dem Titel „Schach C4 von Michael Rakaska, zwei anderen in Büchern und einer Veröffentlichung in Maschinensprache "Sargon II"). Diese Tatsache zeigt, wie schwer es ist, ein solches Programm zu erstellen. Die Rarität der veröffentlichten Schachprogramme sprach für das Abdrucken meines Programmes in diesem Buch.

Das menschliche Gehirn beschäftigte sich für Hunderte von Jahren mit einer schachspielenden Maschine. So ein Automat, von Baron von Kempelen gebaut, hatte seine erste öffentliche Präsentation bei einem Spiel vor dem Königshaus in Wien im Jahre 1770 und begeisterte fast ganz Europa. Unglücklicherweise war die ganze Maschine nur ein Trick. Ein in einem Versteck sitzender Mann bediente den schachspielenden Mechanismus. Es war ein sehr erfolgreicher Betrug. Der Nachvollzug der Züge zeigt, daß der Mann wirklich gut schachspielen konnte. Als ich im Britischen Museum etwas in der Geschichte von Spielen erforschte, fand ich in der Bibliothek ein Buch aus dem letzten Jahrhundert, "50 Games Played by the Chess Automaton", das ein eifriger Leser mit Anmerkungen versehen hatte. Die Menschen wurden mit kritischeren Kommentaren bedacht – so z.B. „dieser Zug stempelt den Spieler zu einem blutigen Anfänger!“ – als die Maschine.

Entgegen diesem Ruf konnte die Maschine nie richtig schachspielen. Der erste wirkliche Versuch, eine Maschine zu bauen, die eine komplette Schachpartie spielen konnte (zu Beginn dieses Jahrhunderts wurden Maschinen gebaut, die bestimmte Partien zu Ende spielen konnten), war im Jahr 1949. Am 9. März präsentierte Claude Shannon, ein Forschungsarbeiter der Bell Telefon Laboratorien, Murray Hill, New Jersey, in einer New Yorker Versammlung einen Artikel mit dem Titel "Programming a Computer for Playing Chess".

Wie schon David Levy in seinem faszinierenden Buch "Chess and Computers" (Computer Science Press, Inc., Potomac, Maryland, 1976) aufzeigt, liegt die wirkliche Bedeutung von Shannons Artikel nicht nur darin, daß er der erste war, sondern auch darin, daß viele Ideen Shannons auch in heutigen Programmen noch gefunden werden.

In diesem Artikel sagt Shannon, daß die Anzahl der möglichen Spielbewegungen in einem 40-Züge-Spiel die Zahl 10 zur 120ten Potenz erreicht. Diese Zahl ist von der gleichen Größenkategorie wie die Anzahl der Atome im Universum. Es ist ganz klar, daß ein Programm, das alle diese Züge in Erwägung zöge, sehr lange brauchen würde (sagen wir 10 zur 90ten Potenz Jahre), bevor der erste Bauer seinen ersten Zug zur Brettmitte machen würde.

Dieses Programm schneller spielen zu lassen, erfordert – wie Sie sich sicher vorstellen werden können – eine Anzahl von Kompromissen. Ich tauschte einige langwierige Bewertungen gegen einige trockene und vorgefertigte Spielzüge (wie z.B. den Computer so zu programmieren, daß er Figuren bewegt, um das Brettzentrum so gut wie möglich zu halten; Springer so früh wie möglich in einem Spiel zu aktivieren und um einen Bauern unabhängig von der dem Computer drohenden Gefahr immer zu schlagen. Der König wird so wenig wie möglich bewegt, zumindest in den Beginnphasen des Spieles, wenn ein langsamer und vorsichtiger Anfang sinnvoll erscheint. Figuren mit großer Mobilität – wie der Königin – habe ich einen zufallsgesteuerten Entscheidungsfindungsmechanismus gegeben, der sie zurückhält, quer über das Brett zu fahren, nur weil sie es könnten).

Durch das ganze Programm folgte ich einer klaren Struktur, die den Zugabtausch in der kürzestmöglichen Zeit gestattet. Leider ist die Struktur nicht so transparent, wie ich es mir beim Schreiben des Programmes erhofft hatte. Meine Pläne für ein klares Programm mußte ich angesichts der beim Programmieren auftretenden Schwierigkeiten über Bord werfen. Trotzdem, das Programm spielt relativ schnell, und wenn es auch nicht sehr gut spielt, so stellt es doch einen Gegner dar, der einige Tricks auf Lager hat und Sie immer amüsieren und reizen wird.

Am Ende der Listing der Programmzeilen ist eine Anweisung zur Abänderung des Programmes, so daß es gegen sich selbst spielt. Dies ist eine faszinierende Programmdemonstration und wird insbesondere die Schachexperten unter Ihnen fesseln. Ich versuchte dies einmal mit eingeschaltetem Drucker. Vor der Programmlistung zeige ich Ihnen das Ergebnis, damit Sie selbst beurteilen können, was für ein Gegner der Computer ist.

Diese Automatikschachvariante verwendet die Seitenaustauschoption während des Spieles, die die Seiten nach jedem Zug austauscht. Wenn Sie die Seiten tauschen, wobei Sie dem Computer die Seite geben, auf der Sie spielten, werden die Figuren so getauscht, als ob sie in einem Spiegel auf der Brettmitte abgebildet werden. Das bedeutet, daß eine Königin, die die Seite tauscht, auf das Feld ihrer Farbe kommt, usw. Versuchen Sie den Austausch einige Male während Sie spielen, und das Verfahren dieses Mechanismus wird Ihnen klar werden.

Noch ein Wort, bevor wir zum Demonstrationsprogramm kommen. Beobachten Sie den Computer, damit er nicht falsch spielt! Er wird es nur sehr selten tun. Sollte er aber z.B. sich selbst in Schach begeben, werten Sie es als Aufgabe seitens des Computers. Die Maschine hat gewisse Schwierigkeiten, sich aus dem Zustand des Schach herauszu-mannövrieren. Hier sollten Sie etwas toleranter sein. Die Maschine schafft es vielleicht schon beim nächsten Zug. Jedenfalls können Sie einen solchen Fehler des Computers als Anerkennung Ihrer Überlegenheit werten.

Vor dem Listing zeige ich Ihnen noch einige Stellungen aus dem Spiel, das der Computer gegen sich selbst spielte. Der Computer wechselte die Seiten nach jedem Zug und druckte das Brett nach jedem zweiten Zug.

A B C D E F G H	A B C D E F G H
8 R N B Q K B N R 8	8 R N B Q K B N R 8
7 P P P P P P P 7	7 P P P P P P P 7
6 6	6 6
5 5	5 5
4 4	4 4
3 3	3 3
2 P P P P P P P 2	2 P P P P P P P 2
1 r n b q k b n r 1	1 r n b q k b n r 1
A B C D E F G H	A B C D E F G H

A B C D E F G H	A B C D E F G H
8 R N B Q K B N R 8	8 R N B Q K B N R 8
7 P P P P P P P 7	7 P P P P P P P 7
6 6	6 6
5 5	5 5
4 4	4 4
3 3	3 3
2 P P P P P P P 2	2 P P P P P P P 2
1 r n b q k b n r 1	1 r n b q k b n r 1
A B C D E F G H	A B C D E F G H

A B C D E F G H	A B C D E F G H
8 R N B Q K B N R 8	8 R N B Q K B N R 8
7 P P . . . P P 7	7 P . . . P P 7
6 6	6 6
5 . . P P P P . 5	5 . P P P P . 5
4 4	4 4
3 . P P P P . 3	3 . P P P P . 3
2 P . . . P P P 2	2 P . . . P P P 2
1 r n b q k b n r 1	1 r n b q k b n r 1
A B C D E F G H	A B C D E F G H

Hier ist das Listing des Schachspiels:

```

10 REM Schach
20 GOSUB2970
30 GOTO40
40 GOSUB2580
50 GOSUB2820
60 GOSUB2580
70 REM *****
80 IF A$="E" THEN END
90 IF A$="X" THEN PRINT"SEITENWECHSEL":
GOSUB3540:A$=""
100 IF A$="D" THEN GOSUB3730
110 REM *****
120 FOR Z=1 TO 16:T(Z)=0:NEXT Z
130 U=0
140 PRINT"Bitte warten Sie"
150 FOR Q=1 TO 64:IF A(S(Q))>=BB AND A(S
(Q))<=RB THEN U=U+1:T(U)=S(Q):IF A(S(Q))=K
B KM=S(Q)
160 NEXT Q:IF U<3 THEN GOTO2230
170 GOTO650
180 FOR Q=1 TO U:IF A(T(Q))=KB THEN T(Q)
=T(U):T(U)=KM
190 NEXT Q
200 Q=INT(RND(1)*3)
210 IF A$="S" THEN Q=0
220 IF Q<U THEN Q=Q+1
230 Z=T(Q):GOSUB280
240 IF MM=1 THEN GOSUB2500:GOTO40
250 IF Q<U THEN GOTO220
260 GOTO2360
270 REM *****
280 IF A(Z)=QB THEN GOSUB910
290 IF A(Z)=RB THEN GOSUB1170
300 IF A(Z)=BB THEN GOSUB1420
310 IF A(Z)=NB THEN GOSUB1690
320 IF A(Z)=PB THEN GOSUB2240
330 RETURN
340 REM *****

```

```

350 IF A(X)=107 THEN MM=0:Q=Q+1:RETURN
360 IF X+9>88 THEN GOTO380
370 IF A(X+9)<83 AND A(X+9)>65 AND RND(1
)<.96 THEN RETURN
380 IF X-11<11 THEN GOTO400
390 IF A(X-11)<83 AND A(X-11)>65 AND RND
<.96 THEN RETURN
400 AD=0
410 AY=0
420 AX=X+Q(AY+AD)
430 IF AX<11 OR AX>88 THEN GOTO460
440 AP=A(AX)
450 IF AP=Q OR AP=R AND RND(1)>.8 OR AP=
B AND RND(1)>.5 THEN RETURN
460 AY=AY+1
470 IF AY<8 THEN GOTO420
480 AD=AD+7
490 IF AD<56 THEN GOTO410
500 AY=1
510 AX=X+N(AY)
520 IF AX<11 OR AX>88 THEN GOTO540
530 IF A(AX)=N THEN RETURN
540 AY=AY+1
550 IF AY<9 THEN GOTO510
560 AY=1
570 AX=X+K(AY)
580 IF AX<11 OR AX>88 THEN GOTO600
590 IF (A(AX)=K OR A(AX)=P) AND RND(1)>.
1 THEN RETURN
600 AY=AY+1
610 IF AY<9 THEN GOTO570
620 MM=1
630 RETURN
640 REM *****
650 Z=KM
660 Y=0
670 Y=Y+1
680 X=Z+N(Y)
690 IF X<11 OR X>88 THEN GOTO600

```



```

700 IF A(X)=N THEN GOTO1870
710 IF Y<8 THEN GOTO670
720 REM *****
730 D=0
740 Y=1
750 X=Z+Q(Y+D)
760 IF X<11 OR X>88 THEN GOTO810
770 IF A(X)=B OR A(X)=Q OR A(X)=R THEN G
GOTO1870
780 IF A(X)<>E THEN GOTO810
790 Y=Y+1
800 IF Y<8 THEN GOTO750
810 D=D+7
820 IF D<49 THEN GOTO750
830 X=Z+11
840 IF X>88 THEN GOTO860
850 IF A(X)=P THEN GOTO1870
860 X=Z-11
870 IF X<11 THEN GOTO180
880 IF A(X)=P THEN GOTO1870
890 GOTO180
900 REM *****
910 D=0
920 Y=1
930 X=Z+Q(Y+D)
940 IF X<11 OR X>88 THEN GOTO1000
950 IF A(X)=42 OR A(X)=8B AND A(X)<=RB
THEN GOTO1000
960 IF A(X)=B AND A(X)<=P THEN GOSUB350
:IF MM<>1 THEN GOTO1000
970 IF MM=1 THEN RETURN
980 Y=Y+1
990 IF Y<7 THEN GOTO930
1000 D=D+7
1010 IF D<42 THEN GOTO920
1020 RETURN
1030 REM *****
1040 D=0
1050 Y=1

```

```

1060 X=Z+Q(Y+D)
1070 IF X<11 OR X>88 THEN GOTO1130
1080 IF A(X)<>E THEN GOTO1130
1090 IF RND(1)>.5 THEN GOSUB350:IF MM=0 T
HEN GOTO1130
1100 IF MM=1 THEN RETURN
1110 Y=Y+1
1120 IF Y<8 THEN GOTO1060
1130 D=D+7
1140 IF D<49 THEN GOTO1050
1150 RETURN
1160 REM *****
1170 D=0
1180 Y=1
1190 X=Z+R(Y+D)
1200 IF X<11 OR X>88 THEN GOTO1240
1210 IF A(X)=42 OR A(X)=8B AND A(X)<=RP
THEN GOTO1690
1220 IF A(X)=B AND A(X)<=P THEN GOSUB350
:IF MM=0 THEN GOTO1240
1230 IF MM=1 THEN RETURN
1240 Y=Y+1
1250 IF Y<7 THEN GOTO1190
1260 D=D+7
1270 IF D<21 THEN GOTO1180
1280 RETURN
1290 REM *****
1300 D=0
1310 Y=1
1320 X=Z+R(Y+D)
1330 IF X<11 OR X>88 THEN GOTO1390
1340 IF A(X)<>E THEN GOTO1390
1350 IF RND(1)<.1 THEN GOSUB350
1360 IF MM=1 THEN RETURN
1370 Y=Y+1
1380 IF Y<7 THEN GOTO1320
1390 D=D+7
1400 IF D<21 THEN GOTO1310
1410 RETURN

```

```

1420 IF A(Z)<>BB THEN RETURN
1430 D=0
1440 Y=1
1450 X=Z+B(Y+D)
1460 IF X<11 OR X>88 THEN GOTO1520
1470 IF A(X)=42 OR A(X)=BB AND A(X)<=RB
THEN GOTO1520
1480 IF A(X)=B AND A(X)<=R THEN GOSUB350
:IF MM<>1 THEN GOTO1520
1490 IF MM=1 THEN RETURN
1500 Y=Y+1
1510 IF Y<7 THEN GOTO1450
1520 D=D+7
1530 IF D<21 THEN GOTO1440
1540 RETURN
1550 REM *****
1560 D=0
1570 Y=1
1580 X=Z+B(Y+D)
1590 IF X<11 OR X>88 THEN GOTO1650
1600 IF A(X)<>E THEN GOTO1650
1610 IF RND>.05 THEN GOSUB350:IF MM<>1 TH
EN GOTO1650
1620 IF MM=1 THEN RETURN
1630 Y=Y+1
1640 IF Y<7 THEN GOTO1580
1650 D=D+7
1660 IF D<21 THEN GOTO1570
1670 RETURN
1680 REM *****
1690 Y=1
1700 X=Z+N(Y)
1710 IF X<11 OR X>88 THEN GOTO1750
1720 IF A(X)=42 THEN GOTO1750
1730 IF A(X)=B AND A(X)<=R THEN GOSUB350
1740 IF MM=1 THEN RETURN
1750 Y=Y+1
1760 IF Y<9 THEN GOTO1700
1770 RETURN

```

```

1780 REM *****
1790 Y=0
1800 X=Z+N(INT(RND(1)*8+1))
1810 IF X<11 OR X>88 THEN GOTO1800
1820 IF A(X)=42 THEN GOTO1800
1830 Y=Y+1
1840 IF A(X)=E THEN GOSUB350
1850 IF MM=1 OR Y>20 THEN RETURN
1860 GOTO1800
1870 YK=1
1880 Z=KM
1890 X=Z+K(YK):X1=X
1900 IF X<11 OR X>88 THEN GOTO2200
1910 IF A(X)=42 OR A(X)>65 AND A(X)<83 TH
EN GOTO2200
1920 IF A(X)>97 AND A(X)<115 THEN GOTO2200
1930 Z=X
1940 Y=0
1950 Y=Y+1
1960 X=Z+N(Y)
1970 IF X<11 OR X>88 THEN GOTO1990
1980 IF A(X)=N THEN GOTO2200
1990 IF Y<8 THEN GOTO1950
2000 REM *****
2010 D=0
2020 Y=1
2030 X=Z+B(Y+D)
2040 IF X<11 OR X>88 THEN GOTO2090
2050 IF A(X)=B OR A(X)=0 OR A(X)=P THEN G
OTO2200
2060 IF A(X)<>E THEN GOTO2090
2070 Y=Y+1
2080 IF Y<8 THEN GOTO2030
2090 D=D+7
2100 IF D<49 THEN GOTO2030
2110 X=Z+11
2120 IF X>88 THEN GOTO2140
2130 IF A(X)=P THEN GOTO2200
2140 X=Z-11

```

```

2150 IF X<11 THEN GOTO2170
2160 IF A(X)=P THEN GOTO2200
2170 X=X1:Z=KM
2180 MM=1
2190 GOSUB2500:GOTO40
2200 YK=YK+1
2210 Z=KM
2220 IF YK<9 THEN GOTO1880
2230 PRINT"Ich gebe auf, Meister!":END
2240 X=Z+9
2250 IF A(X)>=B AND A(X)<=R THEN MM=1:IF
A(X)=P AND RND(1)<.2 THEN MM=0
2260 IF MM=1 THEN RETURN
2270 IF Z=12 THEN RETURN
2280 X=Z-11
2290 IF A(X)>=B AND A(X)<=R THEN MM=1:IF
A(X)=P AND RND(1)<.2 THEN MM=0
2300 RETURN
2310 REM *****
2320 IF Z-10*(INT(Z/10))=7 AND A(Z-1)=E A
ND A(Z-2)=E AND (A(Z-13)=E OR A(Z-13)=42)
AND (A(Z+7)=E OR A(Z+7)=42) THEN X=Z-2:MM=
1:RETURN
2330 IF A(Z-1)=E AND A(Z-12)<98 AND A(Z+8
)<98 THEN X=Z-1:MM=1:RETURN
2340 IF RND(1)<.05 AND A(Z-1)=E THEN X=Z-
1:MM=1
2350 RETURN
2360 Q=INT(RND(1)*RND(1)*5):IF Q>U THEN G
OTO2360
2370 IF Q<U THEN Q=Q+1
2380 Z=T(Q)
2390 IF A(Z)=PB THEN GOSUB2320
2400 IF A(Z)=NB THEN GOSUB1790
2410 IF A(Z)=BB THEN GOSUB1560
2420 IF A(Z)=RB THEN GOSUB1300
2430 IF A(Z)=QB THEN GOSUB1040
2440 IF A(Z)=KB AND A<>"C" AND RND(1)<.0
7 THEN GOSUB1870

```

```

2450 IF MM=0 AND Q<U THEN GOTO2370
2460 IF MM=1 THEN GOSUB2500:GOTO40
2470 UK=UK+1:IF UK>8 THEN GOTO2230
2480 GOTO2360
2490 REM *****
2500 IF A(Z)=KB AND A<>"S" AND RND(1)>.1
THEN MM=0:GOTO2360
2510 IF A(Z)=PB AND ((X-10*INT(X/10))>Z-10
*INT(Z/10) OR ABS(X-Z)>11) THEN MM=0:U=U+
1:GOTO2360
2520 IF A(X)=K THEN PRINT"SCHACH":MM=0:U=
U+1:GOTO2360
2530 A(X)=A(Z):A(Z)=E
2540 PRINT"Ich fahre von ";
2550 FZ=INT(Z/10):PRINT CHR$(FZ+64):Z-10*
FZ;" nach ";FX=INT(X/10):PRINT CHR$(FX+64
):X-10*FX:FOR Q=1 TO 3000:NEXT Q
2560 RETURN
2570 REM *****
2580 CLS
2590 GOSUB2670
2600 FOR X=8 TO 1 STEP -1
2610 PRINT TAB(10);X;" ";
2620 FOR Y=10 TO 80 STEP 10
2630 IF A(Y+1)=PB THEN A(Y+1)=QB
2640 IF A(Y+8)=P THEN A(Y+8)=Q
2650 PRINT CHR$(A(X+Y)):" ";
2660 NEXT Y:PRINT:X:NEXT X:MM=0
2670 PRINT:PRINT TAB(12);"A B C D E F G H
":PRINT
2680 REM *****
2690 RETURN
2700 Z=KM
2710 QK=0
2720 M=Z+K(QK)
2730 IF M<11 OR M>88 THEN GOTO2780
2740 IF A(M)=42 OR A(M)>65 AND A(M)<83 OR
MM=0 THEN GOTO2780
2750 X=M

```

```

2760 KM=Z
2770 RETURN
2780 IF OK<8 THEN GOTO2720
2790 IF A$(<)"S" THEN RETURN
2800 GOTO2230
2810 REM *****
2820 PRINT
2830 INPUT"VON (BUCHST,ZAHL) ":A$
2840 IF LEN(A$)<2 THEN GOTO2820
2850 INPUT "NACH ":B$
2860 IF LEN(B$)<2 THEN GOTO2850
2870 X=10*(ASC(A$)-64)+VAL(RIGHT$(A$,1))
2880 Y=10*(ASC(B$)-64)+VAL(RIGHT$(B$,1))
2890 PRINT"Geben Sie ein: S - Schach"
2900 PRINT"          D - Spielbrett
drucken"
2910 PRINT"          X - Seitenwechs
el"
2920 PRINT"          E - Spielende"
2930 INPUT "oder RETURN fuer weiter ":A$
2940 IF A(Y)>75 AND A(Y)<=92 THEN GOSUB3
660
2950 A(Y)=A(X):A(X)=46:RETURN
2960 REM *****
2970 CLS:PRINT"BITTE SCHALTEN SIE CAPS LO
CK EIN,":PRINT"DANN DRUECKEN SIE RETURN"
2980 N=N+1:IF INKEY$(10)=" " THEN GOTO2980
2990 CLS:PRINT"Bitte warten Sie":REM RAND
OMIZE N
3000 REM
3010 MM=0:A$=""
3020 DIM A(99),R(28),B(28),N(8),O(56),K(8
),Z(88),S(64),T(16)
3030 P=112:R=114:N=110:B=98:O=113:K=107:E
=46
3040 PB=80:RB=82:NB=87:BB=64:OB=81:KB=75
3050 FOR Z=1 TO 99:A(Z)=-99:NEXT Z
3060 REM *****
3070 FOR Z=1 TO 64:READ X,Y:A(X)=Y:NEXT Z

```

```

3080 DATA 18,82,28,78,38,66,48,81
3090 DATA 58,75,68,66,78,78,88,82
3100 DATA 17,80,27,80,37,80,47,80
3110 DATA 57,80,67,80,77,80,87,80
3120 DATA 16,46,26,46,36,46,46,46
3130 DATA 56,46,66,46,76,46,86,46
3140 DATA 15,46,25,46,35,46,45,46
3150 DATA 55,46,65,46,75,46,85,46
3160 DATA 14,46,24,46,34,46,44,46
3170 DATA 54,46,64,46,74,46,84,46
3180 DATA 13,46,23,46,33,46,43,46
3190 DATA 53,46,63,46,73,46,83,46
3200 DATA 12,112,22,112,32,112,42,112
3210 DATA 52,112,62,112,72,112,82,112
3220 DATA 11,114,21,110,31,98,41,113
3230 DATA 51,107,61,98,71,110,81,114
3240 REM *****
3250 RESTORE 3270
3260 FOR Z=1 TO 8:READ N(Z):NEXT Z
3270 DATA 19,-19,21,-21,-8,8,12,-12
3280 FOR Z=1 TO 28:READ R(Z):NEXT Z
3290 DATA 10,20,30,40,50,50,50
3300 DATA -1,-2,-3,-4,-5,-5,-5
3310 DATA -10,-20,-30,-40,-50,-50,-50
3320 DATA 1,2,3,4,5,5,5
3330 RESTORE 3350
3340 FOR Z=1 TO 28:READ B(Z):NEXT Z
3350 DATA -11,-22,-33,-44,-55,-55,-55
3360 DATA 11,22,33,44,55,55,55
3370 DATA 9,18,27,36,45,45,45
3380 DATA -9,-18,-27,-36,-45,-45,-45
3390 RESTORE 3290
3400 FOR Z=1 TO 56:READ O(Z):NEXT Z
3410 FOR Z=1 TO 8:READ K(Z):NEXT Z
3420 DATA 1,11,9,10,-10,-9,-11,-1
3430 FOR Z=1 TO 64:READ S(Z):NEXT Z
3440 DATA 46,56,36,66,47,57,45,55
3450 DATA 37,67,35,65,28,78,27,77
3460 DATA 44,54,26,76,38,68,17,87

```

```

3470 DATA 18,88,34,64,25,75,16,86
3480 DATA 48,24,74,15,85,14,84,43
3490 DATA 53,33,62,23,73,52,42,62
3500 DATA 32,83,13,72,22,12,82,41
3510 DATA 51,31,61,21,71,11,81,58
3520 CLS:RETURN
3530 REM *****
3540 FOR Z=11 TO 88:Z(Z)=A(Z):NEXT Z
3550 FOR Z=11 TO 88:X=Z-10*INT(Z/10)
3560 IF X=0 OR X=9 THEN GOTO3580
3570 A(Z)=Z(Z+9-X*2)
3580 NEXT Z
3590 FOR Z=11 TO 88:M=A(Z)
3600 IF M>=8 THEN A(Z)=A(Z)+PB-P
3610 IF M<=8 AND M>=BB THEN A(Z)=A(Z)-PB
+P
3620 NEXT Z
3630 GOSUB2580
3640 RETURN
3650 REM *****
3660 CM=INT(RND(1)*4)+1
3670 ON CM GOSUB 3690,3700,3710,3720
3680 FOR J=1 TO1000:NEXT J:RETURN
3690 PRINT"Bravo!":RETURN
3700 PRINT"Guter Zug!":RETURN
3710 PRINT"Superzug Meister!":RETURN
3720 PRINT"Sie haben mich erwischt...":RE
TURN
3730 LPRINT "*****
*****"
3740 GOSUB3800
3750 FOR X=8 TO 1 STEP-1
3760 LPRINT TAB(5):X:" ";
3770 FOR Y=10 TO 80 STEP 10
3780 LPRINT CHR$(A(X+Y)):" ";
3790 NEXT Y:LPRINT X:NEXT X
3800 LPRINT:LPRINT TAB(9):"A B C D E F G
H":LPRINT
3810 RETURN

```

Wenn Sie wollen, daß der Computer gegen sich selbst spielt, ändern Sie Zeile 2580 wie folgt:

```
2580 CLS:PRINT"Spieler ";SP;" von oben"
```

...und ändern Sie die ersten zehn Zeilen des Programmes folgendermaßen:

```

15 SP=1
20 GOSUB2970
30 GOTO60
40 GOSUB2580
50 A$="X"
60 IF SP=2 THEN SP=1:GOTO80
70 SP=2
80 IF A$="S" THEN END
90 IF A$="X" THEN PRINT "SEITENWECHSEL"
:GOSUB3540:A$=""
100 IF SP=2 THEN GOSUB3730

```

Wenn Sie wollen, daß der Computer den Drucker während des Spieles ab und zu einschaltet, um Ihnen Momentaufnahmen des Spieles zu liefern, fügen Sie folgende Zeilen hinzu:

```

3730 IF RND(1)>.08 THEN RETURN
3735 LP. "*****
*****"
3740 GOSUB3800

```

Wie das Programm arbeitet:

Um zu wissen, welche Teile Sie betrachten müssen, wenn Sie das Programm verbessern wollen, müssen wir uns zuerst einige Abschnitte des Schachprogrammes ansehen.

Wie ich am Beginn des Buches erklärte, beginnt das Programm mit einer Serie von Subroutinen. Diese werden im Zuge des Spieles nach und nach durchlaufen. Nach der ersten REM-Anweisung, die das Programm identifizierte, geht der Computer zu der Subroutine von Zeile 2970, die die Variablen initialisiert. Danach springt Zeile 30 zu Zeile 60, die die Subroutine von Zeile 2580, die das Schachbrett druckt, aufruft. Wie alle anderen Routinen in diesem Kapitel, die ein Spielbrett drucken, beginnt diese Routine mit dem Befehl, den Bildschirm zu löschen (CLS). Sie sollten jedoch CLS-Kommandos durch einen HOME-Befehl oder

einen ähnlichen Befehl (wie z.B. PRINT AT 0, PRINT AT 0,0 oder LOCATE 1,1) ersetzen. Dann wird nämlich einfach das Brett über das alte Bild gedruckt. Dieses ergibt einen sehr anschaulichen Eindruck, wie sich die Figuren bewegen. Das ist viel beeindruckender, als den Bildschirm jedes Mal zu löschen und das Brett neu drucken zu lassen.

Nachdem das Brett gedruckt wurde, fährt das Programm nach Zeile 60 mit den Zügen des Computers fort. Zunächst wird A\$, eine Stringvariable, die vom Spieler nach jedem Zug neu definiert wird, auf ihren Status überprüft. Ist A\$ "S", so hat der Spieler seinen Wunsch nach der Beendigung des Programmes geäußert, und das Programm endet. Ist A\$ jedoch "X" (für Austausch), springt das Programm zu der Subroutine in Zeile 3540, um die Seiten zu tauschen. Wenn A\$ aber "P" ist, so fährt das Programm mit der Subroutine von Zeile 3730 fort, und die Stellung wird auf dem Drucker ausgedruckt.

Doch nun wird es mit dem Spielen ernst. Zuerst wird das T-Feld (das die Stellung der Spielfiguren des Computers beinhaltet) mit Nullen gefüllt, und die Figurenzählvariable (U) wird auf Null gesetzt. Die Schleife der Zeilen 150 und 160 fragt jedes Quadrat auf dem Brett ab, zählt die Figuren (U wird immer erhöht, wenn eine Figur gefunden wird) und vermerkt deren Standort in den Elementen des T-Feldes. Wird der König gefunden (Variable KB, für "king black"), so wird dieser Variablen die Variable KM (für "king marker") gleichgesetzt. So weiß der Computer zu jeder Zeit, wo sein König ist. Ist U kleiner als drei (Zeile 160), so geht der Computer zu Zeile 2230 und gibt das Spiel auf. Er hat keine Endspielstrategie und ist daher hilflos, wenn er zu wenig Figuren hat.

In Zeile 170 springt das Programm zu 650, wo eine lange Routine prüft, ob der König im Schach steht. Jede Sektion dieses Programmteiles überprüft die Gefahr, die von einem bestimmten Spielstein drohen könnte. Der erste Teil von 650 bis 710 überprüft die Gefahr, die von den Springern drohen könnte. Die nächsten Zeilen, 730 bis 810, halten nach hinterlistigen Königinnen, Läufern, Türmen und Bauern Ausschau. Wurde keine Bedrohung gefunden, so wird das Programm durch die zahlreichen GOTO's in diesem Teil nicht umdirigiert, und Zeile 890 sendet es zurück zu 180, wo ein Zug erzeugt wird.

Die Zeilen 180 und 190 verändern den Inhalt des T-Feldes so, daß der König das letzte Element in diesem ist. Im allgemeinen bewegt der Computer die Figuren unabhängig von ihrer Stellung in diesem Feld. Ist der König also ganz am Ende dieses Feldes, wird ihn der Computer nur dann bewegen, wenn er feststellt, daß es keinen anderen, wichtigeren Zug gibt. Wie ich schon vor dem Programmlisting erwähnte, ist dies eines meiner groben und vorgefertigten Spielprinzipien. Es basiert auf dem Prinzip, daß es unwahrscheinlich für den König ist, sich in Gefahr zu begeben, wenn er sich so wenig wie möglich bewegt. Das stimmt natürlich nicht immer. Es stellt jedoch für die Art von Schach, die das Programm spielt, eine annehmbare Verallgemeinerung dar. Sicherlich ist sie weniger gefährlich als die Alternative, bei der der König nur so herumflitzen und sich dadurch ständig in Gefahr begeben würde.

In Zeile 210 wird nun Q entweder auf 0, 1 oder 2 gesetzt oder auch in Zeile 210 auf 0 gestellt, wenn der menschliche Spieler "S" für Schach anzeigt. Q bestimmt, wo im T-Feld der Computer nach einem Zug zu suchen beginnt. Hier wird das Zufallsmoment ins Spiel gebracht, so daß der Computer nicht immer dasselbe tut, wenn er mit ähnlichen Situationen konfrontiert wird. Q wird um eins erhöht, wenn Q kleiner als U (das ist die gesamte Anzahl der Figuren, die der Computer noch hat) ist. Danach wird Z gleich dem Element des T-Feldes gesetzt, das der Ort der Computerfigur auf dem Brett ist.

Nun wird eine Subroutine von Zeile 280 aufgerufen. Wenn Sie sich die Zeilen 280 bis 320 ansehen, werden Sie bemerken, daß diese bestimmen, mit welcher Figur sie sich beschäftigt. Der Computer spielt schwarz. Um dem Ablauf des Listings besser folgen zu können: QB steht für die Königin (queen black), RB ist der Turm, BB der Läufer, NB der Springer und PB der Bauer. In dieser kleinen Subroutine findet der Computer also jene Figuren, die er selber noch hat, und fährt dann in den relevanten Subroutinen fort, um ein mögliches Schlagen mit der jeweiligen Figur zu ermitteln.

Jetzt sollten Sie schon eine Ahnung davon haben, wie das Programm arbeitet. Grob gesprochen sucht es seine Figuren, wenn kein Schach gegeben wurde, reiht die Figuren durch die auf dem Brett bestimmte Anordnung auf (das wird am Ende der Programmlisting erklärt werden), setzt den König an das Ende dieser Reihe und sucht dann nach der Möglichkeit, jede einzelne Figur zu schlagen.

Die wichtigste Variable (nach KM, der Königsmarkierung) ist MM (steht für Maschinenzug). Sie werden MM in Zeile 240 finden. Zu Beginn wird MM auf Null gesetzt, doch wenn ein Zug gefunden wurde, wird sie eins. Zeile 240 überprüft MM, und wenn sie eins ist, weiß der Computer, daß ein Zug gefunden wurde, und springt zu der Subroutine von Zeile 2500. Hier untersucht er, ob er (a) nicht versucht, seinen König ohne wirklichen Anlaß zu bewegen; (b) nicht einen Bauern äußerst unintelligent bewegt; (c) nicht gerade dabei ist, den gegnerischen König zu schlagen. Jeder dieser drei Fälle bewirkt, daß MM auf Null zurückgesetzt und in Zeile 230 ein neuer Zug gesucht wird. Findet der Computer Fall 'C', druckt er SCHACH auf den Bildschirm und sucht sodann nach seinem Zug.

Findet er diese drei Fälle nicht vor, macht er den eigentlichen Zug (Zeile 2530 nimmt die nötigen Veränderungen im A-Feld, jenem Feld, das die jeweilige Spielkonstellation beinhaltet, vor). Danach teilt er Ihnen durch die Zeilen 2540 und 2550 seinen Zug mit und setzt das Programm mit dem Neudrucken des Brettes fort.

Ist jedoch MM nicht gleich eins, wenn der Computer zu Zeile 240 kommt, überprüft er in Zeile 250, ob er alle Figuren benützt hat (was der Fall sein wird, wenn Q gleich U ist). Findet er, daß er nicht alle verwendet hat, geht er auf Zeile 220 zurück, Q wird um eins erhöht, und der ganze Prozeß beginnt von Neuem. Ist Q gleich U, erkennt der Computer, daß er keinen legalen Zug machen kann, und springt zu 2360, um einen Zug zu suchen, bei dem er nicht schlägt. Der nächste Programmteil

sendet, wie wir schon besprochen, den Programmablauf zu Kontrollabschnitten der einzelnen Figuren.

Betrachten wir den Abschnitt nach Zeile 2360, so sehen wir, daß Q zunächst einer Zufallszahl zwischen Null und Vier zugeordnet wird. Die doppelte Verwendung von RND bewirkt, daß eher niedere Nummern innerhalb dieser Bandbreite gewählt werden. Z wird dem Qten Element des T-Feldes gleichgesetzt (wie schon zuvor, als wir nach „schlagenden“ Zügen gesucht haben). Dann senden die Zeilen von 2390 bis 2440 den Computer zu jenen Programmteilen, die die Züge der einzelnen Figuren kontrollieren.

Da alle diese Abschnitte nach einem ähnlichen Schema arbeiten, werden wir nicht jeden einzelnen betrachten. Wenn Sie das Prinzip von einem verstanden haben, können Sie sehr leicht ableiten, wie die anderen arbeiten.

Nachdem wir schon zu Beginn dieses Kapitels in diesem Buch über die Art und Weise, wie sich Springer bewegen, gesprochen haben, werden wir uns den Teil, der die Springer steuert, genauer betrachten. Sehen Sie sich den Programmabschnitt nach Zeile 1790 an. Wenn Sie den Initialisierungsteil des Programmes betrachten, werden Sie sehen, daß für jede Figur ein Feld errichtet wurde, das die möglichen Züge für diese Figur enthält (in Form einer numerischen Verschiebung vom Ausgangsfeld her). Das Q-Feld enthält die möglichen Züge der Dame, das B-Feld die des Läufers, usw. Daher wissen wir, daß das N-Feld, das in Zeile 1800 erscheint, die legalen Züge des Springers beinhaltet.

In dieser Zeile wird die Variable X gleich Z (das Feld, auf dem die Figur steht) plus einem zufällig ausgewählten Element der N-Matrix gesetzt. Zeile 1810 prüft, ob dieses mögliche Zielfeld nicht schon außerhalb des Brettes liegt, was der Fall wäre, wenn der Wert kleiner als 11 oder größer als 88 ist, oder aber gleich 42, was ebenso bedeuten würde daß es nicht mehr innerhalb des Brettes liegt. Y ist die Anzahl der möglichen Züge. Diese wird in Zeile 1830 um eins erhöht, danach überprüft der Computer die Zahl in A(X), die das Zielfeld darstellt. Findet er den Wert 'E' (für „leer“), weiß er, daß er auf dieses Feld ziehen kann. Aber er stolpert nicht unbedingt nur deshalb auf dieses Feld A(X), weil es leer ist.

Zuerst prüft er in der Subroutine von Zeile 350, ob dieses Zielfeld bedroht wird. Danach wird der Inhalt von MM abgerufen. Ist er eins, so hat das Programm für den in Erwägung gezogenen Zug grünes Licht gegeben, und der Computer fährt mit jenem Programmteil fort, der den Zug ausführt. Wenn Y größer als 20 ist, was 20 erfolglose Zugversuche bedeutet, springt der Computer zurück, so daß die Chancen einer anderen Figur probiert werden. Treffen alle beide Fälle nicht zu, sucht der Computer einen neuen Zufallszug für den Springer in Zeile 1800.

Prinzipiell wird jeder Zug so erzeugt und getestet. Der Springer ist etwas einfacher als andere Figuren zu testen, da er über im Wege stehende Spielsteine springen kann. Für alle anderen Figuren muß zuerst geprüft werden, ob ein freier Weg für ihre beabsichtigten Züge besteht. Auch das ist allerdings nicht schwer zu erreichen.

In Zeile 2970 beginnt die Initialisierung. Zunächst verlangt der Computer vom Spieler „BITTE SCHALTEN SIE CAPS LOCK EIN. DRÜCKEN SIE DANN RETURN“. Der Computer zählt, wie lange dies dauert (er verwendet N als Zähler), und benutzt diesen Wert, um den Zufallsgenerator zu füttern. Das betrifft nun alle Programme in diesem Buch: sollte Ihre Zufallsgeneratorfunktion anders arbeiten, ändern Sie diese Zeile, um sie Ihrem speziellen BASIC-Dialekt anzupassen. In Zeile 3000 werden alle Variablen als INTEGER (ganzzahlig) definiert. Es maximiert die Geschwindigkeit, mit der das Programm abläuft. Sie können es aber ohne Schaden weglassen, insbesondere, wenn Ihr Computer kein DEFINT hat. MM (Maschinenzug, Sie erinnern sich) wird in Zeile 3010 auf Null und A\$ (das Mitteilungen des Spielers an den Computer enthält, wie z.B. den Wunsch, die Seiten zu tauschen) auf den Nullstring gesetzt. Nun werden eine Anzahl von Matrizen dimensioniert. A beinhaltet das Brett selbst, R die Züge des Turmes, B die des Läufers, N die des Springers, Q die der Königin, K die des Königs, Z den Seitentauschmechanismus, S die Vorrangigkeit, mit der die Reihenfolge bestimmt wird, in der die Felder nach Figuren abgesucht werden, und T die Stellung der Figuren, bevor ein Zug gemacht wurde.

Die nächsten beiden Zeilen initialisieren die Variablen für die Figuren des Computers und des Spielers wie folgt:

P – Bauer des Spielers
R – Turm des Spielers
N – Springer des Spielers
B – Läufer des Spielers
Q – Königin des Spielers
K – König des Spielers
E – ein leeres Feld
PB – Bauer des Computers
PR – Turm des Computers
PN – Springer des Computers
BB – Läufer des Computers
QB – Königin des Computers
KB – König des Computers

Die nächste Zeile (3050) füllt das A-Feld (welches das Brett beinhaltet) mit einem fingierten Wert (–99). Dadurch wird erreicht, daß jedes Feld, dem im Zuge des Programmes noch kein Wert zugeordnet wurde, übersprungen wird. Übrigens verwendet dieses Programm ein dem am Beginn dieses Kapitels beschriebenen ähnliches Brett. Allerdings liegt es „auf der Seite und mit Vorne als Hinten“ und sieht daher folgendermaßen aus:

```

18 28 38 48 58 68 78 88 *8
17 27 37 47 57 67 77 87 *7
16 26 36 46 56 66 76 86 *6
15 25 35 45 55 65 75 85 *5
14 24 34 44 54 64 74 84 *4
13 23 33 43 53 63 73 83 *3
12 22 32 42 52 62 72 82 *2
11 21 31 41 51 61 71 81 *1
*****
A B C D E F G H

```

Mit der Z-Schleife (in Zeile 3070) füllt die nächste Sektion des Programmes die Felder mit der Startaufstellung zum Beginn des Spieles. Der nächste Teil ist ziemlich leicht zu durchschauen: 3260 setzt die Werte für die Bewegungen des Springers in das N-Feld ein, 3280 macht das gleiche für den Turm und 3340 für den Läufer. Der DATA-Zeiger wird dann auf Zeile 3290 zurückgestellt (zum Beginn der Information über den Turm), damit das Q-Feld (für die Königin) sowohl mit den Daten für die Bewegungen des Turmes als auch mit denen über die Bewegungen des Läufers gefüllt werden kann. Dies deshalb, da die Königin natürlich die Züge beider Figuren vollführen kann.

Die Zeile 3430 füllt das S-Feld mit den DATA-Werten in den Zeilen 3430 bis 3510. Diese stellen wahrscheinlich den schwerwiegendsten Grund dar, warum das Programm so schnell abläuft, wenn es einmal gestartet wurde. Diese Serie von Zahlen ist die Reihenfolge, in der die Felder des Brettes überprüft werden. Sie können diese Reihenfolge herausfinden, wenn Sie jene Serie mit dem „numerierten Brett“ auf der vorhergehenden Seite vergleichen. Es beginnt mit den Feldern vor der Beginnposition der beiden Bauern in der Brettmitte, um dann links und rechts zum Bretttrand fortzuschreiten, danach kommen die Felder, auf denen wahrscheinlich das Spiel eröffnet wird. Haben Sie sich einmal durch diese Liste hindurchgearbeitet, werden Sie vielleicht die Sequenz modifizieren wollen. Dieses Feld bestimmt mehr als jedes andere die Art und Weise, wie der Computer spielen wird. Sollten Sie mit der von mir aufgestellten Sequenz nicht einverstanden sein, sollten Sie diese auf alle Fälle ändern und die Auswirkungen dieser Programmänderung auf die Spielart des Computers beobachten.

Die Routine, die die Seiten tauscht, liegt von 3540 bis 3640. Der Teil zwischen 3660 und 3720 erzeugt eine Anzahl von Kommentaren, wenn der Computer eine Figur verliert („Toller Zug, Meister!“ und Ähnliches). Der letzte Teil zwischen 3660 und 3720 des Programmes wirft die aktuelle Stellung auf dem Drucker aus.

Gomoku

Sie werden wahrscheinlich finden, daß Gomoku ein sehr leicht zu erlernendes, aber ein fast unmöglich zu gewinnendes Spiel ist. In diesem Programm, das auf einem von Graham Charlton geschriebenen basiert, spielt der Computer außerordentlich gut.

Sie müssen versuchen, fünf Ihrer Figuren (die M's) in einer Reihe in jeder beliebigen Richtung anzuordnen. Natürlich versucht der Computer dasselbe.

Sehen Sie sich nun die Brettpositionen von dem Beginn eines Spieles, das ich gegen dieses Programm spielte, an. Das genauere Betrachten der Ausdrucke zeigt Ihnen, wie sich das Programm entwickelt und wie Sie es spielen müssen:

```

      1 2 3 4 5 6 7 8      1 2 3 4 5 6 7 8
1 . . . . . . . . 1 . . . . . . . . 1
2 . . . . . . . . 2 . . . . . . . . 2
3 . . . . . . . . 3 . . . . . . . . 3
4 . . . . . . . . 4 . . . . . . . . 4
5 . . . . . . . . 5 . . . . M . . . 5
6 . . . . . . . . 6 . . . . . . . . 6
7 . . . . . . . . 7 . . . . . . . . 7
8 . . . . . . . . 8 . . . . . . . . 8
      1 2 3 4 5 6 7 8      1 2 3 4 5 6 7 8

```

```

      1 2 3 4 5 6 7 8      1 2 3 4 5 6 7 8
1 . . . . . . . . 1 . . . . . . . . 1
2 . . . . . . . . 2 . . . . . . . . 2
3 . . . . . . . . 3 . . . . . . . . 3
4 . . . . . . . . 4 . . . . . . . . 4
5 . . . . M . . . 5 . . . . M . . . 5
6 . . . . C . . . 6 . . . . C . . . 6
7 . . . . . . . . 7 . . M . . . . . 7
8 . . . . . . . . 8 . . . . . . . . 8
      1 2 3 4 5 6 7 8      1 2 3 4 5 6 7 8

```



```

1 2 3 4 5 6 7 8
1 . . . . . 1
2 . . . . . 2
3 . . . . . 3
4 . . . . . 4
5 . . . M . 5
6 . . C C . 6
7 . M . . . 7
8 . . . . . 8
1 2 3 4 5 6 7 8

```

```

1 2 3 4 5 6 7 8
1 . . . . . 1
2 . . . . . 2
3 . . . . . 3
4 . . . . . 4
5 . . . M . 5
6 . . C M C . 6
7 . M . . . 7
8 . . . . . 8
1 2 3 4 5 6 7 8

```

```

1 2 3 4 5 6 7 8
1 . . . . . 1
2 . . . . . 2
3 . . . . . 3
4 . . . C . 4
5 . . . M . 5
6 . . C M C . 6
7 . M . . . 7
8 . . . . . 8
1 2 3 4 5 6 7 8

```

```

1 2 3 4 5 6 7 8
1 . . . . . 1
2 . . . . . 2
3 . . . . . 3
4 . . . C . 4
5 . . . M . 5
6 . . C M C . 6
7 . M M . . 7
8 . . . . . 8
1 2 3 4 5 6 7 8

```

Hier ist das Listing für Ihr eigenes Gomoku-Programm:

```

10 REM GOMOKU
20 GOSUB750
30 GOSUB130
40 GOSUB240
50 GOSUB130
60 GOSUB320
70 GOSUB130
80 IF L>3 THEN PRINT:PRINT"ICH GEWINNE!"
":END
90 GOTO40
100 E=A
110 E=E+N:IF A(E)<>Z THEN RETURN
120 K=K+1:GOTO110
130 CLS

```

```

140 PRINT:PRINT:PRINT
150 PRINT TAB(10);"1 2 3 4 5 6 7 8"
160 FOR A=1 TO 8:PRINT TAB(8);A;" ";
170 FOR B=2 TO 9
180 PRINT CHR$(A*(A*10+B));" ";
190 NEXT B
200 PRINT:A
210 NEXT A
220 PRINT TAB(10);"1 2 3 4 5 6 7 8"
230 RETURN
240 PRINT:PRINT
250 PRINT"Bitte geben Sie Ihren Zug ein."
..
260 INPUT G
270 G=G+1
280 IF G<12 OR G>89 OR A(G)<>46 THEN GOTO260
290 Z=H
300 A(G)=Z
310 RETURN
320 A=G
330 L=0
340 FOR X=1 TO 4:K=0:N=X(X)
350 GOSUB100
360 N=-N:GOSUB100
370 IF K>L THEN L=K
380 NEXT X
390 IF L>3 THEN PRINT:PRINT"Sie gewinnen"
!":END
400 T=1
410 IF T<>2 THEN Z=C
420 IF T=2 THEN Z=H
430 G=0:H=0:L=0
440 FOR A=12 TO 89
450 M=0
460 IF A(A)<>46 THEN GOTO570
470 FOR X=1 TO 4:K=0:N=X(X)
480 GOSUB100
490 N=-N:GOSUB100

```

```

500 IF K>L THEN H1=0:L=K
510 IF L<>K THEN GOTO540
520 IF T=1 AND L<4 OR (T=2 OR T=3) AND L
<2 THEN GOTO540
530 M=M+1
540 NEXT X
550 IF M<=H1 THEN GOTO570
560 H1=M:G=A
570 NEXT A
580 IF H1<>0 THEN GOTO650
590 T=T+1:IF T<>4 THEN GOTO410
600 A=1
610 G=INT(RND(1)*77)+13
620 IF A(G)=46 THEN GOTO650
630 A=A+1:IF A<100 THEN GOTO610
640 PRINT:PRINT"Ich gebe auf, Meister!":
END
650 A(G)=C
660 Z=C:A=G:L=0
670 FOR X=1 TO 4
680 K=0
690 N=X(X)
700 GOSUB100
710 N=-N:GOSUB100
720 IF K>L THEN L=K
730 NEXT X
740 RETURN
750 CLS
760 DIM A(100),X(4)
770 FOR C=1 TO 8
780 FOR B=2 TO 9
790 A(C*10+B)=46
800 NEXT B
810 NEXT C
820 FOR Q=1 TO 4
830 READ Z:X(Q)=Z
840 NEXT Q
850 DATA 1,9,10,11
860 H=ASC("M"):C=ASC("C")

```

```

870 PRINT:PRINT"Geben Sie J ein wenn Sie
beginnen"
880 PRINT"wollen, N wenn nicht"
890 N=0
900 N=N+1
910 A$=INKEY$(10)
920 IF A$<>"J" AND A$<>"j" AND A$<>"N" A
ND A$<>"n" THEN GOTO900
930 REM RANDOMIZE N
940 CLS
950 IF A$="J" OR A$="j" THEN RETURN
960 FOR J=1 TO INT(RND(1)*12)+1
970 READ Z
980 NEXT J
990 A(Z)=C
1000 RETURN
1010 DATA 34,35,44,46,47,54,55,56,57,66

```

Vier-in-einer-Reihe

Das Ziel des Spieles ist es, wie schon der Name sagt, vier Ihrer Steine (den M's) in einer Linie in jeder beliebigen Richtung zu bringen, bevor es der Computer (der C's verwendet) schafft.

Durch die Auswahl der Spalte, in die Sie Ihren Stein setzen wollen, geben Sie Ihre Zugentscheidung bekannt. Der Stein fällt dann auf die niedrigste freie Position in dieser Spalte.

Der Computer spielt dieses Spiel ziemlich gut. Auch ist er überraschend schnell, wenn man bedenkt, wie oft er die Schleifen des Programmes durchlaufen muß.

Als ich die erste handgeschriebene Version dieses Programmes beendet hatte, war ich nicht sehr zufrieden. Es schien mir, als hätte ich eine Methode der „nackten Gewalt“ gewählt, um das Problem zu lösen. Ich war mir sicher, es müsse einen gerisseneren Weg geben. Trotzdem beendete ich das Programm völlig und gab es dem Computer ein. Obwohl es etwas komisch spielte, gewann es doch das erste Spiel, und ich wußte, daß ich trotz der Programmiermethode gut gearbeitet hatte. Als ich später darüber nachdachte, fiel mir auf, daß, wenn das Programm korrekt strukturiert war (was es war), wenn es keine redundanten Befehle enthielt (und das hat es, soweit ich sehen kann, nicht), es schnell und gut lief, es kein Herumfeilen mehr brauchte, nur um es

noch trickreicher zu machen. Leicht nachzuvollziehende Programmabschnitte sind immer besser, als überaus klug verknüpfte Schritte, die, zugegeben, weniger Platz beanspruchen und vielleicht um einige Mikrosekunden schneller laufen, aber dafür fast unmöglich zu modifizieren sind.

Dieses Programm hier ist also meine „Nackte Gewalt“-Version. Die überaus zahlreichen REM-Bemerkungen sollten es Ihnen sehr leicht machen herauszufinden, was jeder Programmteil bewirkt. Außerdem sollte es sich auch als sehr leicht zu verbessern herausstellen, wenn Sie erst einmal einige Partien in der jetzigen Form gespielt haben.

In dieser Version von **Vier in einer Reihe** gibt der Computer immer seinem Gegner den ersten Zug. Dann führt er seinen Eröffnungszug auf diesen zurück. Vielleicht wollen Sie das Programm so verändern, daß auch der Computer das Spiel eröffnen kann.

Bevor wir uns das Programmlisting von **Vier** in einer Reihe ansehen, habe ich hier ein Spiel, das gegen das Programm spielte:

Ihr Zug...

Auf welche Spalte wollen Sie
Ihren Stein setzen ?4

1 2 3 4 5 6 7

Bitte warten Sie auf meinen Zug...

Ihr Zug...

Auf welche Spalte wollen Sie
Ihren Stein setzen ?5

.
.
.
.
.
.	.	.	C	M	.	.
.	.	M	M	C	.	.
1	2	3	4	5	6	7

Bitte warten Sie auf meinen Zug...

Ihr Zug...

Auf welche Spalte wollen Sie
Ihren Stein setzen ?3

A 7x7 grid of dots. The bottom row of dots is numbered 1 through 7 from left to right. The fourth column of dots is labeled with a 'C' at the top and 'M M' at the bottom. The grid is used to illustrate the definition of the C and M sets.

Bitte warten Sie auf meinen Zug...

Ihr Zug...

Auf welche Spalte wollen Sie
Ihren Stein setzen ??

.
.
.
.
.
.	.	.	C	M	.	.
.	C	M	M	C	.	M
1	2	3	4	5	6	7

Bitte warten Sie auf meinen Zug...

Ihr Zug...

Auf welche Spalte wollen Sie
Ihren Stein setzen ?4

.
.
.
.
.
.	.	.	M	C	.	.
.	.	.	C	M	.	.
.	C	M	M	C	.	M
1	2	3	4	5	6	7

Bitte warten Sie auf meinen Zug..

Ihr Zug...

Auf welche Spalte wollen Sie
Ihren Stein setzen ?6

```

. . . . .
. . . . .
. . . . .
. . C C .
. . M C .
. . M C M .
. C M M C M M
1 2 3 4 5 6 7

```

Bitte warten Sie auf meinen Zug...

Ihr Zug...

Auf welche Spalte wollen Sie
Ihren Stein setzen ?3

.
.
.
.
.	.	.	C	.	.	.
.	.	.	M	C	.	.
.	.	M	C	M	.	.
.	C	M	M	C	.	M
1	2	3	4	5	6	7

Bitte warten Sie auf meinen Zug...

Ihr Zug...

Auf welche Spalte wollen Sie
Ihren Stein setzen ?6

```

. . . . .
. . . . .
. . . C .
. . C C .
. . M C .
. M C M M
. C M M C M M
1 2 3 4 5 6 7

```

Bitte warten Sie auf meinen Zug...

Ich habe Sie besiegt, Mensch!

Das ist das Listing für Vier in einer Reihe:

```

10 REM VIER-IN-EINER-REIHE
20 GOSUB1090:REM INITIALISIEREN
30 GOSUB860:REM SPIELBRETT AUSDRUCKEN
40 GOSUB680:REM GEWINNTEST
50 GOSUB980:REM MENSCHENZUG
60 GOSUB860:REM SPIELBRETT AUSDRUCKEN
70 GOSUB680:REM GEWINNTEST
80 GOSUB110:REM COMPUTERZUG
90 GOT030

100 REM *****
110 REM COMPUTERZUG
120 PRINT:PRINT"Bitte warten Sie auf mei

```

```

nen Zug..."
130 B=10
140 B=B+1
150 IF A(B)=-9 THEN GOTO180
160 IF A(B)=C THEN X=C:GOTO210
170 IF A(B)=H THEN X=H:GOTO210
180 IF B<77 THEN GOTO140
190 GOTO480
200 REM *****
*****
210 REM VIER IN EINER REIHE GEFAHR/MOEGL
ICHKEIT?
220 REM QUER
230 IF A(B+1)=X AND A(B+2)=X AND A(B+3)=
AND A(B+13)<>E THEN ZUG=B+3:GOTO650
240 IF A(B-1)=X AND A(B-2)=X AND A(B-3)=
AND A(B+7)<>E THEN ZUG=B-3:GOTO650
250 IF A(B+1)=X AND A(B+2)=X AND A(B-1)=
AND A(B+9)<>E THEN ZUG=B-1:GOTO650
260 IF A(B-1)=X AND A(B+2)=X AND A(B+1)=
AND A(B+11)<>E THEN ZUG=B+1:GOTO650
270 IF A(B+1)=X AND A(B-1)=X AND A(B+2)=
AND A(B+12)<>E THEN ZUG=B+2:GOTO650
280 IF A(B+1)=X AND A(B-1)=X AND A(B-2)=
AND A(B+8)<>E THEN ZUG=B-2:GOTO650
290 IF A(B-1)=X AND A(B-2)=X AND A(B+1)=
AND A(B+11)<>E THEN ZUG=B+1:GOTO650
300 REM VERTIKAL
310 IF B>20 THEN IF A(B-10)=X AND A(B-20
)=X AND A(B+10)=E AND A(B+20)<>E THEN ZUG=
B+10:GOTO650
320 REM DIAGONAL
330 IF A(B+11)=X AND A(B+22)=X AND A(B-1
1)=E AND A(B-1)<>E THEN ZUG=B-1:GOTO650
340 IF A(B+9)=X AND A(B+18)=X AND A(B-9)
=E AND A(B+1)<>E THEN ZUG=B-9:GOTO650
350 REM *****
360 REM DREIERGRUPPE MACHEN/BLOCKIEREN?
370 REM QUER

```

```

380 IF A(B+1)=X AND A(B+2)=E AND A(B+12)
<>E THEN ZUG=B+2:GOTO650
390 IF A(B+1)=X AND A(B-1)=E AND A(B+9)<
>E THEN ZUG=B-1:GOTO650
400 IF A(B-1)=X AND A(B-2)=E AND A(B+8)<
>E THEN ZUG=B-2:GOTO650
410 REM VERTIKAL
420 IF A(B+10)=X AND A(B-10)=E AND A(B)<
>E THEN ZUG=B-10:GOTO650
430 REM DIAGONAL
440 IF A(B+9)=X AND A(B-9)=E AND A(B+1)<
>E THEN ZUG=B-9:GOTO650
450 IF B>11 THEN IF A(B+11)=X AND A(B-11
)=E AND A(B-1)<>E THEN ZUG=B-11:GOTO650
460 GOTO180
470 REM *****
480 REM EINELNE ZUEGE
490 FOR N=1 TO 3
500 M(N)=0
510 NEXT N
520 ZAEHL=0
530 FOR B=11 TO 77
540 IF A(B)<>C AND A(B)<>H THEN GOTO600
550 IF A(B+1)=E AND A(B+11)<>E THEN ZAEH
L=ZAEHL+1:M(ZAEHL)=B+1
560 IF A(B-1)=E AND A(B+9)<>E THEN ZAEHL
=ZAEHL+1:M(ZAEHL)=B-1
570 IF A(B-10)=E AND A(B)<>E THEN ZAEHL=
ZAEHL+1:M(ZAEHL)=B-10
580 IF A(B-11)=E AND A(B-1)<>E THEN ZAEH
L=ZAEHL+1:M(ZAEHL)=B-11
590 IF A(B-9)=E AND A(B+1)<>E THEN ZAEHL
=ZAEHL+1:M(ZAEHL)=B-9
600 NEXT B
610 IF ZAEHL<>0 THEN GOTO640
620 PRINT:PRINT"Wir sollten es ein Unent
schieden nennen"
630 PRINT:PRINT:PRINT:END
640 ZUG=M(INT(RND(1)*ZAEHL)+1)

```

```

650 A(ZUG)=C
660 RETURN
670 REM *****
680 REM GEWINNTEST
690 X=H
700 B=10
710 B=B+1
720 IF A(B)<>X THEN GOTO770
730 IF A(B+1)=X AND A(B+2)=X AND A(B+3)=
X THEN GOTO800
740 IF B>30 THEN IF A(B-10)=X AND A(B-20
)=X AND A(B-30)=X THEN GOTO800
750 IF B>33 THEN IF A(B-11)=X AND A(B-22
)=X AND A(B-33)=X THEN GOTO800
760 IF B>27 THEN IF A(B-9)=X AND A(B-18)
=X AND A(B-27)=X THEN GOTO800
770 IF B<77 THEN GOTO710
780 IF X=H THEN X=C:GOTO700
790 RETURN
800 REM SIEG GEFUNDEN
810 PRINT:PRINT
820 IF X=H THEN PRINT"Sie haben mich ges
chlagen, Mensch!"
830 IF X=C THEN PRINT"Ich habe Sie besie
gt, Mensch!"
840 PRINT:PRINT:PRINT:END
850 REM *****
860 REM SPIELBRETT AUSDRUCKEN
870 CLS:PRINT:PRINT
880 FOR K=10 TO 70 STEP10
890 PRINT TAB(5):
900 FOR J=1 TO 7
910 PRINT CHR$(A(K+J));" ";
920 NEXT J
930 NEXT K
940 PRINT TAB(5);"1 2 3 4 5 6 7"
950 PRINT:PRINT
960 RETURN
970 REM *****

```

```

980 REM SPIELERZUG
990 PRINT"Ihr Zug...":PRINT
1000 PRINT"Auf welche Spalte wollen Sie"
1010 INPUT "Ihren Stein setzen ";J
1020 Z=J
1030 Z=Z+10
1040 IF A(Z+10)=E THEN GOTO1030
1050 IF A(Z)=E THEN A(Z)=H:RETURN
1060 PRINT"Das koennen Sie nicht!"
1070 GOTO1000
1080 REM *****
1090 REM INITIALISIEREN
1100 CLS
1110 REM RANDOMIZE
1120 DIM A(109),M(30),P(6)
1130 E=ASC(".")
1140 H=ASC("M"):C=ASC("C")
1150 FOR B=1 TO 109
1160 A(B)=E
1170 D=B-10*INT(B/10)
1180 IF D=0 OR D>7 OR B<11 OR B>77 THEN A
(B)=-9
1190 NEXT B
1200 RETURN

```

Dame

Das Damespiel hat eine lange und ehrenvolle Geschichte. R.C. Bell (in seinem Buch 'Discovering Old Board Games', Shire Publication, Aylesbury, UK, 1980) behauptet, es wäre um 1100 erfunden worden, wahrscheinlich im Süden Frankreichs mit Backgammonsteinen auf einem Schachbrett, wobei die Alquerque-Methode des Schlagens verwendet wurde. Die Enzyklopädie des Sports, der Spiele und der Vergangenheit (Fleetway House, London, c. 1935) siedelt es sogar noch früher an: Formen dieses Spieles waren schon im alten Ägypten, Griechenland und Rom bekannt. Es war in der Mitte des siebzehnten Jahrhunderts so verbreitet wie heute. Ungeachtet seines Alters ist es ein sehr populäres Spiel. In vielen europäischen Ländern gibt es unterschiedliche Varianten. Das Kontinentale

Damespiel' zum Beispiel wird auf einem Brett mit 100 Quadraten und 20 Steinen pro Spieler gespielt. Diese Art wurde im frühen 18. Jahrhundert entwickelt.

Dieses Programm spielt die wahrscheinlich bekannteste Version. Obwohl das Fehlen einer Endspielstrategie oft zu dramatischen Zusammenbrüchen in der Endphase des Spiels führt, spielt es flink und relativ gut.

Jeder versucht, des anderen Steine zu nehmen (das heißt, über diese zu springen, um sie zu fangen, und sie dann vom Brett zu nehmen) oder sie so in die Enge zu treiben, daß der Gegner sie nicht mehr bewegen kann. Meist wird das Spiel auf einem Brett mit 64 quadratischen Feldern gespielt, die abwechselnd hell und dunkel sind. Das Spielbrett dieses Programmes besteht (wie Sie anhand des Demonstrationsspiels sicherlich erkennen) aus einer Anzahl von Punkten. Die Steine werden durch M's (für Mensch) und C's dargestellt.

Zu Beginn des Spieles fangen Sie von unten nach oben und der Computer von oben nach unten zu spielen an. In der momentanen Programmfassung beginnt der Computer zu ziehen. Wenn Sie das Spiel einleiten wollen, so löschen Sie Zeile 50.

Alle Züge müssen diagonal gemacht werden. Die einzelnen Steine können sich nur vorwärts, d.h. in die Richtung der gegnerischen Seite, bewegen. Erreicht ein Stein die letzte Reihe auf der anderen Brettseite, wird er in eine 'Dame' umgewandelt. Ihre Damen werden durch K's dargestellt, die des Computers durch Dollarzeichen. Damen können sich sowohl vorwärts als auch rückwärts bewegen. Haben Sie einen Stein geschlagen – durch das Überspringen dieses in ein leeres Feld unmittelbar hinter dem geschlagenen Stein – dürfen Sie den Zug fortsetzen, wenn Sie weiter schlagen können.

Hier ist eine Anzahl von Brettpositionen aus dem Spiel, das ich gegen das Programm spielte:

COMPUTER: 0
MENSCH: 0

```

1 2 3 4 5 6 7 8
8 . C . C . C . C 8
7 C . . C . C . C 7
6 . C . C . C . C 6
5 . . . . . 5
4 . . . . . 4
3 M . M . M . M . 3
2 . M . M . M . M 2
1 M . M . M . M . 1
1 2 3 4 5 6 7 8

```

COMPUTER: 0
MENSCH: 0

```

1 2 3 4 5 6 7 8
8 . C . C . C . C 8
7 C . C . C . C . 7
6 . . C . C . C 6
5 . C . . . 5
4 . M . . . 4
3 M . . M . M . 3
2 . M . M . M . M 2
1 M . M . M . M . 1
1 2 3 4 5 6 7 8

```

COMPUTER: 0
MENSCH: 0

```

1 2 3 4 5 6 7 8
8 . C . C . C . C 8
7 C . . C . C . 7
6 . C . C . C . C 6
5 . C . . . 5
4 . M . . . 4
3 M . M . M . M . 3
2 . M . . M . M 2
1 M . M . M . M . 1
1 2 3 4 5 6 7 8

```

COMPUTER: 0
MENSCH: 0

```

1 2 3 4 5 6 7 8
8 . C . C . C . C 8
7 C . . C . C . 7
6 . C . C . . C 6
5 . C . C . . 5
4 . M . M . . 4
3 M . . M . M . 3
2 . M . . M . M 2
1 M . M . M . M . 1
1 2 3 4 5 6 7 8

```

COMPUTER: 1
MENSCH: 0

```

1 2 3 4 5 6 7 8
8 . C . C . C . C 8
7 C . . C . C . 7
6 . C . C . . C 6
5 . C . . . 5
4 . M . . . 4
3 M . C . M . M . 3
2 . M . . M . M 2
1 M . M . M . M . 1
1 2 3 4 5 6 7 8

```

COMPUTER: 1
MENSCH: 1

```

1 2 3 4 5 6 7 8
8 . C . C . C . C 8
7 C . . C . C . 7
6 . C . C . . C 6
5 . C . . . 5
4 . M . M . . 4
3 M . . M . M . 3
2 . . . M . M 2
1 M . M . M . M . 1
1 2 3 4 5 6 7 8

```

Hier ist das komplette Listing von Dame:

```

10 REM DAME
20 GOSUB1010
30 REM LOESCHEN SIE ZEILE 50, WENN DER
40 REM MENSCHLICHE SPIELER BEGINNEN SOL
L
50 GOT090

```

```

60 REM *****
70 GOSUB600
80 GOSUB760
90 GOSUB600
100 GOSUB130
110 GOTO70
120 REM *****
130 FOR X=1 TO 10:S(X)=0:NEXT X
140 SC=0:A=89
150 A=A-1
160 IF Q(A)<>C AND Q(A)<>CD THEN GOTO240
170 B=0:IF A<29 THEN B=2
180 B=B+1
190 R=A+N(B)
200 IF R>88 OR R<11 THEN GOTO240
210 IF (Q(R)=M OR Q(R)=MD) AND Q(R+N(B))
=L THEN GOTO280
220 IF Q(R)=L THEN IF (Q(R-11)<>M AND Q(
R-11)<>MD) THEN IF (Q(R-9)<>M AND Q(R-9)<>
MD AND Q(R+9)<>MD) THEN IF ((Q(R+22)<>MD O
R Q(R+18)<>MD) AND (Q(R+9)<>C OR Q(R+9)<>C
D OR Q(R+11)=C OR Q(R+11)=CD)) AND Q(R+11)
<>MD THEN GOSUB400
230 IF B<2 OR (Q(A)=CD AND B<4) THEN GOT
O180
240 IF A>11 THEN GOTO150
250 FL=0:IF Q(22)=C OR Q(24)=C OR Q(26)=
C OR Q(28)=C THEN GOSUB1270
260 IF FL=1 THEN GOTO570
270 GOTO420
280 Q(R+N(B))=Q(A):Q(R)=L:Q(A)=L
290 CO=CO+1
300 GOSUB600
310 A=L+N(B)
320 B=0
330 B=B+1
340 IF (A+2*N(B)<11 OR A+2*N(B)>88) AND
B<4 THEN GOTO330
350 R=A+N(B)

```

```

360 IF Q(R)=C AND B>3 THEN RETURN
370 IF (Q(R)=M OR Q(R)=MD) AND Q(R+N(B))
=L THEN GOTO280
380 IF B<2 OR (Q(A)=CD AND B<4) THEN GOT
O330
390 RETURN
400 IF SC<10 THEN SC=SC+1
410 S(SC)=100*A+B+20:RETURN
420 IF SC=0 THEN GOTO470
430 XC=INT(RND(1)*SC)+1
440 A=INT(S(XC)/100)
450 R=A+N(S(XC)-100*A-20)
460 GOTO570
470 SC=SC+1:A=INT(RND(1)*88)+1
480 IF Q(A)<>C AND Q(A)<>CD THEN GOTO550
490 B=0
500 B=B+1
510 R=A+N(B)
520 IF R>88 OR R<11 THEN GOTO540
530 IF Q(R)=L THEN GOTO570
540 IF B<2 OR Q(A)=CD AND B<4 THEN GOTO5
00
550 IF SC<300 THEN GOTO470
560 PRINT:PRINT"Ich gebe auf":END
570 Q(R)=Q(A):Q(A)=L
580 RETURN
590 REM *****
600 CLS:PRINT:PRINT
610 PRINT"      COMPUTER: ";CO
620 PRINT"      MENSCH: ";ME:PRINT:PRIN
T
630 PRINT"  1 2 3 4 5 6 7 8"
640 FOR F=80 TO 10 STEP -10
650 PRINT ;F/10;" ";
660 FOR G=1 TO 8:PRINT CHR$(Q(F+G));" ";
:NEXT G
670 PRINT ;F/10:NEXT F
680 PRINT"  1 2 3 4 5 6 7 8"
690 IF CO=12 OR ME=12 THEN GOTO710

```

```

700 RETURN
710 IF ME=12 THEN PRINT"Sie haben gewonn
en"
720 IF CO=12 THEN PRINT"Ich habe gewonne
n"
730 PRINT"Danke fuer das Spiel":END
740 REM *****
750 REM 99 FUER AUFGEBEN
760 PRINT:PRINT
770 PRINT"Geben Sie Ihren Zug ein:"
780 INPUT"Von: ":A
790 IF A=99 THEN GOTO730
800 IF Q(A)<>M AND Q(A)<>MD THEN GOTO780
810 INPUT"Nach: ":B
820 IF Q(B)<>L THEN GOTO810
830 Q(B)=Q(A):Q(A)=L
840 REM *****
850 FOR T=11 TO 17:IF Q(T)=C THEN Q(T)=C
D
860 NEXT T
870 FOR T=82 TO 88:IF Q(T)=M THEN Q(T)=M
D
880 NEXT T
890 REM *****
900 IF ABS(A-B)<12 THEN RETURN
910 TY=RND(1)
920 IF TY<.3 THEN PRINT"Guter Zug":GOSUB
1340
930 IF TY>.7 THEN PRINT"Hat mich erwisch
t!":GOSUB1340
940 ME=ME+1:Q((A+B)/2)=L:GOSUB600
950 FOR T=82 TO 88:IF Q(T)=M THEN Q(T)=M
D
960 NEXT T
970 PRINT:INPUT"Koennen Sie nocheinmal s
pringen (J/N) ":A$
980 IF A$<>"J" AND A$<>"j" THEN RETURN
990 A=B:GOTO800
1000 REM *****

```

```

1010 REM INITIALISIEREN
1020 CLS:PRINT"Druucken Sie eine Taste":N
=1
1030 N=N+1:IF INKEY$(10)="" THEN GOTO1030
1040 CLS:REM RANDOMIZE N
1050 PRINT"Bitte warten Sie"
1060 DIM Q(99),N(4),S(10)
1070 M=ASC("M"):MD=ASC("D")
1080 C=ASC("C"):CD=ASC("#")
1090 L=32:NI=-99
1100 FOR A=1 TO 99:Q(M)=NI:NEXT A
1110 FOR A=1 TO 64
1120 READ D,G
1130 Q(D)=G:NEXT A:FOR I=1 TO 99:IF Q(I)=
72 THEN Q(I)=M
1135 NEXT I
1140 DATA 81,46,82,67,83,46,84,67,85,46,8
6,67,87,46
1150 DATA 88,67,71,67,72,46,73,67,74,46,7
5,67,76,46
1160 DATA 77,67,78,46,61,46,62,67,63,46,6
4,67
1170 DATA 65,46,66,67,67,46,68,67,51,32,5
2,46
1180 DATA 53,32,54,46,55,32,56,46,57,32,5
8,46
1190 DATA 41,46,42,32,43,46,44,32,45,46,4
6,32
1200 DATA 47,46,48,32,31,72,32,46,33,72,3
4,46,35,72
1210 DATA 36,46,37,72,38,46,21,46,22,72,2
3,46,24,72
1220 DATA 25,46,26,72,27,46,28,72,11,72,1
2,46,13,72
1230 DATA 14,46,15,72,16,46,17,72,18,46
1240 FOR A=1 TO4:READ N(A):NEXT A
1250 DATA -11,-9,11,9
1260 CO=0:ME=0:RETURN
1270 IF Q(22)=C AND Q(11)=L THEN A=22:P=1

```



```

1:FL=1:RETURN
1280 IF Q(22)=C AND Q(13)=L THEN A=22:R=1
3:FL=1:RETURN
1290 IF Q(24)=C AND Q(13)=L THEN A=24:R=1
3:FL=1:RETURN
1300 IF Q(24)=C AND Q(15)=L THEN A=24:R=1
5:FL=1:RETURN
1310 IF Q(26)=C AND Q(15)=L THEN A=26:R=1
5:FL=1:RETURN
1320 IF Q(26)=C AND Q(17)=L THEN A=26:R=1
7:FL=1:RETURN
1330 RETURN
1340 FOR Q=1 TO 1000:NEXT Q:RETURN

```

Shogun

Shogun basiert auf dem japanischen Brettspiel Hasam Shogi. In diesem Spiel, das meist in einer Ecke eines Go-Brettes gespielt wird, darf man seine Steine in einer geraden Linie in jede Richtung ziehen (vorwärts, zurück oder zur Seite). Diagonale Bewegungen sind nicht erlaubt. Man kann aber auch über einen anderen Stein, den eigenen oder den gegnerischen, in einer geraden Linie springen.

Der übersprungene Stein wird nicht vom Brett genommen. Ein Stein wird geschlagen, indem er zwischen zwei gegnerischen Steinen 'eingeklemmt' wird. Sie müssen also einen Ihrer Steine auf beide Seiten (ausgenommen in der Diagonale) eines Steines des Computers bringen. Sodann wird der Stein aus dem Spiel genommen. Ein Stein wird nicht geschlagen, wenn er zwischen zwei gegnerische Steine gezogen wird.

Es gibt keine zusätzlichen Züge als Belohnung für einen eingenommenen Stein. Im ursprünglichen Spiel gewinnt man, wenn man alle Steine des Gegners vom Brett entfernt hat. Das wäre jedoch ein sehr langsame und langwieriges Spiel. Es würde, gegen den Computer gespielt, eine Ewigkeit dauern. Daher habe ich das Programm so gestaltet, daß der erste Spieler, der sechs Steine schlägt, das Spiel gewinnt.

Sie werden finden, daß das Spiel Sie am besten lehrt, es zu spielen, wenn Sie die Computerzüge beobachten. Vergessen Sie nicht, daß Sie über Ihre eigenen Steine und die des Computers springen dürfen. Das stellt eine empfohlene Taktik dar, um am Beginn des Spieles die Steine auf das Brett hinaus und es damit in Fluß zu bringen.

Hier ist das Listing von Shogun:

```

10 REM Shogun
20 REM Ihr Computer muss in CAPS LOCK Modus sein
30 GOSUB790
40 GOSUB90
50 GOSUB460
60 GOSUB430
70 GOSUB430
80 GOTO40
90 REM SCLAGEN
100 A=99
110 IF A(A)<>C THEN GOTO190
120 IF A(A-10)=L THEN IF A(A-9)=M THEN IF A(A-8)=C THEN B=A-10:GOTO350
130 IF A(A-10)=L THEN IF A(A-11)=M THEN IF A(A-12)=C THEN B=A-10:GOTO350
140 IF A(A-10)=L THEN IF A(A+11)=M THEN IF A(A+12)=C THEN B=A-10:GOTO350
150 B=1
160 IF A+2*(B)<11 OR A+2*(B)>99 THEN GOTO180
170 IF A(A+C(B))=L AND A(A+2*(B))=M AND A(A+3*(B))=C THEN A(A+2*(B))=L:CS=CS+1:GOTO340
180 IF B<4 THEN B=B+1:GOTO160
190 IF A>11 THEN A=A-1:GOTO110
200 REM Nicht geschlagen
210 ZAEHL=0
220 ZAEHL=ZAEHL+1
230 A=RND(1)*89+11
240 IF A(A)=C THEN GOTO270
250 IF ZAEHL<200 THEN GOTO220
260 PRINT"Meister des Shogun,":PRINT"der Sieg gehoert Ihnen!":END
270 B=1
280 IF A+2*(B)<11 THEN GOTO300
290 IF (A(A+C(B))=C OR A(A+C(B))=M) AND A(A+2*(B))=L THEN B=A+2*(B):GOTO350

```

```

300 IF A(A+C(B))=L THEN GOTO330
310 IF B<4 THEN B=B+1:GOTO280
320 GOTO250
330 REM Computerzug
340 B=A+C(B)
350 B1=B-10*INT(B/10)
360 A(B)=C:A(A)=L
370 IF B1>7 THEN GOTO390
380 IF A(B+1)=M AND A(B+2)=C THEN A(B+1)
=L:CS=CS+1
390 IF B1<3 THEN GOTO410
400 IF A(B-1)=M AND A(B-2)=C THEN A(B-1)
=L:CS=CS+1
410 IF A>89 THEN GOTO430
420 IF A(B+10)=M AND A(B+20)=C THEN A(B+
10)=L:CS=CS+1
430 IF A<29 THEN RETURN
440 IF A(B-10)=M AND A(B-20)=C THEN A(B-
10)=L:CS=CS+1
450 RETURN
460 REM Spielbrettausdruck
470 CLS:REM Fuegen Sie hier HOME ein, we
nn Ihr Computer es aufweist
480 PRINT TAB(15);"1 2 3 4 5 6 7 8 9"
490 FOR I=90 TO 10 STEP -10
500 PRINT TAB(13)CHR$(I/10+64);" ";
510 FOR N=1 TO 9
520 PRINT CHR$(A(I+N)):" ";
530 NEXT N
540 PRINT CHR$(I/10+64)
550 NEXT I
560 PRINT TAB(15);"1 2 3 4 5 6 7 8 9"
570 PRINT:PRINT,"Computer: ";CS
580 PRINT," Mensch: ";MS
590 IF CS>6 OR MS>6 THEN GOTO610
600 RETURN
610 IF CS>MS THEN PRINT:PRINT"ich habe g
ewonnen!":END
620 PRINT:PRINT"Sie haben gewonnen!":END

```

```

630 REM Spielerzug
640 INPUT"Von (Buchst,Zahl) ";A$
650 IF A$="E" THEN END
660 IF LEN(A$)<>2 THEN GOTO640
670 PRINT"Von ";A$;" nach ";:INPUT B$
680 IF LEN(B$)<>2 THEN GOTO670
690 A=10*(ASC(A$)-64)+VAL(RIGHT$(A$,1))
700 B=10*(ASC(B$)-64)+VAL(RIGHT$(B$,1))
710 Y=VAL(RIGHT$(B$,1))
720 A(B)=M:A(A)=L
730 IF A(B+1)=C AND A(B+2)=M AND Y<=7 TH
EN A(B+1)=L:MS=MS+1
740 IF A(B-1)=C AND A(B-2)=M AND Y>=3 TH
EN A(B-1)=L:MS=MS+1
750 IF B>79 THEN GOTO770
760 IF A(B+10)=C AND A(B+20)=M THEN A(B+
10)=L:MS=MS+1
770 IF B>=31 THEN IF A(B-10)=C AND A(B-2
0)=M THEN A(B-10)=L:MS=MS+1
780 RETURN
790 REM Initialisieren
800 REM Basiszahl fuer Zufallszahlengene
rator
810 PRINT"Druucken Sie eine Taste"
820 N=1
830 N=N+1
840 IF INKEY$(10)="" THEN GOTO830
850 REM RANDOMIZE N
860 CLS
870 DIM A(129),C(4)
880 M=77:C=67:L=42
890 FOR Z=11 TO 29
900 IF Z=20 THEN Z=21
910 A(Z)=M
920 NEXT Z
930 FOR Z=31 TO 79
940 IF 10*INT(Z/10)=Z THEN Z=Z+1
950 A(Z)=L
960 NEXT Z

```

```

970 FOR Z=81 TO 99
980 IF Z=90 THEN Z=91
990 A(Z)=C
1000 NEXT Z
1010 MS=0
1020 CS=0
1030 FOR Z=1 TO 4
1040 READ C(Z)
1050 NEXT Z
1060 DATA -10,-1,1,10
1070 GOSUB450
1080 RETURN

```

Awari

Awari gehört zu den Stein-in-Gruben-Spielen, die allgemein unter dem Namen 'Mancala' bekannt sind. Das Spiel, das von Afrika bis zu den Philippinen gespielt wird, hält nun durch einen sehr gerissenen Computergegner Einzug in Ihr Heim.

Wie Sie aus dem Demonstrationsspiel, das nach diesen Ausführungen beginnt, ersehen können, beginnt das Spiel mit sechs 'Gruben' oder 'Löchern' (die Buchstaben A bis F und G bis L), die jeder Spieler vor sich hat. Ihre Löcher sind jene von G bis L. Jedes Loch enthält zu Spielbeginn drei Samen. Indem Sie nun eine Grube Ihrer Seite auswählen, nehmen Sie alle Samen aus ihr heraus und säen sie in jede folgende Grube, an der Sie im Uhrzeigersinn vorbeikommen. In die Gruben an den beiden Enden des Brettes, die zu Spielbeginn Null anzeigen, können Sie keinen Samen säen.

Wenn der letzte Samen gegenüber einer leeren Grube zu liegen kommt, so bekommen Sie alle Samen aus dieser letzten Grube. Diese werden nun in Ihr 'Heimfeld', die Null auf der linken Seite des Brettes, gelegt. Das Heimfeld des Computers ist die Null auf der rechten Seite des Brettes.

Das Spiel geht so lange weiter, bis eine Seite total leer ist, so daß ein Spieler nicht mehr ziehen kann. Der Spieler, der zu diesem Zeitpunkt die meisten Samen in seinem Heimfeld hat, ist der Sieger. Mit etwas Übung werden Sie lernen, den Computer zu besiegen, obwohl er in diesem Spiel sehr gut ist. Erwarten Sie nicht zu viele Siege in den ersten Spielen. Wenn Sie den Zug des Computers nicht wissen wollen, bevor er ihn ausführt, entfernen Sie INPUT Z\$ in Zeile 270. Sie bekommen eine maximale Spielgeschwindigkeit, wenn Sie die Schleife in Zeile 40 löschen (der Computer spielt allerdings auch in dieser Form sehr flink).

Hier sind die Anfangszüge aus einem Spiel, das ich gegen meinen eigenen Computer spielte:

A	B	C	D	E	F		A	B	C	D	E	F
3	3	3	3	3	3		3	3	3	3	0	4
0						0	0					0
3	3	3	3	3	3		3	3	3	3	4	4
L	K	J	I	H	G		L	K	J	I	H	G

Ich ziehe von E

Von welchem Loch aus ?H

A	B	C	D	E	F		A	B	C	D	E	F
3	3	3	3	0	4		3	0	4	4	1	4
0						0	0					0
4	4	4	4	0	4		4	4	4	4	0	4
L	K	J	I	H	G		L	K	J	I	H	G

Ich ziehe von B

Von welchem Loch aus ?I

A	B	C	D	E	F		A	B	C	D	E	F
4	0	4	4	1	4		0	1	5	5	2	4
0						0	0					0
5	5	5	0	0	4		5	5	5	0	0	4
L	K	J	I	H	G		L	K	J	I	H	G

Ich ziehe von A

Von welchem Loch aus ?L

A	B	C	D	E	F		A	B	C	D	E	F
1	2	6	6	3	4		1	2	0	0	4	5
0						0	0					7
0	5	5	0	0	4		0	5	5	1	1	5
L	K	J	I	H	G		L	K	J	I	H	G

Ich ziehe von C

Von welchem Loch aus ?G

Hier ist das Listing von Awari:

```

10 REM Awari
20 GOSUB930:REM Initialisieren
30 GOSUB770:REM Spielbrett ausdrucken
40 FOR P=1 TO 500:NEXT P
50 GOSUB160:REM Computerzug
60 GOSUB770:REM Spielbrett ausdrucken
70 GOSUB620:REM Spielerzug
80 CW=0:MW=0
90 FOR C=1 TO 12
100 IF C<7 THEN CW=CW+A(C)
110 IF C>6 THEN MW=MW+A(C)
120 NEXT C
130 IF CW=0 OR MW=0 THEN GOTO510
140 GOTO30
150 REM *****
160 REM Computerzug
170 GM=0:C=0
180 C=C+1
190 IF A(C)=0 THEN GOTO180
200 Z=C+A(C)
210 IF Z>12 THEN Z=Z-12
220 IF Z>6 THEN IF A(Z-6)<>0 AND A(Z)=0
AND A(Z-6)>GM THEN GM=C
230 IF Z<7 THEN IF A(Z+6)<>0 AND A(Z)=0
AND A(Z+6)>GM THEN GM=C
240 IF C<6 THEN GOTO180
250 IF GM=0 THEN GOTO370
260 C=GM
270 PRINT"Ich ziehe von ";CHR$(64+C):INP
UT Z#
280 FOR Z=C TO C+A(C)
290 IF Z>12 THEN A(Z-12)=A(Z-12)+1
300 IF Z<13 THEN A(Z)=A(Z)+1
310 NEXT Z
320 Z=C+A(C)-1:IF Z>12 THEN Z=Z-12
330 A(C)=0
340 B(2)=B(2)+A(13-Z):A(13-Z)=0
350 RETURN

```

```

360 REM *****
370 REM Zug ohne Punkte
380 W=0
390 W=W+1
400 C=INT(RND(1)*6)+1
410 IF A(C)<>0 THEN GOTO440
420 IF W<20 THEN GOTO390
430 GOTO510
440 PRINT"Ich ziehe von ";CHR$(64+C):INP
UT Z#
450 FOR Z=C TO C+A(C)
460 IF Z<13 THEN A(Z)=A(Z)+1
470 IF Z>12 THEN A(Z-6)=A(Z-6)+1
480 NEXT Z
490 A(C)=0:GOTO350
500 REM *****
510 REM Spielende
520 GOSUB770
530 PRINT:PRINT"Das ist das Ende des Spi
eles"
540 PRINT
550 IF B(1)>B(2) THEN PRINT"Sie sind der
Sieger!"
560 IF B(1)<B(2) THEN PRINT"Und ich bin
der Sieger!"
570 IF B(1)=B(2) THEN PRINT"Sieht nach U
nentschieden aus!"
580 PRINT:PRINT"Meine Punkte: ";B(2)
590 PRINT"Ihre Punkte: ";B(1)
600 END
610 REM *****
620 REM Spielerzug
630 INPUT"Von welchem Loch aus ";A#
640 B=ASC(A#)-64
650 IF B<7 OR B>12 THEN GOTO630
660 C0=B:Z=B+A(B):IF Z>12 THEN Z=Z-12
670 R=A(Z)
680 FOR Z=B TO B+A(B)
690 IF Z>12 THEN A(Z-12)=A(Z-12)+1

```

```

700 IF Z<13 THEN A(Z)=A(Z)+1
710 NEXT Z
720 Z=B+A(B)-1:IF Z>12 THEN Z=Z-12
730 IF R=0 THEN B(1)=B(1)+A(13-Z):A(13-Z)=0
740 A(CO)=0
750 RETURN
760 REM *****
770 REM Spielbrett ausdrucken
780 CLS:PRINT:PRINT:PRINT
790 PRINT "  A B C D E F":PRINT "
";
800 FOR C=1 TO 6
810 PRINT " ";A(C);" ";
820 NEXT C
830 PRINT:PRINT ;B(1)"
";B(2):PRINT " ";
840 FOR C=12 TO 7 STEP -1
850 PRINT " ";A(C);" ";
860 NEXT C
870 PRINT:PRINT "  L K J I H G"
880 PRINT:PRINT
890 RETURN
900 PRINT"Ich ziehe von "CHR$(64+GM)
910 C=GM
920 REM *****
930 REM Initialisieren
940 CLS
950 REM RANDOMIZE
960 DIM A(12).B(2)
970 FOR C=1 TO 12
980 A(C)=3
990 NEXT C
1000 RETURN

```

Springeraufmarsch

Springeraufmarsch gehört zu den wenigen Spielen, von denen ich behaupten kann, sie ganz erfunden zu haben. Ich habe mir zur Gewohnheit gemacht, Bücher zu lesen, um Ideen zu bekommen. Daher nehme ich oft, wenn ich mich nach Entspannung sehne, ein Buch mit auf meine Spaziergänge in den Wäldern rund um London. Als ich eines Sonntagnachmittags nahe Ruislip Lido spazierend ging, dachte ich über ein eben gelesenes Kapitel in einem Buch nach. Dieses behauptete, daß Schach von einem indischen Spiel abstammt, bei dem der Wurf eines Würfels bestimmte, mit welcher Figur man zu ziehen hatte. Da gebar ich die Idee von einem Spiel, in dem alle Figuren Schach-Springer sind, und ein Würfel verwendet wird, um den zu bewegendenden Springer zu bestimmen. Aus dieser Idee entstand **Springeraufmarsch**. Es wird auf einem Brett mit sieben mal sieben Feldern gespielt. Der Computer schreibt Ihnen die Figur vor, die Sie bewegen müssen. Er bedient sich dazu einer zweiziffrigen Koordinate (die erste Ziffer ist die horizontale und die zweite die vertikale Koordinate). Danach überläßt er Ihnen die Entscheidung, wohin Sie ziehen wollen. Alle Figuren werden wie die Springer eines Schachspiels bewegt. Auch werden alle Züge untersucht, ob sie überhaupt möglich sind, und dann erst freigegeben. Man schlägt, wie bei Schach, indem man auf eine gegnerische Figur springt. Der Gewinner ist, wer als erster fünf der sieben Figuren des Gegners geschlagen hat. Es gibt keine Mehrfachzüge. Sie werden herausfinden, daß der Computer gut und schnell spielt.

Hier ist das Listing von Springeraufmarsch:

```

10 REM Springeraufmarsch
20 GOSUB760:REM Initialisieren
30 GOSUB500:REM Spielbrett ausdrucken
40 IF ME=5 OR CO=5 THEN GOTO680:REM Spielende
50 GOSUB300:REM Computerzug
60 GOSUB500:REM Brett ausdrucken
70 IF ME=5 OR CO=5 THEN GOTO680:REM Spielende
80 GOSUB110:REM Spielerzug
90 GOTO30
100 REM *****
110 REM Spielerzug
120 @=0
130 R=INT(RND(1)*66)+11

```

```

140 Q=Q+1
150 IF Q=500 THEN GOTO680
160 IF M(R)<>77 THEN GOTO130
170 PRINT"Sie muessen mit dem Stein auf
;"R;" ziehen"
180 INPUT N
190 IF N=99 THEN Q=500:GOTO680
200 REM *** Gueltigkeit des Zuges testen
***
210 P=0
220 CT=1
230 IF R+Z(CT)=N THEN P=1
240 IF CT<8 THEN CT=CT+1:GOTO230
250 IF P=0 THEN PRINT"Ungueltiger Zug":G
OTO180
260 IF M(N)=67 THEN ME=ME+1:PRINT"Guter
Zug!":FOR I=1 TO 500:NEXT I
270 M(R)=46:M(N)=77
280 RETURN
290 REM *****
300 REM Computerzug
310 Q1=0
320 Q1=Q1+1
330 K=INT(RND(1)*66)+11
340 IF Q1=500 THEN GOTO680
350 IF M(K)<>67 THEN GOTO320
360 PRINT"Ich muss mit dem Stein auf ";K
;" ziehen"
370 W=1
380 IF K+Z(W)<11 OR K+Z(W)>77 THEN GOTO4
00
390 IF M(K+Z(W))=77 THEN PRINT"Erwischt!
!":C0=C0+1:FOR P=1 TO 500:NEXT P:GOTO450
400 IF W<8 THEN W=W+1:GOTO380
410 W=1
420 IF (K+Z(W)<11 OR K+Z(W)>77) AND W<8
THEN W=W+1:GOTO420
430 IF M(K+Z(W))<>46 AND W<8 THEN W=W+1:
GOTO420

```

```

440 IF W=8 AND M(K+Z(W))<>46 THEN Q1=500
:GOTO680
450 X=K:Y=K+Z(W)
460 M(X)=46:M(Y)=67
470 RETURN
480 REM *****
490 REM Spielbrett ausdrucken
500 CLS:PRINT:PRINT
510 PRINT TAB(8);"Meine Punkte ":C0
520 PRINT TAB(8);"und Ihre      ":ME
530 PRINT
540 PRINT TAB(8);"1 2 3 4 5 6 7"
550 PRINT TAB(8);"-----"
560 FOR I=70 TO 10 STEP -10
570 PRINT TAB(5);I/10;" ";
580 FOR N=1 TO 7
590 PRINT CHR$(M(I+N));" ";
600 NEXT N
610 PRINT :I/10
620 NEXT I
630 PRINT TAB(8);"-----"
640 PRINT TAB(8);"1 2 3 4 5 6 7"
650 PRINT
660 RETURN
670 REM *****
680 REM Spielende
690 GOSUB490
700 IF ME=5 THEN PRINT"Sie haben mich ge
schlagen, Mensch"
710 IF C0=5 THEN PRINT"Dieser Sieg ist d
er erste Schritt":PRINT"in unserem Plan, d
ie":PRINT"Weltherrschaft zu uebernehmen!"
720 IF Q=500 THEN PRINT"Ich nehme Ihre A
ufgabe an"
730 IF Q1=500 THEN PRINT"Ich gebe das Sp
iel auf"
740 END
750 REM *****
760 REM Initialisieren

```

```

770 CLS:PRINT"Bitte warten Sie ... Mensc
h"
780 DIM M(99),Z(8)
790 X=0:O1=0:O=0
800 REM RANDOMIZE
810 ME=0:C0=0:REM Punkte
820 FOR A=1 TO 99
830 IF A>77 OR A=70 OR A=60 OR A=68 OR A
=69 OR A=50 OR A=59 OR A=40 OR A=48 OR A=4
9 THEN GOTO880
840 IF A=30 OR A=38 OR A=39 OR A=20 OR A
=28 OR A=29 OR A=18 OR A=19 OR A<11 THEN G
OTO880
850 M(A)=46
860 IF A>70 AND A<78 THEN M(A)=67
870 IF A>10 AND A<18 THEN M(A)=77
880 NEXT A
890 FOR A=1 TO 8:READ Z(A):NEXT A
900 DATA -8,-21,-12,-19,19,12,21,8
910 RETURN

```

Reversi/Othello

Das letzte Programm ist **Reversi**, das auch oft **Othello** genannt wird. Es wurde im späten 18. Jahrhundert erfunden. Man spielt es auf einem Brett mit acht mal acht Feldern. Es werden Steine mit verschiedenen Farben auf ihrer Ober- und Unterseite verwendet. Das Spiel beginnt, indem ein Spieler vier Steine auf die Quadrate in der Mitte des Brettes legt. Von da an legen die Spieler abwechselnd je einen Stein auf das Brett. Liegen gegnerische Steine zwischen zwei eigenen, dann werden diese Steine 'umgedreht' und zu eigenen Steinen gemacht. Und so funktioniert es. Nehmen wir an, eine Reihe Steine sieht so aus:

XXXX
...und Sie wollen Ihren Stein (das O) an das Ende der Reihe legen:
OXXO
...dann würden die Computersteine umgedreht, so daß die Zeile nach Ihrem Zug so aussehen würde:
OOOO

Othello ist ein registrierter Markenname von Gabriel Industries, Inc., USA und Mine of Information, UK.

Das Spiel schreitet so lange fort, bis alle Felder des Brettes gefüllt sind, oder einer der Spieler nicht mehr ziehen kann. Wie Sie sehen, kann das Glück in diesem Spiel sehr launenhaft sein. So können lange Zeilen, wie z.B. auf den Diagonalen, mit einem einzigen Zug geändert werden. Sollten Sie einmal nicht ziehen können, teilen Sie dies dem Computer durch die Eingabe von Null mit.

Reversi wird Ihnen sicher Spaß machen. In einem Artikel im 'Creative Computing Magazine' (Juni 1981, S.188) behauptet David Levy, daß es eines der besten Spiele ist, die je erfunden wurden. Die Regeln sind nämlich in wenigen Minuten zu erlernen, jedoch kann es Jahre dauern, bis man zu einem Meister in diesem Spiel wird. Weiter stellt er fest, daß es, obwohl sehr viel einfacher als Schach, viel komplizierter als Dame ist.

Obwohl das moderne Spiel **Othello** heißt, es wurde 1971 von Goro Hasegawa in Tokio erfunden, liegt der einzige Unterschied zwischen diesem und **Reversi** in der Aufstellung der ersten vier Steine zu Beginn des Spieles (Sie können dies anhand des Demonstrationsspieles nach diesen Ausführungen sehen).

Im London des Jahres 1880 behaupteten zwei Männer – Lewis Waterman und John W. Mollett – die Erfinder des Spieles zu sein. Stephen Kimmel meint in einem Artikel im 'Creative Computing Magazine' vom Juli 1981, S.94, daß seiner Meinung nach Waterman den größeren Anspruch hat, als Erfinder von Reversi zu gelten. Er hat nämlich als erster Details in einer Serie von Artikeln im Magazin 'The Queen' (das sich hauptsächlich mit interessanten Geschneissen für Damen' beschäftigte) veröffentlicht.

Unabhängig davon, wem nun die Ehre zukommt, das Spiel erfunden zu haben, Hasegawa, Waterman oder Mollett, ist es ein tolles Spiel. Das Programm, basierend auf einem von Graham Charlton geschriebenen, hat eine kraftvolle Verteidigung. Diese Demonstration zeigt die Beginnphasen des Spieles:

COMPUTER IST X
MENSCH IST O

1	2	3	4	5	6	7	8
1	1
2	2
3	3
4	.	.	X	O	.	.	4
5	.	.	O	X	.	.	5
6	6
7	7
8	8
1	2	3	4	5	6	7	8

Computer: 2

Mensch: 2

COMPUTER IST X
MENSCH IST O

1	2	3	4	5	6	7	8
1	1
2	2
3	3
4	.	.	X	O	.	.	4
5	.	X	X	X	.	.	5
6	6
7	7
8	8
1	2	3	4	5	6	7	8

Computer: 4

Mensch: 1

COMPUTER IST X
MENSCH IST O

1	2	3	4	5	6	7	8
1	1
2	2
3	3
4	.	.	X	O	.	.	4
5	.	X	O	X	.	.	5
6	.	O	6
7	7
8	8
1	2	3	4	5	6	7	8

Computer: 3

Mensch: 3

COMPUTER IST X
MENSCH IST O

1	2	3	4	5	6	7	8
1	1
2	2
3	3
4	.	.	X	O	.	.	4
5	.	X	X	X	.	.	5
6	.	O	X	.	.	.	6
7	7
8	8
1	2	3	4	5	6	7	8

Computer: 5

Mensch: 2

Und hier das Listing von Reversi:

```

10 REM REVERSI
20 GOTO740
30 PRINT"MEIN ZUG..."
40 S=0:T=X:M=0
50 FOR A=2 TO 9:FOR B=2 TO 9
60 IF A(A,B)<>46 THEN GOTO210
70 Q=0
80 FOR C=-1 TO 1:FOR D=-1 TO 1
90 K=0:F=A:G=B
100 IF A(F+C,G+D)<>S THEN GOTO130
110 K=K+1:F=F+C:G=G+D
120 GOTO100
130 IF A(F+C,G+D)<>T THEN GOTO150
140 Q=Q+K
150 NEXT D:NEXT C
160 IF A=2 OR A=9 OR B=2 OR B=9 THEN Q=Q/2
*2
170 IF A=3 OR A=8 OR B=3 OR B=8 THEN Q=Q/2
180 IF (A=2 OR A=9) AND (B=3 OR B=8) OR
(A=3 OR A=8) AND (B=2 OR B=9) THEN Q=Q/2
190 IF Q<M OR (RND(1)<.3 AND Q=M) THEN G
OTO210
200 M=Q:P=A:N=B
210 NEXT B:NEXT A
220 IF M=0 AND R=0 THEN GOTO690
230 IF M=0 THEN GOTO250
240 GOSUB580
250 GOSUB370
260 INPUT"GEBEN SIE IHREN ZUG EIN ":R
270 REM 0 UM ZU PASSEN
280 S=X:T=O:REM BUCHSTABE O
290 IF R=0 THEN GOTO350
300 IF R<11 OR R>88 THEN GOTO260
310 R=R+11
320 P=INT(R/10)
330 N=R-10*P
340 GOSUB580

```



```

350 GOSUB370
360 GOTO30
370 REM SPIELBRETT AUSDRUCKEN
380 C=0:M=0
390 CLS:REM ODER 'HOME'
400 PRINT
410 PRINT"      COMPUTER IST X"
420 PRINT"      MENSCH IST O"
430 PRINT
440 PRINT TAB(2);"1 2 3 4 5 6 7 8"
450 FOR B=2 TO 9:PRINT ;B-1;" ";
460 FOR D=2 TO 9
470 PRINT CHR$(A(B,D));" ";
480 IF A(B,D)=X THEN C=C+1
490 IF A(B,D)=O THEN M=M+1
500 NEXT D
510 PRINT ;B-1
520 NEXT B
530 PRINT TAB(2);"1 2 3 4 5 6 7 8"
540 PRINT:PRINT
550 PRINT"Computer: ";C
560 PRINT:PRINT"Mensch: ";M
570 RETURN
580 FOR C=-1 TO 1:FOR D=-1 TO 1
590 F=P:G=N
600 IF A(F+C,G+D)<>S THEN GOTO630
610 F=F+C:G=G+D
620 GOTO600
630 IF A(F+C,G+D)<>T THEN GOTO670
640 A(F,G)=T
650 IF P=F AND N=G THEN GOTO670
660 F=F-C:G=G-D:GOTO640
670 NEXT D:NEXT C
680 RETURN
690 GOSUB370
700 IF C>M THEN PRINT"ICH BIN DER MEISTE
R!"
710 IF M>C THEN PRINT"SIE SIND DER MEIST
ER!"

```

```

720 IF M=C THEN PRINT"ES IST EIN UNENTSC
HIEDEN!"
730 END
740 CLS
750 X=ASC("X"):O=ASC("O"):REM BUCHSTABE
O, NICHT O
760 DIM A(10,10)
770 FOR B=1 TO 10:FOR C=1 TO 10
780 IF B<>1 AND C<>1 AND B<>10 AND C<>10
THEN A(B,C)=ASC(".")
790 NEXT C:NEXT B
800 A(5,5)=X:A(6,6)=X
810 A(6,5)=O:A(5,6)=O
820 R=0
830 PRINT"WOLLEN SIE BEGINNEN,"
840 PRINT"J ODER N ";
850 INPUT W$
860 GOSUB370
870 IF W$="J" OR W$="j" THEN GOTO260
880 GOTO30

```

Abenteuer

Ab in die Gründe der Finsternis! Bewaffnet nur mit Ihrem Computer und Ihrem kühnen Verstand haben Sie beschlossen, es mit den Mächten der Finsternis aufzunehmen.

Sie könnten auf einer verödeten Insel, in einem von Spuk heimgesuchten Haus, in einem Verlies tief im Inneren der Erde oder gefangen in einem Höhlenlabyrinth auf einem Planeten, irgendwo in einem Sternensystem weit weg sein.

Das und tausende mehr sind die Schauplätze, an denen die Abenteuerspiele stattfinden. Kommen Sie jetzt mit uns, wenn wir die Aufregungen der Abenteuerwelt erleben.

Das Wort Abenteuer bezeichnet jenen Typ von Computerspiel, bei dem sich der Spieler durch eine andere Wirklichkeit bewegt. In dieser „anderen Welt“ müssen Ungeheuer bekämpft, Schätze gefunden, Karten gezeichnet und Rätsel gelöst werden.

Viele Leute meinen, daß die Abenteuerspiele zu den aufregendsten Spielen zählen, die mit einem Computer gespielt werden können. Wenn Sie sich für Spiele interessieren, in denen Sie eine Rolle verkörpern, wie TSR Hobbys **Dungeon and Dragons** (das Spiel und der Name sind von TSR gesetzlich geschützt), dann wird Ihnen das Spielen von Abenteuern sicherlich viel Freude bereiten.

Eine Eigenschaft von echten Abenteuerspielen ist, daß die von ihnen geschaffene Wirklichkeit konstant ist. Damit meine ich, daß die Welt innerhalb des Abenteuerprogramms statisch ist, abgesehen von den speziellen Vorfällen in dem Spiel, so wie z.B. ein Erdbeben oder ein Zauberspruch. Diese Welt verändert sich also nicht zufällig. Wenn Sie auf dem Weg durch einen Wald also an einer knorrigen alten Eiche mit den Initialen Ihres Freundes in der Rinde vorbeikommen, so sollte diese Eiche auch auf Ihrem Rückweg noch an dem gleichen Ort stehen. In einem gut aufgebauten Abenteuer bleiben die Flüsse in ihren Betten, die Mauern von Verliesen bewegen sich nicht geisterhaft jedes Mal, wenn Sie ihnen den Rücken kehren, und Dinge, die Sie in einer Höhle deponierten, tauchen nicht plötzlich Hunderte von Metern davon entfernt wieder auf (außer sie wurden mit einer geheimnisvollen Gabe bedacht, die ihnen dies ermöglicht).

Das Erstellen einer Karte gehört zu den wirklich notwendigen Begabungen eines Abenteuerspielers. Sich den Weg durch eine imaginäre (aber stabile) Umwelt zu erkämpfen, Ungeheuer zu attackieren, Schätze zu sammeln, Rätsel zu lösen, während Sie Stiegen steigen und Stromschnellen entlanggetrieben werden, verästelter Tunnelwerk zu erforschen, sich in Labyrinthen zu verirren usw., ist nur dann faszinierend, wenn diese Welt kartographisch darstellbar ist. So sollten Sie z.B. erwar-

ten können, daß die Ausgänge eines Tunnels immer an denselben Stellen liegen. Dadurch können Sie, wenn Sie im Laufe des Spieles von einer anderen Richtung wieder an einen Ausgang oder Eingang kommen, die von Ihnen schon gesammelten Hinweise auf den Verlauf des Tunnels verwenden. Alle 'Welten', die von den Spielern in diesem Kapitel des Buches geschaffen werden, sind kartographisch erfaßbar. Sie geben Ihnen ab und zu sogar einen Einblick über die Vorgänge aus der Vogelperspektive.

Das Rollenspiel ist ein Teil des Lebens. Unbestritten gibt es in Ihrem Alltagsleben, bis zu einer gewissen Grenze, Situationen, die Sie zwingen, eine Rolle anzunehmen. Unglücklicherweise erfordern die meisten Berufe ein Verhalten, als wären die Bedürfnisse und Ziele der Firma die der Belegschaft. Genauso, als wäre der Enthusiasmus, den Sie jeden Tag auf die Kunden und Ihre Mitarbeiter ausstrahlen, Ihr eigener tief innerer usw. Sie werden herausfinden, daß Sie gelernt haben, in neuen, schwierigen Situationen Ihre Spannung durch Selbstsicherheit zu kompensieren. Mühelos können Sie dann erkennen – wenn Sie es gekonnt machen –, daß Ihre Umwelt die von Ihnen angenommene Rolle als Ihre eigene erkennt.

Das Rollenspiel in einer erdachten Umwelt eines Abenteuerspieles ist weitaus fesselnder und auch sehr weit von Ihrem Alltagsleben entfernt. Wenn Sie es genau betrachten, dann sind auch Spiele wie 'Räuber und Gendarm' eine Übung im Rollenspiel.

Wir alle haben also schon Erfahrung im Rollenspiel. Auch wenn diese Erfahrung nicht so unterhaltsam ist, wie der langwierige Versuch, eine hübsche Prinzessin aus einem Traum zu befreien. Die Abenteuer in diesem Buch sind auf keinen Fall so ausgefeilt wie diejenigen, die Sie für Ihren Computer kaufen können. Aber sie sind sorgfältiger und zusammenhängender organisiert als viele, die mir schon untergekommen sind. Als zusätzlichen Vorteil geben Sie Ihnen eine Richtschnur, wie Sie Ihre eigenen Abenteuerprogramme schaffen können.

Rollenspiele, die noch andere Personen miteinbeziehen, wie z.B. TSR's D&D, das ich schon erwähnte, haben normalerweise eine große Anpassungsfähigkeit der Spielentwicklung im Verlaufe des Spieles. Unter der Kontrolle eines guten Verliesaufsehers kann sich so ein Rollenspiel in eine Richtung entwickeln und Situationen hervorbringen, die man sich zu Beginn des Spieles nicht erträumt hätte. Leider kann ein Computerrollenspiel nicht so anpassungsfähig sein. Wenn man ein Abenteuerspiel mit einem Computer spielt, versucht man, Rätsel zu lösen, die der Programmierer geschaffen hat. Der Computer kann nur dazu verwendet werden, die Anfangsphase des Spieles zu erstellen, zu bestimmen helfen, wo bestimmte Objekte deponiert werden und vielleicht, wohin eine bestimmte Stiege führt. Der Programmierer jedoch gibt der Umgebung im Programm ihre grobe Gestalt.

In unserem Programm **Festung der Zwergenfürsten** können Sie dies sehen. Hier müssen Sie sich in einem Labyrinth zurechtfinden, um den dort versteckten Zwergenschatz zu finden. Eigentlich versteckt der

Computer den Schatz und ändert den zweidimensionalen Irrgarten bis zu einem gewissen Grad ab, aber das ist auch schon alles. Natürlich gibt der Computer die Rückmeldungen an den Spieler, und seine Aufgabe hierbei ist sehr wertvoll.

Früher wurden Feldzüge am grünen Tisch wiederaufgerollt. Napoleonische Schlachten wurden auf großen Karten mit den Flüssen und Wäldern darauf gewonnen oder verloren, wenn Spieler Schlachtverläufe nachspielten oder neue erfanden. Kriegsspieler fanden, daß der Spaß am Spielen erhöht werden konnte, wenn sie den einzelnen Individuen, mit denen die kämpften, mehr Aufmerksamkeit schenken, als ganzen Gruppen von Soldaten. Diese Aufmerksamkeit auf die individuelle Identität führt direkt zu den Rollenspielen, die wir heute haben, da die Spieler entdeckten, wie befriedigend es ist, die Rolle einer bestimmten Figur zu übernehmen.

Unsere Verliese und Drachen entwickelten sich direkt aus diesen Kriegsspielen, in denen die Spieler in individuelle Persönlichkeiten auf dem Schlachtfeld schlüpfen. Zwei Spieler, Dave Anson und E. Gary Gygax, schufen eine vollkommen abgeschlossene andere Realität. Im Jahre 1974 erschien ihr Werk als ein Drei-Bücher-Set mit dem Titel **Dungeons and Dragons**. Die Spiele wurden ein enormer Erfolg. Heute verbringen hunderttausende Menschen einen großen Teil ihrer Zeit damit, als Geister, Gnome, Zauberer und Hexenmeister in die Tiefen der Erde vorzudringen, zu erforschen, kämpfen, erobern und manchmal auch, um zu sterben.

Die Originalidee von D&D hatte bald eine Fülle von Nachahmern, wie z.B. **Runequest** (Chaosium Ltd., Albany, California), **Bushido** und **Chivalry & Sorcery** (Fantasy Games Unlimited, Roslin, New York), **Tunnels & Trolls** (Flying Buffalo, Inc., Scottsdale, Arizona), **Traveller** (Game Designers Workshop, Bloomington, Illinois) and **Heroes of Olympus** (Task Force Games, Amerillo, Texas).

Will Crowther und Jon Woods schrieben im Jahre 1976 Computergeschichte, als sie der Welt erstes Rollenspiel – einfach 'Abenteuer' genannt – in einen Großcomputer der Stanford-Universität eingaben. (Obwohl die Entwicklung des Programmes heute beiden Männern zugeschrieben wird, hat eigentlich Crowther das Programm geschrieben und Woods es verbessert.) Obwohl dieses Spiel bald zu einem Kult in ganz Amerika wurde, haben sich die Tore für das Abenteuerspiel am Computer erst mit der Einführung der Mikrocomputer geöffnet. Dadurch wurde Millionen Menschen der Zugang zu Computern und zur Computerzeit ermöglicht. Billigere Speicher erlaubten das Entstehen hochentwickelter Abenteuerspiele. Dadurch werden die Wünsche der Abenteuerspieler sehr gut befriedigt.

Crowther und Woods haben wahrscheinlich besser gearbeitet, als sie vermuteten. Trotz der Vielfalt der zur Verfügung stehenden Programme verkauft sich das Original 'Abenteuer' immer noch sehr gut. Mindestens fünf Firmen verkaufen heute Versionen dieses Originals alleine für den Apple-Computer.

Dieses Original wurde in FORTRAN geschrieben. Als es in eine freie Programmaustausch-Bibliothek eingegliedert wurde (DECUS = Digital Equipment Computer User's Society, von deren Mitglieder die meisten DEC's PDP-11-System verwendeten), stieg es bald zum meist verwendeten Programm in dieser Bibliothek auf. Zwei Jahre später wurde ein anderes Abenteuerprogramm, 'Verlies', in die Bibliothek aufgenommen. Dieses Programm spannt die Idee aus, 'Abenteuer' viel weiter als Crowther's Programm. Geschrieben wurde es von Tim Anderson, Marc Blank, Bruce Daniels und Dave Lebling, der gesamten Programming Technology Division der MIT Labors für Computertechnik.

In dem Addison-Wesley-Buch über Apple-Computer-Programme (bearbeitet von Jeffrey Stanton und John Dickey, The Book Co., Lawn-dale, California, 1981) sind der Beschreibung von Fantasie-, Rollen-, Abenteuer- und Kriegsspielen für den Apple II 29 Seiten gewidmet. Ist die Unterstützung auf diesem Gebiet der Programme für diesen Computer so gut, bedeutet das, daß auch andere verbreitete Computer gut versorgt sind.

Obwohl Sie viele Abenteuerspiele für Ihren Computer kaufen können, sollten Sie doch den Nervenkitzel, Ihr eigenes Programm zu schreiben, erleben. Die Programme in diesem Buch geben Ihnen sicherlich eine Basis, auf der Sie weiter aufbauen können. Ich schlage vor, daß Sie sie eingeben, um sie dann nach Herzenslust weiterzuentwickeln (zumindest so lange, bis Sie keinen Speicherplatz mehr haben).

Die Attraktion für Text-Programme hält trotz des Trends zu den graphischen Abenteuerspielen weiter an. Genauso, wie Fernsehen das Radio nicht verdrängen konnte, weil eben die Vorstellungskraft stärker ist als alles, was der Mensch schaffen kann, so haben auch die graphischen Spiele den Wert der auf Text basierenden, klassischen Spiele nicht nehmen können. Bei diesen auf Text basierenden Programmen wird die Vorstellungskraft der Spieler mehr angeregt, phantastische Welten entstehen zu lassen, als jeder noch so kluge Programmierer mit den ihm zur Verfügung stehenden hoch entwickelten Graphikfähigkeiten heutiger Computer fähig wäre zu erschaffen.

Wahrscheinlich werden Sie dennoch der Meinung sein, daß Sie die Darstellungen dieser Texte attraktiver gestalten können, wenn Sie Farbe und Geräusche nach Ihrem Ermessen und den Fähigkeiten Ihres Computers dazumischen. Dadurch erhöhen Sie sicher auch den Spaß am Spielen.

Eine der führenden Firmen, die Abenteuerprogramme herstellen, ist die von Scott Adams gegründete Firma Adventure International. Seine Firma hat die einfallsreichsten Stände des gesamten Spielprogramm-sektors auf Computerefachmessen. Es ist gut zu wissen, daß seine Programme genauso einfallsreich sind wie seine Messestände. Sie sollten unbedingt das Vergnügen erleben, ein Scott-Adams-Abenteuerprogramm unverfälscht gespielt zu haben; und Sie wissen dann auch, wie weit die Entwicklung in dieser Branche schon fortgeschritten ist.

Als ich Scott bei einer Computermesse in Atlanta begegnete, fand ich, daß er ein ruhiger, freundlicher und reservierter Mann war. Irgendwie habe ich ihn mir eher wie einen Krieger aus einem seiner Verliese oder als einen Drachen aus einem seiner Höhlensysteme vorgestellt. Etwas in seinem Benehmen jedoch entsprach meinen Vorstellungen. Es war Halloween, und Scott und seine Männer waren alle als Charaktere aus ihren Programmen verkleidet. Dieselbe Einstellung zum Leben, die ihn seinen Leuten befehlen läßt, verkleidet zur Arbeit zu kommen, durchdringt auch seine Programme. Vielleicht ist sein Programm **Abenteuerland** ein guter Einstieg, bevor Sie sich seinen anderen Programmen und solchen anderer Firmen zuwenden.

Sowohl Avalon Hill's Programm **Empire of the Over-Mind** als auch Sentient Software's **Cyborg** gehören zu den Klassikern in der Branche und verdienen es, genauer betrachtet zu werden. Diese Aufzählung könnte beinahe unendlich fortgesetzt werden. Natürlich gibt es auch sehr gute neue Programme. Um herauszufinden, was es noch so alles für Sie und Ihren Computer gibt, sollten Sie ein Computermagazin und/oder den 'Spieler' in einem Computer-Club befragen.

Ich glaube, es ist nun Zeit, Sie mit den Abenteuerprogrammen in diesem Buch vertraut zu machen. Wenn Sie mehr über dieses Gebiet wissen wollen, auch um Ihnen beim Schreiben Ihres eigenen Programmes zu helfen, verweise ich Sie noch auf einige Bücher aus dem üppigen Angebot in dieser Richtung:

- Putting Adventure in Adventure Games -
Artikel in Creative Computing, August 1981, von Robert Plamondon
- Graphic Adventures on the Atari -
Artikel in Creative Computing, August 1982, von John Anderson
- Fantasy Games (Teil Eins und Zwei) -
Artikel in Creative Computing, März 1981 (Teil Eins) und Mai 1981 (Teil Zwei), von David Lubar
- An Adventure in Small Computer Game Simulation -
Artikel von Scott Adams (als er noch ganz unbekannt war), in Creative Computing im August 1979
- Kings of the Castle Walls -
Artikel in Practical Computing, März 1982, von Dennis Ellis
- Fantasy Role-Playing Games -
Holmes, J. Eric (Hippocrene Books, Inc., New York, 1981)
- What is Dungeons and Dragons? -
Butterfield John, Parker Philip und Honigmann David (Penguin Books, New York, 1982)
- Dicing with Dragons - An Introduction to Role-Playing Games -
Livingstone Ian (Routledge & Kegan Paul, London Melbourne und Henley)

Festung der Zwergenfürsten

Tief ins Innere der Erde werden Sie kommen, weit hinein ins Zwergen-
hinterland. Gefahr lauert überall, während Sie hinabsteigen, doch Ihre
Gier treibt Sie an. Seitdem Sie in einem staubigen Stapel von unkatalo-
gisierten Schriften in Raum 546B des Britischen Museums eine ver-
gilbte, fast unleserliche Karte eines Goldschatzes der Zwerge fanden,
sind Sie von der Idee besessen, diesen zu finden.

Während Sie in dem Labyrinth nach unten steigen, wird Ihnen klar, daß die Zwergenfürsten, die das Gold hier vor 7.389 Jahren versteckten, schon lange ausgerottet sind. Daher droht Ihnen eher von dem Höhlensystem selbst als von den Wächtern der Festung Gefahr.

Sysleri selbst aus von der wachsenden Lesung sich in Festung der Zwerge fürsten befinden Sie sich in Höhlen, in denen das Gold versteckt ist. Jedesmal, wenn Sie dieses Spiel spielen, kann das Gold an drei verschiedenen Stellen sein. Die einzige Information, die Sie für Ihren Fortschritt erhalten, bekommen Sie von dem Zwerge Quellenmeter, den Sie bei Ihrem Herumirren in Höhlensystemen gefunden haben. Dieser gibt Ihnen nach jedem Zug Hinweise, wie Sie sich in bezug auf das Gold bewegen.

Bevor Sie jedoch aus diesen Informationen klug werden können, müssen Sie lernen, diese richtig zu verstehen.

Als eine andere Information erhalten Sie die trockene Erklärung, in welche Richtung Sie sich von Ihrer momentanen Position aus bewegen können. Nach jedem Zug („Schritt“) zeigt Ihnen der Bildschirm folgendes an:

SCHRITT NUMMER 49

SCHRITT NUMMER
NORDEN: MAUER

NORDEN: MAUER
SÜDEN: OFFEN

SUDEN: OFFEN
OSTEN: MAUER

OSTEN: MAUER
WESTEN: OFFEN

WESTEN. OFFEN
DAS ZWERGENQUELLMETER ZEIGT 10

Dann werden Sie aufgefordert, Ihren nächsten Schritt wie folgt einzugeben:

In welche Richtung wollen Sie gehen?

N – Nord, S – Süd, O – Ost, W – West, H – Hilfe

N-1
? W

Sie können, wie Sie wahrscheinlich schon erraten haben, nicht durch Wände gehen.

Es ist überflüssig zu sagen, daß das Ziel des Spieles das schnelle Finden des Zwergenschatzes ist. Es gibt eine Karte des Höhlensystems, die Sie, um Ihnen zu helfen, von Zeit zu Zeit abrufen können. Die Karte sieht so aus (die Wände als $\#$ und Ihre Position mit $***$):

[illegible]

Doch es gibt zwei Haken, wenn Sie eine Karte dieser Art abrufen. Erstens wird zwar Ihre Position, nicht aber die des Goldes angezeigt. Zweitens werden Sie immer, wenn Sie „H“ eingeben, um einen Blick auf die Karte zu werfen, mit 15 Strafschritten belegt. Deshalb ist es in Ihrem eigenen Interesse, die vom Programm angebotene Hilfe so selten wie möglich in Anspruch zu nehmen.

Obwohl alle Höhlen, die von diesem Programm erstellt werden, einige Gemeinsamkeiten haben (das bewirken die DATA-Anweisungen von 870 bis 960), sind die Höhlen von einem zum anderen Programmablauf doch nicht identisch. Genauso kann auch das Gold, wie ich auch schon vorher erwähnte, an drei verschiedenen Orten in der Höhle sein. Die Ausgabe des Programmes in bezug auf die Richtungen, nach denen Sie sich von Ihrer momentanen Position aus bewegen können, ist so abänderbar, daß Sie ein „dreidimensionales Labyrinth“ erhalten könnten. Von Schritt zu Schritt wissen Sie, in welche Richtung sich die möglichen Durchgänge (d. h. keine Mauern) befinden. Mann würde eine Subroutine brauchen, die ein Bild der Situation unmittelbar vor Ihnen erzeugt. Diese müßte aus einer Reihe von Bildern (eines mit freiem Weg nach vorne, aber Wänden links und rechts von Ihnen; ein anderes mit Gängen links und rechts, dem Weg vor Ihnen aber versperrt; usw.) wählen, um ein kontinuierliches Bild der Szenerie um Sie herum wiederzugeben. Das kann sehr eindrucksvoll sein und ist sicher wert, ausprobiert und in das Spiel integriert zu werden.

Jetzt aber zum Listing von **Festung der Zwerge**, damit Sie endlich nach Ihrem Gold suchen können:

```

10 REM FESTUNG DER ZWergenKoenige
20 GOSUB640
30 GOSUB480
40 REM *****
50 M=M+1
60 CLS:PRINT:PRINT
70 PRINT"Schritt Nr. ";M
80 PRINT
90 PRINT"Norden: ";
100 IF A(D+1,E)=S THEN PRINT "OFFEN"
110 IF A(D+1,E)=X THEN PRINT "Wand"
120 PRINT"Sueden: ";
130 IF A(D-1,E)=S THEN PRINT "OFFEN"
140 IF A(D-1,E)=X THEN PRINT "Wand"
150 PRINT"Osten: ";
160 IF A(D,E+1)=S THEN PRINT "OFFEN"
170 IF A(D,E+1)=X THEN PRINT "Wand"
180 PRINT"Westen: ";
190 IF A(D,E-1)=S THEN PRINT "OFFEN"
200 IF A(D,E-1)=X THEN PRINT "Wand"
210 PRINT
220 PRINT"DAS ZWergenQUELLENMETER ZEIGT
AN ";
230 PRINT ;100*ABS(Z-D)+10*ABS(Y-E)
240 REM *****
250 PRINT:PRINT:PRINT
260 PRINT"In welche Richtung wollen Sie"
270 PRINT"gehen ?"
280 PRINT:PRINT"N - Norden, S - Sueden,"
290 PRINT"D - Osten, W - Westen, H - Hi
1fe"
300 INPUT A$
310 IF A$="N" AND A(D+1,E)=X THEN GOTO30
0
320 IF A$="S" AND A(D-1,E)=X THEN GOTO30
0
330 IF A$="O" AND A(D,E+1)=X THEN GOTO30
0
340 IF A$="W" AND A(D,E-1)=X THEN GOTO30

```

```

0
350 IF A$="H" THEN GOSUB480
360 IF A$="N" THEN D=D+1
370 IF A$="S" THEN D=D-1
380 IF A$="O" THEN E=E+1
390 IF A$="W" THEN E=E-1
400 IF Z=D AND Y=E THEN GOTO430
410 GOTO50
420 REM *****
430 PRINT:PRINT"Sie haben das Zwergengol
d in nur"
440 PRINT:M;" Schritten gefunden!"
450 GOSUB500
460 END
470 REM *****
480 REM Hilfe
490 CLS
500 PRINT
510 PRINT"Norden"
520 FOR B=15 TO 1 STEP -1
530 FOR C=1 TO 15
540 IF A(B,C)=X THEN PRINT"#";
550 IF B=D AND C=E THEN PRINT"*":GOTO57
0
560 IF A(B,C)=S THEN PRINT " ";
570 NEXT C:PRINT:NEXT B
580 PRINT"Sueden"
590 M=M+15
600 FOR J=1 TO 2000:NEXT J
610 CLS:A(D,E)=S
620 RETURN
630 REM *****
640 CLS
650 N=1:PRINT"Druecken Sie eine Taste"
660 N=N+1:IF INKEY$(10)="" THEN GOTO660
670 REM RANDOMIZE N
680 CLS
690 DIM A(15,15)
700 B=INT(RND(1)*3)+1

```

```

710 Z=14:Y=14
720 IF B=2 THEN Y=2
730 IF B=3 THEN Z=2
740 X=1:S=2
750 FOR B=1 TO 15:FOR C=1 TO 15
760 A(B,C)=X:IF RND(1)>.8 THEN A(B,C)=S
770 IF C<2 OR C>14 OR B<2 OR B>14 THEN A
(B,C)=X
780 NEXT C:NEXT B
790 D=2:E=2
800 FOR F=1 TO 68
810 READ B,C
820 A(B,C)=S
830 NEXT F
840 M=-15
850 RETURN
860 REM *****
870 DATA 2,2,2,3,2,4,2,5,2,6,2,7
880 DATA 3,7,4,7,5,7,5,6,5,5,4,5,3,6,3
890 DATA 7,3,7,4,7,5,7,6,7,7,8,7,9,9,8
900 DATA 9,9,10,8,10,7,10,6,10,5,10,4,8,
8
910 DATA 10,3,11,3,12,3,13,3,14,3,14,2,7
,10
920 DATA 6,10,5,10,4,10,3,10,2,10,2,11,2
,12
930 DATA 2,13,2,14,6,11,6,12,6,13,6,14,7
,12
940 DATA 14,12,8,12,8,14,9,12,9,13,9,14,
10,12
950 DATA 11,9,11,10,11,11,11,12,12,9,13,
9,13,10
960 DATA 13,11,13,12,13,13,13,14,14,14

```

Herzog von Drachenfurcht

Der Herzog von Drachenfurcht versetzt Sie in eine andere, in sich geschlossene Welt. Diese Welt aber ist voll mit einer Anzahl von furchterregenden Überraschungen wie Gruben, die Treibsand oder Drachen beinhalten, oder magischen Höhlen, die Sie an zufällig ausgewählte Orte in der Drachenwelt transportieren können, und Höhlen, die mit Drachengold gefüllt sind:
Wenn Sie das Programm ablaufen lassen, erhalten Sie folgende Information:

Willkommen in der Welt von Drachenfurcht

Euer Auftrag ist es, die Höhlen von Drachenfurcht zu erforschen und nach Gold zu suchen. Gleichzeitig sollt Ihr versuchen, die furchterlichen Drachen, die in den Höhlen leben, zu erschlagen.

Heil Euch, Graf Martin

Ihr beginnt diese Erforschung mit 25 Einheiten Charisma, Ihr müsst Eure Aufgabe erfüllen, bevor das Charisma abgetragen ist. Ihr verliert eine Einheit pro Bewegung.

DRUECKT EINE TASTE FUER START

Wenn Sie das verdaut haben, können Sie einen kurzen Blick auf die Karte von Drachenwelt werfen:

```

X X X X X X X X X
X D . . . $ . . D X
X . . . . T . $ $ X
X . . . X . . . . X
X . ? . . . . . X
X . . X M . . D . X
X . . X T ? . ? X X
X . T . . . . X T X
X D . . . . T . . X
X X X X X X X X X

```

Auf dieser Karte steht X für eine unüberwindbare Wand oder eine verschlossene Höhle, M (für Mensch) stellt Sie in der Startposition (in ungefährer Mitte des Systems) dar, S sind die Höhlen mit den Schätzen, ? steht für die Höhlen, die Sie an zufällig ausgewählte Orte transportieren, D sind die Drachen und Q teilt Ihnen die Stellen mit Treibsand mit. Augenscheinlich ist diese Welt sowohl voller Freuden als auch Gefahren.

Wenn Sie ein alter Vertrauter der Computerspiele sind, dann werden Sie wohl erkennen, daß **Der Herzog von Drachenfurcht** eine Weiterentwicklung der **Hunt the Wumpus**-Spiele ist. Wumpus ist eines der großen Standardcomputerspiele, in dem Sie sich in einem Höhlensystem befinden und nach einer oder mehreren mythischen Bestien suchen, die in der Einzahl Wumpus heißen. Für alle, die daran interessiert sind, heißt die Mehrzahl Wumpii, wie uns versichert wurde. Grob gesprochen wird jedoch in diesen Spielen ein Gitter aufgestellt, und Objekte, Leute, Effekte, Monster und was Sie immer wollen, werden in diesem Gitter platziert. Wenn Sie wollen, können ab und zu die Inhalte spezieller Zellen ihren Standort wechseln. Auch der Spieler kann natürlich innerhalb des Gitters seinen Standort ändern.

Im allgemeinen gibt es keine sichtbare Generalkarte des Systems, doch stellen manche Programme (so wie **Der Herzog von Drachenfurcht**) diese Information für ihre Spieler zur Verfügung. Das erste Wumpus-Spiel wurde 1975 von Gregory Yob geschrieben. In David Ahls hervorragendem Buch 'More BASIC Computergames' (Creative Computing Press, Morristown, New Jersey, 1979) erzählt Herr Yob, wie er die People's Computer Company in Menlo Park, California, besuchte, um die dort in Entwicklung befindlichen Programme zu sehen. Er bemerkte, daß dort drei 'Jagt den irgendwas'-Programme geschrieben wurden, jedes für ein 10x10 Gitter, und jedes gab dem Spieler Rückmeldungen der Art: „Nein, nein... versuchen Sie Nordwesten“.

Etwas enttäuscht kehrte Herr Yob an sein einsames Time-Sharing-Terminal zurück (ja, damals in der alten Zeit, gebildeter Leser, hatten die Menschen noch keine kleinen Computer, über die sie zu jeder Zeit verfügen konnten) und erarbeitete das allererste 'Jagt den Wumpus'-Programm. Herrn Yobs Kreatur lebte in einem dodekadsch geformten Höhlensystem.

Herrn Yobs Erfindergeist machte hier jedoch nicht halt. Er bevölkerte die Höhle mit magischen Fledermäusen, die den Spieler an zufällig ausgewählte Plätze im Höhlensystem transportieren, er versah die Höhlen mit bodenlosen Löchern und ließ die Pfeile, die man zu Beginn des Spieles erhielt, auch um die Ecken fliegen.

Jetzt werden Sie sicher erkennen, wie dieses Gerüst zum Herzstück der Wumpus-Programme wurde, die sich über die ganze Welt verbreiteten. Auch stellt es den Leitfaden für den **Herzog von Drachenfurcht** dar. Dieses Programm gibt dem Spieler mehr Kontrolle über den Verlauf des Programmes, fügt Schätze in Höhlen hinzu und setzt dem Spiel ein Zeitlimit.

Außerdem haben Sie noch ein 'magisches Amulett', das in den Höhlen um Sie herum sehen kann. Leider sind Sie aber kein besonderer Künstler, wenn es darum geht, magische Kräfte zu wecken. Daher arbeitet das Amulett auch nicht sehr zufriedenstellend. Es kann Ihnen nur den Inhalt einer der acht Höhlen um Sie herum mitteilen. Damit nicht genug, es kann Ihnen auch nicht sagen, welche der Höhlen es meint. Doch ist es nicht so schlimm, wie es klingen mag. Sie werden überrascht sein, wie viel Nützliches Sie aus der begrenzten Information des Amuletts ableiten können. Außerdem erhalten Sie ja auch einen kurzen Einblick in das Höhlensystem am Beginn des Spieles und ab und zu während des Spieles.

Wie Sie im Demonstrationsablauf nach der Einführung sehen können, haben Sie nach jedem Zug auch die Möglichkeit, 'K' für 'kämpfen' einzugeben. Geben Sie 'K' ein, so springt das Programm zur Subroutine in Zeile 930. Zuerst überprüft diese, ob Sie nicht schon alle Ihre Pfeile verschossen haben, und wenn nicht, dann haben Sie nun die Wahl, in welche Richtung (von Ihrer momentanen Position aus) Sie Ihren Pfeil abfeuern wollen. Eine erfolgreiche Tötung bringt Ihnen eine ansehnliche Geldsumme.

Hier sind einige Momentaufnahmen des ablaufenden Programmes:

```

GRAF Martin, SIE SIND IN HOEHLE 55
Euer Amulett zeigt an, dass
eine stabile Wand
in der Naeh e ist

```

```

IHR HABT NOCH 25 EINHEITEN CHARISMA

```


Was wollt Ihr nun tun?
 N - nach Norden, S - nach Sueden
 O - nach Osten, W - nach Westen
 K - mit einem Drachen kaempfen, E - Ende

?S

Schrecklich...Treibsand!
 Schrecklich...Treibsand!
 Schrecklich...Treibsand!
 Schrecklich...Treibsand!
 Schrecklich...Treibsand!
 Schrecklich...Treibsand!
 Schrecklich...Treibsand!
 Schrecklich...Treibsand!
 Schrecklich...Treibsand!
 Schrecklich...Treibsand!
 Schrecklich...Treibsand!

Euer ganzes Charisma ist
aufgebraucht....

```

X X X X X X X X X
X D . . . $ . . D X
X . . . . T . $ $ X
X . . . X . . . X
X . ? . . . . X
X . . X . . . D . X
X . . X M ? . ? X X
X . T . . . . X T X
X D . . . . T . . X
X X X X X X X X X

```

Hier das Listing, um Ihnen zu ermöglichen, der Herzog von Drachenfurcht zu werden:

```

10 REM Der Graf von Drachenfurcht
20 GOSUB1270
30 CLS:PRINT:PRINT:PRINT
40 GOSUB1190
50 Q=INT(RND(1)*7)
60 IF Q=0 AND E<>55 THEN GOSUB1190
70 CLS:PRINT:PRINT:PRINT"GRAF ";A$;" S
IE SIND IN HOEHLE ";E
80 IF G>0 THEN PRINT"SIE HABEN $";G;" I
N GOLD"
90 GOSUB760
100 PRINT:PRINT"IHR HABT NOCH ";25-H;" E
INHEITEN CHARISMA"
110 PRINT:PRINT"Was wollt Ihr nun tun?"
120 PRINT"N - nach Norden, S - nach Sued
en"
130 PRINT"O - nach Osten, W - nach West
en"
140 PRINT"K - mit einem Drachen kaempfen
, E - Ende":PRINT
150 INPUT Z$:U=0:IF Z$(">"N" AND Z$(">"S"
AND Z$(">"O" AND Z$(">"W" AND Z$(">"K" AND Z$
(">"E" THEN GOTO150
160 IF Z$="N" AND A(E-10)=88 OR Z$="S" A
ND A(E+10)=88 OR Z$="O" AND A(E+1)=88 OR Z
$="W" AND A(E-1)=88 THEN PRINT"Dahin koenn
t Ihr nicht gehen!":GOTO150
170 IF Z$="E" THEN Q=9:GOTO1160
180 A(E)=46:IF Z$="N" THEN E=E-10
190 IF Z$="S" THEN E=E+10
200 IF Z$="O" THEN E=E+1
210 IF Z$="W" THEN E=E-1
220 IF Z$="K" THEN GOSUB920
230 IF A(E)=63 THEN GOSUB310:REM MAGISCH
240 IF A(E)=68 THEN GOSUB400:REM DRACHE
250 IF A(E)=84 THEN GOSUB550:REM TREIRSA
ND

```

```

260 IF A(E)=36 THEN GOSUB630:REM GOLD
270 H=H+1:IF H=25 THEN Q=9:GOTO1150
280 GOSUB1600
290 GOTO50
300 REM *****
310 REM MAGISCH
320 PRINT:PRINT"Graf ";A$; ", Ihr seid in
eine magische"
330 PRINT"Hoehle gestolpert und werden n
un"
340 PRINT"in eine eine andere Hoehle gez
aubert."
350 GOSUB1600
360 A(E)=46
370 E=INT(RND(1)*76)+12:IF A(E)=88 THEN
GOTO370
380 RETURN
390 REM *****
400 REM DRACHE
410 PRINT"Ihr seid in eine Drachenhoehe
"
420 PRINT"gekommen...sprecht Eure Gebete
"
430 GOSUB1600
440 M=RND(1):IF M<.2 THEN PRINT"Er ist d
avongeflogen":RETURN
450 PRINT"Er wacht auf...er hat Euch ges
ehen"
460 GOSUB1600
470 IF M>.8499999 THEN PRINT"aber er hat
kuerzlich gespeist":PRINT"und schlaeft wi
eder ein":RETURN
480 PRINT"und nun greift er an..."
490 GOSUB1600
500 IF M>.95 THEN PRINT"aber Ihr verteid
igt Euch...und siegen":RETURN
510 PRINT"Gute Nacht, Graf ";A$
520 GOSUB1600
530 Q=9:GOTO1150

```

```

540 REM *****
550 REM TREIBSAND
560 FOR J=1 TO 12
570 FOR K=1 TO J:PRINT " ";:NEXT
580 PRINT"Schrecklich...Treibsand!"
590 FOR O=1 TO 90:NEXT O
600 NEXT J
610 Q=9:H=0:GOTO1150
620 REM *****
630 REM SCHATZ
640 K=INT(RND(1)*100)+100
650 FOR J=1 TO 12
660 FOR Z=1 TO J:PRINT " ";:NEXT Z
670 PRINT"Schatz ! ! !"
680 FOR O=1 TO 200:NEXT O
690 NEXT J
700 GOSUB1600
710 PRINT:PRINT"Sie haben den Schatz ein
es Drachen"
720 PRINT"gefunden. Wert: $";K
730 G=G+K
740 RETURN
750 REM *****
760 REM AMULETTDETAILS
770 Y=1
780 L=A(E+P(Y))
790 IF L<>46 THEN GOTO820
800 IF Y<8 THEN Y=Y+1:GOTO780
810 IF L=46 THEN RETURN
820 PRINT"Euer Amulett zeigt an, dass"
830 IF L=88 THEN PRINT"eine stabile Wand
"
840 IF L=63 THEN PRINT"eine magische Hoe
hle"
850 IF L=68 THEN PRINT"ein Drache"
860 IF L=81 THEN PRINT"Treibsand"
870 IF L=36 THEN PRINT"Gold"
880 PRINT"in der Naehel ist"
890 GOSUB1600

```

```

900 RETURN
910 REM *****
920 REM DRACHEN ANGREIFEN
930 PRINT
940 PF=PF-1:IF PF=0 THEN PRINT"Ihr habt
alle Ihre":PRINT"Pfeile verbraucht...":GOS
UB1600:RETURN
950 PRINT"Ihr habt ";PF;" Pfeile im Koec
her"
960 SS=0
970 PRINT"In welche Richtung wollt"
980 INPUT "Ihr schiessen (N,S,O oder W)
";S$
990 IF S$="N" AND A(E-10)=68 THEN SS=1:Y
T=E-10
1000 IF S$="S" AND A(E+10)=68 THEN SS=1:Y
T=E+10
1010 IF S$="O" AND A(E+1)=68 THEN SS=1:YT
=E+1
1020 IF S$="W" AND A(E-1)=68 THEN SS=1:YT
=E-1
1030 PRINT
1040 IF SS=0 THEN PRINT"Da war kein Drach
e...":PRINT"Ihr habt einen Pfeil verschwen
det":GOTO1140
1050 PRINT"Bravo, Graf ";A$
1060 PRINT"Ihr habt einen wilden"
1070 PRINT"Drachen getroffen"
1080 GOSUB1600
1090 IF RND(1)>.3 THEN GOTO1130
1100 PRINT"Ihr habt ihn erlegt!":A(YT)=46
:K=INT(RND(1)*100)+100
1110 PRINT
1120 PRINT"Ihr werdet mit $";K;" belohnt"
:G=G+K:GOTO1140
1130 PRINT"Aber Ihr habt ihn nur verwunde
t..."
1140 GOSUB1600:RETURN
1150 IF H<1 THEN PRINT"Euer ganzes Charis

```

```

ma ist":PRINT"....aufgebraucht...":GOSUB1
600:GOTO1170
1160 PRINT"Ihr habt ";25-H;" Einheiten Ch
arisma"
1170 IF G>0 THEN PRINT"Ihr habt $";G;" in
Gold angesammelt"
1180 PRINT:PRINT
1190 A(E)=77
1200 FOR J=1 TO 100
1210 PRINT CHR$(A(J));" ";
1220 IF 10*INT(J/10)=J THEN PRINT
1230 NEXT J
1240 GOSUB1600
1250 IF Q=9 THEN END
1260 RETURN
1270 CLS:PRINT:PRINT:PRINT
1280 PRINT"Willkommen in der Welt von Dra
chenfurcht"
1290 PRINT:PRINT"Euer Auftrag ist es, die
Hoehlen"
1300 PRINT"von Drachenfurcht zu erforsche
n und"
1310 PRINT"nach Gold zu suchen. Gleichzei
tig sollt"
1320 PRINT"Ihr versuchen, die fuerchterli
chen"
1330 PRINT"Drachen, die in den Hoehlen le
ben, zu":PRINT"erschlagen."
1340 INPUT "Wie ist Euer Name ";A$
1350 CLS:PRINT:PRINT"Heil Euch, Graf ";A$
1360 PRINT:PRINT"Ihr beginnt diese Erfors
chung"
1370 PRINT"mit 25 Einheiten Charisma, Ihr
"
1380 PRINT"muesst Eure Aufgabe erfuellen,
"
1390 PRINT"bevor das Charisma abgetragen
ist."
1400 PRINT"Ihr verliert eine Einheit pro

```

```

Bewegung."
1410 N=1:PRINT:PRINT"DRUECKT EINE TASTE F
UER START"
1420 N=N+1:IF INKEY$(10)="" THEN GOTO1420
1430 CLS:REM RANDOMIZE N
1440 PRINT"Bitte wartet, Graf ";A$
1450 DIM A(100),P(8):H=0:G=0:L=0:G=0:PF=6
1460 FOR B=1 TO 100:A(B)=46
1470 IF B<12 OR B>90 OR 10*INT(B/10)=B OR
10*INT(B/10)=B-1 THEN A(B)=88
1480 NEXT B
1490 FOR B=1 TO 5:RESTORE:FOR D=1 TO 5
1500 Z=INT(RND(1)*76)+12:IF A(Z)=88 THEN
GOTO1500
1510 READ A(Z)
1520 NEXT D:NEXT B
1530 DATA 88,63,68,84,36
1540 FOR B=1 TO 8:READ P(B):NEXT B
1550 DATA -11,-10,-9,-1,1,9,10,11
1560 E=55
1570 RETURN
1580 REM SCHALTEN SIE CAPS LOCK
1590 REM VOR DEM SPIEL EIN
1600 FOR O=1 TO 3000:NEXT O:RETURN

```

Simulationen

Simulationen sind Verfahren, eine vorgetäuschte Wirklichkeit zu erzeugen. Im Gegensatz zu den Abenteuern, in denen die vom Computer erschaffenen Welten oft traumartig und magisch sind, stehen die durch Computersimulationen erzeugten Welten der wirklichen Welt näher. In einer Simulation verändert der Computer Variablen nach Formeln, die Sie bestimmt haben. Er kontrolliert die Entwicklung der von Ihnen ins Rollen gebrachten Situationskette und nimmt den Platz der Umwelt ein, die auf Ihre Eingaben reagiert.

Simulationen versuchen, das wirkliche Leben nachzuvollziehen. Da es aber schwer ist, die Wirklichkeit wahrheitsgetreu wiederzugeben und einzugrenzen, müssen wir einige sehr drastische Vereinfachungen vornehmen, bevor wir eine arbeitsfähige Simulation produzieren können. Trotz dieser Vereinfachungen kann eine gut geschriebene Simulation – wie Sie gleich sehen werden – das Leben sehr wahrheitsgetreu wiedergeben. Ursache und Wirkung sind jeweils so genau miteinander verknüpft wie genau Ihre Formeln dafür sind. Der Zufallsgenerator kann die Aufgabe übernehmen, Dinge wie unterschiedliches Wetter, Bevölkerungszahlen, das Molekularverhalten in Gasen oder die Versetzung einer klaren Flüssigkeit mit einer farbigen Tinte zu simulieren.

In diesem Kapitel haben wir drei Simulationen. Eine davon, **Herrin von Xenophobia**, bewegt sich knapp an der Grenze der Absurdität. Es ist aber lustig zu spielen, wenn Sie an dem Schicksal eines gesamten Planeten herumbasteln. Die beiden anderen sind etwas ernster, und die eingebauten Formeln produzieren Ergebnisse, die eher an das Alltagsleben erinnern. In **Ländliche Geschäfte** haben Sie die Aufsicht über einen Bauernhof mit launischen Arbeitern. In **Geschäftsführer** kontrollieren Sie eine Firma, die alle möglichen Produkte, angefangen von Kazoos über Tropenhelme und Himmelschaken, herstellen kann. Trotz der Ernstheit dieser beiden Spiele bereiten sie auch sehr viel Spaß. Das werden Sie sicher bemerken, wenn Sie auf Ihrem Computer spielen.

Wollen Sie mehr über das Schreiben von Simulationsprogrammen wissen, so empfehle ich Ihnen das Buch 'Designing Classroom Simulations' von Glenn Pate und Hugh Parker, jr. Creative Computing Magazine hat auch zwei sehr interessante Artikel veröffentlicht. Diese sind 'How to write a computer simulation', von dem früheren Herausgeber des Magazins, David Ahl bei der 'Conference on Computers in the Undergraduate Curricula' in Pullman, Washington, im Juni 1974, und 'Strategies for Successful Simulation', von Bruno B. Wolff, jr., im August 1981. Es ist wirklich ratsam, eine Bibliothek nach diesen alten Ausgaben zu durchforschen, da Sie sehr viel aus ihnen lernen können. Besonders

interessant in Wolffs Artikel sind vier Gleichungen, die Ursache und Wirkung miteinander verknüpfen und sehr leicht an ein von Ihnen geschriebenes Programm angepaßt werden können.

Herrin von Xenophobia

Wir beginnen die Untersuchung der Simulationen mit der dümmsten aus unserem Angebot, **Herrin von Xenophobia**. Das Programm basiert auf einem Listing, das von Alastair Gourley, einem talentierten, in Glasgow lebenden Programmierer, geschrieben wurde. Es besteht keine Veranlassung, Ihnen das Programm zu erklären, da es alle Anweisungen selbst gibt. Hier ist ein Teil eines Programmlaufes:

Herrin von Xenophebia, ein Bericht
vom Amt fuer Information, betreffend
dem Zustand Ihres Planeten in diesem
Jahre des Heiles, 1995

Die Bevoelkerung des Planeten ist 348

Die gemeinen Untertanen koennten
gute 95 Acres bewirtschaften

Ihr Staatsschatz enthaelt Gold,
Edelsteine und Muenzen im Werte von 1118

Zeit, ein Dekret herauszugeben...

Wieviel Land soll in diesem Jahr
bewirtschaftet werden ?27

Wieviel wollen Sie fuer Nahrung
fuer die Untertanen ausgeben ?2400

Tja, Xeno, Sie haben sich und
unseren kleinen Planeten in
feine Schwierigkeiten gebracht.

Der Staatsschatz ist bankrott!!

Wer ist da wohl schuld ????????

Wenn Sie nocheinmal herrschen
wollen, druecken Sie 'J' oder
'N', um diese Farce zu beenden...

Und hier ist das Listing:

```

10 REM Herrin von Xenophobia
20 GOSUB700:REM Initialisieren
30 REM *****
40 FOR J=1 TO 20
50 CLS
60 PRINT:PRINT:PRINT
70 PRINT"Herrin von Xenophebia, ein Ber
  icht"
80 PRINT"vom Amt fuer Information, betr
  effend"
90 PRINT"dem Zustand Ihres Planeten in
  diesem"
100 PRINT"Jahre des Heiles, ";1994+J
110 PRINT:PRINT
120 PRINT"Die Bevoelkerung des Planeten
  ist ";INT(B)
130 GOSUB880
140 PRINT"Die gemeinen Untertanen koennt
  en"
150 PRINT"gute ";L;" Acres bewirtschafte
  n"
160 GOSUB880
170 PRINT"Ihr Staatsschatz enthaelt Gold
  ,"
180 PRINT"Edelsteine und Muenzen im Wert
  e von ";INT(U)
190 GOSUB880
200 PRINT"Zeit, ein Dekret herauszugeben
  "
```

```

...
210 PRINT:PRINT"Wieviel Land soll in die
sem Jahr"
220 INPUT"bewirtschaftet werden ";W
230 U=U-W*10
240 IF U<1 THEN GOTO780
250 L=L+W
260 GOSUB880
270 PRINT:PRINT"Wieviel wollen Sie fuer
Nahrung"
280 INPUT"fuer die Untertanen ausgeben "
;W
290 U=U-10*W
300 IF U<1 THEN GOTO780
310 R=INT(RND(1)*10)+1
320 GOSUB880
330 IF B-W*R*5>B/4 THEN PRINT"Um Gottes
Willen! Es ist nicht":PRINT"genug Nahrung
fuer alle da...":GOSUB580
340 B=B+RND(1)*(W*R*5-B)
350 IF B>149 THEN GOTO410
360 GOSUB880
370 PRINT"Die Bevoelkerung von Xenophobi
a ist"
380 PRINT"auf ";B;" zusammengeschmolzen,
das"
390 PRINT"ist nicht genug, Eure Xeno"
400 GOTO480
410 U=U+B*L/93
420 NEXT J
430 PRINT:PRINT:Tja, Eure Xenophob
ie, das ist das"
440 PRINT"Ende Ihrer 20 jaehrigen Herrsc
haft"
450 PRINT"ueber unseren kleinen Planeten
. Sie"
460 PRINT"konnten $";U;" ansammeln, das
ist"
470 PRINT"nicht so schlecht, finde ich..

```

```

.
480 GOSUB880
490 PRINT:PRINT"Wenn Sie nocheinmal herr
schen"
500 PRINT"wollen, druecken Sie 'J' oder"
510 PRINT"'N', um diese Farce zu beenden
..."
520 A$=INKEY$(10)
530 IF A$(">"N" AND A$(">"n" AND A$(">"J" A
ND A$(">"j" THEN GOTO520
540 IF A$="N" OR A$="n" THEN PRINT"Ihr W
unsch ist mein Befehl,":PRINT"O Herrin von
Xenophobia":END
550 RUN
560 END
570 REM *****
580 PRINT:PRINT"Schrecklich, Eure Xenoni
a"
590 PRINT"Es hat einen Aufstand gegeben!
!"
600 PRINT"Die Untertanen revoltieren"
610 PRINT"(Das musste ich einfuegen!)"
620 GOSUB880
630 IF RND(1)<.2 THEN PRINT"und Ihre Pol
izisten konnten sie":PRINT"nicht aufhalten
":GOTO480
640 PRINT"aber Ihre ruecksichtslose Poli
zei hat"
650 PRINT"diesen Schwachsinn beendet"
660 GOSUB880
670 RETURN
680 NEXT J
690 REM *****
700 REM Initialisation
710 REM RANDOMIZE
720 CLS
730 B=200+INT(RND(1)*300)
740 U=700+INT(RND(1)*500)
750 L=70+INT(RND(1)*50)

```

```

760 RETURN
770 REM *****
780 REM Bankrott
790 PRINT:PRINT"Tja, Xeno, Sie haben sich
und"
800 PRINT"unseren kleinen Planeten in"
810 PRINT"feine Schwierigkeiten gebracht
."
820 GOSUB880
830 PRINT:PRINT"Der Staatsschatz ist bankrott!!"
840 GOSUB880
850 PRINT:PRINT"Wer ist da wohl schuld ?
?????"
860 GOT0490
870 REM *****
880 REM Pause
890 FOR Y=1 TO 1500:NEXT Y
900 PRINT
910 RETURN

```

Ländliche Geschäfte

Haben Sie als Herrin von Xenophobia überlebt, dann sind Sie vielleicht an etwas Erdgebundenerem – einen Bauernhof zu organisieren – interessiert. In „Ländlichen Geschäften“, das sehr frei auf einem Programm von Stephen Glenn, ebenfalls aus Glasgow, basiert, haben Sie eine schwierige Aufgabe. Sie müssen sich mit einer sehr feinfühligem Arbeiterschaft herumschlagen, um Ihren Lebensunterhalt zu verdienen (oder zumindest Weizen, Mais und Gerste nach Hause zu bringen). Auch dieses Programm versorgt Sie mit allen nötigen Anweisungen während des Programmablaufes und lehrt Sie die beste Strategie, um den größtmöglichen Gewinn zu machen, und wie Sie die verlangten 20 Jahre auf dem Bauernhof überleben, um das Spiel zu gewinnen. Einige kleine Hinweise können jedoch nicht schaden. Obwohl Sie Geld sparen können, wenn Sie Ihren Arbeitern weniger zahlen, werden Sie merken, daß Sie dann von ihnen im nächsten Jahr im Stich gelassen werden. Die Ernte, die Sie jedes Jahr einbringen, hängt bis zu einem gewissen Grad auch von der Anzahl der Feldarbeiter ab.

Ein anderer Punkt, den Sie beachten sollten, ist der unterschiedliche Ertrag, den Sie von den drei verschiedenen Getreidearten, die Sie anbauen können, erhalten. Es liegt also in Ihrem eigenen Interess herauszufinden, welche die ertragreichste ist, und sich dann auf diese zu konzentrieren.

Seien Sie nicht entmutigt, wenn Sie die ersten paar Male Schiffbruch erleiden. Sie müssen einige spezielle Fähigkeiten entwickeln, um Ihre Farm in Schwung zu halten, und das kann unter Umständen ein paar schlechte Jahre bedeuten.

Hier ist ein Teil eines „ländlichen Geschäftes“:

Sie haben \$7628 im Jahr 1

Sie beschaeftigen 471
Arbeiter, die auf
466 Acres arbeiten

Ernte:- 0 Korn
 0 Gerste
 0 Weizen

Allgemeine Kosten fuer
Bearbeitung eines Acres
werden \$10 sein, so
koennen Sie dieses Jahr
maximal 466 Acres bearbeiten

Wieviel Land wollen Sie bearbeiten ?100

Und hier ist das Programm, damit Sie ein glücklicher Bauer werden können:

```

10 REM Laendliche Geschaefte
20 REM RANDOMIZE
30 JR=1
40 GE=INT(RND(1)*1000+7000)
50 LA=INT(RND(1)*1000+100)
60 AC=INT(RND(1)*200+300)
70 GK=INT(RND(1)*5)+8:KX=GK

```

```

80 GS=0:WE=0:K0=0
90 GOSUB170
100 GOSUB290
110 IF JR=10 THEN GOTO750
120 IF GE<1 THEN GOTO790
130 IF LA<1 THEN GOTO820
140 JR=JR+1
150 GK=INT(GK*1.125):REM Allgemeinkosten
    pro Jahr um 12.5% erhoeht
160 GOTO80
170 REM Erneuern
180 GOSUB870
190 PRINT"Sie haben $";GE;" im Jahr ";JR
200 GOSUB880
210 PRINT"Sie beschaeftigen ";LA
220 PRINT"Arbeiter, die auf"
230 PRINT";AC;" Acres arbeiten"
240 GOSUB880
250 PRINT"Ernte:-",K0;" Korn"
260 PRINT",GS;" Gerste"
270 PRINT",WE;" Weizen"
280 RETURN
290 PRINT:PRINT"Allgemeine Kosten fuer"
300 PRINT"Bearbeitung eines Acres"
310 PRINT"werden $";GK;" sein, so"
320 PRINT"koennen Sie dieses Jahr"
330 PRINT"maximal ";
340 MAX=INT(GE/GK):IF MAX>AC THEN MAX=AC
350 PRINT";MAX;" Acres bearbeiten"
360 PRINT:PRINT"Wieviele Land wollen Sie
    bearbeiten ";
370 INPUT L
380 IF L>MAX THEN GOTO370
390 GE=GE-L*GK
400 GOSUB170
410 PRINT:PRINT"Wieviele wollen Sie jedem
    Arbeiter"
420 INPUT "bezahlen ";W
430 IF W*LA>GE THEN GOTO410

```

```

440 GE=GE-LA*W
450 GOSUB170
460 P=10
470 PRINT"Welchen Anteil (von 10) wollen
    "
480 PRINT"Sie auf Kornanbau konzentriere
    n ";
490 INPUT KA
500 IF KA>P THEN GOTO490
510 P=P-KA
520 PRINT:PRINT"Wieviele von den verbleib
    enden ";P
530 PRINT"von 10, wollen Sie fuer Weizen
    anbau"
540 INPUT "verwenden ";WA
550 IF WA>P THEN GOTO540
560 P=P-WA
570 GOSUB870
580 PRINT"Bitte warten Sie ein Jahr..."
590 FOR Z=1 TO 2000:NEXT Z
600 GS=INT(P*LA*W*3/100000)
610 K0=INT(KA*LA*W*2.7/17000)
620 WE=INT(WA*LA*W*1.4/9300)
630 T=GS+K0+WE
640 GOSUB170
650 PRINT:PRINT"Die Ernte betrug ";T;" T
    onnen"
660 VE=INT(((.5+8.7*GS+5.94*K0+2.2*WE)*(G
    K-KX+1)))
670 IF GE=0 AND K0=0 AND WE=0 THEN VE=0
680 FOR Z=1 TO 2000:NEXT Z
690 PRINT:PRINT"Ihre Gesamteinnahmen"
700 PRINT"waren $";VE
710 GE=GE+VE
720 LA=INT(LA-LA/(W+.01))
730 FOR Z=1 TO 2000:NEXT Z
740 RETURN
750 GOSUB870
760 PRINT"Sie haben 10 Jahre lang"

```



```

770 PRINT"ueberlebt. Bravo!"
780 END
790 GOSUB870
800 PRINT"Sie sind pleite!"
810 END
820 GOSUB870
830 PRINT"Sie haben keine Arbeiter, und"
840 PRINT"mussten so Ihre Farm"
850 PRINT"verkaufen"
860 END
870 CLS
880 PRINT:PRINT
890 RETURN

```

Geschäftsführer

Endlich kommen wir zu dem Hauptwerk dieses Kapitels, **Geschäftsführer!**

Auch dieses Programm versorgt Sie während des Programmablaufes mit jeder nötigen Information. Das Ziel dieses Spieles ist es, Ihre Fabrik so lange produzieren zu lassen, bis Sie \$10.000 erwirtschaftet haben (dazu zählen Ihre Aktien und Ihr Kapital). Sie werden sich mit widerspenstigen Gewerkschaften auseinandersetzen, die Ihre Wünsche nach Entlassung von Personal ablehnen werden und die immer Lohnerhöhungen wollen, denen Sie nicht zustimmen können. Die Arbeiter erreichen fast nie ihr Plansoll in der Produktion, und es gibt Rohstofflieferanten, die ihre Preise genauso gerne erhöhen, wie die Gewerkschaften mehr Lohn verlangen.

Sie werden bald erkennen, daß bei diesem Programm, im Gegensatz zu den beiden anderen, die nur eine sehr grobe Annäherung an die wirkliche Welt erreichen, das Schicksal Ihrer Fabrik für Sie sehr wichtig wird. Sicherlich werden Sie herausfinden, wie Sie die Ihnen zur Verfügung stehenden Mittel für maximale Einnahmen einsetzen müssen. Bedenken Sie jedoch, daß, obwohl Sie in der Bestimmung des Verkaufspreises sehr frei sind, eine Erhöhung des Preises die Kaufwilligkeit Ihrer Kunden herabsetzt.

Hier sind einige Beispiele von einem Programmlauf des „Geschäftsführers“:

Bericht der Verkaufsabteilung,
fuer Woche 1

Fluessiges Kapital ist \$96
Laufende Kosten sind \$103 pro Woche

Ihr Lager enthaelt 359 Schilder
im Werte von je \$11
im Werte von 3949

Sie werden fuer \$11 verkauft
und \$9 Kosten um eines herzustellen

Sie haben 10 Angestellte,
jedem zahlen Sie \$16, die
Lohnkosten dieser Woche sind 160

Jeder Angestellte kann 10
Schilder pro Woche herstellen,
insgesamt also 100

Ihr Gesamtlager von Schilder
ist 359

Bitte warten Sie auf einen
Verkaufsbericht...

Es wurden insgesamt 162
Schilder verkauft

Das Einkommen aus diesem
Verkauf ist \$1782

Sie haben die Chance, Ihren Preis
zu erhoehen. Sie verkaufen
Schilder jetzt um \$11

Um wieviel Prozent wollen Sie
den Preis erhoehen ?3

Schilder werden nun um \$11.33 verkauft

Jetzt können Sie Ihren Geschäftsführungsambitionen mit diesem Listing nachkommen:

```

10 REM GESCHAEFTSFUEHRER
20 GOSUB1670:REM Initialisieren
30 WE=WE+1
40 GOSUB930:REM Bericht
50 GOSUB1300:REM Mitarbeiter
60 GOSUB930:REM Bericht
70 GOSUB1130:REM Produktion
80 GOSUB930:REM Bericht
90 GOSUB730:REM Verkauf
100 GOSUB140:REM Probleme
110 CA=CA-WA*WO-RC
120 GOTO30
130 REM *****
140 REM Probleme
150 CLS
160 IF RND(1)<.45 THEN GOTO260
170 A=INT(RND(1)*7)+1
180 PRINT:PRINT:PRINT
190 PRINT"Die Gewerkschaften verlangen"
200 PRINT"eine Lohnerhoehung von ";A;"%"
210 WA=INT(100*(WA+WA*A/100))/100
220 GOSUB1840
230 PRINT:PRINT"Gehalt pro Angestelltem
ist nun ";WA
240 GOSUB1840
250 CLS
260 IF RND(1)<.81 THEN GOTO410
270 PRINT:PRINT:PRINT
280 PRINT"Ein Feuer in Ihrem Lagerhaus h
at"
290 PRINT"einiges an Ware vernichtet."
300 PRINT"Bitte warten Sie auf den Beric
ht"
310 PRINT"ueber die Zerstoeerungen..."
320 GOSUB1840
330 A=INT(RND(1)*ST/2)+1
340 ST=ST-A

```

```

350 PRINT:PRINT"Es wurden ";A;" ";A$
360 PRINT"vernichtet. Sie hatten $";A*SP
370 PRINT"Verkaufswert"
380 GOSUB1840
390 PRINT"Warenlager enthaelt"
400 PRINT"nun ";ST;" ";A$
410 IF RND(1)>.3 THEN GOTO560
420 CLS
430 PRINT:PRINT:PRINT
440 PRINT"Ihr Hauptlieferant hat eine"
450 PRINT"drastische Preiserhoehung ange
kuendigt..."
460 GOSUB1840
470 A=INT(RND(1)*100*CO/7)/100
480 IF A<.01 THEN GOTO470
490 PRINT:PRINT"Die Kosten um ";A$;" her
zustellen,"
500 PRINT"sind um $";A;" gestiegen"
510 GOSUB1840
520 CO=CO+A
530 PRINT:PRINT"Es kostet nun $";CO;" ,"
540 PRINT"um eines herzustellen..."
550 GOSUB1840
560 IF RND(1)<.65 AND MA<SP THEN RETURN
570 CLS
580 PRINT:PRINT:PRINT
590 PRINT"Sie haben die Chance, Ihren Pr
eis"
600 PRINT TAB(4);"zu erhoeuen. Sie verka
ufen"
610 PRINT A$;" jetzt um $";SP
620 GOSUB1840
630 PRINT
640 PRINT"Um wieviel Prozent wollen Sie"
650 INPUT"den Preis erhoeuen ";A
660 RE=RE+A
670 SP=INT(100*(SP+A*SP/100))/100
680 GOSUB1840
690 PRINT:PRINT A$;" werden nun um $";SP

```

```

;" verkauft"
700 GOSUB1840
710 RETURN
720 REM *****
730 REM Verkauf
740 PRINT:PRINT"Ihr Gesamtlager von ";A$
750 PRINT"ist ";ST
760 GOSUB1840
770 PRINT:PRINT"Bitte warten Sie auf ein
en"
780 PRINT"Verkaufsbericht..."
790 A=INT(RND(1)*ST/(VF/1000))+1
800 IF A>ST THEN GOTO790
810 CLS
820 PRINT:PRINT
830 PRINT"Es wurden insgesamt ";A
840 PRINT A$;" verkauft"
850 ST=ST-A
860 ZA=A*SP
870 PRINT:PRINT"Das Einkommen aus diesem
"
880 PRINT"Verkauf ist $";ZA
890 CA=INT(A*SP*100)/100+CA
900 GOSUB1840
910 RETURN
920 REM *****
930 REM Bericht fuer den Geschaeftsfuehr
er
940 CLS
950 IF CA+ST<1 THEN GOTO1510:REM Pleite
960 IF CA+ST>9999 THEN PRINT"Sie haben $
10000 erwirtschaftet":PRINT"und koennen si
ch nun zur Ruhe setzen...":GOSUB1840:GOTO1
590
970 PRINT:PRINT"Bericht der Verkaufsabte
ilung,"
980 PRINT TAB(6);"fuer Woche ";WE
990 PRINT:PRINT"Fluessiges Kapital ist $
";INT(CA*100)/100

```

```

1000 PRINT"Laufende Kosten sind $";RC;" p
ro Woche"
1010 PRINT:PRINT"Ihr Lager enthaelt ";ST;
" ";A$
1020 PRINT TAB(6);"im Werte von je $";SP
1030 PRINT TAB(6);"im Werte von ";INT(ST*
SP*100)/100
1040 PRINT:PRINT"Sie werden fuer $";SP;"
verkauft"
1050 PRINT"und $";CO;" Kosten um eines he
rzustellen"
1060 PRINT:PRINT"Sie haben ";WO;" Angeste
llte,"
1070 PRINT"jedem zahlen Sie $";WA;" , die"
1080 PRINT"Lohnkosten dieser Woche sind "
;WA*WO
1090 PRINT:PRINT"Jeder Angestellte kann "
;PR
1100 PRINT A$;" pro Woche herstellen,"
1110 PRINT"insgesamt also ";PR*WO
1120 RETURN
1130 INPUT"Wieviele wollen Sie herstellen
";MA
1140 IF MA=0 THEN RETURN
1150 PRINT
1160 IF MA*CO>CA THEN PRINT"Sie haben nic
ht genug Geld":GOTO1130
1170 IF MA>PR*WO THEN PRINT"Sie haben nic
ht genug Leute":PRINT"angestellt":GOTO1130
1180 PRINT"Jawohl...Produktionsziel in Wo
che ";WE
1190 PRINT"ist ";MA;" ";A$
1200 MA=INT(MA-RND(1)*MA/5)
1210 GOSUB1840
1220 PRINT:PRINT"Die Anzahl von wirklich"
1230 PRINT"hergestellten ";A$;" war in"
1240 PRINT"Woche ";WE;" ";MA;"..."
1250 ST=ST+MA
1260 CA=CA-CO*MA

```

```

1270 GOSUB1840
1280 RETURN
1290 REM *****
1300 REM Angestellte
1310 PRINT"Wieviele Leute wollen Sie"
1320 INPUT"einstellen ";A
1330 WO=WO+A
1340 PRINT:PRINT"Die Zahl Ihrer Angestell
ten"
1350 PRINT"umfasst nun ";WO
1360 GOSUB1840
1370 IF A>0 THEN RETURN
1380 GOSUB930
1390 PRINT"Wieviele Angestellte wollen"
1400 INPUT"Sie feuern ";A
1410 IF A=0 THEN GOTO1480
1420 IF A>WO THEN GOTO1390
1430 A=INT(RND(1)*A+1)
1440 GOSUB1840
1450 PRINT:PRINT"Die Gewerkschaften erlau
ben"
1460 PRINT"Ihnen ";A;" loszuwerden"
1470 WO=WO-A
1480 GOSUB1840
1490 RETURN
1500 REM *****
1510 REM Das schlimme Ende !!
1520 PRINT:PRINT"Sie sind pleite !!!"
1530 GOSUB1840
1540 PRINT:PRINT"Diese Schande!!"
1550 GOSUB1840
1560 PRINT:PRINT"Sie haben das Geschaef
t fuer ";WE
1570 PRINT"Wochen in Gang gehalten"
1580 GOSUB1840
1590 PRINT"Druicken Sie 'J' wenn Sie eine
n"
1600 PRINT"weiteren Versuch als Geschaef
sfuehrer"

```

```

1610 PRINT"starten wollen ('N' fuer Ende)
"
1620 A#=INKEY$(10)
1630 IF A#="" THEN GOTO1620
1640 IF A#="J" OR A#="j" THEN RUN
1650 END
1660 REM *****
1670 REM Initialisieren
1680 REM RANDOMIZE
1690 FOR Z=1 TO RND(1)*8+1
1700 READ A#
1710 NEXT Z
1720 CA=500+INT(RND(1)*500)
1730 ST=100+INT(RND(1)*500)
1740 SP=10+INT(RND(1)*5)
1750 CO=7+INT(RND(1)*5)
1760 IF CO>SP THEN GOTO1750
1770 WO=7+INT(RND(1)*10)
1780 WA=12+INT(RND(1)*SP)
1790 PR=5+INT(RND(1)*6)
1800 RC=100+INT(RND(1)*20)
1810 WE=0
1820 VF=1:REM VF ist der Verkaufsfaktor
1830 RETURN
1840 REM Pausenunterprogramm
1850 FOR Z=1 TO 3000:NEXT Z:RETURN
1860 DATA "Schilder","Fahrraeder","Harmon
ikas"
1870 DATA "Kazoos","Pulte","Moleskins"
1880 DATA "Plastikbeutel","Tropenhelme","
Lufthaken"
1890 DATA "Barklingeln"

```

Würfelspiele

Ob mit großen, kleinen oder gar keinen Einsätzen, Würfelspiele erfreuten sich immer schon großer Beliebtheit. Ob die Würfel aus Stein oder Holz waren oder aber aus den Knochen einer eben erlegten Jagdbeute geschnitzt wurden, ob sie vier oder sechs wichtige Seiten hatten, Würfel waren zu allen Zeiten eine willkommene Abwechslung für zahllose Menschen.

Der Würfel, physisches Demonstrationsobjekt der Widerspenstigkeit des Zufalls, wurde auch in der Zukunftsvorhersage verwendet. Wie jedes Orakel trägt er damit wahrscheinlich auch etwas Verantwortung für den Lauf der Geschichte.

In Troja spielten die Griechen mit Würfeln, wie auch Mark Anton in Alexandria. Das mittelalterliche Europa wurde von Italien und Frankreich aus mit dem Würfelspiel bekannt gemacht, und die Verwendung der sechsseitigen Würfel hat sich bis heute erhalten. In diesem Kapitel des Buches untersuchen wir, wie wir den Zufallsgenerator Ihres Computers verwenden müssen, damit er die physischen Würfel ersetzt. Sie werden bald bemerken, daß die Würfel Ihres Computers um nichts weniger verführerisch sind als echte, obwohl sie ausschließlich im Programm existieren und keine physische Gestalt haben.

Das Spiel gegen einen Computer anstelle eines menschlichen Gegenspielers verändert die 'Atmosphäre' des Würfelspieles etwas. Daß Sie nicht zu zahlen brauchen, wenn Sie verlieren, ist ein echter Vorteil!

Eine große Anzahl von Würfelprogrammen ist auf dem Markt. Diese haben die beeindruckendsten Namen. Hier sind einige der weitverbreitetsten:

- | | |
|---------------------|------------------------|
| - Fahrt nach Boston | - Jahrestag |
| - Schummeln | - Schnapp den Yankee |
| - Fünzig | - Hilf deinem Nachbarn |
| - Klippe | - Tot umfallen |
| - Schwein | - Schieß die Schachtel |
| - Chicago | - Baseball |
| - Sechszwanzig | - Basketball |
| - Einundzwanzig | - Pokerwürfel |
| - Rund um die Uhr | - Lügnerwürfel |

Die Würfelspiele, die wir in diesem Buch vorstellen werden, sind Schlangenaugen, Eins-und-zwanzig, Sieben/Elf, Drunter und Drüber, Keine Ursache, Chemin de Computer.

Chemin de Computer

Ich bin sicher, Sie haben schon erraten, daß **Chemin de Computer** auf Chemin de Fer basiert. In diesem Spiel werfen Sie abwechselnd mit dem Computer fünf Würfel und addieren die Augenzahlen. Sie versuchen, eine größere Augenzahl zu erreichen als der Computer. Das Spiel ist jedoch nicht so einfach, daß Sie nur die Augenzahlen zusammenzählen müssen. Jeder Würfel, der eine Fünf oder eine Zwei zeigt, muß nochmals gewürfelt werden. Außerdem wird nur die letzte Nummer der geworfenen Zahl gewertet (z.B. Sie warfen 27 in Summe, so wird nur 7 gewertet; bei einer Summe von 13 gilt nur 3). Es gibt drei spezielle Summen: 7 (ein Natural), 8 (La Petite) und 9 (La Grande). Diese Namen werden nur dann verwendet, wenn 7, 8 oder 9 beim ersten Wurf schon erzielt werden, d.h. daß jene Würfel, die 5 oder 2 zeigen, noch nicht nochmals geworfen wurden. Sie werden es übrigens in diesen Fällen auch nicht. Sie und der Computer halten immer bei einer 7, 8 oder 9 auf den ersten Wurf der fünf Würfel. Wenn Sie sich das Programmlisting ansehen, so bemerken Sie, daß, nachdem der Zufallsgenerator in Zeile 20 aktiviert wird, der Bildschirm gelöscht und die Spiele zählende Variable (SPIEL) initialisiert und auf Null gesetzt wird. Gleiches geschieht auch mit den Variablen für Ihre Punktezahl und für die des Computers. Nun geht das Programm zu Zeile 330, wo das eigentliche Spiel beginnt.

Wie in vielen anderen Programmen in diesem Buch hat auch dieses Programm eine Routine mit einer Warteschleife am Ende des Listings. Diese wird einige Male während des Ablaufens des Programmes aufgerufen, um die Geschwindigkeit, mit der das Programm fortschreitet, zu verbessern. Außerdem gibt sie Ihnen in vielen Fällen die Möglichkeit, den Bildschirm abzulesen, bevor das Programm weiterläuft, den Bildschirm löscht und Sie im Dunklen tippen läßt. In diesem Programm gibt es zwei Schleifen. Eine ist etwas länger als die andere und druckt außerdem einige Leerzeilen auf den Bildschirm, bevor sie zum Hauptstamm des Programmes zurückkehrt.

Zeile 330 verursacht eine Verzögerung und löscht dann den Bildschirm, um danach den Zähler SPIEL in der nächsten Zeile zu erhöhen. In Zeile 350 wird Ihnen mitgeteilt, das wievielte Spiel gerade läuft. Weil der Computer die Bank spielt, die immer den ersten Wurf hat, beginnt er zu würfeln. Jetzt wird die Subroutine in Zeile 70, die das Werfen der Würfel bewirkt, abgerufen. Sie sehen, daß der Schleifenzähler G in Zeile 80 dazu verwendet wird, bei jedem Schleifendurchlauf die Variable A auf einen zufällig ausgewählten Wert zwischen eins und sechs zu setzen. Bleibt ein Würfel mit einer Zwei oder einer Fünf liegen (Zeile 90), dann wird dieser auf Null gesetzt, bevor der Wert durch die nächste Zeile ausgedruckt wird.

Die einzelnen Würfe werden in Variable D addiert, und danach werden die Zehnerstellen subtrahiert (z.B. aus 27 wird zunächst 17 und dann 7). Dies geschieht in Zeile 150. Die sich ändernde Gesamtsumme wird jedesmal auf den Bildschirm gedruckt. Nach einer kurzen Pause druckt der Computer „Summe des ersten Wurfes...“ und überprüft dann in den Zeilen 180 bis 200, ob einer der besonderen Würfe aufgetreten ist. Wenn ja, dann wird eine dazu passende Bemerkung gedruckt. Wenn nein, geht der Computer zu Zeile 210, wo er testet, ob Würfel nochmals geworfen werden müssen (C wird immer, wenn 2 oder 5 geworfen wird, um 1 erhöht, dazu Zeile 90). Müssen keine Würfel wiederholt werden, so geht der Computer zu Zeile 410 und druckt „Meine Gesamtsumme ist...“. Ist jedoch C nicht Null, so durchläuft das Programm eine andere Schleife (240 bis 300), um jene Würfel nochmals zu werfen. Noch einmal, jeder 2 oder 5 zeigende Würfel wird beiseite gelegt (Zeile 280). Nachdem der Computer geworfen hat, kommen Sie an die Reihe. Eigentlich ist Ihre Aufgabe eher gering. Nachdem Sie RETURN gedrückt haben (Zeile 440), übernimmt es wieder der Computer, mittels der selben Subroutinen am Beginn des Programms die Würfe für Sie auszuführen und Ihre Punktezahl zu errechnen.

Nachdem Sie beiden Ihren Versuch hatten, entscheidet der Computer, ob er Sie oder Sie ihn geschlagen haben, oder ob ein Unentschieden vorliegt. Das Ziel von **Chemin de Computer** ist es, in neun Runden die größte Punktezahl zu haben (ein Unentschieden zählt nicht). Zeile 670 teilt Ihnen mit: „Die Summen in Chemin de Computer sind bis jetzt...“ und, sollten noch keine neun Runden gespielt worden sein, macht der Computer Bemerkungen zum Spiel, bevor er zum Hauptprogramm zurückkehrt.

Wenn neun gültige Runden gespielt wurden (Zeile 700), tritt Zeile 750 in Aktion. Diese Routine sagt Ihnen: „Na, alter Knabe, wir sind am Ende des Spieles nach neun Punkterunden angelangt...“ und ermittelt dann den endgültigen Gewinner.

Wenn Sie dieses Spiel angehen, befinden Sie sich in guter Gesellschaft. Baccarat, von dem Chemin de Fer abgeleitet wurde, ist während der Amtsperiode von Charles VIII gegen 1490 von Italien nach Frankreich gebracht worden. Das italienische Spiel heißt „Baccara“.

Ein Spiel läuft so ab:

```
>>> Das ist Spiel 1 <<<<
```

```
*****
Nun, wuerfle ich als Bank...
*****
21446
15
5
```

Gesamtzahl beim ersten Wurf ist 5

1 muessen nochmals geworfen werden
7

So ist mein Gesamtstand 2

RETURN um zu wuerfeln ?

24431
12
2

Gesamtzahl beim ersten Wurf ist 2

1 muessen nochmals geworfen werden
7
Ihr Gesamtstand ist also 9

Computer Mensch

2 9

Sie sind diesmal der Sieger!

Der Gesamtstand bisher in
Chemin de Computer ist
0 fuer mich, und 1 fuer Sie...

Und Sie duerften in Front liegen!

Hier das Listing:

```

10 REM Chemin
20 REM RANDOMIZE
30 CLS
40 SPIEL=0
50 B1=0:P1=0
60 GOTO330
70 D=0:C=0
80 FOR G=1 TO 5:A=INT(RND(1)*6)+1
90 IF A=2 OR A=5 THEN C=C+1
100 PRINT ;A;
110 GOSUB920
120 IF A=2 OR A=5 THEN A=0
130 D=D+A:NEXT G
140 PRINT:PRINT ;D;
150 IF D>9 THEN D=D-10:PRINT:PRINT ;D:GO
TO150
160 GOSUB890
170 PRINT:PRINT"Gesamtzahl beim ersten W
urf ist ";D
180 IF D=9 THEN PRINT"und das ist La Gra
nde...":RETURN
190 IF D=8 THEN PRINT"und das ist La Pet
ite...":RETURN
200 IF D=7 THEN PRINT"und das ist ein Na
tural...":RETURN
210 IF C=0 THEN RETURN
220 GOSUB920
230 PRINT:PRINT ;C;" muessen nochmals ge
worfen werden"
240 FOR A=1 TO C
250 GOSUB920
260 E=INT(RND(1)+6)+1
270 PRINT ;E;
280 IF E=2 OR E=5 THEN E=0
290 D=D+E
300 NEXT A
310 IF D>9 THEN D=D-10:GOTO310
320 RETURN

```

```

330 GOSUB920:CLS
340 SPIEL=SPIEL+1
350 PRINT:PRINT">>>> Das ist Spiel ";SPI
EL;" <<<<:PRINT
360 PRINT"*****
*****"
370 PRINT"Nun, wuerfle ich als Bank..."
380 PRINT"*****
*****"
390 GOSUB70
400 GOSUB890
410 PRINT:PRINT"So ist mein Gesamtstand
";D
420 GOSUB890
430 PRINT"*****
*****"
440 INPUT"RETURN um zu wuerfeln ";A$
450 PRINT"*****
*****"
460 J=D
470 GOSUB920
480 GOSUB70
490 PRINT:PRINT"Ihr Gesamtstand ist also
";D
500 PRINT:PRINT
510 GOSUB920
520 PRINT"Computer", "Mensch"
530 GOSUB890
540 PRINT ;J,;D
550 GOSUB920:PRINT
560 PRINT"$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$"
570 PRINT TAB(5);
580 IF J=D THEN PRINT"Das ist ein Unents
chieden":GOTO620
590 IF J<D THEN PRINT"Sie sind";:P1=P1+1
600 IF J>D THEN PRINT"Ich bin";:B1=B1+1
610 PRINT" diesmal der Sieger!"
620 PRINT"$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

```

```

$$$$$$$$$$$$"
630 PRINT:PRINT
640 GOSUB890
650 PRINT"Der Gesamtstand bisher in"
660 PRINT"Chemin de Computer ist"
670 PRINT ;B1;" fuer mich, und ";P1;" fu
er Sie..."
680 PRINT:PRINT
690 GOSUB920
700 IF B1+P1=9 THEN GOTO750
710 IF B1>P1 THEN PRINT"Sieht so aus, al
s ob ich fuehre!"
720 IF B1<P1 THEN PRINT"Und Sie duerften
in Front liegen!"
730 GOSUB890
740 GOTO330
750 REM Spielende
760 PRINT:PRINT
770 PRINT"Tja, alter Knabe, wir sind"
780 PRINT"am Ende des Spieles nach"
790 PRINT"neun Punkterunden angelangt..."
"
800 GOSUB890
810 IF P1>B1 THEN PRINT"Und Sie haben mi
ch":PRINT"schlagen koennen!":GOTO830
820 IF P1<B1 THEN PRINT"Und wieder einma
l ist diese":PRINT"maechtige Maschine der
Champion!!"
830 GOSUB890
840 PRINT"Danke fuer das Spiel, alter"
850 PRINT"Knabe, wir muessen es mal"
860 PRINT"wieder tun, wenn Sie sich"
870 PRINT"im Glueck fuehlen....."
880 END
890 FOR Z=1 TO 1000:NEXT Z
900 PRINT:PRINT
910 RETURN
920 FOR Z=1 TO 500:NEXT Z:RETURN

```


Sieben/Elf

Sieben/Elf ist ein anderes nettes Würfelspiel. Hier ändert sich das Ziel des Spieles von Mal zu Mal. Basierend auf **Schummeln** haben Sie auch hier eine „Zielnummer“ (in **Schummeln** heißt sie „Punkt“), die Sie zu erreichen trachten.

Schummeln ist eine amerikanische Abart des alten englischen Würfelspiels **Risiko**. In **Sieben/Elf** wurde das schwierige Wetten aus **Schummeln** weggelassen. Was bleibt, ist der reine Spaß an dem Spiel. Anstatt sich den Kopf darüber zu zerbrechen, worauf Sie wetten sollen, können Sie sich ernsthaft auf das Beten für den richtigen Wurf konzentrieren. In diesem Spiel spielen Sie gegen sich selbst. Damit Sie zu jeder Zeit während des Spieles wissen, wie es um Sie steht, werden Ihre gewonnenen und verlorenen Runden angeschrieben.

Sie werfen zwei Würfel auf ein Mal. Sie verlieren die Runde, wenn Sie beim ersten Wurf 2, 3 oder 12 erhalten. Das Beste, was Ihnen passieren kann, ist eine 7 oder 11 beim ersten Wurf. Damit haben Sie diese Runde auch schon mit einem einzigen Wurf gewonnen (daher auch der Name **Sieben/Elf**).

Jede andere Zahl, die Sie beim ersten Wurf erhalten, wird zu Ihrer „Zielnummer“. Sie müssen versuchen, diese nochmals zu erhalten, und zwar bevor Sie eine 7 oder 11 erzielen. Schaffen Sie es, Ihre „Zielnummer“ nochmals zu werfen, gewinnen Sie die Runde. Erhalten Sie jedoch eine 7 oder 11, bevor Sie Ihre „Zielsumme“ werfen können, verlieren Sie. Hier sind einige Momentaufnahmen des fortgeschrittenen Programmes:

Ihre bisherige Ausbeute:

Siege:0 Niederlagen:0

Sie haben \$100

Das ist Runde 1

Druecken Sie 'W' um zu wuerfeln

Sie haben 2 geworfen

und 5

In 7/11 haben Sie geworfen:7

Und so haben Sie gewonnen

Ihre bisherige Ausbeute:

Siege:1 Niederlagen:0

Sie haben \$115

Das ist Runde 2

Druecken Sie 'W' um zu wuerfeln

Sie haben 3 geworfen

und 5

In 7/11 haben Sie geworfen:8

Ihre Zielzahl ist 8

Druecken Sie 'W' um zu wuerfeln

Sie haben 6 geworfen

und 5

In 7/11 haben Sie geworfen:11

Und so haben Sie verloren

Ihre bisherige Ausbeute:

Siege:1 Niederlagen:1

Sie haben \$110

Und hier das Listing von **Sieben/Elf**:

```
10 REM SIEBEN ELF
20 B$="In 7/11 haben Sie geworfen:"
30 S=0:G=0:N=0:G=105
40 CLS
50 PRINT:PRINT"Ihre bisherige Ausbeute:"
"
60 PRINT"Siege: ";S;" Niederlagen: ";N
70 R=R+1
80 G=G-5
90 PRINT"Sie haben $";G
100 PRINT:PRINT"Das ist Runde ";R
110 GOSUB200
120 IF A=7 OR A=11 THEN GOTO300
130 IF A=2 OR A=3 OR A=12 THEN PRINT"und
das ist das Ende der Runde":GOTO320
140 P=A
```

```

150 CLS:PRINT:PRINT"Ihre Zielzahl ist ";
P
160 GOSUB200
170 IF A=P THEN GOTO300
180 IF A=7 OR A=11 THEN GOTO310
190 GOSUB340:GOTO150
200 Q=0
210 PRINT:PRINT"Druecken Sie 'W' um zu w
uerfeln"
220 Q=Q+S:IF INKEY$(10)=" " THEN GOTO220
230 C=INT(RND(1)*6)+1:PRINT"Sie haben ";
C;" geworfen"
240 GOSUB340
250 B=INT(RND(1)*6)+1:PRINT"und ";B
260 A=B+C
270 PRINT:PRINT B$;A
280 GOSUB340
290 RETURN
300 PRINT"Und so haben Sie gewonnen":S=S
+1:G=G+20:GOTO320
310 PRINT"Und so haben Sie verloren":N=N
+1
320 GOSUB340
330 GOTO40
340 FOR T=1 TO 2500:NEXT T
350 RETURN

```

Keine Ursache

Der englische Name dieses Spieles – **No Sweat** – kommt von einem altenglischen Namen für ein ähnliches Spiel, von dem dieses abstammt. Als dieses Spiel unter dem Namen 'Sweathcloth' in England gespielt wurde, verwendete man drei Würfel in einem hölzernen Schuh. Die am weitesten verbreitete Abart dieses Spieles finden wir heute in amerikanischen Spielkasinos, wo es, aufgrund der Ausrüstung, **Vogelkäfig** (Birdcage) heißt. Bei **Vogelkäfig** werden drei Würfel in einem metallenen Käfig, der sich um eine horizontale Achse drehen läßt, umhergewirbelt. Nach dem

Wetten wird der Käfig gedreht. Dieses deshalb, damit die Würfel ordentlich geworfen werden, da sie kein Kasinoangestellter mehr berührt. Bei **Vogelkäfig** und **Keine Ursache** zu wetten, geht einfach und schnell. Obwohl Sie glauben werden, daß der Zufall auf Ihrer Seite, der des Spielers, ist, werden Sie bald sehen, wie leicht es ist, hier sein letztes Hemd zu verlieren.

Sie wetten einen bestimmten Betrag (siehe Zeilen 50 und 60) bis zu der Höhe Ihres Guthabens (repräsentiert durch die Variable G für Geld – siehe Zeilen 30 und 250) und wählen dann eine Zahl zwischen 1 und 6. Der Betrag Ihrer Wette wird von Ihrem Guthaben subtrahiert, und dann werden die drei Würfel geworfen. Für jeden, der Ihre Zahl zeigt, erhalten Sie den Betrag Ihrer Wette.

Wie Sie sehen, müssen Sie zumindest einen Würfel mit Ihrer Nummer werfen, um keinen Verlust zu haben. Erst bei zwei, die Ihre Nummer zeigen, gewinnen Sie. Da man die gleiche gewählte Nummer gleich zweimal erzielen muß, hat das Kasino die besseren Chancen in diesem Spiel.

Nachdem der Zufallsgenerator in Zeile 20 eingeschaltet wird, die Geldvariable (G) in Zeile 30 auf 30 für Sie gesetzt wird, geht der Computer zu Zeile 240 und druckt:

Sie haben jetzt \$ 30.

Danach legt das Spiel los.

Zeile 50 fragt Sie, wieviel Sie wetten wollen, und dementsprechend wird Variable A belegt. Ihr Wettbetrag wird nun in Zeile 60 mit Ihrem Gesamtbesitz verglichen, damit Sie nicht mehr verwetten, als Sie haben. Nachdem Sie diese Hürde geschafft haben, zieht Zeile 70 diesen Betrag von Ihrem Guthaben ab, und der Computer verlangt dann, daß Sie die Wette eingeben (Zeile 50).

Die C-Schleife (Zeilen 100 bis 180) würfelt nun, vergleicht jeden Wurf mit Ihrer Nummer und teilt Ihnen die geworfene Zahl mit. Ihre Gewinne, sollte es welche geben, werden auch in dieser Schleife zusammengezählt.

Nach jeder Runde wird Ihr Geld mit 250 verglichen (haben Sie mehr als \$ 250, so haben Sie die Bank gesprengt und dürfen nicht mehr spielen) und mit 0 haben Sie kein Geld mehr, so KÖNNEN Sie nicht mehr spielen). Haben Sie jedoch nicht mehr als \$ 250 und nicht weniger als 0, dann können Sie Ihr Spiel fortsetzen. Hier ist **Keine Ursache** in voller Aktion:

```

*****
      Sie haben nun $30
*****
Wieviel wollen Sie setzen ?20

```

Auf welche Zahl setzen Sie ?5

```

Wuerfel 1 fiel auf 3
*****
Sie haben nun $10
*****

Wuerfel 2 fiel auf 5
So gewinnen Sie $20
*****
Sie haben nun $30
*****

Wuerfel 3 fiel auf 6
*****
Sie haben nun $30
*****

*****
Sie haben nun $30
*****
Wieviel wollen Sie setzen ?10

Auf welche Zahl setzen Sie ?6

Wuerfel 1 fiel auf 3
*****
Sie haben nun $20
*****

Wuerfel 2 fiel auf 5
*****
Sie haben nun $20
*****

Wuerfel 3 fiel auf 2
*****
Sie haben nun $20
*****

```

Hier ist das Listing, damit Sie und Ihr Computer ein oder zwei Runden spielen können:

```

10 REM KEINE URSACHE
20 REM RANDOMIZE
30 G=30
40 CLS:GOSUB240
50 INPUT"Wieviel wollen Sie setzen ";A
60 IF A>G THEN GOTO50
70 G=G-A:PRINT:PRINT
80 INPUT"Auf welche Zahl setzen Sie ";B
90 IF B<1 OR B>6 THEN GOTO80
100 FOR C=1 TO 3
110 W=0
120 GOSUB280
130 D=INT(RND(1)*6)+1
140 PRINT:PRINT"Wuerfel ";C;" fiel auf "
;D
150 IF D=B THEN W=A:PRINT"So gewinnen Si
e $";W
160 G=G+W
170 GOSUB240
180 NEXT C
190 GOSUB280:GOSUB280
200 IF G>250 THEN GOTO310
210 IF G>0 THEN GOTO40
220 PRINT"Das Spiel ist vorbei, weil Sie
":PRINT"pleite sind!"
230 GOTO220
240 PRINT"*****
*****"
250 PRINT" Sie haben nun $";G
260 PRINT"*****
*****"
270 RETURN
280 FOR P=1 TO 1000
290 NEXT P
300 RETURN
310 FOR J=1 TO 30
320 PRINT"Sie haben $250 ueberschritten!

```

```
"
330 PRINT TAB(J); "Bravo !"
340 NEXT J
```

Eins-und-Zwanzig

Unsere Würfelspiele setzen wir nun mit **Eins-und-Zwanzig** fort. Sicher haben Sie schon beim Lesen des Namens in der Einführung erkannt, daß es sich hierbei um die Würfelversion des Kartenspiels **Pontoon** handelt.

Das Spiel ist einfach, verlangt aber einige kühle Überlegungen und die Fähigkeit, das Ergebnis des nächsten Wurfes zu erraten. In **Eins-und-Zwanzig** spielen Sie gegen den Computer.

Sie dürfen immer als Erster würfeln. Sie würfeln, so oft Sie wollen. Dabei versuchen Sie, so knapp wie möglich an 21 heranzukommen, jedoch nicht mehr zu erzielen. Wann immer Sie wollen, können Sie aufhören zu werfen.

Der Computer gewinnt die erste Runde automatisch, wenn Sie über 21 kommen (wenn Sie 'gesprengt' sind). Ein Spiel besteht aus fünf Runden. Natürlich gewinnt der Sieger der meisten Runden das Spiel. Nachdem der Computer einige Variable in den Zeilen 20 und 30 initialisiert (darunter MP für die Punktezahl des Spielers und CP für die des Computers) und den Bildschirm in Zeile 40 gelöscht hat, verlangt er von Ihnen „W' für würfeln, 'S' für stehen...“. Der Ausdruck „stehen“ bedeutet, daß Sie mit der bis dahin erreichten Punktezahl zufrieden sind und dabei bleiben wollen. Nun geben Sie dem Computer die Chance, sein Glück zu versuchen. Bald werden Sie herausgefunden haben, daß die Geschwindigkeit, mit der Sie das 'W' oder 'S' eingeben, die geworfene Zahl beeinflusst. Die Verzögerung zwischen dem „W...“ und dem Moment, an dem Sie wirklich 'W' drücken, wird zur Erzeugung einer Zahl (N) verwendet. Diese Zahl zusammen mit Ihrem Punktestand setzt den Zufallsgenerator für den nächsten Wurf in Gang.

Die Zeilen 50 bis 170 kontrollieren Ihren Wurf, aus dem der Computer nur aussteigt, wenn Sie sich entscheiden, stehen zu bleiben. Die ganze Logik des Computerwerks steckt in Zeile 180. Dort wird entschieden, ob das Risiko einer Pleite bei dem Versuch, Ihre Punktezahl einzustellen, den Versuch wert ist oder nicht. Der Wurf wird dann in den Zeilen 190 bis 250 entschieden.

Hat der Computer entschieden, daß er genug davon hat, ermittelt er in den Zeilen 260, 280 und 290, wer gewonnen hat. Sind beide Punktezahlen gleich oder über 21, dann wird diese Runde als „tote Runde“ gewertet. Zeile 320 teilt Ihnen dieses mit. Weder CP noch MP werden dann erhöht. Zeile 280 erkennt den Sieg des Computers, zählt 1 zu CP

dazu und druckt „Ich gewinne“, wohingegen die nächste Zeile MP um 1 erhöht und „Sie gewinnen“ druckt.

In **Chemin de Computer** versuchten Sie, die Mehrzahl von neun gültigen Runden zu gewinnen. So auch bei **Eins-und-Zwanzig**, wo ein Spiel allerdings nur fünf Runden dauert. Ob Sie es geschafft haben, überprüft Zeile 290, und sollte es zutreffen, springt das Programm zu Zeile 450. Dort wird dann das Ergebnis, versehen mit ein oder zwei sarkastischen Bemerkungen, gedruckt.

Wurden noch nicht fünf gültige Runden abgeschlossen, so bittet Sie Zeile 400 „Warten Sie auf die nächste Runde...“, und nach einer kurzen Pause (Zeile 430) kann die nächste Runde durch Sie eröffnet werden. Sehen wir uns nun einen Teil des Programmes in Aktion an:

'W' fuer Wuerfeln, 'S' fuer stehen

Sie haben eine 1 geworfen
und Ihre Summe ist 1

'W' fuer Wuerfeln, 'S' fuer stehen

Sie haben eine 3 geworfen
und Ihre Summe ist 4

'W' fuer Wuerfeln, 'S' fuer stehen

Sie haben eine 5 geworfen
und Ihre Summe ist 9

'W' fuer Wuerfeln, 'S' fuer stehen

Sie haben eine 2 geworfen
und Ihre Summe ist 11

'W' fuer Wuerfeln, 'S' fuer stehen

Sie haben eine 2 geworfen
und Ihre Summe ist 13

'W' fuer Wuerfeln, 'S' fuer stehen

Sie haben eine 2 geworfen
und Ihre Summe ist 15

'W' fuer Wuerfeln, 'S' fuer stehen

Sie haben eine 4 geworfen
und Ihre Summe ist 19

'W' fuer Wuerfeln, 'S' fuer stehen

Sie haben eine 4 geworfen
und Ihre Summe ist 23

'W' fuer Wuerfeln, 'S' fuer stehen

Ihre Summe: 23

Ich habe gewonnen!!

Nach dieser Runde ist der Stand:
Sie:0 und ich:1

Das Listing wird es Ihrem Computer ermöglichen, Sie in Eins-und-Zwanzig zu fordern und wahrscheinlich zu schlagen:

```

10 REM EINS-UND-ZWANZIG
20 MP=0:CP=0
30 M=0:C=0:N=0
40 CLS
50 PRINT"'W' fuer Wuerfeln, 'S' fuer stehen"
60 A$=INKEY$(10)
70 N=N+1
80 IF A$<>"S" AND A$<>"s" AND A$<>"W" AND A$<>"w" THEN GOTO60
90 IF A$="S" OR A$="s" THEN CLS:PRINT"Ihre Summe: ";M:GOTO180

```

```

100 REM RANDOMIZE (N+M)
110 GOSUB430
120 R=INT(RND(1)*6)+1
130 PRINT:PRINT"Sie haben eine ";R;" geworfen"
140 M=M+R
150 PRINT"und Ihre Summe ist ";M
160 GOSUB430
170 PRINT:GOTO50
180 IF C>M AND C<22 OR C>21 OR M>21 OR M=21 AND C=21 THEN GOTO260
190 R=INT(RND(1)*6)+1
200 GOSUB430
210 PRINT:PRINT"Ich habe eine ";R;" geworfen"
220 C=C+R
230 PRINT"und meine Summe ist ";C
240 GOSUB430
250 GOTO180
260 IF M=C OR M>21 AND C>21 THEN GOTO320
270 GOSUB510
280 IF (C>M OR M>21) AND C<22 THEN PRINT,"Ich habe";:CP=CP+1
290 IF (C<M OR C>21) AND M<22 THEN PRINT,"Sie haben";:MP=MP+1
300 PRINT" gewonnen!!"
310 GOTO330
320 PRINT"Diese Runde ist ein alter Schuch...":PRINT"keine Punkte"
330 GOSUB430
340 GOSUB510
350 PRINT"Nach dieser Runde ist der Stand:"
360 GOSUB430
370 PRINT,"Sie: ";MP;" und ich: ";CP
380 GOSUB510
390 IF CP+MP=5 THEN GOTO450
400 PRINT:PRINT"Warten Sie auf die naechste Runde..."

```

```

410 GOSUB430:GOSUB430
420 PRINT:PRINT:GOTO30
430 FOR I=1 TO 900:NEXT
440 RETURN
450 PRINT:PRINT"Das ist das Spielende"
460 PRINT:PRINT"Endstand:"
470 PRINT,"Sie: ";MP
480 PRINT,"Ich: ";CP:PRINT
490 IF CP>MP THEN PRINT"Und ich habe Sie
geschlagen, Mensch!":END
500 PRINT"Und, bemerkenswert, eine bloss
er Mensch":PRINT"hat die Maschine besiegt!
":END
510 PRINT:PRINT"*****
*****":PRINT
520 RETURN

```

Schlangenaugen

Schlangenaugen verlangt etwas mehr kühles Denken unter Druck. Sie werfen abwechselnd mit dem Computer zwei Würfel. Die Augenzahlen der beiden Würfel werden zusammengezählt und zu der Gesamtpunktzahl addiert.

Sie können so oft würfeln, wie Sie wollen. Werfen Sie jedoch eine 7, dann haben Sie automatisch verloren. Wie Sie sehen können, verlangt das Programm von Ihnen die Entscheidung, ob Sie – durch Vorsicht dazu getrieben – nicht mehr würfeln sollen und die Runde vielleicht wegen zu geringer Punkte verlieren, oder ob Sie – von Ihrer Gier getrieben – auf eine höchstmögliche Punktzahl spekulieren und das Risiko eingehen, eine 7 zu erhalten.

Wie auch in **Eins-und-Zwanzig** hat hier die Zeit, die Sie dazu benötigen, 'RETURN' zu drücken und damit zu würfeln, Einfluß auf die Punktzahl, die Sie erreichen. Sie können sehen, daß die Variable N, die in 120 auf Null gesetzt wird, in 120 bis 150 bei jedem Durchlauf der Schleife 140/150 um eins erhöht wird, bis Sie 'w', 'W', 'e' oder 'E' drücken. Drücken Sie 'w' oder 'W' und zeigen damit, daß Sie würfeln wollen, springt das Programm zu der Subroutine in Zeile 370, die den Bildschirm löscht (380) und dann die beiden Würfel (X und Y) wirft. Die Q-Schleife wird 20mal durchlaufen. Die innere Schleife der Zeile 430 bewirkt eine stetige Verlangsamung während der Abarbeitung, bis

endlich in Zeile 420 die Resultate der Würfe gedruckt werden. Die Variable Z wird gleich der Summe der beiden Würfe gesetzt (in Zeile 480), bevor der Computer zu Zeile 700 kommt, eine Pause einlegt und eine Zeile auf den Bildschirm druckt. Dann kehrt er zu der Folgezeile – zu der, mit der die Würfelroutine beginnt – am Anfang des Programmes zurück (180). Hier überprüft der Computer, ob die Summe dieses Wurfes nicht 7 ist (Zeile 180). Wenn nein, addiert er den letzten Wurf zu Ihrer Gesamtpunktzahl.

Wenn Sie jedoch durch INKEY\$ Ihrem Wunsch Ausdruck geben, nicht mehr weiterzuwürfeln, springt der Computer auf Zeile 220. Nachdem er „Bitte warten“ gedruckt hat, geht der Computer zu der Verzögerungsschleife und der Druckroutine. Danach verwendet er dieselbe Routine wie der Spieler und wirft seine Würfel. Kommt er von dieser Routine zurück, so prüft er zunächst, ob die Summe des Wurfes 7 ist. Wenn nicht, addiert auch er den Wurf zu seiner Gesamtpunktzahl. In der Entscheidung, ob er nochmals würfeln soll, beschreitet der Computer einen ganz einfachen Weg: hat er weniger als sein Gegenspieler, versucht er es. Sie werden finden, daß es relativ einfach ist, Programme wie dieses in Ihren Computer einzugeben. Haben Sie die „mechanischen Routinen“ einmal durchschaut, damit meine ich die, die das eigentliche Würfeln oder das Addieren der Punkte übernehmen, so wird es Ihnen nicht schwerfallen, ein anderes Programm zu entwickeln, das sich dieser Routinen auch bedient. Die meisten Würfelspiele bedürfen nicht viel Intelligenz, und ihre „Strategie“ kann meist auf ein paar IF/THEN-Befehle reduziert werden. Um Ideen für Würfelprogramme zu erhalten, lesen Sie doch Bücher über Würfelspiele, wie z. B. das sehr gute Buch 'Dice Games Old and New' von William E. Tredd (The Oleander Press, New York, 1981). Dort werden Sie auch einfache Anleitungen finden, wie Sie diese Spiele am besten spielen. Aus diesen simplen Anleitungen sollten Sie auch sehr einfach relativ „intelligente Algorithmen“ erstellen können, so daß der Computer dann auch ganz gegen Sie spielen können sollte.

Wenn Sie, wie z. B. bei **Eins-und-Zwanzig**, vereinbaren, daß der Spieler als Erster würfelt, weiß der Computer genau, welches Ziel er anstrebt. Daher hat er einen gewissen Vorteil, der die Dummheit der Maschine ausgleichen hilft.

Hier sehen Sie, wie eine Partie **Schlangenaugen** aussieht:

Bitte warten Sie...

Ihre Summe ist 0

'W' um zu wuerfeln
'E' fuer Ende

Wuerfel eins: 3 Wuerfel zwei: 5

Ihre Summe ist 8

Ihre Summe ist 8

'W' um zu wuerfeln
'E' fuer Ende

Wuerfel eins: 5 Wuerfel zwei: 5

Ihre Summe ist 18

Ihre Summe ist 18

'W' um zu wuerfeln
'E' fuer Ende

Wuerfel eins: 1 Wuerfel zwei: 6

Ich habe gewonnen!

Der Punktestand:

Sie: 0

Ich: 1

Und ich liege voran...

Bitte warten Sie...

Ihre Summe ist 0

'W' um zu wuerfeln
'E' fuer Ende

Wuerfel eins: 5 Wuerfel zwei: 5

Ihre Summe ist 10

Ihre Summe ist 10

'W' um zu wuerfeln
'E' fuer Ende

Wuerfel eins: 5 Wuerfel zwei: 4

Ihre Summe ist 19

Ihre Summe ist 19

'W' um zu wuerfeln
'E' fuer Ende

Wuerfel eins: 3 Wuerfel zwei: 1

Ihre Summe ist 23

Ihre Summe ist 23

'W' um zu wuerfeln
'E' fuer Ende

Bitte warten

Wuerfel eins: 2 Wuerfel zwei: 6

Meine Summe ist 8
Ihre Summe ist 23

Bitte warten

Wuerfel eins: 1 Wuerfel zwei: 2

Meine Summe ist 11
Ihre Summe ist 23

Bitte warten

Wuerfel eins: 5 Wuerfel zwei: 5

Meine Summe ist 21
Ihre Summe ist 23

Bitte warten

Wuerfel eins: 5 Wuerfel zwei: 5

Meine Summe ist 31
Ihre Summe ist 23

Ich habe gewonnen!

Der Punktestand:
Sie: 0
Ich: 2

Und hier ist das Listing für das Spiel:

```

10 REM SCHLANGENAUGEN
20 CLS
30 M=0:CZ=0
40 MP=0:CP=0
50 PRINT"Bitte warten Sie..."
60 GOSUB700
70 CLS:PRINT:PRINT
80 PRINT"Ihre Summe ist ";MP:PRINT
90 GOSUB710
100 PRINT"'W' um zu wuerfeln"
110 PRINT"'E' fuer Ende"
120 N=0
130 W$=INKEY$(10)
140 N=N+1
150 IF W$<>"E" AND W$<>"e" AND W$<>"W" A
ND W$<>"w" THEN GOTO130
160 IF W$="E" OR W$="e" THEN GOTO220
165 REM RANDOMIZE N
170 GOSUB370
180 IF Z=7 THEN GOTO350
190 MP=MP+Z
200 PRINT:PRINT"Ihre Summe ist ";MP
210 GOTO60
220 PRINT:PRINT:PRINT"Bitte warten"
230 GOSUB700
240 GOSUB370
250 IF Z=7 THEN GOTO330
260 CP=CP+Z

```

```

270 PRINT:PRINT"Meine Summe ist ";CP
280 PRINT"Ihre Summe ist ";MP
290 IF CP<MP THEN GOTO220
300 IF CP=MP THEN PRINT"Unentschieden!"
310 IF CP>MP THEN GOTO350
320 GOTO510
330 PRINT:PRINT"Sie haben gewonnen!";M=I
+1
340 GOTO510
350 PRINT:PRINT"Ich habe gewonnen!";CZ=
Z+1
360 GOTO510
370 REM Wuerfeln
380 CLS
390 FOR Q=1 TO 20
400 X=INT(RND(1)*6)+1
410 Y=INT(RND(1)*6)+1
420 PRINT:PRINT"Wuerfel eins: ";X;" W
uerfel zwei: ";Y
430 FOR P=1 TO 3*Q:NEXT P
440 NEXT Q
450 CLS
460 GOSUB710
470 PRINT"Wuerfel eins: ";X;" Wuerfel
zwei: ";Y
480 Z=X+Y
490 GOSUB700
500 RETURN
510 PRINT:PRINT"Der Punktestand:"
520 PRINT,"Sie: ";M
530 PRINT,"Ich: ";CZ
540 IF CZ+M=9 THEN GOTO580
550 GOSUB710
560 IF M>CZ THEN PRINT"Sie fuehren!"
570 IF CZ>M THEN PRINT"Und ich liege v
an..."
580 GOSUB700
590 CLS
600 IF CZ+M=9 THEN GOTO620

```

```

610 GOTO40
620 PRINT:PRINT:PRINT
630 PRINT"Tja, mein Lieber, das ist da
640 PRINT"Ende des Spiels..."
650 PRINT:PRINT:PRINT
660 PRINT"Ihr Endstand war ";M
670 PRINT"und meiner ";CZ:PRINT
680 IF M<CZ THEN PRINT"Ich bin der Sie
r!!":END
690 PRINT"Sie sind der Sieger!!":END
700 FOR P=1 TO 3000:NEXT P:RETURN
710 PRINT"-----":P
NT:RETURN

```

Drunter und Drüber

Drunter und Drüber ist eine Computerabart des Würfelspiels **Unter und über sieben** (Under and over Seven). In diesem Spiel wetten Sie auf die Wahrscheinlichkeit, daß das Ergebnis des Wurfes mit zwei Würfeln

- kleiner als 7,
- genau 7 oder
- größer als 7 ist.

Auf den ersten Blick ist auch dieses ein Spiel, das sehr gute Chancen für den Spieler verspricht. Wenn Sie aber dieses Spiel mit zwei perfekten Würfeln bis in alle Ewigkeit spielen würden, fänden Sie, daß Ihre Verluste Ihre Gewinne um fast 17% übertreffen würden.

Die Struktur des Programmes ist nicht schwer zu verfolgen. Nachdem der Zufallsgenerator in Zeile 20 aktiviert und der Bildschirm in Zeile 30 gelöscht wurde, gibt Ihnen der Computer ein Anfangskapital von 30 (für \$ 30), mit dem die Variable G (für Geld) bedacht wird. Zeile 50 springt zur Subroutine von 450, in der Ihre Finanzlage gedruckt wird („Sie haben jetzt \$ 30“). Danach wird in Zeile 60 die Verzögerungssubroutine der Zeilen 470, 480 und 490 aufgerufen.

Die Zeilen 70 bis 120 fragen Sie nach Ihrer Wette und sagen Ihnen auch, wie Sie diese durchführen müssen (drücken Sie 'A', um unter 7, 'B', um auf und 'C', um über 7 zu wetten). Die Zeilen 130 und 140 lesen durch Verwendung von INKEY\$ die Eingabe und weisen alle Tasten zurück, die nicht A, B oder C sind.

Haben Sie eine gültige Wahl getroffen, so erfahren Sie durch die Zeilen 150 bis 180 die Auszahlungsquoten der einzelnen Wettmöglichkeiten,

und Zeile 200 fragt Sie nach dem Betrag, den Sie wetten wollen. Natürlich können Sie nicht mehr verwetten, als Sie haben, nicht in diesem Spiel. Zeile 220 prüft daher Ihre Wette (A) gegen Ihr Kapital und, sollten Sie nicht genug haben, wird „Soviel haben Sie nicht!“ ausgedruckt.

Der nächste Programmteil würfelt. Er ordnet den Variablen B und C Zufallszahlen zwischen 1 und 6 zu und addiert sie in Zeile 280 und ordnet diese D zu. Zeile 300 teilt Ihnen diese Summe mit.

Das Ergebnis

Das Ergebnis wird durch die Routine der Zeilen 310 bis 350 bestimmt. Zunächst wird in Zeile 310 ein Verlust angenommen, der Variablen 'W' für 'Gewinn' wird der negative Wert Ihrer Wette zugeordnet. Später wird dann dieser Wert, wenn wirklich ein Gewinn eingetreten ist, in den richtigen Betrag umgeändert. Wie Sie sehen können, erhalten Sie Ihren vierfachen Einsatz, wenn Sie richtig erraten haben, daß der Wurf eine 7 ergibt (Zeile 320), und den einfachen, wenn Sie über oder unter 7 korrekt erraten haben (Zeilen 330 und 340).

Die Zeilen 370 und 380 teilen Ihnen Ihren Gewinn oder Verlust mit, Zeile 420 prüft, ob Sie noch über Spielkapital verfügen. Wenn Sie noch Geld besitzen, so wird Ihnen noch eine Spielrunde angeboten. Haben Sie keines mehr, dann werden Sie mit einer gewissen Heftigkeit verabschiedet.

Hier ist Drunter und Drüber in Aktion:

Sie haben nun \$30

Gut, Spieler, Zeit Ihre Wette abzugeben.....

'A' fuer Wette unter 7

'B' fuer Wette auf 7

'C' fuer Wette ueber 7

Die Gewinnausschuetzung ist:-

A - bringt 1 zu 1

B - bringt 4 zu 1

C - bringt 1 zu 1

Wieviel wollen Sie setzen ?30

Wuerfel eins fiel auf 6

Wuerfel zwei fiel auf 3

Die Summe ist daher 9

Und Sie verlieren \$30

Sie haben nun \$0

Sie sind total pleite, mein Bester,
und so muss ich das Casino schliessen.

Es gibt keinen Platz fuer Taugenichtse!

Hier ist das Listing, um Ihren Computer in ein Würfelfas zu verwandeln:

```
10 REM DRUNTER UND DRUEBER
20 REM RANDOMIZE
30 CLS
40 G=30
50 GOSUB450
60 GOSUB470
70 PRINT"Gut, Spieler, Zeit Ihre"
80 PRINT"Wette abzugeben....."
90 GOSUB470
100 PRINT"'A' fuer Wette unter 7"
110 PRINT"'B' fuer Wette auf 7"
120 PRINT"'C' fuer Wette ueber 7"
130 A$=INKEY$(10)
```

```
140 IF A$<>"A" AND A$<>"a" AND A$<>"B" A
ND A$<>"b" AND A$<>"C" AND A$<>"c" THEN GO
TO 130
150 PRINT:PRINT"Die Gewinnausschuetzung
ist!:-"
160 PRINT" A - bringt 1 zu 1"
170 PRINT" B - bringt 4 zu 1"
180 PRINT" C - bringt 1 zu 1"
190 GOSUB470
200 INPUT"Wieviel wollen Sie setzen ";A
210 GOSUB470
220 IF A>G THEN PRINT"Soviel haben Sie n
icht!":PRINT:GOTO200
230 B=INT(RND(1)*6)+1
240 PRINT,"Wuerfel eins fiel auf ";B
250 GOSUB470
260 C=INT(RND(1)*6)+1
270 PRINT,"Wuerfel zwei fiel auf ";C
280 D=C+B
290 GOSUB470
300 PRINT,"Die Summe ist daher ";D
310 W=-A
320 IF D=7 AND (A$="B" OR A$="b") THEN W
=4*A
330 IF D<7 AND (A$="A" OR A$="a") THEN W
=A
340 IF D>7 AND (A$="C" OR A$="c") THEN W
=A
350 G=G+W
360 GOSUB470
370 IF W>0 THEN PRINT"Sie haben gerade $
";W;" gewonnen"
380 IF W<0 THEN PRINT"Und Sie verlieren
$";-W
390 GOSUB470
400 GOSUB450
410 GOSUB470
420 IF G<1 THEN PRINT"Sie sind total ple
ite, mein Bester,":PRINT"und so muss ich d
```

```

as Casino schliessen.":PRINT:PRINT"Es gibt
keinen Platz fuer Taugenichtse!":END
430 CLS
440 GOTO50
450 PRINT:PRINT"Sie haben nun $";G
460 RETURN
470 FOR P=1 TO 2000:NEXT P
480 PRINT:PRINT
490 RETURN

```

Künstliche Intelligenz

Künstliche Intelligenz ist ein Ziel, das noch nicht erreicht werden konnte. Freilich sind schon Programme geschrieben worden, die dem Computer Verhaltensweisen geben, die als 'intelligent' eingestuft werden könnten. Aber die 'Intelligenz' ist auf ein sehr kleines Gebiet beschränkt und auch nur in diesem kleinen Rahmen wirksam. Ein Computer kann daher nur dann sein Köpfchen zeigen, wenn das Umfeld, in dem er seine Klugheit beweisen muß, sehr begrenzt wurde.

In diesem Kapitel haben wir 4 Programme, die Ihrem Computer den Anschein von Intelligenz geben werden. Gegenüber Leuten, die nicht viel von Computern verstehen, stellen diese Programme sicherlich eindrucksvolle Demonstrationen der Verstandeskraft von Computern dar. Trotz der Aussagen einiger Leute (wie z.B. Carl Sagan "In Defense of Robots", daß der Mensch eigentlich nur ein sehr gut konstruierter Computer ist, erahnen wir, daß zwischen unserem Bewußtsein unserer Existenz und dem vollkommenen Nicht-Erfassen der Computer ihrer eigenen Existenz (zumindest heutzutage) Welten liegen. Mir scheint, daß diese Selbstwahrnehmung doch ein wichtiger Bestandteil zumindest eines Aspektes von Intelligenz ist.

Computer können in begrenzten Gebieten sehr 'intelligent' arbeiten. Sehen Sie sich nur 20x20 cm großen Schachmaschinen an. Innerhalb des begrenzten Bereiches des Schachspieles kann der Eindruck einer soliden Klugheit geschaffen werden.

Innerhalb der begrenzten Welten trifft dies auch auf die Programme in diesem Kapitel zu. Eines davon ist eine Variante von einem der bekanntesten und populärsten Künstliche-Intelligenz-Programme, das je geschrieben wurde: ELIZA. Der Computer übernimmt dabei die Rolle eines Psychiaters und führt mit Ihnen regelrecht ein Gespräch. Joseph Weizenbaum entwickelte ELIZA in der Mitte der 60er Jahre. Er war bemüht, ein Programm zu schaffen, das einen Rogerianischen Analytiker imitieren würde. Es wird vermutet (siehe "You're never alone with a fluffy micro", David Tebbutt, MicroScope, 13. Januar 1983, S. 9), daß Weizenbaum seit damals seine Arbeit an ELIZA bereut hat. Es ist nicht schwer, den Grund dafür zu sehen.

In den euphorischen Tagen, nachdem das Programm veröffentlicht wurde, kamen die wilden und unberechenbaren Behauptungen auf, es wäre nur noch eine Frage von (sehr kurzer) Zeit, wann Computer Psychiater vollständig ersetzen würden. Weiter, so einige unvorsichtige Beobachter, würde der Erfolg ELIZAs den Weg für wirklich intelligente Computer ebnen.

Weizenbaum hat einen Moment daran gezweifelt. Er schreibt, er wäre am Anfang und auch noch später von den Reaktionen der Leute über

sein Programm verblüfft gewesen. In USP auf einem Heimcomputer an MIT geschrieben, konnte jeder an den über den ganzen Campus verstreuten 'time-sharing'-Bildschirmen auf das Programm zugreifen. Weizenbaum sagt, es war sehr interessant zu sehen, daß das Programm sehr oft spät nachts abgerufen wurde und daß geplagte Studenten wirklich ihre Probleme mit dem Programm diskutierten. Auch waren die Zugriffszeiten sehr lang. Die Aufzeichnungen zeigten, daß einige Leute länger als eine Stunde an ELIZA angeschlossen waren. Das Programm durchsucht die Aussage der Person nach 'Schlüsselwörtern' (wie Freund oder Traum) und sucht dann eine passende Antwort aus einer Anzahl zur Verfügung stehender Möglichkeiten. Wörter aus den von den Personen gestellten Sätzen konnten eingebaut oder ganze Sätze herumgedreht werden (wie z.B. „Wegen des Wetters bin ich glücklich“ konnte einfach als „Warum sind Sie wegen des Wetters glücklich“ oder als Feststellung der Form „Sie sind wegen des Wetters glücklich. Was schließen Sie daraus?“ zurückgegeben werden). Eine Anzahl von anderen Antworten (so z.B. „Das ist sehr interessant, bitte fahren Sie fort“) konnte verwendet werden, wenn kein Schlüsselwort erkannt wurde.

Weizenbaums Sekretärin hat mit ihm die 6 Monate, die die Entwicklung des Programmes etwa brauchte, sehr eng zusammengearbeitet. Obwohl die Sekretärin mit dem Programm vertraut war, bemerkte Weizenbaum, daß, wenn er in sein Büro kam und die Sekretärin gerade mit dem Programm spielte, sie verlegen wurde und ihn die Ausdrücke nicht sehen ließ. Als er vorschlug, einen Drucker an den Computer anzuschließen, um die spätabendlichen Gespräche der Studenten mit dem Programm aufzuzeichnen, wurde dies mit Schrecken abgelehnt, als wäre es eine Art elektronischen Voyeurismus. (Wenn Sie mehr über Weizenbaums Arbeit auf dem Sektor der Künstlichen Intelligenz wissen wollen, wird Sie dieses Buch sicherlich interessieren: 'Computer Power and Human Reason', W.H. Freeman & Co., San Francisco, 1976.) Ein weitverbreiteter Stolperstein in der Arbeit der ersten Forscher auf dem Gebiet der künstlichen Intelligenz war, daß das Schreiben eines mit dem menschlichen Folgern wetteifernden Programmes (wie z.B. ein sehr interessantes Dame-Programm von Samuels) die Forscher nicht unbedingt dem Erzeugen von Verhalten, das als intelligent bezeichnet werden könnte, oder dem theoretischen Verstehen des menschlichen Folgerns und Schließens nahebrachte. „Der Fortschritt im Erzeugen intelligenter Verhaltensweisen ist nicht unbedingt auch Fortschritt in der Produktion von Gedanken“. (Normal Whaland, When Is A Program Intelligent?, Creative Computing, im Februar 1981, Seiten 44–49).

Es gibt noch einen anderen Weg, an das Programm heranzugehen. Ich nenne es die „Wenn es quakt wie eine Ente, dann ist es eine Ente“-Annäherung. In dem Buch "The Turing Criterion – Machine Intelligent Programs for the 16K ZX81" (Charlton et al., Interface Publications, London, 1982), das ich herausgab, beantworte ich die Frage „Was ist das Turing-Kennzeichen?“ mit den folgenden Worten:

„1950 schlug der englische Mathematiker und Logiker A. M. Turing etwas vor, das als das 'Turing-Merkmal' für Maschinenintelligenz bekannt werden sollte.

Er sagte, daß, wenn Sie mit etwas am Ende eines Drahtes, das Mensch oder Maschine sein könnte, zu tun haben, und Sie können mit den Antworten, die Sie erhalten, nicht feststellen, ob Sie es mit einem Menschen oder mit einer Maschine zu tun haben, so ist das 'Ding', oder was es auch immer sein mag, per Definition intelligent.“

Auf dieser Basis wäre es möglich, sehr viele Computerprogramme als intelligent zu klassifizieren. Natürlich könnten alle Programme dieses Kapitels diesen Test bestehen. Ganz besonders dann, wenn Sie die Möglichkeit erwägen, daß Ihnen einige sehr dumme Menschen gegenüber sitzen.

Doch jetzt ist es Zeit, die Welt der künstlichen Intelligenz mit folgenden Programmen zu erforschen:

- Eliza
- Elektronisches Gehirn (wo der Computer versucht, einen 'Master-mind'-ähnlichen Code zu knacken, den Sie vorgegeben haben)
- Memory
- Dichter (in dem sich der Computer als Gedichteschreiber profiliert)

Eliza

Wir werden unsere Untersuchung der künstlichen Intelligenz mit einer vereinfachten Version des klassischen Programmes **Eliza**, das schon ausgiebig in der Einführung behandelt wurde, beginnen. Wie wir dort schon sagten, sucht das Programm Ihre Eingaben nach Wörtern ab, die es verwenden kann. Diese Wörter werden dann so an Sie zurückgegeben, daß es scheint, als würde das Programm wirklich mit Ihnen sprechen. Wenn Sie einmal eine Zeit lang damit gespielt haben, werden Sie herausfinden, wie Sie die eindrucksvollsten Antworten erzielen können. (Anmerkung des Übersetzers: Es war mir unmöglich, ein Programm zu schreiben, das der Syntax der deutschen Sprache adäquat war. Deswegen spricht Eliza hier englisch. Sollte dies einem Leser gelingen, bitte Nachricht an den Verlag!). Hier ist ein Teil einer 'Konversation', die ich mit meiner Version von Eliza hatte:

WELCOME TO ANOTHER SESSION WITH
YOUR COMPUTER PSYCHATRIST, ELIZA

WHAT IS YOU FIRST NAME ?MARTIN

WELL, HI THERE MARTIN

IT SURE IS NEAT TO HAVE YOU DROP BY

PRESS MY SPACE BAR WHEN YOU'RE READY TO START

?I HAVE A SERIOUS PROBLEM
THAT IS QUITE INTERESTING!

?I'VE COME TO YOU FOR HELP
I SEE...

?I AM AFRAID OF MICE
HOW LONG HAVE YOU BEEN AFRAID OF MICE

?SINCE BEFORE I TOOK UP COMPUTERS
MARTIN, WHY DO YOU MENTION COMPUTERS?

?IT SEEMED RELEVANT AT THE TIME
WHY ARE YOU SAYING THAT?

?BECAUSE I AM AFRAID OF LITTLE SQUEAKY THINGS
HOW LONG HAVE YOU BEEN AFRAID OF LITTLE SQUEAKY THINGS

?I THINKS WE HAVE BEEN THROUGH THIS BEFORE
DO YOU DOUBT WE HAVE BEEN THROUGH THIS BEFORE

?I SOMETIMES DOUBT EVERYTHING
THAT'S SO...PLEASE CONTINUE...

?MANY PEOPLE HAVE DOUBTS
I UNDERSTAND...

?I BET EVEN YOU HAVE DOUBTS
WHAT DOES THAT SUGGEST TO YOU?

?DO YOU EVER DREAM OF ROBOTS
MARTIN, YES, I EVER DREAM OF ROBOTS

?WELL I NEVER DREAM ABOUT THEM
WHY DID YOU BRING UP THE SUBJECT OF DREAMS?

?I HAVE THIS THING ABOUT MICE
MARTIN, WHAT DOES THAT SUGGEST TO YOU?

?I'M AFRAID THEY'LL COME AND JUST STAY FOR TEA
WHY ARE YOU TELLING ME YOU'RE AFRAID THEY'LL COME AND JUST STAY FOR TEA

?I THOUGHT YOU WERE MEANT TO ASSIST IN AN HOUR OF NEED MARTIN, WHY ARE YOU SAYING THAT?

?HOW CAN I SAY IT MORE CLEARLY
PERHAPS YOU DON'T WANT TO BE ABLE TO SAY IT MORE CLEARLY

?I THINK YOU ARE A FRAUD
WHAT MAKES YOU THINK I AM A FRAUD

?BECAUSE YOU NEVER GIVE ME A STRAIGHT ANSWER
THAT'S SO...PLEASE CONTINUE...

?COME ON TURKEY I WANT A STRAIGHT ANSWER
WHY DO YOU WANT A STRAIGHT ANSWER

?TO HELP ME IN AN HOUR OF NEED
WHAT DOES THAT SUGGEST TO YOU?

?YOUTHAT YOU 'RE PRETTY DUMB FOR A MACHINE
WHAT IS YOUR REACTION TO ME BEING PRETTY DUMB FOR A MACHINE

?IT MAKES ME WANT TO DRINK HEAVILY
MODERATION IN ALL THINGS SHOULD BE THE RULE.

?EVEN IN SEX
I FEEL YOU SHOULD REALLY DISCUSS THAT WITH
A HUMAN.

?BYE
BYE BYE FOR NOW.
SEE YOU AGAIN SOME TIME.

Und hier das Listing, so daß der Computer Ihnen helfen kann, mit sich
selber ins Reine zu kommen.

```

10 REM ELIZA
20 REM ALLE EINGABEN IN GROSSBUCHSTABEN
30 REM KEINE SATZZEICHEN AUSSER APOST
ROPH
40 REM WIE IN DON'T YOU'RE I'M I
'VE YOU'VE
50 GOSUB660:REM INITIALISIEREN
60 PRINT"WELCOME TO ANOTHER SESSION WIT
H"
70 PRINT"YOUR COMPUTER PSYCHATRIST, ELI
ZA"
80 PRINT:INPUT"WHAT IS YOU FIRST NAME "
;Z$
90 CLS:PRINT"WELL, HI THERE ";Z$
100 PRINT:PRINT"IT SURE IS NEAT TO HAVE
YOU DROP BY"
110 PRINT:PRINT"PRESS MY SPACE BAR WHEN
YOU'RE READY TO START"
120 N=N+1
130 IF INKEY$(10)=" " THEN GOTO120
140 CLS:REM RANDOMIZE N
150 GOSUB590:REM EINGABE DES BENUTZERS
160 IF RND(1)<.2 THEN PRINT Z$;" ";
170 GOSUB190:REM EINGABE VERARBEITEN, AN
TWORT AUSGEBEN
180 PRINT:GOTO150
190 REM SCHLUESSELWOERTER ODER -PHRASE F
INDEN

```

```

200 Z=0
210 Z=Z+1
220 D=INSTR(A$,B$(Z))
230 IF D<>0 THEN GOTO260
240 IF Z<35 THEN GOTO210
250 GOSUB440:RETURN
260 REM SCHLUESSELWOERTER VERARBEITEN
270 PRINT C$(Z);" ";
280 IF RIGHT$(C$(Z),1)<"A" THEN PRINT:RE
TURN
290 H=LEN(A$)-D-LEN(B$(Z))
300 IF H>0 THEN A$=RIGHT$(A$,H)
310 Z=0
320 Z=Z+1
330 D=INSTR(A$,F$(Z))
340 IF D<>0 THEN GOTO370
350 IF Z<8 THEN GOTO320
360 PRINT A$:RETURN
370 J$=LEFT$(A$,D-1)+" "+G$(Z)
380 Z=LEN(A$)-LEN(LEFT$(A$,D-1))-LEN(G$(
Z))
390 PRINT J$
400 IF Z>2 THEN L$=RIGHT$(A$,Z-2):IF INS
TR(K$,L$)=0 THEN RETURN
410 IF Z>2 THEN PRINT RIGHT$(A$,Z-2)
420 IF Z<2 THEN PRINT
430 RETURN
440 REM ZUFAELLIGE ANTWORT, KEIN SCHLUES
SELWORT
450 Z=INT(RND(1)*11)+1
460 ON Z GOSUB 480,490,500,510,520,530,5
40,550,560,570,580
470 RETURN
480 PRINT"WHAT DOES THAT SUGGEST TO YOU?
":RETURN
490 PRINT"I SEE...":RETURN
500 PRINT"I'M NOT SURE I UNDERSTAND YOU
FULLY.":RETURN
510 PRINT"CAN YOU ELABORATE ON THAT?":RE
TURN

```

```

TURN
520 PRINT"THAT IS QUITE INTERESTING!":RE
TURN
530 PRINT"THAT'S SO...PLEASE CONTINUE...
":RETURN
540 PRINT"I UNDERSTAND...":RETURN
550 PRINT"WELL, WELL..DO GO ON.":RETURN
560 PRINT"WHY ARE YOU SAYING THAT?":RETU
RN
570 PRINT"PLEASE EXPLAIN THE BACKGROUND
TO THIS REMARK...":RETURN
580 PRINT"COULD YOU SAY THAT AGAIN, IN A
DIFFERENT WAY?":RETURN
590 REM EINGABE DES BENUTZERS
600 INPUT A$:K$=A$
610 IF LEFT$(A$,3)="BYE" THEN PRINT"BYE
BYE FOR NOW.":PRINT"SEE YOU AGAIN SOME TIM
E.":END
620 K=LEN(A$)
630 IF LEFT$(A$,7)="BECAUSE" THEN A$=RIG
HT$(A$,K-7):K=K-7
640 A$=" "+A$+" "
650 RETURN
660 REM INITIALISIEREN
670 CLS:N=0
680 DIM B$(35),C$(35),F$(8),G$(8)
690 REM B$, C$ MIT SCHLUESSELWOERTERN, A
NTWORTEN FUELLEN
700 FOR A=1 TO 35
710 READ B$(A),C$(A)
720 NEXT A
730 RESTORE 1150
740 FOR A=1 TO 8
750 READ F$(A),G$(A)
760 NEXT A
770 RETURN
780 REM SCHLUESSELWOERTER, ANTWORTEN
790 DATA " CAN YOU","PERHAPS YOU WOULD L
IKE TO BE ABLE TO"

```

```

800 DATA " DO YOU","YES, I"
810 DATA " CAN I","PERHAPS YOU DON'T WAN
T TO BE ABLE TO"
820 DATA " YOU ARE","WHAT MAKES YOU THIN
K I AM"
830 DATA " YOU'RE","WHAT IS YOUR REACTIO
N TO ME BEING"
840 DATA " I DON'T","WHY DON'T YOU"
850 DATA " I FEEL","TELL ME MORE ABOUT F
EELING"
860 DATA " WHY DON'T YOU","WHY WOULD YOU
WANT ME TO"
870 DATA " WHY CAN'T I","WHAT MAKES YOU
THINK YOU SHOULD BE ABLE TO"
880 DATA " ARE YOU","WHY ARE YOU INTERES
TED IN WHETHER OR NOT I AM"
890 DATA " I CAN'T","HOW DO YOU KNOW YOU
CAN'T"
900 DATA " SEX","I FEEL YOU SHOULD REALL
Y DISCUSS THAT WITH A HUMAN."
910 DATA " I AM","HOW LONG HAVE YOU BEEN
"
920 DATA " I'M","WHY ARE YOU TELLING ME
YOU'RE"
930 DATA " I WANT","WHY DO YOU WANT"
940 DATA " WHAT","WHAT DO YOU THINK?"
950 DATA " HOW","WHAT ANSWER WOULD PLEAS
E YOU THE MOST?"
960 DATA " WHO","HOW OFTEN DO YOU THINK
OF SUCH QUESTIONS?"
970 DATA " WHERE","WHY DID YOU ASK ME TH
AT?"
980 DATA " WHEN","WHAT WOULD YOUR BEST F
RIEND SAY TO THAT QUESTION?"
990 DATA " WHY","WHAT IS IT THAT YOU REA
LLY WANT TO KNOW?"
1000 DATA " PERHAPS","YOU'RE NOT VERY FIR
M ON THAT!"
1010 DATA " DRINK","MODERATION IN ALL THI

```


NGS SHOULD BE THE RULE."

1020 DATA " SORRY", "WHY ARE YOU APOLOGIZING?"

1030 DATA " DREAM", "WHY DID YOU BRING UP THE SUBJECT OF DREAMS?"

1040 DATA " I LIKE", "IS IT GOOD THAT YOU LIKE"

1050 DATA " MAYBE", "AREN'T YOU BEING A BIT TENTATIVE?"

1060 DATA " NO", "WHY ARE YOU BEING NEGATIVE?"

1070 DATA " YOUR", "WHY ARE YOU CONCERNED ABOUT MY"

1080 DATA " ALWAYS", "CAN YOU THINK OF A SPECIFIC EXAMPLE?"

1090 DATA " THINK", "DO YOU DOUBT"

1100 DATA " YES", "YOU SEEM QUITE CERTAIN. WHY IS THIS SO?"

1110 DATA " FRIEND", "WHY DO YOU BRING UP THE SUBJECT OF FRIENDS?"

1120 DATA " COMPUTER", "WHY DO YOU MENTION COMPUTERS?"

1130 DATA " AM I", "YOU ARE"

1140 REM KONJUGIEREN

1150 DATA " I ", "YOU"

1160 DATA " ARE ", "AM"

1170 DATA " WERE ", "WAS"

1180 DATA " YOU ", "ME"

1190 DATA " YOUR ", "MY"

1200 DATA " I'VE ", "YOU'VE"

1210 DATA " I'M ", "YOU'RE"

1220 DATA " ME ", "YOU"

1230 DATA " AM I ", "YOU ARE"

1240 DATA " AM ", "ARE"

Elektronengehirn

Elektronengehirn ist der Versuch, ein Programm zu schreiben, das die Antwort zu einem Problem auf eine einleuchtend intelligente Weise findet. Der Computer löst ein Problem in der Form eines von Ihnen erdachten numerischen Codes, den er erraten muß. Die Rückmeldungen, die Sie ihm geben, sind in der Form von „weiß“ und „schwarz“. Sie geben ein „weiß“ für eine richtige Ziffer an einer falschen Stelle, wohingegen ein „schwarz“ eine richtige Zahl im Code kennzeichnet. Doch ist die Problemlösung nicht so einfach, da der Computer ja nicht weiß, welche Ziffer welche Rückmeldung bewirkt. Ich habe zwei Programmversionen geschrieben. Eine löst einen dreistelligen und die andere einen vierstelligen Code. Dem Programm liegt eine einfache Arbeitsweise zugrunde. Allerdings war es nicht gerade einfach, dieser simplen Idee dann auch im Programm nachzukommen. Jedesmal, wenn eine Zahl in einem Code, der mit „schwarz“ bewertet wird, aufscheint, wird jede Zahl dieses Codes so gewichtet, daß sie in zukünftigen Vermutungen öfter vorkommt. Je mehr „schwarz“ ein Code bekommt, desto gewichtiger wird dieser Code beurteilt. Eine weitaus geringere Bewertung erhalten die Codes, die nur ein oder mehrere „weiß“ bekommen. Jeder Code, der weder ein „weiß“ noch ein „schwarz“ erhält, hat zur Folge, daß keine der in ihm enthaltenen Ziffern in zukünftigen Vermutungen wieder verwendet wird.

Wie ich schon sagte, stellt die dreistellige Programmversion die Basis dar, auf der das vierstellige Programm geschrieben wurde. Geben Sie zunächst das dreiziffrige ein und speichern Sie es. Machen Sie erst danach die nötigen Änderungen, um es als vierstellig gebrauchen zu können. Nachdem die dreistellige Programmversion recht gut und verhältnismäßig schnell arbeitet, gerät die vierstellige gegen Ende des Programmes beinahe zum Stillstand. Das deshalb, weil es versucht, Codes zu erstellen, die (a) die Gewichtung der früheren Antworten reflektieren und (b) nicht schon vorher in diesem Programmablauf verwendet wurden.

Hier sehen Sie, wie das Programm einen dreistelligen Code knackt:

Versuch Nr.1	Versuch Nr.2
Mein Versuch ist 123	Mein Versuch ist 456
Wieviele Schwarze ?1	Wieviele Schwarze ?0
Und wieviele Weiße ?1	Und wieviele Weiße ?1

Versuch Nr.3	Versuch Nr.4
Mein Versuch ist 789	Mein Versuch ist 421
Wieviele Schwarze ?0	Wieviele Schwarze ?1
Und wieviele Weisse ?0	Und wieviele Weisse ?1
 Versuch Nr.5	 Versuch Nr.6
Mein Versuch ist 524	Mein Versuch ist 521
Wieviele Schwarze ?2	Wieviele Schwarze ?1
Und wieviele Weisse ?0	Und wieviele Weisse ?0
 Versuch Nr.7	 Versuch Nr.8
Mein Versuch ist 523	Mein Versuch ist 124
Wieviele Schwarze ?1	Wieviele Schwarze ?2
Und wieviele Weisse ?1	Und wieviele Weisse ?0
 Versuch Nr.9	
Mein Versuch ist 324	
Wieviele Schwarze ?3	
 Ich habe Ihren Code 324 in nur 9 Versuchen erraten	

Und das Listing, das obiges produzierte:

```

10 REM ELEKTRONENGEHIRN - 3 Ziffern
20 GOSUB490:REM INITIALISIEREN
30 REM VERSUCH ABGEBEN
40 IF VERSUCH<3 THEN FOR Z=1 TO 3:B(Z)=
Z+3*VERSUCH:NEXT Z:GOTO60
50 GOSUB260
60 PRINT:PRINT
70 VERSUCH=VERSUCH+1
80 PRINT:PRINT
90 PRINT"Versuch Nr.";VERSUCH
100 PRINT
110 PRINT"Mein Versuch ist ";B(1);B(2);B
(3)
120 PRINT:PRINT
130 INPUT"Wieviele Schwarze ";S
140 IF S=3 THEN GOTO570
150 PRINT:PRINT
160 INPUT"Und wieviele Weisse ";W
170 IF W+S=3 THEN Q=3:C(1)=B(1):C(2)=B(2
):C(3)=B(3)
180 IF S+W=0 THEN C(B(1))=0:C(B(2))=0:C(
B(3))=0:GOTO30
190 IF S>HILFE THEN FOR Z=1 TO 3:E(Z)=B(
Z):NEXT Z:HILFE=S
200 FOR Z=1 TO 9
210 FOR D=1 TO 3
220 IF B(D)=C(Z) THEN C(Z)=C(Z)+(S+W)*10
O+W*10
230 NEXT D
240 NEXT Z
250 GOTO30
260 REM Drei Ziffern auswaehlen
270 FOR Z=1 TO 3
280 D1=C(INT(RND(1)*Q)+1)
290 IF D1=0 THEN GOTO280
300 D2=C(INT(RND(1)*Q)+1)
310 IF D2=0 THEN GOTO300
320 IF INT(D1/10)>INT(D2/10) THEN B(Z)=D

```

```

1 330 IF INT(D1/10)<INT(D2/10) THEN B(Z)=D
2 340 IF INT(D1/10)=INT(D2/10) THEN B(Z)=D
1 350 IF B(Z)>100 THEN B(Z)=B(Z)-100*INT(B
(Z)/100):GOTO350
360 IF B(Z)>10 THEN B(Z)=B(Z)-10*INT(B(Z
)/10):GOTO360
370 NEXT Z
380 IF B(1)=B(2) OR B(1)=B(3) OR B(2)=B(
3) THEN GOTO270
390 IF HILFE>0 THEN ZAEHLER=0:FOR Z=1 TO
3:IF B(Z)=E(Z) THEN ZAEHLER=ZAEHLER+1
400 IF HILFE>0 THEN NEXT Z:IF ZAEHLER<HI
LFE THEN GOTO270
410 M=100*B(1)+10*B(2)+B(3)
420 K(VERSUCH)=M
430 IF VERSUCH<3 THEN GOTO480
440 ZAEHLER=1
450 ZAEHLER=ZAEHLER+1
460 IF K(ZAEHLER)=M THEN GOTO270
470 IF ZAEHLER<VERSUCH-1 THEN GOTO450
480 RETURN
490 REM INITIALISIEREN
500 VERSUCH=0:Q=9:HILFE=0
510 REM RANDOMIZE
520 DIM B(3),C(9),E(3),K(100)
530 FOR Z=1 TO 9
540 C(Z)=Z
550 NEXT Z
560 RETURN
570 PRINT:PRINT
580 PRINT"Ich habe Ihren Code ";B(1);B(2
);B(3)
590 PRINT"in nur ";VERSUCH;" Versuchen e
rraten"
600 END

```

Nach einigen Veränderungen versucht der Computer jetzt, einen vierstelligen Code zu knacken:

Versuch Nr.1	Versuch Nr.2
Mein Versuch ist 1234	Mein Versuch ist 5678
Wieviele Schwarze ?1	Wieviele Schwarze ?0
Und wieviele Weisse ?1	Und wieviele Weisse ?1
Versuch Nr.3	Versuch Nr.4
Mein Versuch ist 2147	Mein Versuch ist 7812
Wieviele Schwarze ?0	Wieviele Schwarze ?0
Und wieviele Weisse ?1	Und wieviele Weisse ?0
Versuch Nr.5	Versuch Nr.6
Mein Versuch ist 4536	Mein Versuch ist 5369
Wieviele Schwarze ?0	Wieviele Schwarze ?1
Und wieviele Weisse ?3	Und wieviele Weisse ?2
Versuch Nr.7	Versuch Nr.8
Mein Versuch ist 9463	Mein Versuch ist 6934
Wieviele Schwarze ?0	Wieviele Schwarze ?2
Und wieviele Weisse ?4	Und wieviele Weisse ?2

```

Versuch Nr.9                Versuch Nr.10
Mein Versuch ist 9634       Mein Versuch ist 9436

Wieviele Schwarze ?1       Wieviele Schwarze ?0

Und wieviele Weisse ?3     Und wieviele Weisse ?4

Versuch Nr.11
Mein Versuch ist 6394

Wieviele Schwarze ?4

Ich habe Ihren Code 6394
in nur 11 Versuchen erraten

```

Hier ist das Listing, um einen vierstelligen Code zu lösen. Wie gesagt, basiert dieses Programm auf der dreistelligen Version. Achten Sie auf das zusätzliche „1“ in Zeile 420 dieses Programmes sowie auf alle anderen Änderungen:

```

10 REM ELEKTRONENGEHIRN - 4 Ziffern
20 GOSUB510:REM INITIALISIEREN
30 REM VERSUCH ABGEBEN
40 IF VERSUCH=0 THEN FOR Z=1 TO 4:B(Z)=
Z:NEXT Z:GOTO70
50 IF VERSUCH=1 THEN FOR Z=1 TO 4:B(Z)=
Z+4:NEXT Z:GOTO70
60 GOSUB280
70 CLS
80 VERSUCH=VERSUCH+1
90 PRINT:PRINT
100 PRINT"Versuch Nr. ";VERSUCH
110 PRINT
120 PRINT"Mein Versuch ist ";B(1);B(2);B
(3);B(4)

```

```

130 PRINT:PRINT
140 INPUT"Wieviele Schwarze ";S
150 IF S=4 THEN GOTO590
160 IF S=3 THEN GOTO190
170 PRINT:PRINT
180 INPUT"Und wieviele Weisse ";W
190 IF W+S=4 THEN Q=4:C(1)=B(1):C(2)=B(2
):C(3)=B(3):C(4)=B(4)
200 IF W+S=0 THEN C(B(1))=0:C(B(2))=0:C(
B(3))=0:C(B(4))=0:GOTO30
210 IF S>HILFE THEN FOR Z=1 TO 4:E(Z)=B(
Z):NEXT Z:HILFE=S
220 FOR Z=1 TO 9
230 FOR D=1 TO 4
240 IF B(D)=C(Z) THEN C(Z)=C(Z)+(S+W)*10
0+W*10
250 NEXT D
260 NEXT Z
270 GOTO30
280 REM vier Ziffern auswählen
290 FOR Z=1 TO 4
300 D1=C(INT(RND(1)*Q)+1)
310 IF D1=0 THEN GOTO300
320 D2=C(INT(RND(1)*Q)+1)
330 IF D2=0 THEN GOTO320
340 IF INT(D1/10)>INT(D2/10) THEN B(Z)=D
1
350 IF INT(D2/10)>INT(D1/10) THEN B(Z)=D
2
360 IF INT(D2/10)=INT(D1/10) THEN B(Z)=D
1
370 IF B(Z)>100 THEN B(Z)=B(Z)-100*INT(B
(Z)/100):GOTO370
380 IF B(Z)>10 THEN B(Z)=B(Z)-10*INT(B(Z
)/10):GOTO380
390 NEXT Z
400 IF B(1)=B(2) OR B(1)=B(3) OR B(1)=B(
4) OR B(2)=B(3) OR B(2)=B(4) OR B(3)=B(4)
THEN GOTO290

```

```

410 IF HILFE>0 THEN ZAEHLER=0:FOR Z=1 TO
4:IF B(Z)=E(Z) THEN ZAEHLER=ZAEHLER+1
420 IF HILFE>0 THEN NEXT Z:IF ZAEHLER<HI
LFE-1 THEN GOTO290
430 M=1000*B(1)+100*B(2)+10*B(3)+B(4)
440 K(VERSUCH)=M
450 IF VERSUCH<3 THEN GOTO500
460 ZAEHLER=1
470 ZAEHLER=ZAEHLER+1
480 IF K(ZAEHLER)=M THEN GOTO290
490 IF ZAEHLER<VERSUCH-1 THEN GOTO470
500 RETURN
510 REM INITIALISIEREN
520 VERSUCH=0:Q=9:HILFE=0
530 REM RANDOMIZE
540 DIM B(4),C(9),E(4),K(100)
550 FOR Z=1 TO 9
560 C(Z)=Z
570 NEXT Z
580 RETURN
590 PRINT:PRINT
600 PRINT"Ich habe Ihren Code ";B(1);B(2
);B(3);B(4)
610 PRINT"in nur ";VERSUCH;" Versuchen e
rraten"
620 END

```

Memory

Dies ist ein Programm, in dem der Computer zeigt, was für ein ausgezeichnetes Gedächtnis er hat. Er fordert Sie in einer Runde des Kartenspiels „Konzentration“.

Ein Kartenpäckchen wird gemischt und dann auf dem Bildschirm ausgelegt. Sie und der Computer wählen abwechselnd zwei Karten aus. Diese werden umgedreht, und wenn sie ein Paar ergeben, erhält der sie auswählende Spieler einen Punkt. Dieses setzt sich so lange fort, bis alle Karten umgedreht wurden. Natürlich gewinnt derjenige, der die meisten Paare findet.

Am Anfang werden Sie wahrscheinlich das Programm einfacher machen wollen. Dazu ändern Sie das dritte Element jedes Variablenstranges, das die richtig gewählten Karten darstellt, und dieser Mechanismus wird die erratenen Karten vom Bildschirm löschen. Im folgenden das Listing:

```

10 REM Memory
20 GOSUB790:REM Initialisieren/Mischen
30 GOSUB610:REM Ausdruck
40 GOSUB430:REM Menschlicher Versuch
50 GOSUB150:REM Computerversuch
60 IF MP+CP<26 THEN GOTO30
70 PRINT"Das bringt uns zum Ende"
80 PRINT"einer Runde Memory!"
90 PRINT
100 IF MP>CP THEN PRINT"Sie sind der Sie
ger!"
110 IF CP>MP THEN PRINT"Ich bin der Sieg
er, Mensch!"
120 IF CP=MP THEN PRINT"Unentschieden, M
ensch!"
130 END
140 REM *****
150 REM Computerversuch
160 ZUG=0
170 IF R=1 OR RND(1)<GRAD THEN GOTO260
180 X=R
190 X=X-1
200 Y=1
210 Y=Y+1
220 IF D(X)=D(Y) THEN GOTO240
230 IF ASC(A$(D(X)))=ASC(A$(D(Y))) THEN
IF X<>Y THEN IF MID$(A$(D(X)),3,1)="X" AND
MID$(A$(D(Y)),3,1)="X" THEN A=D(X):B=D(Y)
:GOTO340
240 IF Y<X THEN GOTO210
250 IF X-1>R THEN GOTO190
260 REM Zufälliger Versuch
270 ZAEHLER=0
280 ZAEHLER=ZAEHLER+1

```

```

290 IF ZAEHLER=200 THEN GOTO70
300 A=INT(RND(1)*52)+1
310 B=INT(RND(1)*52)+1
320 IF B=A THEN GOTO310
330 IF RIGHT$(A$(A),1)<>"X" OR RIGHT$(A$(B),1)<>"X" THEN GOTO280
340 PRINT"Ich habe die Karten ";A;" und ";B;" gewaehlt"
350 FOR P=1 TO 2500:NEXT P
360 A$(A)=LEFT$(A$(A),2)+" ":A$(B)=LEFT$(A$(B),2)+" "
370 GOSUB610
380 IF LEFT$(A$(A),1)=LEFT$(A$(B),1) THEN PRINT"Und das ist ein Paar!":CP=CP+1
390 IF LEFT$(A$(A),1)<>LEFT$(A$(B),1) THEN PRINT"Aber das ist kein Paar!":A$(A)=LEFT$(A$(A),2)+"X":A$(B)=LEFT$(A$(B),2)+"X":D$(R)=A$(A):D$(R+1)=A$(B):D(R)=A:D(R+1)=B:R=R+2
400 FOR P=1 TO 2000:NEXT P
410 RETURN
420 REM *****
430 REM Menschenversuch
440 ZUG=1
450 PRINT:INPUT"Geben Sie Karte Nr.1 ein ";A
460 IF A<1 OR A>52 THEN GOTO450
470 IF RIGHT$(A$(A),1)=" " THEN PRINT"Diese Karte ist schon vergeben":GOTO450
480 A$(A)=LEFT$(A$(A),2)+" "
490 PRINT:INPUT"Geben Sie Karte Nr.2 ein ";B
500 IF B<1 OR B>52 OR A=B THEN GOTO490
510 IF RIGHT$(A$(B),1)=" " THEN PRINT"Diese Karte ist schon vergeben":GOTO490
520 A$(B)=LEFT$(A$(B),2)+" "
530 GOSUB610
540 IF LEFT$(A$(A),1)=LEFT$(A$(B),1) THEN PRINT"Jawohl, es ist ein Paar":MP=MP+1

```

```

550 IF LEFT$(A$(A),1)<>LEFT$(A$(B),1) THEN PRINT"Nein, es ist kein Paar":A$(A)=LEFT$(A$(A),2)+"X":A$(B)=LEFT$(A$(B),2)+"X"
560 IF LEFT$(A$(A),1)<>LEFT$(A$(B),1) THEN D$(R)=A$(A):D$(R+1)=A$(B):D(R)=A:D(R+1)=B:R=R+2
570 FOR P=1 TO 2000:NEXT P
580 GOSUB610
590 RETURN
600 REM *****
610 REM Ausdruck
620 CLS:PRINT"Meine Punkte sind ";CP;" und Ihre ";MP
630 PRINT:PRINT"Wir spielen auf Stufe ";GRAD*10
640 PRINT" ";
650 FOR Z=1 TO 52
660 IF Z=6 THEN PRINT" ";
670 PRINT Z;" ";
680 B$=RIGHT$(A$(Z),1)
690 IF B$="X" THEN PRINT"-- ";
700 IF B$=" " THEN PRINT LEFT$(A$(Z),2);
710 IF 5*(INT(Z/5))=Z THEN PRINT
720 NEXT Z
730 PRINT:PRINT
750 IF ZUG=1 THEN PRINT"Ihre Karten: ";A;" und ";B
760 IF ZUG=0 THEN PRINT"Meine Karten: ";A;" und ";B
770 RETURN
780 REM *****
790 REM Initialisieren/Mischen
800 REM RANDOMIZE
810 DIM A$(52),D$(500),D(500)
820 CLS:PRINT"Bitte warten Sie, Kartenfa
n!"
830 PRINT:PRINT"Beginnen Sie, Ihre Gedac
htnismuskeln":PRINT"anzuspannen..."

```

```

840 CP=0:MP=0:R=1:ZUG=3
850 FOR B=1 TO 52
860 D$(B)="."
870 READ A$(B)
880 NEXT B
890 REM Moses/Oakford Mischroutine
900 FOR J=52 TO 2 STEP -1
910 Z=INT(RND(1)*J)+1
920 H$=A$(Z)
930 A$(Z)=A$(J)
940 A$(J)=H$
950 NEXT J
960 PRINT:PRINT"Welcher Schwierigkeitsgr
ad (1-10)?"
970 PRINT"(10 ist einfach, 1 ziemlich UN
MOEGlich!)"
980 INPUT GRAD:IF GRAD<1 OR GRAD>10 THEN
GOTO980
990 GRAD=GRAD/10
1000 RETURN
1010 DATA "AHX","2HX","3HX","4HX","5HX","
6HX","7HX","8HX","9HX","ZHX","BHX","DHX","
KHX"
1020 DATA "ATX","2TX","3TX","4TX","5TX","
6TX","7TX","8TX","9TX","ZTX","BTX","DTX","
KTX"
1030 DATA "AKX","2KX","3KX","4KX","5KX","
6KX","7KX","8KX","9KX","ZKX","BKX","DKX","
KKX"
1040 DATA "APX","2PX","3PX","4PX","5PX","
6PX","7PX","8PX","9PX","ZPX","BPX","DPX","
KPX"

```

Dichter

Das nächste ist unser Gedichtschreibprogramm. Es verlangt keine Interaktionen von Ihnen, außer vielleicht das Bewundern der wunderschönen (!) Verse des Programmes. Hier ist eine Demonstration der Verse, die es produziert:

```

THE SENTINEL ENTREATED HELPLESSLY
  IN THE BEAR GARDEN...
SCREAMING FOR A MALEVOLENT SPIRIT,
  TO DEMUR THE DRYAD...
WASTING, THEN GNAWING
  ...RESTRAINING, GNAWING.

```

```

THE CASTELLAN CONJURED NEGLECTFULLY
  IN THE BEAR GARDEN...
HOPING FOR FORTUNE'S WHEEL,
  TO QUENCH THE HOBGOBLIN...
WASTING, THEN CORRODING
  ...IMPOUNDING, BREAKING.

```

```

THE CONVOY BESOUGHT PRIDEFULLY
  IN THE FIELD...
WAITING FOR A ZEALOT,
  TO FORSWEAR THE PIXY...
USING, THEN UNDERMINING
  ...GAGING, WOUNDING.

```

```

THE MENDICANT URGED SLAVISHLY
  IN THE RACECOURSE...
WANTING FOR FORTUNE'S WHEEL,
  TO BACK THE SATUR...
EXHAUSTING, THEN SHAKING
  ...IMPOUNDING, DAMAGING.

```

(Anmerkung des Übersetzers: Da diese Verse ausschließlich von der Neugestaltung des Programmes abhängen und eine wörtliche Übersetzung dieser gekünstelten Worte außerhalb meiner Fähigkeiten liegt, daher auch nicht zielführend wäre, unterlasse ich dieses!)
Folgend das Listing Dichter. Wenn Sie das Programm schon ein paar mal gesehen haben, werden Sie wahrscheinlich die Wörter in den DATA-Anweisungen ändern wollen. Damit verändert sich natürlich auch die Art der Gedichte, die es hervorbringt.

```

10 REM DICHTER
20 CLS
30 DIM A$(12)
40 N=0
50 PRINT "TASTE FUER START"
60 N=N+1: IF INKEY$(10)="" THEN GOTO 60
70 CLS: REM RANDOMIZE N
80 REM *****
90 RESTORE 330: GOSUB 450: A$(1)=B$
100 RESTORE 340: GOSUB 450: A$(2)=B$
110 RESTORE 350: GOSUB 450: A$(3)=B$
120 RESTORE 360: GOSUB 450: A$(4)=B$
130 RESTORE 370: GOSUB 450: A$(5)=B$
140 RESTORE 380: GOSUB 450: A$(6)=B$
150 RESTORE 390: GOSUB 450: A$(7)=B$
160 RESTORE 400: GOSUB 450: A$(8)=B$
170 RESTORE 410: GOSUB 450: A$(9)=B$
180 RESTORE 420: GOSUB 450: A$(10)=B$
190 RESTORE 430: GOSUB 450: A$(11)=B$
200 RESTORE 420: GOSUB 450: A$(12)=B$
210 REM *****
220 PRINT "THE "; A$(1); " "; A$(2); " "; A$(3)
)
230 PRINT "    IN THE "; A$(4); "... "
240 PRINT A$(5); " FOR "; A$(6); ", "
250 PRINT "    TO "; A$(7); " THE "; A$(8); ".
..
260 PRINT " "; A$(9); ", THEN "; A$(10)
270 PRINT "    ..."; A$(11); ", "; A$(12); ".
"
280 PRINT: PRINT
290 REM *****

```

```

300 FOR T=1 TO 3500: NEXT T
310 GOTO 90
320 REM *****
330 DATA "APPLICANT", "SUPPLICANT", "MEND
ICANT", "BEGGAR", "ASPIRANT", "CLAIMANT", "POS
TULANT", "CONVOY", "SENTINEL", "CASTELLAN"
340 DATA "ENTREATED", "DESEECED", "WAITED
", "BESOUGHT", "CONJURED", "PRESSED", "URGED",
"CRAVED", "DEMANDED", "CLAIMED"
350 DATA "SUCCESSFULLY", "COURAGEOUSLY", "
PROGRESSIVELY", "FOOLISHLY", "ENDLESSLY", "HE
LPLESSLY", "HALTINGLY", "SLAVISHLY", "NEGLECT
FULLY", "PRIDEFULLY"
360 DATA "FIELD", "BATTLEFIELD", "PLAYGROU
ND", "CIRCUS RING", "BEAR GARDEN", "ENEMY'S C
AMP", "HIPPODROME", "THEATRE OF WAR", "COURT"
, "RACECOURSE"
370 DATA "WAITING", "ASKING", "HOPING", "WA
TCHING", "PRAYING", "STAYING", "SCREAMING", "W
ANTING", "STARING", "CRYING"
380 DATA "THE FATES", "FORTUNE'S WHEEL", "
HOBSON'S CHOICE", "THE STARS", "AN ASTRAL IN
FLUENCE", "A BLIND BARGAIN", "A SPELLBINDER"
, "A MALEVOLENT SPIRIT", "A ZEALOT", "A DOGMA
TIST"
390 DATA "GUENCH", "RECAANT", "FORSEWAR", "A
BJURE", "RENOUNCE", "RELINQUISH", "WAVER", "DE
MUR", "SWERVE", "BACK"
400 DATA "GHOST", "SPECTRE", "HOBGOBLIN", "
DJINN", "LEPRECHAUN", "PIXY", "FAUN", "SATUR",
"DRYAD", "NAIAD"
410 DATA "WASTING", "SPENDING", "USING", "C
ONSUMING", "RUNNING OUT", "DRYING UP", "EMPTY
ING", "EXHAUSTING", "SQUANDERING", "DISPENSIN
G WITH"
420 DATA "CORRODING", "ERODING", "BLIGHTIN
G", "ROTTING", "GNAWING", "UNDERMINING", "SHAK
ING", "BREAKING", "DAMAGING", "WOUNDING"
430 DATA "RESTRAINING", "COERCING", "CHECK

```


ING", "DETAINING", "ENCLOSING", "IMPOUNDING",
"HANDCUFFING", "MUZZLING", "GAGING", "SUPPRES
SING"

```
440 REM *****  
450 FOR N=1 TO RND(1)*10+1  
460 READ B$  
470 NEXT N  
480 RETURN
```

Nur zum Spaß

Die Programme in diesem Kapitel sind so gestaltet, daß sie in erster Linie Spaß machen sollen. Durch **Las Vegas**, **Caddy**, **Roboterminenfeld** und andere mehr ist für alle gute Unterhaltung gesichert.

Wenn Sie eine Gruppe Kinder haben, die Sie gerne am Computer spielend alleine lassen wollen, so haben wir ein massives Programm, **Aufmarsch von Puzzles**, das eine Zusammenstellung von unterhaltsamen Spielen umfaßt. Sie können erst dann ein Spiel beginnen, wenn Sie das vorhergehende erfolgreich beendet haben. Dies ist eine lustige Art, die Jungen oder Junggebliebenen zu unterhalten. Gehen wir's gleich an!

Dazwischen

Wir beginnen das Kapitel mit dem einfachsten Spiel dieser Reihe, **Dazwischen**. Wie Ihnen auch das Programm erklärt, erzeugt der Computer zwei Zahlen zwischen 1 und 13. Sie müssen dann auf die Möglichkeit setzen, daß die nächste vom Computer 'erdachte' Zahl zwischen den ersten beiden liegt. Es ist sehr einfach zu spielen und überaus lustig. Und so sieht eine Runde **Dazwischen** aus:

```
MEINE ERSTE ZAHL IST 10  
MEINE ZWEITE IST 5
```

```
SIE HABEN $20
```

```
WIEVIEL WETTEN SIE DARAUF,  
DASS MEINE NAECHSTE ZAHL  
ZWISCHEN 10 UND 5 LIEGT ?5
```

```
MEINE ZAHL WAR 8  
BRAVO, SIE GEWINNEN $10
```

MEINE ERSTE ZAHL IST 2
MEINE ZWEITE IST 6

SIE HABEN \$30

WIEVIEL WETTEN SIE DARAUF,
DASS MEINE NAECHSTE ZAHL
ZWISCHEN 2 UND 6 LIEGT ?10

MEINE ZAHL WAR 11
TUT MIR LEID, SIE VERLIEREN \$10

MEINE ERSTE ZAHL IST 12
MEINE ZWEITE IST 10

SIE HABEN \$20

WIEVIEL WETTEN SIE DARAUF,
DASS MEINE NAECHSTE ZAHL
ZWISCHEN 12 UND 10 LIEGT ?5

MEINE ZAHL WAR 8
TUT MIR LEID, SIE VERLIEREN \$5

MEINE ERSTE ZAHL IST 7
MEINE ZWEITE IST 11

SIE HABEN \$15

WIEVIEL WETTEN SIE DARAUF,
DASS MEINE NAECHSTE ZAHL
ZWISCHEN 7 UND 11 LIEGT ?15

MEINE ZAHL WAR 12
TUT MIR LEID, SIE VERLIEREN \$15

DAS SPIEL IST ZU ENDE

SIE SIND PLEITE!

DANKE FUER DAS SPIEL

Und hier das Listing von Dazwischen:

```

10 REM DAZWISCHEN
20 GOSUB330
30 GOSUB80
40 IF D<1 THEN GOTO420
50 GOSUB340
60 GOTO30
70 REM *****
80 PRINT:PRINT:PRINT
90 PRINT"MEINE ERSTE ZAHL IST ";A
100 PRINT"    MEINE ZWEITE IST ";B
110 PRINT
120 PRINT"SIE HABEN $";D
130 PRINT
140 PRINT"WIEVIEL WETTEN SIE DARAUF,"
150 PRINT"DASS MEINE NAECHSTE ZAHL"
160 PRINT"ZWISCHEN ";A;" UND ";B;" LIEGT
";
170 INPUT E
180 IF E>D THEN GOTO170
190 D=D-E
200 GOSUB500
210 PRINT:PRINT"MEINE ZAHL WAR ";C
220 GOSUB500
230 IF NOT(C>A AND C<B OR C<A AND C>B) T
HEN GOTO290
240 PRINT"BRAVO, SIE GEWINNEN $";2*D
250 D=D+3*D
260 GOSUB500

```

```

270 RETURN
280 REM *****
290 PRINT "TUT MIR LEID, SIE VERLIEREN $"
;E
300 GOSUB 500
310 RETURN
320 REM *****
330 D=20
340 CLS
350 A=INT(RND(1)*13)+1
360 B=INT(RND(1)*13)+1
370 IF ABS(A-B)<2 OR ABS(A-B)>6 THEN GOT
0360
380 C=INT(RND(1)*13)+1
390 IF A=C OR B=C THEN GOTO 380
400 RETURN
410 REM *****
420 PRINT
430 PRINT "DAS SPIEL IST ZU ENDE"
440 PRINT
450 PRINT "SIE SIND PLEITE!"
460 PRINT
470 PRINT "DANKE FUER DAS SPIEL"
480 END
490 REM *****
500 FOR Z=1 TO 1000
510 NEXT Z
520 RETURN

```

Nimgrab

Nimgrab erscheint auf den ersten Blick als sehr leicht, aber es ist richtig teuflisch zu spielen und schwer zu gewinnen. Sie wechseln sich mit dem bösen Computer ab und nehmen Objekte von den am Bildschirm abgebildeten weg. Es gibt eine Grenze für die Anzahl der Objekte, die Sie auf einmal wegnehmen dürfen. Diese Grenze ändert sich nicht in einer Runde, wohl aber von Spiel zu Spiel. Der Sieger ist derjenige, der den anderen dazu bringt, das letzte Objekt vom Bildschirm zu nehmen.

Wenn Sie es einmal in Aktion gesehen haben, werden Sie auch sehen, wie Sie es spielen müssen. Dann können Sie beginnen, eine Methode zu erarbeiten, den Computer zu schlagen. Hier folgt eine Runde Nimgrab:

Sie koennen maximal 4 nehmen

```

12
34
5678
9101112131415161718
1920212223

```

Wieviele wollen Sie nehmen?

Also nehmen Sie 4

Sie koennen maximal 4 nehmen

Sie nahmen 4, und ich 4

```

123
456789101112
13
1415

```

Wieviele wollen Sie nehmen?

Also nehmen Sie 4

Sie koennen maximal 4 nehmen

Sie nahmen 4, und ich 1

1

2

345678910

Wieviele wollen Sie nehmen?

Also nehmen Sie 2

Sie koennen maximal 4 nehmen

Sie nahmen 2, und ich 2

123456

Wieviele wollen Sie nehmen?

Also nehmen Sie 4

Ich habe das letzte genommen,
und so haben Sie gewonnen!!

Und hier das Listing des Programmes:

```

10 REM Nimgrab
20 CLS
30 PRINT "S' fuer Spielbeginn..."
40 N=0
50 N=N+1:IF INKEY$(10)="" THEN GOTO50
60 REM RANDOMIZE N
70 CLS
80 M=0:E=0:Z=INT(RND(1)*8)+16
90 IF 2*INT(Z/2)=Z THEN Z=Z+1
100 H=3+INT(RND(1)*2)
110 PRINT "Sie koennen maximal ";H;" nehmen"
120 GOSUB320
130 IF E>0 THEN PRINT "Sie nahmen ";E;"",
und ich ";Q
140 FOR K=1 TO Z
150 PRINT ;K;:IF RND(1)>.8 THEN PRINT
160 NEXT K
170 GOSUB320
180 PRINT "Wieviele wollen Sie nehmen?"
190 E=VAL(INKEY$(10))

```

```

200 IF E<1 OR E>H OR E>Z THEN GOTO190
210 PRINT:PRINT"Also nehmen Sie ";E
220 Z=Z-E
230 GOSUB320
240 IF Z<1 THEN PRINT"Sie haben das letzte genommen,":PRINT"also habe ich gewonnen!!":END
250 Q=Z-1-INT((Z-1)/(H+1))*(H+1)-INT(RND(1)*2)+INT(RND(1)*2)
260 IF Q<1 OR Q>H THEN GOTO250
270 GOSUB320
280 Z=Z-Q
290 IF Z<1 THEN PRINT"Ich habe das letzte genommen,":PRINT"und so haben Sie gewonnen!!":END
300 GOSUB320
310 GOTO110
320 PRINT
330 PRINT"*****"
***"
340 PRINT
350 FOR I=1 TO 200:NEXT I
360 RETURN

```

Caddy

Jetzt ist es an der Zeit, daß Sie und Ihr vielseitiger Computer den Mikrochip-Golfkurs mit unserem nächsten Programm, **Caddy**, in Angriff nehmen.

Der Kurs geht über neun Löcher, und wie Sie während des Ablaufes des Spieles feststellen werden, übernimmt es der Computer freundlichweise, Ihre Punktezah auf der Spielkarte zu vermerken. Nach jedem Loch teilt er Ihnen auch mit, wie es um Sie steht und errechnet auch die durchschnittliche Schlagzahl pro Loch. Alles, was Sie zu tun haben, ist, den Ball zu schlagen. Sollten Sie über Ihr Ziel hinauschießen, so vergewissert sich der Computer automatisch, daß der nächste Schlag zurück zum Loch geht. Sie werden das Spiel ziemlich verzwickelt finden, besonders bei Löchern mit hohen Schwierigkeitsfaktoren.

Hier ist Caddy in Aktion:

<<< Loch Nr. 1 >>>

SCHWIERIGKEITSGRAD DREI

```

      □
##### /#####
#####
#####

```

Schlagstaerke 750

```

      □
##### /#####
#####
#####

```

Nach diesem Schlag haben Sie 1 Punkte

<<< Loch Nr. 1 >>>

SCHWIERIGKEITSGRAD DREI

```

      □
##### /#####
#####
#####

```

Schlagstaerke 725

```

      □
##### /#####
#####
#####

```

Nach diesem Schlag haben Sie 2 Punkte

<<< Loch Nr. 1 >>>

SCHWIERIGKEITSGRAD DREI

 #####
 #####

Schlagstaerke 78

 #####
 #####

Nach diesem Schlag haben Sie 3 Punkte

<<< Loch Nr. 1 >>>

SCHWIERIGKEITSGRAD DREI

 #####
 #####

Schlagstaerke 75

 #####
 #####

Nach diesem Schlag haben Sie 4 Punkte

<<< Loch Nr. 1 >>>

SCHWIERIGKEITSGRAD DREI

 #####
 #####

Schlagstaerke 72

Sie haben es geschafft!!

 #####
 #####

Nach diesem Schlag haben Sie 5 Punkte

Punkte fuers erste Loch: 5

Punktestand bei diesem Loch: 27

<<< Loch Nr. 5 >>>

SCHWIERIGKEITSGRAD DREI

 #####
 #####

Schlagstaerke 72

Sie haben es geschafft!!

 #####
 #####

Nach diesem Schlag haben Sie 6 Punkte

Spielzusammenfassung:

Loch 1 brauchte 5 Schlaege

Loch 2 brauchte 6 Schlaege

Loch 3 brauchte 12 Schlaege

Loch 4 brauchte 4 Schlaege

Loch 5 brauchte 6 Schlaege

Der Durchschnitt ist momentan 6

Hier ist das Listing, Golfprofi:

```

10 REM Caddy
20 DIM X(9):CO=0
30 REM RANDOMIZE
40 FOR Z=1 TO 9
50 SC=0
60 J=INT(RND(1)*12)
70 Q=INT(RND(1)*3)+3
80 IF Q=5 THEN Q$="FUENF"
90 IF Q=4 THEN Q$="VIER"
100 IF Q=3 THEN Q$="DREI"
110 CLS:PRINT:PRINT
120 IF Z=2 THEN PRINT"Punktestand bei di
esem Loch: ";X(1)
130 IF Z>2 THEN PRINT"Punktestand bei di
esem Loch: ";K
140 PRINT"<<< Loch Nr. ";Z;" >>>"
150 PRINT:PRINT"SCHWIERIGKEITSGRAD ";Q$
160 GOSUB430
170 PRINT:INPUT"Schlagstaerke ";A
180 IF J>26 THEN A=-A
190 J=J+INT(A/(RND(1)*Q+1))
200 IF J=26 THEN GOSUB490
210 IF J<>26 THEN GOSUB430
220 SC=SC+1
230 PRINT:PRINT"Nach diesem Schlag haben
Sie ";SC;" Punkte"
240 FOR P=1 TO 500:NEXT P
250 IF J<>26 THEN GOTO110
260 C=C+SC
270 X(Z)=SC
280 IF Z=1 THEN GOTO390
290 K=0
300 PRINT"Spielzusammenfassung:"
310 FOR J=1 TO Z
320 K=K+X(J)
330 PRINT"Loch ";J;" brauchte ";X(J);" S
chlaege"
340 FOR M=1 TO 600:NEXT M

```

```

350 NEXT J
360 IF Z<9 THEN PRINT:PRINT"Der Durchsch
nitt ist momentan ";INT((K+.5)/Z)
370 FOR P=1 TO 2000:NEXT P
380 IF Z>1 THEN PRINT:PRINT"Punkte fuer
";Z;" Loecher: ";C
390 IF Z=1 THEN PRINT:PRINT"Punkte fuer's
erste Loch: ";C
400 FOR M=1 TO 2000:NEXT M
410 NEXT Z
420 GOTO560
430 IF J>30 THEN J=30
440 PRINT TAB(J+2);"o"
450 PRINT"##### /#
#####"
460 PRINT"##### ##
#####"
470 PRINT"#####
#####"
480 RETURN
490 PRINT:PRINT:PRINT"Sie haben es gesch
afft!!"
500 FOR P=1 TO 200:NEXT P
510 PRINT"##### /#
#####"
520 PRINT"##### o #
#####"
530 PRINT"#####
#####"
540 FOR P=1 TO 4000:NEXT P
550 RETURN
560 PRINT:PRINT"Ende dieser Runde, Golfe
r!"
570 PRINT:PRINT"Ihre Punkte: ";C
580 PRINT"und Ihr Lochdurchschnitt war "
;INT((C+.5)/9)
590 PRINT:PRINT
600 PRINT"'J' fuer eine weitere Runde,":
PRINT"'N' fuer Ende"

```

```

610 A$=INKEY$(10)
620 IF A$<>"J" AND A$<>"j" AND A$<>"N" A
ND A$<>"n" THEN GOTO610
630 IF A$="J" OR A$="j" THEN RUN
640 PRINT:PRINT"OK, danke fuer das Spiel
, Champion"

```

Aufmarsch von Puzzles

Es besteht keine Veranlassung, **Aufmarsch von Puzzles** näher zu erklären. Die Regeln sind alle im Programm enthalten, das eine Reihe von lustigen Aktivitäten und Spielen bringt. Wie Sie beim Spielen herausfinden werden, müssen Sie zuerst ein Spiel fertiggestellt haben, bevor Sie mit dem nächsten beginnen können.

Das Listing ist schon lang, aber Sie können sich durch den Gedanken an die friedliche Zeit, wenn Sie das Programm einmal im Computer haben, überzeugen, daß es die Mühe wert ist. Wenn Ihnen z.B. ein wohlmeinender Freund erklärt, wie toll es ist, daß Sie einen Heimcomputer haben, aber was um alles in der Welt Sie damit machen, brauchen Sie nur dieses Programm zu laden und sich ein Buch zur Hand nehmen, das Sie in der nächsten Stunde lesen wollen. Der Computer wird es übernehmen, Ihren Freund zu unterhalten. Hier ist das laufende Programm:

Denken Sie an eine Zahl zwischen
eins und dreihundsechzig

Ich werde Ihnen fuerf
Bildschirme voll Zahlen zeigen.

Wenn Ihre Zahl auf dem Schirm
ist, druecken Sie 'J' (=Ja),
wenn nicht, einfach 'N'.

Ich werde Ihnen am Ende des
Spieles die Zahl nennen, an
die Sie gedacht haben.

```

1 3 5 7 9 11 13 15 17
19 21 23 25 27 29 31
33 35 37 39 41 43 45
47 49 51 53 55 57 59
61 63

```

Druecken Sie 'J' wenn Ihre Zahl
hier ist, 'N' wenn nicht

```

2 3 6 7 10 11 14 15
18 19 22 23 26 27 30
31 34 35 38 39 42 43
46 47 50 51 54 55 58
59 62 63

```

Druecken Sie 'J' wenn Ihre Zahl
hier ist, 'N' wenn nicht

```

4 5 6 7 12 13 14 15
20 21 22 23 28 29 30
31 36 37 38 39 44 45
46 47 52 53 54 55 60
61 62 63

```


Druecken Sie 'J' wenn Ihre Zahl
hier ist, 'N' wenn nicht

8 9 10 11 12 13 14 15
24 25 26 27 28 29 30
31 40 41 42 43 44 45
46 47 56 57 58 59 60
61 62 63

Druecken Sie 'J' wenn Ihre Zahl
hier ist, 'N' wenn nicht

16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31 48 49 50 51 52
53 54 55 56 57 58 59
60 61 62 63

Druecken Sie 'J' wenn Ihre Zahl
hier ist, 'N' wenn nicht

32 33 34 35 36 37 38
39 40 41 42 43 44 45
46 47 48 49 50 51 52
53 54 55 56 57 58 59
60 61 62 63

Druecken Sie 'J' wenn Ihre Zahl
hier ist, 'N' wenn nicht

Ihre Zahl war....

23 !

Nun ist es Zeit fuer
etwas voellig anderes!

Ich werde an eine Zahl zwischen
eins und 40 denken, und Sie
haben nur sechs Versuche, um
sie herauszufinden...wenn Sie es
schaffen, koennen Sie das
naechste Spiel angehen.

Wenn Sie es nicht schaffen,
erhalten Sie ein leichteres
Raetsel aufzuloesen...so
lange, bis es sogar etwas so
dummes wie ein Mensch
loesen koennte...

Bitte warten Sie...

OK, ich denke an eine Zahl
zwischen eins und 40

Versuch 1
720
Zu niedrig!

Versuch 2
730
Zu niedrig!

Versuch 3
735
Zu niedrig!

Versuch 4
737
Zu niedrig!

Versuch 5
739

Wau!!! Wumm!!

Sie haben es geschafft...

Nun ist es Zeit, erstaunt
zu sein...

Befolgen Sie diese Anleitung
und druecken Sie RETURN nachdem
Sie sie ausgefuehrt haben...

Beginnen Sie, indem Sie Ihr
Alter mit 2 multiplizieren...
?

Nun addieren sie fuenf...
?

Multiplizieren Sie das mit 50...
?

Ziehen Sie nun 365 davon ab...
?

Addieren Sie nun die Menge von
Kleingeld in Ihrer Tasche...
?

Geben Sie nun die errechnete
Zahl ein...
71508

Sie haben 23 Kleingeld...

und sind 16 Jahre alt...

Nun etwas, um Ihr Gehirn
wieder auf normal zu bringen...

In diesem Spiel beginnen wir
mit zwei Bloecken von vier
Gegenstaenden in einer Reihe,
mit einem Space dazwischen,
wie z.B. folgendes:

\$\$\$\$ ****

Sie muessen Sie in folgende
Ordnung bringen:
**** \$\$\$\$
und das in der kuerzesten
Zeit...

Sie koennen nur in ein leeres
Feld ruecken, oder ueber ein
Stueck in ein leeres Feld
springen...ich zeige es
Ihnen jetzt...

\$\$\$\$ ****
\$\$\$ \$****
\$\$\$* \$\$\$
\$\$\$* \$*

und so weiter...

Druecken Sie RETURN, wenn
Sie bereit sind...
?

Das war Zug 40

\$ \$ \$ \$ * * * *
1 2 3 4 5 6 7 8 9

Sie haben 0 richtig
Welchen Stein wollen Sie bewegen ?

Bravo...Sie haben es in
nur 40 Zuegen geschafft!

Das bringt uns zum Ende des
Raetselaufmarsches

Geben Sie 'J' ein um nocheinmal
zu spielen, 'N' fuer Ende...

Danke, dass Sie soviel Zeit
gehabt haben, mit mir zu
spielen...es hat Spass gemacht!

Auf baldiges Wiedersehen...

Und hier ist das lange Listing, das ich Ihnen versprach:

```

10 REM Aufmarsch der Puzzles
20 GOSUB2550
30 X=.5:N=0
40 PRINT"Denken Sie an eine Zahl zwisch
en"
50 PRINT"eins und dreiundsechzig"
60 GOSUB2570
70 PRINT"Ich werde Ihnen fuenf"
80 PRINT"Bildschirme voll Zahlen zeigen
."
90 GOSUB2570
100 PRINT"Wenn Ihre Zahl auf dem Schirm"
110 PRINT"ist, druecken Sie 'J' (=Ja),"
120 PRINT"wenn nicht, einfach 'N'."
130 GOSUB2570
140 PRINT"Ich werde Ihnen am Ende des"
150 PRINT"Spieles die Zahl nennen, an"
160 PRINT"die Sie gedacht haben."
170 GOSUB2580:GOSUB2580
180 GOSUB2550
190 PRINT"1 3 5 7 9 11 13 15 17"
200 PRINT"19 21 23 25 27 29 31"
210 PRINT"33 35 37 39 41 43 45"
220 PRINT"47 49 51 53 55 57 59"
230 PRINT"61 63"
240 GOSUB620
250 PRINT" 2 3 6 7 10 11 14 15"
260 PRINT"18 19 22 23 26 27 30"
270 PRINT"31 34 35 38 39 42 43"
280 PRINT"46 47 50 51 54 55 58"
290 PRINT"59 62 63"
300 GOSUB620
310 PRINT" 4 5 6 7 12 13 14 15"
320 PRINT"20 21 22 23 28 29 30"
330 PRINT"31 36 37 38 39 44 45"
340 PRINT"46 47 52 53 54 55 60"
350 PRINT"61 62 63"
360 GOSUB620

```

```

370 PRINT"8 9 10 11 12 13 14 15"
380 PRINT"24 25 26 27 28 29 30"
390 PRINT"31 40 41 42 43 44 45"
400 PRINT"46 47 56 57 58 59 60"
410 PRINT"61 62 63"
420 GOSUB620
430 PRINT"16 17 18 19 20 21 22"
440 PRINT"23 24 25 26 27 28 29"
450 PRINT"30 31 48 49 50 51 52"
460 PRINT"53 54 55 56 57 58 59"
470 PRINT"60 61 62 63"
480 GOSUB620
490 PRINT"32 33 34 35 36 37 38"
500 PRINT"39 40 41 42 43 44 45"
510 PRINT"46 47 48 49 50 51 52"
520 PRINT"53 54 55 56 57 58 59"
530 PRINT"60 61 62 63"
540 GOSUB620
550 GOSUB2570:GOSUB2570
560 GOSUB2550
570 PRINT"Ihre Zahl war...."
580 GOSUB2570
590 PRINT",N;" !"
600 GOSUB2570
610 GOTO720
620 X=X+X
630 GOSUB2570
640 PRINT"Druecken Sie 'J' wenn Ihre Zah
1"
650 PRINT"hier ist, 'N' wenn nicht"
660 A$=INKEY$(10)
670 IF A$<>"N" AND A$<>"n" AND A$<>"J" A
ND A$<>"j" THEN GOTO660
680 IF A$="J" OR A$="j" THEN N=N+X
690 GOSUB2550
700 RETURN
710 REM *****
720 GOSUB2550
730 PRINT"Nun ist es Zeit fuer"

```

```

740 PRINT"etwas voellig anderes!"
750 GOSUB2570
760 W=40
770 PRINT"Ich werde an eine Zahl zwische
n"
780 PRINT"eins und ";W;" denken, und Sie
"
790 PRINT"haben nur sechs Versuche, um"
800 PRINT"sie herauszufinden...wenn Sie
es"
810 PRINT"schaffen, koennen Sie das"
820 PRINT"naechste Spiel angehen."
830 GOSUB2570
840 PRINT"Wenn Sie es nicht schaffen,"
850 PRINT"erhalten Sie ein leichteres"
860 PRINT"Raetsel aufzuloesen...so"
870 PRINT"lange, bis es sogar etwas so"
880 PRINT"dummes wie ein Mensch"
890 PRINT"loesen koennte..."
900 PRINT:PRINT
910 PRINT"Bitte warten Sie..."
920 GOSUB2570
930 GOSUB2550
940 PRINT"OK, ich denke an eine Zahl"
950 PRINT"zwischen eins und ";W
960 N=INT(RND(1)*W)+1
970 J=0
980 J=J+1
990 PRINT:PRINT"Versuch ";J
1000 INPUT Q
1010 IF Q=N THEN GOTO1150
1020 IF Q<1 OR Q>W THEN GOTO1000
1030 IF Q<N THEN PRINT"Zu niedrig!"
1040 IF Q>N THEN PRINT"Zu hoch!"
1050 IF J<6 THEN GOTO980
1060 PRINT:PRINT"Ziemlich hoffnungslos..."
"
1070 PRINT"Ich dachte an ";N
1080 IF W=5 THEN PRINT"Sie sind ein hoffn

```

```

ungsloser Fall...":PRINT"versuchen wir etw
as anderes":GOSUB2570:GOTO1200
1090 PRINT
1100 PRINT"Tja, ich glaube, ich muss"
1110 PRINT"Ihnen einen weiteren Versuch"
1120 PRINT"geben..."
1130 W=W/2
1140 GOTO940
1150 GOSUB2570
1160 PRINT"Wau!!! Wumm!!"
1170 GOSUB2570
1180 PRINT"Sie haben es geschafft..."
1190 GOSUB2570
1200 GOSUB2550
1210 REM *****
1220 PRINT"Nun ist es Zeit, erstaunt"
1230 PRINT"zu sein..."
1240 GOSUB2570
1250 PRINT"Befolgen Sie diese Anleitung"
1260 PRINT"und druecken Sie RETURN nachde
m"
1270 PRINT"Sie sie ausgefuehrt haben..."
1280 GOSUB2570
1290 GOSUB2550
1300 PRINT"Beginnen Sie, indem Sie Ihr"
1310 PRINT"Alter mit 2 multiplizieren..."
1320 INPUT A$
1330 GOSUB2550
1340 PRINT"Nun addieren sie fuenf..."
1350 INPUT A$
1360 GOSUB2550
1370 PRINT"Multiplizieren Sie das mit 50.
.."
1380 INPUT A$
1390 GOSUB2550
1400 PRINT"Ziehen Sie nun 365 davon ab..."
"
1410 INPUT A$
1420 GOSUB2550

```

```

1430 PRINT"Addieren Sie nun die Menge von
"
1440 PRINT"Kleingeld in Ihrer Tasche..."
1450 INPUT A$
1460 GOSUB2550
1470 PRINT"Geben Sie nun die errechnete"
1480 PRINT"Zahl ein..."
1490 INPUT A
1500 A=A+115
1510 B=INT(A/100)
1520 A=A-100*B
1530 GOSUB2550
1540 PRINT"Sie haben ";A;" Kleingeld..."
1550 GOSUB2570
1560 PRINT"und sind ";B;" Jahre alt..."
1570 GOSUB2570
1580 REM *****
1590 GOSUB2550
1600 PRINT"Nun etwas, um Ihr Gehirn"
1610 PRINT"wieder auf normal zu bringen..
"
1620 GOSUB2570
1630 PRINT"In diesem Spiel beginnen wir"
1640 PRINT"mit zwei Bloecken von vier"
1650 PRINT"Gegenstaenden in einer Reihe,"
1660 PRINT"mit einem Space dazwischen,"
1670 PRINT"wie z.B. folgendes:"
1680 PRINT
1690 PRINT"$$$$ ****"
1700 GOSUB2570
1710 PRINT"Sie muessen Sie in folgende"
1720 PRINT"Ordnung bringen:"
1730 PRINT"**** $$$"
1740 PRINT"und das in der kuerzesten"
1750 PRINT"Zeit..."
1760 GOSUB2570
1770 PRINT"Sie koennen nur in ein leeres"
1780 PRINT"Feld ruecken, oder ueber ein"
1790 PRINT"Stueck in ein leeres Feld"

```

```

1800 PRINT"springen...ich zeige es"
1810 PRINT"Ihnen jetzt..."
1820 GOSUB2570:GOSUB2570
1830 FOR J=1 TO 3
1840 CLS
1850 PRINT"$$$$ ****"
1860 FOR Z=1 TO 1000:NEXT Z:CLS
1870 PRINT"$$$ *****"
1880 FOR Z=1 TO 1000:NEXT Z:CLS
1890 PRINT"$$$$$ ***"
1900 FOR Z=1 TO 1000:NEXT Z:CLS
1910 PRINT"$$$$$$$ **"
1920 NEXT J
1930 PRINT:PRINT"und so weiter..."
1940 GOSUB2570
1950 PRINT"Druecken Sie RETURN, wenn"
1960 PRINT"Sie bereit sind..."
1970 INPUT A$
1980 X=-1:M=9
1990 DIM A(M)
2000 FOR Q=1 TO M
2010 IF Q<5 THEN A(Q)=ASC("$")
2020 IF Q=5 THEN A(Q)=ASC(" ")
2030 IF Q>5 THEN A(Q)=ASC("*")
2040 NEXT Q
2050 GOSUB2550
2060 C=0
2070 X=X+1
2080 IF X=0 THEN GOTO2100
2090 PRINT"Das war Zug ";X
2100 PRINT
2110 FOR Q=1 TO M
2120 PRINT CHR$(A(Q));" ";
2130 IF Q<5 AND A(Q)=ASC("*") THEN C=C+1
2140 IF A(Q)=ASC(" ") THEN H=Q
2150 IF Q>5 AND A(Q)=ASC("$") THEN C=C+1
2160 NEXT Q
2170 PRINT:PRINT"1 2 3 4 5 6 7 8 9"
2180 PRINT

```

```

2190 PRINT:PRINT"Sie haben ";C;" richtig"
2200 GOSUB2240
2210 IF C=8 THEN GOTO2280
2220 PRINT:PRINT
2230 GOTO2050
2240 INPUT"Welchen Stein wollen Sie beweg
en ";T
2250 K=A(T)
2260 A(T)=ASC(" "):A(H)=K
2270 RETURN
2280 PRINT:PRINT"Bravo...Sie haben es in"
2290 PRINT"nur ";X;" Zuegen geschafft!"
2300 GOSUB2570
2310 GOSUB2550
2320 REM *****
2330 PRINT"Das bringt uns zum Ende des"
2340 PRINT"Raetselaufmarsches"
2350 GOSUB2570
2360 PRINT"Geben Sie 'J' ein um nocheinma
l"
2370 PRINT"zu spielen, 'N' fuer Ende..."
2380 A$=INKEY$(10)
2390 IF A$(">"N" AND A$(">"n" AND A$(">"J" A
ND A$(">"j" THEN GOTO2380
2400 IF A$="J" OR A$="j" THEN GOTO2480
2410 GOSUB2550
2420 PRINT"Danke, dass Sie soviel Zeit"
2430 PRINT"gehabt haben, mit mir zu"
2440 PRINT"spielen...es hat Spass gemacht
!"
2450 GOSUB2570
2460 PRINT"Auf baldiges Wiedersehen..."
2470 END
2480 GOSUB2570
2490 PRINT"OK, wir starten nocheinmal..."
2500 GOSUB2570
2510 CLEAR
2520 GOTO10
2530 END

```

```

2540 REM *****
2550 REM Pause, Cls, Print
2560 CLS
2570 PRINT:PRINT:PRINT
2580 FOR Z=1 TO 2000:NEXT Z
2590 RETURN

```

Roboterminenfeld

Jetzt wird es Zeit, sich dem Schrecken, den Gefahren und dem Horror des **Roboterminenfeldes** zu stellen. Da spazieren Sie eines schönen Tages nichts Böses ahnend ganz friedlich in der Sonne und finden sich plötzlich umgeben von einem hohen Zaun. Inmitten dieses eingezäunten Gebietes suchen einige irre Roboter Sie zu vernichten.

Die Roboter erscheinen auf dem Bildschirm als Dollarzeichen, Sie werden durch das M dargestellt, und die Sterne (*) sind Minen, die sowohl für Sie als auch für die Roboter tödlich sind. Die Roboter, sehr sehr ungeschickt programmiert, können zwar Sie, nicht aber die Minen aufspüren. Daher liegt Ihre Rettung darin, Minen zwischen sich und die Roboter zu bekommen. Dadurch rennen die Roboter, wenn sie auf Sie zukommen, in die Minen und fliegen in die Luft.

Doch haben auch die Roboter ein As im Ärmel. Sie können gemeinsam marschieren, aber nur das Abbild eines Roboters ist sichtbar. So kann es ohne weiteres passieren, daß Sie glauben, nur von einem Roboter bedroht zu werden, doch in allerletzter Sekunde teilt dieser sich in zwei oder mehrere, um auf beiden Seiten an einer Mine vorbeizugehen. Das ist kein Spiel für Leute, die das stille Leben lieben.

Sie können sich nach Norden, Süden, Osten oder Westen (wobei Norden der obere Bildschirmrand ist) dadurch bewegen, daß Sie die Anfangsbuchstaben der gewünschten Richtung eingeben (N; wenn Sie nach Norden wollen, O; wenn Sie nach Osten wollen, etc.).

Nur um Ihnen zu zeigen, daß Sie wirklich nicht die besten Chancen haben, präsentiere ich Ihnen hier ein Spiel:

```

X X X X X X X X X X X X X X
X . . . . . * . $ . . X
X . * . * . . . . . * . X
X . . . . . * . . . * . X
X . * . . . . . . . . X
X . . . * . . . . . . . X
X . . . * . . . . . . . X
X . . . * . . . . . . . X
X . * . * . . . . . . X
X . . * . . . . . * . . X
X . . . . . M . . . . X
X . . . * . . . . * . . X
X . . . . . $ . . . . X
X . . . . . $ . . . . X
X X X X X X X X X X X X X X

```

```

X X X X X X X X X X X X X X
X . . . . . * . . . . X
X . * . * . . . . $ . * . X
X . . . . . * . . . . X
X . . . . . * . . . . X
X . * . . . . . . . * . X
X . . . . . * . . . . X
X . . * . * . . . . . X
X . * . * . . . . . X
X . . . . . M . * . . X
X . . . * . * . . . * . X
X . . . . . $ $ . . . X
X . . . . . . . . . X
X X X X X X X X X X X X X X

```

```

X X X X X X X X X X X X X X
X . . . . . * . . . . X
X . * . * . . . . . * . X
X . . . . . $ . . . . X
X . . . . . * . . . * . X
X . * . . . . . . . * . X
X . . . * . . . . . . X
X . . . * . . . . . . X
X . * . * . . . . . X
X . . . * . . . . . X
X . * . * . . . . M * . X
X . . . . . . . * . . X
X . . . * . * . $ $ . * . X
X . . . . . $ . . . . X
X . . . . . . . . . X
X X X X X X X X X X X X X X

```

```

X X X X X X X X X X X X X X
X . . . . . * . . . . X
X . * . * . . . . . * . X
X . . . . . . . . . * . X
X . . . . . * . $ . * . X
X . * . . . . . . . * . X
X . . . * . . . . . . X
X . . . * . . . . . . X
X . * . * . . . M . . X
X . . . . . . . * . . X
X . . . . . $ . * . . X
X . . . * . * . $ . * . X
X . . . . . . . . . X
X . . . . . . . . . X
X X X X X X X X X X X X X X

```


[illegible]

```

X X X X X X X X X X X
X X   *   *   *   *   X
X   *   *   .   .   *   X
X   .   .   .   .   .   X
X   .   .   .   *   *   X
X   *   .   .   .   *   X
X   .   .   *   .   .   X
X   .   *   *   .   .   X
X   *   *   *   .   .   X
X   *   *   .   .   *   X
X   .   *   *   .   *   X
X   .   .   *   *   .   X
X X X X X X X X X X X

```

[illegible]

Und hier ein Spiel, in dem die menschliche Rasse (repräsentiert durch mich an der Computerkonsole) ein glücklicheres Schicksal hatte:

[illegible]

Kaputte Roboter: 1

[illegible]

Kaputte Roboter: 1

[illegible]

Kaputte Roboter: 2

```

X X X X X X X X X X X X X X
X . . . * . . . * . . . * . X
X . . . * . . . . . * . . . X
X . . . . . . . . . . . . . X
X . . . . . * . . . . . . . . X
X . $ $ . . . * . . . . . . . X
X . . . . . . . . . . * . . . X
X . * . . . . . . . . . . . . X
X . . . * . . . . . . . . . . X
X * M . . . * . . . . . . . . X
X . . . . . . . . . . . . . . X
X . . . . . * . . . . . . . . X
X . . . . . . . . . . . . * . X
X . . . . . . . . . . . . . . X
X X X X X X X X X X X X X X
Kaputte Roboter: 2

```

```

X X X X X X X X X X X X X X
X . . . * . . . * . . . * . X
X . . . * . . . . . * . . . X
X . . . . . . . . . . . . . X
X . . . . . * . . . . . . . . X
X . $ . . . . . . . . * . . . X
X . * . . . . . . . . . . . . X
X . . . * . . . . . . . . . . X
X * M . . . * . . . . . . . . X
X . . . . . . . . . . . . . . X
X . . . . . * . . . . . . . . X
X . . . . . . . . . . . . * . X
X . . . . . . . . . . . . . . X
X X X X X X X X X X X X X X
Kaputte Roboter: 4

```

```

X X X X X X X X X X X X X X
X . . . * . . . * . . . * . X
X . . . * . . . . . * . . . X
X . . . . . * . . . . . . . . X
X . . . * . . . . . . . . . . X
X . * . . . . . . . . . . . . X
X . . . . . . . . . . * . . . X
X . * . . . . . . . . . . . . X
X * M . . . * . . . . . . . . X
X . . * . . . . . . . . . . . X
X . . . * . . . . . . . . . . X
X . . . . . . . . . . . . * . X
X . . . . . . . . . . . . . . X
X X X X X X X X X X X X X X
Sie haben die Roboter abgewehrt!!

```

Dies ist das Listing von Roboterminefeld:

```

10 REM Roboter-Minenfeld
20 GOTO250
30 REM Roboter bewegen
40 KAPUTT=0
50 FOR E=1 TO 4
60 IF A(B(E),C(E))=42 THEN KAPUTT=KAPUT
T+1:GOTO200
70 X=B(E):Y=C(E)
80 IF B(E)<D THEN B(E)=B(E)+1
90 IF B(E)>D AND RND(1)>.3 THEN B(E)=B(
E)-1
100 IF C(E)<F AND RND(1)>.3 THEN C(E)=C(
E)+1
110 IF C(E)>F THEN C(E)=C(E)-1
120 IF B(E)<2 THEN B(E)=2
130 IF B(E)>14 THEN B(E)=14
140 IF C(E)<2 THEN C(E)=2
150 IF C(E)>14 THEN C(E)=14
160 A(X,Y)=4632
170 IF A(B(E),C(E))=42 THEN KAPUTT=KAPUT
T+1:GOTO200
180 IF A(B(E),C(E))=77 THEN A(B(E),C(E))
=36:GOSUB300:GOTO900
190 A(B(E),C(E))=36 193
200 NEXT E
210 IF KAPUTT<CH THEN KAPUTT=CH
220 CH=KAPUTT
230 IF KAPUTT=4 THEN GOSUB300:GOTO960
240 RETURN
250 GOSUB530:REM INITIALISIEREN
260 GOSUB300:REM MINENFELD AUSDRUCKEN
270 GOSUB30:REM ROBTER BEWEGEN
280 GOSUB420:REM MENSCH BEWEGEN
290 GOTO260
300 REM MINENFELD AUSDRUCKEN
310 CLS
320 IF KAPUTT>0 THEN PRINT"Kaputte Robot
er: ";KAPUTT:PRINT

```

```

330 IF KAPUTT=0 THEN PRINT:PRINT
340 PRINT:PRINT:PRINT
350 FOR B=1 TO 15
360 FOR C=1 TO 15
370 PRINT CHR$(A(B,C));" ";
380 NEXT C
390 PRINT
400 NEXT B
410 RETURN
420 REM SPIELERZUG
430 A(D,F)=46 32
440 A$=INKEY$(10)
450 IF A$="" THEN GOTO440 NEXT:A(D,F)=216
460 IF A$="N" AND D>2 THEN D=D-1
470 IF A$="S" AND D<14 THEN D=D+1
480 IF A$="O" AND F<14 THEN F=F+1
490 IF A$="W" AND F>2 THEN F=F-1
500 IF A(D,F)=42 THEN GOSUB300:GOTO980
510 A(D,F)=77 26 75
520 RETURN
530 REM INITIALISIEREN
540 DIM A(15,15),B(4),C(4)
550 CLS
560 PRINT:PRINT"Bitte warten Sie einen M
oment..."
570 CH=0
580 KAPUTT=0
590 REM RANDOMIZE
600 REM WAENDE AUFSTELLEN
610 FOR B=1 TO 15
620 FOR C=1 TO 15
630 A(B,C)=46 21
640 IF B=1 OR B=15 OR C=1 OR C=15 THEN A
(B,C)=88 19
650 NEXT C:NEXT B
660 REM MINEN VERGRABEN
670 FOR B=1 TO 20
680 C=INT(RND(1)*13)+1
690 D=INT(RND(1)*13)+1

```

```

700 IF A(C,D)=88 24 THEN GOTO680
710 A(C,D)=42 275
720 NEXT B
730 DATA 4,4,13,8,8,3,12,7
740 REM ROBOTER EINSCHALTEN
750 FOR E=1 TO 4
760 D=INT(RND(1)*13)+2
770 F=INT(RND(1)*13)+2
780 IF A(D,F)<>46 THEN GOTO760
790 B(E)=D:C(E)=F
800 A(B(E),C(E))=36 75
810 NEXT E
820 REM MENSCH ABWERFEN
830 D=INT(RND(1)*13)+2
840 F=INT(RND(1)*13)+2
850 IF A(D,F)<>46 THEN GOTO820
860 A(D,F)=77 26 75
870 RETURN
880 REM MENSCH AUF D,F
890 REM ROBOTER AUF B(E),C(E)
900 REM SPILENDE
910 CLS:PRINT"Ein Roboter hat Sie erwisc
ht!!"
920 CH=0:KAPUTT=0:GOSUB340
930 A$=""
940 IF INKEY$(10)<>" THEN GOTO940
950 GOSUB560:GOTO260
960 CLS:PRINT"Sie haben die Roboter abge
wehrt!!"
970 GOTO920
980 CLS:PRINT"Sie sind in eine Mine gela
ufen!!"
990 GOTO920

```

Drei in einer Reihe

Ich bin sicher, daß **Drei-in-einer-Reihe** keiner Vorstellung bedarf. Es ist ein Programm, das ganz gut, allerdings nicht in seiner vollkommen vorhersehbaren Art spielt. Daher haben Sie die gerechte Chance, einige wenige Spiele zu gewinnen.

Das klingt nicht nach einem sehr guten Angebot. Es wurden allerdings sehr viele **Drei-in-einer-Reihe**-Programme geschrieben, gegen die Sie bestenfalls unentschieden spielen können, wenn Sie nicht sogar immer von der Maschine geschlagen werden.

Dieses Programm entscheidet bei jedem Spiel, wer den Eröffnungszug erhält. Es kontert flink auf Ihre Züge. Sie müssen nur die Nummer des Feldes, in das Sie ziehen wollen, eingeben.

Sehen wir uns **Drei-in-einer-Reihe** nun in Aktion an:

Druecken Sie eine Taste,
wenn Sie fertig sind
Ich habe den ersten Zug

Geben Sie Ihren Zug ein...

1 2 3 - - -
4 5 6 - - -
7 8 9 - - -

1 2 3 X - -
4 5 6 - 0 -
7 8 9 - - -

1 2 3 - - -
4 5 6 - 0 -
7 8 9 - - -

1 2 3 X - 0
4 5 6 - 0 -
7 8 9 - - -

Geben Sie Ihren Zug ein...

Geben Sie Ihren Zug ein...

1 2 3 X - 0
4 5 6 - 0 -
7 8 9 X - -

1 2 3 X - 0
4 5 6 0 0 X
7 8 9 X - -

1 2 3 X - 0
4 5 6 0 0 -
7 8 9 X - -

1 2 3 X 0 0
4 5 6 0 0 X
7 8 9 X - -

Geben Sie Ihren Zug ein...

Ich habe den ersten Zug

1 2 3 X 0 0
4 5 6 0 0 X
7 8 9 X X -

1 2 3 - - -
4 5 6 - - -
7 8 9 - - -

1 2 3 X 0 0
4 5 6 0 0 X
7 8 9 X X 0

1 2 3 - - -
4 5 6 - 0 -
7 8 9 - - -

Es ist ein Unentschieden!

Geben Sie Ihren Zug ein...

```

1 2 3   X - -
4 5 6   - 0 -
7 8 9   - - -

```

```

1 2 3   X - 0
4 5 6   - 0 -
7 8 9   - - -

```

Geben Sie Ihren Zug ein...

```

1 2 3   X - 0
4 5 6   0 0 -
7 8 9   X - X

```

```

1 2 3   X - 0
4 5 6   0 0 0
7 8 9   X - X

```

Ich bin der Sieger!

Hier ist das Listing:

```

10 REM DREI-IN-EINER-REIHE
20 CLS
30 N=0
40 PRINT"Druecken Sie eine Taste,"
50 PRINT"wenn Sie fertig sind"
60 N=N+1:IF INKEY$(10)=" " THEN GOTO60
70 CLS
80 DIM A(9)
90 REM RANDOMIZE N
100 CLS:FOR N=1 TO 9:A(N)=0:NEXT N
110 IF RND(1)>.5 THEN PRINT"ich habe den
ersten Zug":FOR J=1 TO 2000:NEXT J:CLS:GO
TO150
120 GOSUB740
130 GOSUB470
140 GOSUB660
150 GOSUB740
160 GOSUB470
170 IF A(5)=0 THEN A(5)=1:GOTO120
180 REM Reihe/Block vervollstaendigen
190 D=1
200 B=1
210 IF B=1 THEN X=1:Y=2:Z=3
220 IF B=2 THEN X=1:Y=4:Z=7
230 IF B=3 THEN X=1:Y=5:Z=9
240 IF B=4 THEN X=3:Z=7
250 C=1
260 IF A(X)=D AND A(Y)=D AND A(Z)=0 THEN
A(Z)=1:GOTO120
270 IF A(X)=D AND A(Y)=0 AND A(Z)=D THEN
A(Y)=1:GOTO120
280 IF A(X)=0 AND A(Y)=D AND A(Z)=D THEN
A(X)=1:GOTO120
290 IF B=1 THEN X=X+3:Y=Y+3:Z=Z+3
300 IF B=2 THEN X=X+1:Y=Y+1:Z=Z+1
310 IF C<3 THEN C=C+1:GOTO260
320 IF B<4 THEN B=B+1:GOTO220
330 IF D<2 THEN D=D+1:GOTO200

```

```

340 REM Zufallszug
350 B=1
360 D=INT(RND(1)*9)+1
370 IF A(C)=0 THEN A(C)=1:GOTO120
380 B=B+1
390 IF B<21 THEN GOTO360
400 B=0
410 B=B+1
420 IF A(B)=0 THEN A(B)=1:GOTO120
430 IF B<9 THEN GOTO410
440 GOSUB740
450 PRINT:PRINT"Es ist ein Unentschieden
!"
460 GOTO640
470 REM Gewinnntest
480 FOR B=1 TO 4
490 IF B=1 THEN X=1:Y=2:Z=3
500 IF B=2 THEN X=1:Y=4:Z=7
510 IF B=3 THEN X=1:Y=5:Z=9
520 IF B=4 THEN X=3:Z=7
530 FOR C=1 TO 3
540 IF A(X)=A(Y) THEN IF A(Y)=A(Z) THEN
IF A(X)<>0 THEN GOTO600
550 IF B=1 THEN X=X+3:Y=Y+3:Z=Z+3
560 IF B=2 THEN X=X+1:Y=Y+1:Z=Z+1
570 NEXT C
580 NEXT B
590 RETURN
600 REM Der Sieger!
610 PRINT
620 IF A(X)=1 THEN PRINT"Ich bin der Sie
ger!"
630 IF A(X)=2 THEN PRINT"Sie sind der Si
eger!"
640 FOR I=1 TO 2000:NEXT I
650 GOTO100
660 REM Spielerzug
670 PRINT:PRINT"Geben Sie Ihren Zug ein.
.."

```

```

680 A$=INKEY$(10)
690 IF A$<"1" OR A$>"9" THEN GOTO680
700 B=VAL(A$)
710 IF A(B)<>0 THEN GOTO680
720 A(B)=2
730 RETURN
740 REM Ausdruck
750 CLS
760 PRINT:PRINT:PRINT:PRINT
770 PRINT"1 2 3  ";
780 FL=0
790 FOR B=1 TO 9
800 IF A(B)=0 THEN FL=1
810 IF A(B)=0 THEN PRINT" - ";
820 IF A(B)=1 THEN PRINT" 0 ";
830 IF A(B)=2 THEN PRINT" X ";
840 IF B=3 THEN PRINT:PRINT:PRINT"4 5 6
";
850 IF B=6 THEN PRINT:PRINT:PRINT"7 8 9
";
860 NEXT B
870 PRINT:PRINT
880 IF FL=0 THEN GOTO450
890 RETURN

```

Las Vegas

Am Ende dieses Kapitels haben wir ein Spiel mit einem ‚einarmigen Banditen‘ für Sie. Bevor Sie den Hebel der Maschine betätigen (was Sie hier durch Drücken der Leertaste durchführen), brauchen Sie nur entscheiden, wieviel Sie wetten wollen, und die Glücksräder beginnen abzuheben.

Ihre Gewinne sind, wie Sie sicher erwartet haben, von der Wahrscheinlichkeit der einzelnen erscheinenden Kombination abhängig. Während sich Ihr Reichtum oder Ihre Armut abzeichnet, hält der Computer mit Ihnen eine kleine Konversation.

Und jetzt das Listing:

```

10 REM Las Vegas
20 GOSUB1100:REM Initialisieren
30 GOSUB870:REM Spielereingabe
40 GOSUB520:REM Slotmaschine betätigen
50 IF GELD<1 THEN GOT090
60 IF GELD>2500 THEN GOT0290
70 GOT030
80 REM *****
90 REM Peite
100 REM *****
110 GOSUB410
120 PRINT"Das ist das Ende des Spiels, o
"
130 PRINT"einst maechtiger Spieler..."
140 GOSUB410
150 PRINT"Sie sind total pleite!!"
160 GOSUB410
170 PRINT"Druecken Sie 'J', wenn Sie ein
en"
180 PRINT"weiteren Durchgang"
190 PRINT"!!!! LAS VEGAS !!!!"
200 PRINT"spielen wollen (oder 'N' fuer
Ende)"
210 A$=INKEY$(10)
220 IF A$<>"J" AND A$<>"j" AND A$<>"N" A
ND A$<>"n" THEN GOT0210
230 IF A$="J" OR A$="j" THEN RUN
240 PRINT:PRINT"OK, Gluecksspieler..."
250 GOSUB410
260 PRINT"Danke fuer das Spiel!"
270 END
280 REM *****
290 REM Bank gesprengt
300 REM *****
310 GOSUB410
320 PRINT"Bravo, Spieler!!"
330 GOSUB410
340 PRINT"Sie haben unser Hauslimit"

```

```

350 PRINT"erreicht und wir muessen Sie"
355 PRINT"daher hinauswerfen"
360 GOSUB410
370 PRINT"Leute mit Glueck wie Sie mache
n"
380 PRINT"unserem Casino einen schlechte
n"
385 PRINT"Namens....."
"
390 GOTO160
400 REM *****
410 REM Pause
420 REM *****
430 FOR P=1 TO 2000:NEXT P
440 PRINT:PRINT
450 RETURN
460 REM *****
470 REM Pause 2
480 REM *****
490 FOR P=1 TO 2000:NEXT P
500 RETURN
510 REM *****
520 REM Slotmachine betaetigen
530 REM *****
540 CLS
550 GOSUB410
560 PRINT"/
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
570 PRINT TAB(5);" * ";
580 FOR M=1 TO 3
590 GOSUB460
600 A=INT(RND(1)*43)
610 IF A<2 THEN PRINT A$(4);:C(M)=1
620 IF A>1 AND A<6 THEN PRINT A$(3);:C(M)
)=2
630 IF A>5 AND A<12 THEN PRINT A$(1);:C(
M)=3
640 IF A>11 AND A<20 THEN PRINT A$(2);:C
(M)=4

```

```

650 IF A>19 AND A<31 THEN PRINT A$(5);:C
(M)=5
660 IF A>30 THEN PRINT A$(6);:C(M)=6
670 PRINT " * ";
680 NEXT M
690 GOSUB410
700 SIE=0
710 IF C(1)+C(2)+C(3)=3 THEN PRINT"Drei
GOLD!!!":GOSUB410:PRINT"Das ist ein Jackpo
t!!!":SIE=SIE+9:GOTO750
720 IF C(1)=C(2) AND C(3)=C(2) AND C(1)=
2 THEN PRINT"Drei Glocken!!!":SIE=SIE+3.9:
GOTO750
730 IF C(1)=C(2) AND C(3)=C(2) AND C(1)<
>1 AND C(2)<>3 THEN PRINT"Drei von einer S
orte":SIE=SIE+3.5:GOTO750
740 IF C(1)=C(2) OR C(1)=C(3) OR C(2)=C(
3) THEN PRINT">> EIN PAAR <<":SIE=SIE+.7
750 IF C(3)=2 THEN PRINT"Glocke am Ende
bringt Bonus!":SIE=SIE+.6
760 IF C(1)=3 AND C(3)=3 THEN PRINT"Ein
Apfel auf jeder Seite ist gut":SIE=SIE+.5
770 IF C(1)=4 AND C(2)=3 AND C(3)=4 THEN
PRINT"DIESE ALTE 'Kirsche,Glocke,Kirsche'
KOMINATION":PRINT"IST EINE MEINER LIEBLI
NGE!":SIE=SIE+.4
780 GOSUB410
790 SIE=INT(SIE/EIN)
800 IF SIE>0 THEN PRINT"Und Sie haben $"
;SIE;" gewonnen!":GELD=GELD+SIE
810 IF SIE=0 THEN PRINT"Und Sie haben $"
;EIN;" verloren":GELD=GELD-EIN
820 GOSUB410
830 IF GELD>0 THEN PRINT"Sie haben jetzt
$";GELD
840 GOSUB460
850 RETURN
860 REM *****
870 REM Spielereingabe

```

```

880 REM *****
890 CLS
900 GOSUB410
910 IF GELD<300 THEN PRINT"Hallo, Spiele
r!"
920 IF GELD>299 AND GELD<600 THEN PRINT"
Heute haben Sie Glueck!"
930 IF GELD>599 AND GELD<900 THEN PRINT"
Die Gluecksfee muss Ihnen heute hold sein!
"
940 IF GELD>899 AND GELD<1200 THEN PRINT
"Das Schicksal ist heute extrem freundlich
"
950 IF GELD>1199 THEN PRINT"Es ist schoe
n, einen Experten an der Arbeit zu sehen!"
960 GOSUB470
970 PRINT:PRINT"Sie haben $";GELD
980 PRINT:INPUT"Wieviel wollen Sie setze
n ";EIN
990 IF EIN>GELD THEN PRINT"Soviel haben
Sie nicht!":GOTO980
1000 GOSUB410
1010 PRINT"Gut mein Herr, $";EIN;" sind d
er Einsatz"
1020 GOSUB410
1030 PRINT"Druecken Sie SPACE um zu spiel
en"
1040 IF INKEY$(10)<>" " THEN GOTO1040
1050 FOR T=1 TO 40
1060 PRINT TAB(T/2);"***** Bitte warte
n *****"
1070 PRINT
1080 NEXT T
1090 RETURN
1100 REM *****
1110 REM Initialisieren
1120 REM *****
1130 CLS

```



```

1140 DIM A$(6),C(6)
1150 REM RANDOMIZE
1160 GELD=250
1170 FOR B=1 TO 6
1180 READ A$(B)
1190 NEXT B
1200 RETURN
1210 DATA "$ Apfel $","#Kirsche#","* Gloc
ke*","!! GOLD !!","<< Zitro >>","ää Mango
üü"

```

Spaß mit Ihrem Drucker

Celestia

Das erste der beiden Programme dieses Kapitels ist **Celestia**. Sie können es entweder nur auf dem Bildschirm oder gleichzeitig auf dem Bildschirm und auf dem Drucker laufen lassen. Es erzeugt eine unendliche Serie von sich entwickelnden Mustern.

Die Muster entfalten sich nach den Regeln des berühmten Computerspiels **Life** (entwickelt von John Conway in seiner Zeit am Gonville und Caius College in Cambridge, UK). Martin Gardener verbreitete es durch seinen Artikel in *Scientific American* vom Oktober 1970 in der ganzen Welt.

In **Life** werden Zellen geboren, wachsen und sterben wieder nach den von Conway entwickelten Regeln. Man stellt sich vor, daß die Zellenkolonie auf einem Gitter wächst. Jede Zelle wird von 8 anderen umgeben. Das Stadium dieser 8 anderen Zellen bestimmt, was mit der betreffenden Zelle in zukünftigen Generationen passiert.

Die Gesetze, die die Entwicklung der Zellen bestimmen, sind folgende:

- Hat eine Zelle 2 oder 3 Zellen um sich, dann überlebt sie zur nächsten Generation.
- Sind drei und genau drei volle Zahlen neben einer leeren, so wird eine Zelle in der nächsten Generation in diesen leeren Raum 'geboren'.
- Jede Zelle, deren 4 benachbarte Zellen belegt sind, stirbt in der nächsten Generation.

Verzweifeln Sie nicht. Sie brauchen diese Gesetze nicht zu wissen. Der Computer befolgt sie ganz von selber. Diese Regeln produzieren viel attraktivere Muster, als Sie sich beim Lesen der Regel erträumen können. Sie sind auch (beinahe) unvorhersehbar. Es gibt noch einen Trick in diesem Programm, der es sogar noch eindrucksvoller macht. Immer, wenn eine Kolonie entsteht, druckt **Celestia** eigentlich 4 Kolonien. Die Originalkolonie ist in einem Quadrat des Bildschirms, die anderen 3 Viertel des Schirms enthalten Spiegelungen dieses Originals.

Wie Sie sehen können, ist **Celestia** unheimlich fesselnd:

[illegible][illegible][illegible]


```

380 IF A(X,Y)=B AND C<>3 AND C<>2 THEN B
(X,Y)=E
390 IF A(X,Y)=E AND C=3 THEN B(X,Y)=B
400 NEXT Y:NEXT X
410 FOR X=2 TO 9:FOR Y=2 TO 19
420 A(X,Y)=B(X,Y)
430 NEXT Y:NEXT X
440 RETURN
450 REM Initialisieren
460 CLS
470 PRINT"Druecken Sie die Leertaste":PR
INT"wenn Sie bereit sind"
480 N=1
490 IF INKEY$(10)="" THEN N=N+1:GOTO490
500 CLS:REM RANDOMIZE N
510 PRINT"Wollen Sie eine Hardcopy"
520 PRINT"auf dem Drucker (J oder N)"
530 A$=INKEY$(10):IF A$<>"N" AND A$<>"n"
AND A$<>"J" AND A$<>"j" THEN GOTO530
540 IF A$="J" OR A$="j" THEN N=0
550 CLS
560 B=ASC("*"):E=ASC(" ")
570 DIM A(10,20),B(10,20)
580 FOR X=2 TO 9:FOR Y=2 TO 19
590 A(X,Y)=E
600 IF RND(1)>.45 THEN A(X,Y)=B
610 B(X,Y)=A(X,Y)
620 NEXT Y:NEXT X
630 RETURN
640 LPRINT " -----
-----":REM 6 Spaces, 34 mal -
650 LPRINT:LPRINT:LPRINT" ";
660 FOR X=2 TO 9
670 FOR Y=2 TO 19
680 LPRINT CHR$(A(X,Y));
690 NEXT Y
700 FOR Y=19 TO 2 STEP-1
710 LPRINT CHR$(A(X,Y));
720 NEXT Y

```

```

730 LPRINT:LPRINT" ";
740 NEXT X
750 FOR X=9 TO 2 STEP-1
760 FOR Y=2 TO 19
770 LPRINT CHR$(A(X,Y));
780 NEXT Y
790 FOR Y=19 TO 2 STEP-1
800 LPRINT CHR$(A(X,Y));
810 NEXT Y
820 LPRINT:LPRINT" ";
830 NEXT X
840 LPRINT
850 RETURN

```

Plakat

Wie Sie aus dem Demonstrationslauf sehen können, nimmt **Plakat** eine Mitteilung jeder Länge, natürlich nur bis zu der Ihnen von Ihrem System vorgegebenen Maximallänge einer Variablen an, und verwandelt sie in eine der Länge nach auf das Druckerpapier gedruckte Mitteilung. Sie können die gedruckten Buchstaben dann zur Dekoration an Ihren Wänden verwenden, sie ausschneiden und Posters damit bekleben, oder was Ihnen Ihre Vorstellungskraft sonst noch vorschlägt damit anstellen.

Sie brauchen keine weiteren Anweisungen, um großartige Ergebnisse wie die folgenden zu erhalten:


```

":RETURN
210 LPRINT" GGGGG":LPRINT"G      G":LPRIN
T"G      G":LPRINT"G G      G":LPRINT" GG      G":
RETURN
220 LPRINT"HHHHHHH":LPRINT"      H":LPRINT"
H":LPRINT"      H":LPRINT"HHHHHHH":RETURN
230 LPRINT:LPRINT"I      I":LPRINT"IIIIII
I":LPRINT"I      I":LPRINT:RETURN
240 LPRINT:LPRINT"JJJ":LPRINT"J      J":L
PRINT"J      J":LPRINT"JJJJJJ":RETURN
250 LPRINT"KKKKKKK":LPRINT"      K":LPRINT"
K K":LPRINT" K      K":LPRINT"K      K":RETU
RN
260 LPRINT"LLLLLLL":FOR Z=1 TO 4:LPRINT"
L":NEXT Z:RETURN
270 LPRINT"MMMMMMM":LPRINT"      M":LPRIN
T"      MM":LPRINT"      M":LPRINT"MMMMMMM":RE
TURN
280 LPRINT"NNNNNNN":LPRINT"      NN":LPRIN
T"      NN":LPRINT" NN":LPRINT"NNNNNNN":RETU
RN
290 LPRINT" OOOOO":LPRINT"O      O":LPRIN
T"O      O":LPRINT"O      O":LPRINT" OOOOO":
RETURN
300 LPRINT"PPPPPP":LPRINT"      P      P":LPRIN
T"      P      P":LPRINT"      P      P":LPRINT"      PP":
RETURN
310 LPRINT" QQQQQ":LPRINT"Q      Q":LPRIN
T"Q      Q      Q":LPRINT"QQ      Q":LPRINT" QQQQQ":
RETURN
320 LPRINT"RRRRRR":LPRINT"      R      R":LPRIN
T"      RR      R":LPRINT"      R      R":LPRINT"R      RR":
RETURN
330 LPRINT" S SS":LPRINT"S      S      S":LPRIN
T"S      S      S":LPRINT"S      S      S":LPRINT" SS      S":
RETURN
340 LPRINT"      T":LPRINT"      T":LPRIN
T"TTTTTTT":LPRINT"      T":LPRINT"      T
":RETURN

```

```

350 LPRINT" UUUUUU":LPRINT"U":LPRINT"U":
LPRINT"U":LPRINT" UUUUUU":RETURN
360 LPRINT"      VVVV":LPRINT"      VV":LPRINT"V
":LPRINT"      VV":LPRINT"      VVVV":RETURN
370 LPRINT" WWWWWW":LPRINT"W":LPRINT"WWW
WW":LPRINT"W":LPRINT" WWWWWW":RETURN
380 LPRINT"X      X":LPRINT"      XX      XX":LPRIN
T"      X":LPRINT"      XX      XX":LPRINT"X      X":RET
URN
390 LPRINT"      YY":LPRINT"      Y":LPRINT
"YYYY":LPRINT"      Y":LPRINT"      YY":RETUR
N
400 LPRINT"ZZ      Z":LPRINT"Z      Z      Z":LPRIN
T"Z      Z      Z":LPRINT"Z      Z      Z":LPRINT"Z      ZZ
":RETURN
410 LPRINT"SS":LPRINT"SS":LPRINT:LPRINT:
LPRINT:RETURN
420 LPRINT"SS      SSSS":LPRINT"SS      SSSS":LPRIN
T:LPRINT:LPRINT:RETURN
430 LPRINT:LPRINT:RETURN

```

Weltallspiele

Gutes altes Weltall, die letzten Grenzen. Keine Sammlung von Computerspielen wäre komplett ohne Weltallspiele. Leider verwenden die meisten die speziellen Graphikeigenschaften der einzelnen Computer, um Bewegung und Umfeld darzustellen. Natürlich wollten wir diese Spielzusammenstellung so allgemein wie möglich halten. Daher konnten wir die Verwendung eines bestimmten Computers und Bildschirms nicht voraussetzen. PEEKs und POKEs gehören zu den am wenigsten abänderbaren Teilen jedes Programmes.

Daher beschränken sich die Weltallspiele in dieser Sammlung entweder nur auf eine Textausgabe oder sie verwenden eine symbolische Graphikausgabe (z.B. mit [*] für das Raumschiff). Sogar diese sehr beschränkte Methode funktioniert überraschend gut. Haben Sie das Programm einmal in Ihrem Computer, so sollte es nicht zu schwierig sein und auch nicht zu lang dauern, bis Sie die Ausgabe an die ganz speziellen Fähigkeiten Ihres Computers angepaßt haben.

Mondlandung I und II

Von unserem Mondlandungsprogramm gibt es 2 Versionen. Ich schlage vor, daß Sie mit dem einfacheren der beiden, einfach **Mondlandung** genannt, beginnen. Sind Sie dann damit vertraut und landen Sie schon zwei- von dreimal erfolgreich auf dem Mond, dann können Sie mit der **Echtzeitlandung** Ihre Meisterprüfung ablegen. Dieses Programm verwendet INKEY\$, um die Tastatur zu kontrollieren. Es erfordert schnelles Denken, um das Schiff vor einem Unglück zu bewahren.

Die Aufgabenstellung dieser beiden Programme dürfte ziemlich bekannt sein. Sie landen auf dem Mond und haben nur begrenzte Treibstoffreserven. Aus irgendwelchen Gründen ist Ihr Bordcomputer unfähig, Ihnen andere Daten als den Stand des Fluges zu übermitteln. Die Information, die Sie bekommen, ist korrekt, aber beschränkt.

Die Daten, die Ihnen übermittelt werden, sind die Höhe über der Mondoberfläche, der Rest in den Kraftstofftanks und die Geschwindigkeit, mit der Sie fallen. Sie müssen die richtige Menge 'Schub' eingeben, um sicherzugehen, daß Sie sanft auf der Oberfläche aufsetzen. Der Treibstoff ist beschränkt. Sie müssen damit gut haushalten, um noch genug für die letzte Flugphase, das sanfte Aufsetzen, zu haben. Eine Liste mit den besten Ergebnissen (Zeilen 370 und 380) erhöht die Attraktivität

des Spieles. So kommen Sie und Ihre Freunde ab und zu auf das Programm zurück, um zu versuchen, eine bessere „Galaktische Wertung“ zu bekommen und einen weniger tiefen Mondkrater zu hinterlassen. Dies ist das Listing für die einfachere Version der beiden Spiele:

```

10 REM Mondlandung
20 BP=-10000
30 REM RANDOMIZE
40 A=-20-INT(RND(1)*60):REM Startgeschw
windigkeit
50 B=1200+INT(RND(1)*380):REM Hoehe
60 C=320+INT(RND(1)*90):REM Treibstoff
70 CLS
80 PRINT:PRINT:PRINT
90 B=INT(B):A=INT(A):C=INT(C)
100 PRINT"Hoehe:";B;TAB(15);"Geschwindig
keit:";A
110 PRINT"Treibstoff:";C
120 FOR Q=1 TO 16-B/100
130 PRINT
140 NEXT Q
150 PRINT TAB(5+RND(1)*3-RND(1)*3);"<*>"
160 FOR Q=16-B/100 TO 16
170 PRINT
180 NEXT Q
190 FOR P=1 TO 1000:NEXT P
200 INPUT"Schub ";S
210 FOR P=1 TO 1000:NEXT P
220 IF S>C THEN S=C
230 C=C-S
240 B=B+A+(S-5)/2
250 A=A+(S-5)/2
260 IF C<1 AND B>100 THEN GOTO300
270 IF ABS(B)<20 AND ABS(A)<15 THEN PRIN
T"Sie sind sicher gelandet":PRINT:PRINT"Br
avo, furchtloser Kapitaen":PUNKTE=C*234:GO
TO340
280 IF B>19 THEN GOTO70
290 IF C>1 THEN GOTO310
300 PRINT"Sie haben keinen Treibstoff me

```

```

hr"
310 PRINT"Ihr Schiff schlug mit ";ABS(A)
;" kmh auf"
320 PRINT"und einen neuen Krater mit ";I
NT(ABS(A*45));" m Tiefe"
330 PUNKTE=100-ABS(A)
340 FOR P=1 TO 1000:NEXT P
350 PRINT:PRINT
360 PRINT"Ihre galaktische Einstufung is
t ";PUNKTE
370 IF PUNKTE>BP THEN BP=PUNKTE
380 PRINT:PRINT"Beste Einstufung bisher:
";BP
390 PRINT"Bitte warten"
400 FOR P=1 TO 5000:NEXT P
410 GOTO40

```

Diese Version ist nur für erfahrene Weltallvagabunden:

```

10 REM Echtzeitlandung
20 BP=-10000
30 REM RANDOMIZE
40 A=0
50 B=1200+INT(RND(1)*380):B1=B:REM Hoeh
e
60 C=720+INT(RND(1)*90):REM Treibstoff
70 CLS
80 PRINT:PRINT:PRINT
90 B=INT(B):A=INT(A):C=INT(C)
100 IF B>B1 THEN B=B1
110 PRINT"Hoehe:";B;TAB(15);"Geschwindig
keit:";A
120 PRINT"Treibstoff:";C
130 FOR Q=1 TO 16-B/100
140 PRINT
150 NEXT Q
160 PRINT TAB(5+RND(1)*3-RND(1)*3);"<*>"
170 FOR Q=16-B/100 TO 16
180 PRINT

```



```

190 NEXT Q
200 S#=INKEY$(10):S=VAL(S#)*1.7
210 IF S>C THEN S=0
220 C=C-S
230 B=B+A+(S-5)/2
240 A=A+(S-5)/2
250 IF C<1 AND B>100 THEN GOTO290
260 IF ABS(B)<20 AND ABS(A)<15 THEN PRINT "Sie sind sicher gelandet":PRINT:PRINT "Bravo, furchtloser Kapitän":PUNKTE=C*234:GOTO330
270 IF B>19 THEN GOTO270
280 IF C>1 THEN GOTO300
290 PRINT "Sie haben keinen Treibstoff mehr"
300 PRINT "Ihr Schiff schlug mit ";ABS(A); " kmh auf"
310 PRINT "und einen neuen Krater mit ";INT(ABS(A*45)); " m Tiefe"
320 PUNKTE=100-ABS(A)
330 FOR P=1 TO 1000:NEXT P
340 PRINT:PRINT
350 PRINT "Ihre galaktische Einstufung ist ";PUNKTE
360 IF PUNKTE>BP THEN BP=PUNKTE
370 PRINT:PRINT "Beste Einstufung bisher: ";BP
380 PRINT "Bitte warten"
390 FOR P=1 TO 5000:NEXT P
400 IF INKEY$(10)<>" THEN GOTO400
410 GOTO40

```

Hyperkrieg

Hyperkrieg ist eine große Weltraumsimulation. Sie patrouillieren in einem würfelförmigen Allsektor mit einer Seitenlänge von 10 Parsec. Der Feind ist die Rasse der Dosznti. Es gibt nur ein Dosznti-Schiff in Ihrem Sektor. Es bewegt sich langsam durch den Sektor, während Sie spielen. Obwohl Sie das Schiff an sich nicht zerstören können, so können Sie doch eine Anzahl von Dosznti töten. Das Ziel des Spieles ist es eigentlich, so viele Fremde wie möglich zu töten, bevor Ihre Energiebänke leer sind. Obwohl es viel, viel mehr Doszntis als Menschen in diesem Sektor gibt, schießen die Fremden eher schlecht. Damit sind Ihre Chancen, einen Haufen von ihnen auszulöschen, bevor Ihre Mission zu Ende geht, ziemlich groß.

Sie können nur dann auf ein feindliches Schiff schießen, wenn Sie weniger als drei Einheiten von dessen Position entfernt sind. Der Versuch, auf Doszntis zu schießen, die außerhalb der Reichweite sind, führt nur zu der Meldung Ihrer Mannschaft „Das Dosznti-Schiff ist nicht in der Reichweite unserer Geschütz“.

Ihnen steht nur ein begrenzter Energievorrat zur Verfügung und jeder Treffer durch das feindliche Raumschiff verkleinert ihn.

Ich werde Ihnen jetzt nicht mehr über das Spiel erzählen, denn das würde den Spaß am Spielen nehmen. Das Verhalten des Alls ändert sich nicht von Spiel zu Spiel. Sie sollten daher die Tricks des Weltraumkrieges bald erlernt haben und so immer bessere Ergebnisse erzielen. Das Spiel wird Ihnen in seinem Verlaufe die Regeln erklären. Der Bildschirm ist so gestaltet, daß die Vorfälle so eindeutig wie möglich dargestellt werden.

Hier ein Ausschnitt aus einem „Hyperkrieg“:

Statusbericht der Schiffsoberkontrolle

Energie in Haupt- und Hilfsbank:49

Sternzeituhr zeigt an:50

>>Galaktische Schiffskoordinaten:2 7 8

>>>Doszintischiffkoordinaten:2 8 1

Das Doszintischiff ist im Osten

vor uns, Sir

Was ist Ihr Befehl, Kapitaen?

N, S, O, W
V(oran), Z(urueck)
H(yperspace)
L(aser)

Dieses Listing verhilft Ihnen, Ihre Computerkonsole in den Bordcomputer Ihres Raumschiffes zu verwandeln, wenn Sie in den 10-Parsec-Würfel vorstoßen, um dem Doszinti-Flaggschiff den Garaus zu machen:

```

10 REM Hyperwar
20 BP=0
30 GOSUB1470
40 GOSUB1200
50 IF L<0 THEN GOT0670
60 GOSUB1720
70 L=L-.25
80 ZE=ZE-1
90 PRINT"Was ist Ihr Befehl, Kapitaen?"
100 PRINT"  N, S, O, W"
110 PRINT"  V(oran),Z(urueck)"
120 PRINT"  H(yperspace)"
130 PRINT"  L(aser)"
140 INPUT Z$
150 IF Z$="N" OR Z$="n" THEN X=X-1
160 IF Z$="S" OR Z$="s" THEN X=X+1
170 IF Z$="O" OR Z$="o" THEN Y=Y+1
180 IF Z$="W" OR Z$="w" THEN Y=Y-1
190 IF Z$="V" OR Z$="v" THEN Z=Z-1
200 IF Z$="Z" OR Z$="z" THEN Z=Z+1
210 IF Z$="L" OR Z$="l" THEN GOSUB490
220 IF Z$="H" OR Z$="h" THEN GOSUB1580
230 IF X<1 THEN X=1
240 IF Y<1 THEN Y=1
250 IF Z<1 THEN Z=1
260 IF X>10 THEN X=10
270 IF Y>10 THEN Y=10
280 IF Z>10 THEN Z=10
290 GOSUB910

```

```

940 PRINT"Alarmstufe ROT! Alarmstufe ROT!"
950 GOSUB1710:IF RND(1)<.25 THEN GOSUB14
80:PRINT"Alarmstufe GELB!":GOSUB1710:RETUR
N
960 PRINT"Doszinti schiessen auf uns, Si
r"
970 GOSUB1710
980 PRINT"Alarmstufe ROT! Alarmstufe ROT!"
990 GOSUB1710
1000 IF RND(1)>.7 THEN GOT01150
1010 PRINT"Doszintischuss hat unser Schif
f"
1020 PRINT"getroffen, Kapitaen"
1030 BE=INT(RND(1)*9)+2
1040 GOSUB1710
1050 PRINT"Oberkontrolle meldet einen"
1060 PRINT"Beschaedigungsgrad von ";BE
1070 PRINT"des ";
1080 IF BE=2 THEN PRINT"Bugteiles, Sir"
1090 IF BE>2 AND BE<4 THEN PRINT"Manscha
ftssektors, Kapitaen"
1100 IF BE>3 AND BE<7 THEN PRINT"Hauptant
riebes, Sir"
1110 IF BE>6 THEN PRINT"Energielagerraums
, Kapitaen"
1120 L=L-BE
1130 GOSUB1680
1140 RETURN
1150 GOSUB1710
1160 PRINT"Der Doszintischuss hat unser"
1170 PRINT"Schiff verfehlt, Kapitaen"
1180 GOSUB1710
1190 RETURN
1200 REM Statusbericht
1210 CLS
1220 PRINT"*****"
*****

```

```

1230 PRINT:PRINT"Statusbericht der Schiff
soberkontrolle"
1240 GOSUB1710
1250 PRINT"Energie in Haupt- und Hilfsban
k:";L
1260 IF L<3 THEN PRINT"Warnung...Energies
piegel ist":PRINT"gefaehrlich niedrig!"
1270 PRINT"Sternzeituhr zeigt an:";ZE
1280 IF ZE<8 THEN PRINT"!!!!Warnung - Mi
ssionszeit wird knapp!!!!!"
1290 IF ZE<1 THEN GOTO680
1300 PRINT"*****"
1310 IF T>0 THEN PRINT:PRINT"Zerstoeungs
punkte:";T
1320 PRINT">>Galaktische Schiffskoordina
ten:";X;" ";Y;" ";Z
1330 PRINT:PRINT">>>Doszintischiffkoordi
naten:";A;" ";B;" ";C
1340 PRINT:PRINT"Das Doszintischiff ist "
;
1350 IF A<>X OR B<>Y THEN PRINT"im";
1360 IF A<X THEN PRINT" Norden";
1370 IF A>X THEN PRINT" Sueden";
1380 IF B>Y THEN PRINT" Osten";
1390 IF B<Y THEN PRINT" Westen";
1400 PRINT
1410 IF C=Z THEN PRINT"unseres Schiffes,
Sir"
1420 IF C>Z THEN PRINT"hinter uns, Kapita
en"
1430 IF C<Z THEN PRINT"vor uns, Sir"
1440 PRINT"*****"
1450 RETURN
1460 REM *****
*
1470 REM Initialisieren
1480 CLS

```

```

1490 REM RANDOMIZE
1500 L=35+INT(RND(1)*30):T=0:ZE=50
1510 A=INT(RND(1)*10)+1
1520 B=INT(RND(1)*10)+1
1530 C=INT(RND(1)*10)+1
1540 X=INT(RND(1)*10)+1
1550 Y=INT(RND(1)*10)+1
1560 Z=INT(RND(1)*10)+1
1570 RETURN
1580 REM Hyperspace
1590 X=INT(RND(1)*10)+1
1600 Y=INT(RND(1)*10)+1
1610 Z=INT(RND(1)*10)+1
1620 FOR J=1 TO 40
1630 PRINT TAB(J);"*":PRINT
1640 FOR H=1 TO J:NEXT H
1650 NEXT J
1660 CLS
1670 RETURN
1680 FOR P=1 TO 2000:NEXT P
1690 CLS
1700 GOTO1720
1710 FOR P=1 TO 2000:NEXT P
1720 PRINT:PRINT
1730 RETURN

```

Gehirnspiele

Ich schätze, 90% aller Spiele könnten ‚Gehirnspiele‘ genannt werden, ohne daß die Definition zu weit ausgedehnt würde. Sie verlangen ja schließlich von Ihnen eine gewisse Verstandesstärke und Anpassungsfähigkeit.

Die Programme in diesem Kapitel unter der Überschrift ‚Gehirnspiele‘ sind deshalb hier, weil sie etwas mehr intellektuelle Anstrengung erfordern als einige andere Programme in diesem Buch.

Hier finden Sie **Fastermind**, wo Sie einen vierstelligen, vom Computer aufgestellten Code herausfinden müssen, oder Sie stellen Ihre mathematischen Fähigkeiten auf die Probe, wenn Sie die verzauberten Sterne in **Idaho-Sterne** zu lösen versuchen. In **Henker** geben Sie alles daran, ein vom Computer gewähltes Wort zu erraten, damit Sie Ihren Hals retten können.

Genauso werden Sie in **Kubist** versuchen, einen verdrehten Würfel wieder zu ordnen, oder in **Murmelerde Marmor** ein paar ruhige Minuten finden (allgemein ist dieses Spiel unter **Solitaire** bekannt, wurde hier jedoch wegen der Murmeln, die es im „wirkliche Leben“ verwendet, umbenannt), oder aber unter dem Druck einer ruhelosen Uhr versuchen, Atome in einem Zyklotron aufzustöbern.

Hat Ihr Gehirn nach all dem noch etwas Schmalz übrig, so können Sie in **Flipper** eine Aufgabe übernehmen, die unmöglich zu lösen erscheint. Oder Sie versuchen, einen Algorithmus auszuarbeiten, um die Probleme in **Vertauschereien** mit den wenigst möglichen Zügen zu lösen. Nachdem Ihr Verstand durch all diese Programme in diesem Kapitel geschärft wurde, versuchen Sie in **Suche nach Daryl**, eine Karte eines Höhlensystems zu erstellen. Erwarten Sie nicht, daß dieses leicht ist. Konzentration und Geduld sind Voraussetzung, um es zu lösen.

Fastermind

Wir beginnen mit etwas Einfachem, **Fastermind**. Sie kennen diese Art von Spiel wahrscheinlich schon. Sie müssen einen Code, der von Ihrem Gegenspieler entwickelt wurde, knacken.

In diesem Programm erstellt der Computer einen vierstelligen Code. Dazu verwendet er die Zahlen 1-9 und nimmt aber nie zweimal die gleiche Zahl innerhalb des Codes. Nun haben Sie acht Versuche, diesen Code zu erraten.

Jede Vermutung erhält eine Rückmeldung in der Form von 'schwarz' oder 'weiß', wobei Sie ein 'schwarz' für eine erratene Nummer an ihrer richtigen Stelle erhalten und ein 'weiß' für eine erratene Nummer, aber an der falschen Stelle.

Nachfolgend ein Durchgang:

Druecken Sie eine Taste
Fastermind

Wenn Sie aufgefordert werden, geben
Sie eine 4-stellige Zahl ein und
druecken Sie RETURN.

Ziffern koennen wiederholt werden.

Sie haben 8 Versuche, um
den Code zu brechen.

Geben Sie Versuch Nr.1 ein:
?1234
Weiss

Geben Sie Versuch Nr.2 ein:
?5678
Weiss

Geben Sie Versuch Nr.3 ein:
?9009
Schwarz

Geben Sie Versuch Nr.4 ein:

?9156
Weiss Weiss

Geben Sie Versuch Nr.5 ein:

?9178
Weiss

Geben Sie Versuch Nr.6 ein:

?3478
Schwarz

Geben Sie Versuch Nr.7 ein:

?4477
Schwarz

Geben Sie Versuch Nr.8 ein:

?4433
Schwarz

Sie haben es nicht geschafft...
Die Loesung ist 6469

Und hier das Listing, damit Ihr Gehirn mit Fastermind endlich etwas zu arbeiten bekommt:

```
10 REM Fastermind
20 CLS
30 N=1
40 PRINT"Druecken Sie eine Taste"
50 N=N+1
60 IF INKEY$(10)="" THEN GOT050
70 REM RANDOMIZE N
80 CLS
90 PRINT"Fastermind"
100 PRINT
110 PRINT"Wenn Sie aufgefordert werden,
geben"
120 PRINT"Sie eine 4-stellige Zahl ein u
```

```

nd"
130 PRINT"druecken Sie RETURN."
140 PRINT
150 PRINT"Ziffern koennen wiederholt wer-
den."
160 PRINT
170 PRINT"Sie haben 8 Versuche, um"
180 PRINT"den Code zu brechen."
190 FOR Z=1 TO 6000:NEXT Z
200 CLS
210 DIM B(4),D(4)
220 H=0
230 FOR A=1 TO 4
240 B(A)=INT(RND(1)*9)+1
250 NEXT A
260 FOR C=1 TO 8
270 PRINT
280 PRINT"Geben Sie Versuch Nr.;"C;" ein
:"
290 INPUT X
300 IF X>9999 OR X<1000 THEN GOTO280
310 D(1)=INT(X/1000)
320 D(2)=INT((X-1000*D(1))/100)
330 D(3)=INT((X-1000*D(1)-100*D(2))/10)
340 D(4)=INT(X-1000*D(1)-100*D(2)-10*D(3))
350 FOR E=1 TO 4
360 IF D(E)<>B(E) THEN GOTO410
370 PRINT" Schwarz ";
380 B(E)=B(E)+10
390 D(E)=D(E)+20
400 H=H+1
410 NEXT E
420 IF H=4 THEN GOTO620
430 FOR F=1 TO 4
440 D=D(F)
450 FOR G=1 TO 4
460 IF D<>B(G) THEN GOTO500
470 PRINT" Weiss ";

```

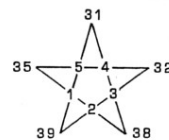
```

480 B(G)=B(G)+10
490 GOTO510
500 NEXT G
510 NEXT F
520 FOR G=1 TO 4
530 IF B(G)<10 THEN GOTO550
540 B(G)=B(G)-10
550 NEXT G
560 H=0
570 PRINT
580 NEXT C
590 PRINT:PRINT"Sie haben es nicht gesch-
afft..."
600 PRINT"Die Loesung ist ";B(1);B(2);B(
3);B(4)
610 END
620 PRINT:PRINT"Bravo, Superhirn!"
630 PRINT
640 PRINT:PRINT"Sie haben die Loesung in
nur"
650 PRINT ;C;" Versuchen gefunden"

```

Idaho-Sterne

Idaho-Sterne sind Sterne mit einer besonderen Eigenschaft. Sie haben nämlich Zahlen an den Schnittpunkten der sie bildenden Linien, die summiert immer denselben Wert ergeben, unabhängig von der ausgewählten Linie. Hier ein Beispiel eines solchen Sternes:



Wie Sie sehen, ergibt jede Linie des Sternes die gleiche Summe. In diesem Programm erzeugt der Computer die Idaho-Sterne, läßt aber 2 oder 3 Zahlen als Nullen, damit Sie diese mit so wenig Versuchen wie möglich herausfinden.
Hier das Programm in Aktion:

```

      0
    7  10  0  0
      1    9
        6
      5    4
  
```

Das ist Durchgang Nr.1

Geben Sie eine Zahl ein, von der Sie glauben, dass sie Teil des Sternes ist ?-2

```

      0
    7  10  0  -2
      1    9
        6
      5    4
  
```

Sie haben bis jetzt 8 Punkte

Das ist Durchgang Nr.2

Geben Sie eine Zahl ein, von der Sie glauben, dass sie Teil des Sternes ist ?3

```

      0
    7  10  3  -2
      1    9
        6
      5    4
  
```

Sie haben bis jetzt 9 Punkte

Das ist Durchgang Nr.3

Geben Sie eine Zahl ein, von der Sie glauben, dass sie Teil des Sternes ist ?2

```

      2
    7  10  3  -2
      1    9
        6
      5    4
  
```

Jawohl, Sie haben es geschafft!

Und in nur 3 Durchgaengen...
Druecken Sie 'J', wenn Sie einen
weiteren Idahostern versuchen wollen,
wollen, 'N' fuer Ende.....

OK, danke fuer das Spiel

Dies ist das Listing des Programmes, damit Sie Ihre eigenen Idaho-Sterne machen können. Ändern Sie die 3 in Zeile 630 auf eine höhere Zahl, so erhalten Sie schwieriger zu lösende Sterne:

```

10 REM Idahostern
20 GOSUB480:REM Stern errechnen
30 GOSUB340:REM Stern ausdrucken
40 GOSUB70:REM Versuch abfragen
50 GOTO30
60 REM *****
70 REM Versuch abfragen
80 IF PUNKTE>0 AND PUNKTE<10 THEN PRINT
  "Sie haben bis jetzt ";PUNKTE;" Punkte":PR
  INT
  90 DUR=DUR+1
  100 PRINT"Das ist Durchgang Nr.";DUR:PRI
  NT
  110 PRINT"Geben Sie eine Zahl ein, von d
  er"
  120 PRINT"Sie glauben, dass sie Teil des
  "
  125 INPUT"Sternes ist ";G
  130 PUNKTE=0
  140 FOR J=1 TO 10
  150 IF G=A(J) THEN B(J)=A(J)
  160 IF B(J)<>0 THEN PUNKTE=PUNKTE+1
  170 NEXT J
  180 FOR P=1 TO 2000:NEXT P
  190 IF PUNKTE<10 THEN RETURN
  200 REM *****

```

```

210 GOSUB340
220 PRINT:PRINT"Jawohl, Sie haben es ges
  chafft!"
230 PRINT:PRINT"Und in nur ";DUR;" Durch
  gaengen..."
240 FOR P=1 TO 2000:NEXT P
250 PRINT"Druecken Sie 'J', wenn Sie ein
  en"
260 PRINT"weiteren Idahostern versuchen
  wollen,"
270 PRINT"wollen, 'N' fuer Ende.....
  ."
280 A$=INKEY$(10)
290 IF A$<>"N" AND A$<>"n" AND A$<>"J" A
  ND A$<>"j" THEN GOTO280
300 IF A$="J" OR A$="j" THEN RUN
310 PRINT:PRINT"OK, danke fuer das Spiel
  "
320 END
330 REM *****
340 REM Stern ausdrucken
350 CLS:PRINT:PRINT
360 PRINT TAB(11);B(1)
370 PRINT:PRINT
380 PRINT TAB(4);B(2);" ";B(3);" ";B
  (4);" ";B(5)
390 PRINT
400 PRINT TAB(7);B(6);" ";B(7)
410 PRINT
420 PRINT TAB(10);B(8)
430 PRINT
440 PRINT TAB(4);B(9);" ";B(10)
450 PRINT:PRINT
460 RETURN
470 REM *****
480 REM Stern berechnen
490 CLS
500 REM RANDOMIZE
510 DIM A(10),B(10)

```



```

530 DUR=0:PUNKTE=0
540 A=INT(RND(1)*9)+1
550 B=INT(RND(1)*9)+1
560 C=INT(RND(1)*9)+1
570 D=INT(RND(1)*9)+1
580 E=INT(RND(1)*9)+1
590 IF A=B OR A=C OR A=D OR A=E OR B=C O
R B=D OR B=E OR C=D OR C=E OR D=E THEN GOT
0550
630 X=INT(RND(1)*3)+1
640 A(1)=X
650 A(2)=X-B+C+D
660 A(3)=A+E
670 A(4)=A+D
680 A(5)=X-B-C+E
690 A(6)=A
700 A(7)=A+C
710 A(8)=A+B
720 A(9)=X-2*B+2*D+E
730 A(10)=X-2*B-C+D+2*E
740 FOR J=1 TO 10
750 B(J)=A(J)
760 IF A(J)=0 THEN RUN
770 NEXT J
780 B(INT(RND(1)*10)+1)=0
790 B(INT(RND(1)*10)+1)=0
800 B(INT(RND(1)*10)+1)=0
810 RETURN

```

Henker

Wie Sie sicher erkennen, ist dies eine Computervariation des alten, auf Papier gespielten Spieles **Henker**. Der Computer wählt in diesem Spiel ein Wort aus seinem Speicher (damit meine ich die DATA-Anweisungen von 490 bis 590), und es liegt dann an Ihnen, es zu erraten. Die Anzahl der Versuche, die Ihnen zugestanden werden, hängen von der Länge des Wortes ab (siehe Zeilen 120 und 170). Sehen wir uns nun einen hilflosen Menschen (mich) an, der versucht, das vom Computer gewählte Wort zu erraten:

Druecken Sie 'H', wenn Sie bereit sind, dem Henker gegenueberzutreten

Sie muessen das Wort des Henkers in nur 12 Versuchen erraten

.....

Sie haben 0 richtige Buchstaben

Noch 12 Versuche...

Ihr naechster Versuch ?HHHHHA

.....A.

Sie haben 1 richtige Buchstaben

Noch 12 Versuche...

Ihr naechster Versuch ?E

.E....A.

Sie haben 2 richtige Buchstaben

Noch 12 Versuche...

Ihr naechster Versuch ?S

.E....A.

Sie haben 2 richtige Buchstaben

Noch 11 Versuche...

Ihr naechster Versuch ?K

.E....A.

Sie haben 2 richtige Buchstaben

Noch 10 Versuche...

Ihr naechster Versuch ?M

ME....A.

Sie haben 3 richtige Buchstaben

Noch 10 Versuche...

Ihr naechster Versuch ?N

ME....AN

Sie haben 4 richtige Buchstaben

Noch 10 Versuche...

Ihr naechster Versuch ?R

MER...AN

Sie haben 5 richtige Buchstaben

Noch 10 Versuche...

Ihr naechster Versuch ?I

MERI.IAN

Sie haben 7 richtige Buchstaben

Noch 11 Versuche...

Ihr naechster Versuch ?D

MERIDIAN

Wau! Sie haben die Exekution
wieder einmal aufgeschoben
Sie haben es in 9 Versuchen geschafft

Das Wort des Henkers war MERIDIAN

Das Listing wird es Ihnen ermöglichen, Ihren Hals für die Vergrößerung
Ihres Wortschatzes zu riskieren:

```

10 REM Henker
20 CLS
30 N=0:Y=0
40 PRINT"Druecken Sie 'H', wenn Sie ber
eit"
50 PRINT"sind, dem Henker gegenueberzut
reten"
60 N=N+1:IF INKEY$(10)<>"H" AND INKEY$(
10)<>"h" THEN GOTO60
70 REM RANDOMIZE N
80 FOR G=1 TO RND(1)*22+1
90 READ A$
100 NEXT G
110 GOSUB470
120 N=LEN(A$):DIM B(N),D(N)
130 FOR G=1 TO N
140 B(G)=ASC(MID$(A$,G,1))
150 D(G)=B(G)
160 NEXT G
170 Q=INT(N+N/2+.5)

```

```

180 CLS:PRINT:PRINT"Sie muessen das Wort
des Henkers in"
190 PRINT"nur ";Q;" Versuchen erraten"
200 GOSUB470
210 FOR J=1 TO Q:Y=Y+1
220 GOSUB390
230 IF H=N THEN GOTO340
240 PRINT:PRINT:PRINT"Noch ";Q+1-J;" Ver
suche..."
250 PRINT:INPUT"Ihr naechster Versuch ";
C$
260 F=ASC(C$)
270 FOR G=1 TO N
280 IF D(G)=F THEN D(G)=0:J=J-1
290 NEXT G:NEXT J
300 GOSUB470
310 GOSUB390
320 PRINT:PRINT:Tut mir leid, aber
Sie muessen nun gehen!"
330 GOTO370
340 PRINT:PRINT"Wau! Sie haben die Exeku
tion"
350 PRINT"wieder einmal aufgeschoben"
360 PRINT"Sie haben es in ";Y-1;" Versuc
hen geschafft"
370 PRINT:PRINT"Das Wort des Henkers war
";A$
380 END
390 H=0:CLS:PRINT:PRINT
400 FOR E=1 TO N
410 IF B(E)=D(E) THEN PRINT".";
420 IF B(E)<>D(E) THEN PRINT CHR$(B(E));
:H=H+1
430 NEXT E
440 PRINT:PRINT:IF H<>N THEN PRINT"Sie h
aben ";H;" richtige Buchstaben"
450 PRINT
460 RETURN
470 FOR P=1 TO 1000:NEXT P

```

```

480 RETURN
490 DATA "MERIDIAN", "VERDIENST", "SEEJUNG
FER", "LUSTBARKEIT"
500 DATA "AUFSEHER", "OXYDATIONSMITTEL", "
SAUERSTOFF", "HANDGREIFLICH", "UNORTHODOX"
510 DATA "INFERNO", "LOBREDE", "PARADOX", "
FASAN"
520 DATA "KRAWALL", "AUSSCHUSS", "SAKRAMEN
T", "KAVALLERIST", "SCHEMATISCH"
530 DATA "SEDIMENT", "TATTERGREIS", "ENTHA
LTSAM", "TELESKOP"

```

Kubist

Das ist eine zweidimensionale Version des Spieles mit dem farbigen Würfel, das die ganze Welt von Ungarn aus eroberte. Am Beginn sieht der Würfel so aus:

```

1 1 2 2
1 1 2 2
3 3 4 4

```

Nach einigen Veränderungen durch den Computer sieht er etwa so aus:

```

1 2 1 2
3 3 3 2
3 1 4 1
4 2 4 4

```

Ihre Aufgabe ist, ihn mit so wenig Zügen wie möglich in sein Ausgangsstadium zurückzuführen. Die Zahlen auf dem Würfel rotieren in Vierergruppen im Uhrzeigersinn.

Das bedeutet, daß $\begin{matrix} 1 & 2 & & 3 & 1 \\ & & & & 4 \end{matrix}$, nachdem es rotierte, so aussieht $\begin{matrix} & & & & 3 & 1 \\ & & & & 4 & 2 \end{matrix}$.

Um die Elemente des Würfels zu bewegen, geben Sie eine Zahl ein. Der Schlüssel dazu ist wie folgt:

```

2 3 4
6 7 8
10 11 12

```

Doch sehen wir uns die Auswirkungen einiger Drehungen an dem schon vorher gezeigten Würfel an:

Geben Sie Ihre Wahl ein (2-12), aber ohne 5 und 9? 6

```

1 2 1 2
3 3 3 2
1 3 4 1
4 2 4 4

```

Ihre Würfeldrehung 1

Geben Sie Ihre Wahl ein (2-12), aber ohne 5 und 9? 6

```

1 2 1 2
1 3 3 2
3 3 4 1
4 2 4 4

```

Ihre Würfeldrehung 2

Das ist das Listing für Kubist:

```

10 REM Wuerfel
20 DIM A(4),B(16)
30 REM RANDOMIZE
40 CLS
50 B=1:F=0
60 FOR D=1 TO 4
70 C=0
80 IF B=1 THEN C=1
90 IF B=3 THEN C=2
100 IF B=9 THEN C=3
110 IF B=11 THEN C=4
120 B(B)=C
130 B(B+1)=C
140 B(B+4)=C
150 B(B+5)=C
160 C=0
170 IF B=1 THEN C=2
180 IF B=3 THEN C=6
190 IF B=9 THEN C=2
200 B=B+C
210 NEXT D
220 GOSUB410
230 PRINT:PRINT

```

```

240 IF F<11 THEN PRINT"Ich drehe..."
250 F=F+1
260 FOR P=1 TO 1800:NEXT P
270 IF F<11 THEN X=INT(RND(1)*12)+1
280 IF F>10 THEN PRINT"Geben Sie Ihre Wah-
    l ein (2-12)":INPUT"aber ohne 5 oder 9 ";
    X
290 IF X<2 OR X=5 OR X=9 OR X>12 THEN GO
    TO270
300 A(1)=B(X)
310 A(2)=B(X+4)
320 A(3)=B(X+3)
330 A(4)=B(X-1)
340 B(X)=A(4)
350 B(X+4)=A(1)
360 B(X+3)=A(2)
370 B(X-1)=A(3)
380 GOSUB410
390 F=F+1
400 GOTO270
410 CLS
420 PRINT:PRINT:PRINT
430 FOR B=1 TO 16
440 PRINT ;B(B);
450 IF B/4=INT(B/4) THEN PRINT
460 NEXT B
470 PRINT:PRINT
480 IF F<10 THEN PRINT:PRINT"Drehung Nr.
";F
490 IF F>10 THEN PRINT"Ihre Wuerfeldrehu-
ng ";F-10:PRINT
500 IF F<11 THEN RETURN
510 P=0
520 IF B(1)=1 AND B(2)=1 AND B(3)=2 AND
    B(4)=2 AND B(5)=1 AND B(6)=1 AND B(7)=2 AN
    D B(8)=2 THEN P=P+1
530 IF B(9)=3 AND B(10)=3 AND B(11)=4 AN
    D B(12)=4 AND B(13)=3 AND B(14)=3 AND B(15
    )=4 AND B(16)=4 THEN P=P+1

```

```

540 IF P<>2 THEN RETURN
550 PRINT"Sie haben es in ";F-10;" Drehu-
ngen geschafft"
560 PRINT"Bravo, Wuerfler!"

```

Murmelnder Marmor

Unser nächstes Gehirnspiel ist eine Version des Spieles, bei dem man Murmeln auf einem Brett bewegt und das Sie wahrscheinlich unter dem Namen *Solitaire* kennen. Am Anfang des Spieles ist nur das Loch in der Mitte des Brettes leer. Man kann über Murmeln in leere Löcher springen und nimmt dann die übersprungenen Murmeln aus dem Spiel. Die Idee des Spieles ist, es nur mit einer Murmel im Loch in der Mitte zu beenden.

Es gibt sehr viele Vermutungen über die Abstammung dieses Spieles. Die interessanteste ist wahrscheinlich jene, die besagt, daß die Idee zu dem Spiel von einem in der Bastille in Einzelhaft sitzenden Gefangenen stammt. Dieser soll damit die Monotonie seiner Lage bekämpft haben. Ob diese Geschichte jetzt wahr ist oder nicht, es gibt keinen Zweifel, daß man richtig süchtig nach diesem Spiel werden kann, wenn man einen idiotensicheren Weg zur Lösung finden will.

So sieht der Bildschirm am Beginn aus:

Geben Sie die Seitenkoordinate zuerst ein

```

99 fuer Aufgeben
1 2 3 4 5 6 7
  0 0 0      1
  0 0 0      2
0 0 0 0 0 0 3
0 0 0 * 0 0 4
0 0 0 0 0 0 5
  0 0 0      6
  0 0 0      7

```

Bisherige Zuege: 0

Einige Züge später könnte er so aussehen:

Welche Murmel wollen Sie bewegen ?
 Geben Sie die Seitenkoordinate zuerst ein
 99 fuer Aufgeben
 1 2 3 4 5 6 7
 0 0 0 1
 0 * 0 2
 0 * 0 0 0 0 3
 0 0 * 0 0 0 0 4
 0 0 0 0 0 0 5
 * 0 0 6
 * 0 0 7

Bisherige Zuege: 4

Es sind 28 am Brett
 Welche Murmel wollen Sie bewegen ?41
 41 nach ?43
 Geben Sie die Seitenkoordinate zuerst ein
 99 fuer Aufgeben
 1 2 3 4 5 6 7
 0 0 0 1
 0 * 0 2
 0 * 0 0 0 0 3
 * * 0 0 0 0 4
 0 0 0 0 0 0 5
 * 0 0 6
 * 0 0 7

Bisherige Zuege: 5

Es sind 27 am Brett

Hier ist das Listing, damit Sie das Problem selbst lösen können:

```

10 REM Murmelnder Marmor
20 GOSUB380
30 GOSUB230
40 REM Zug annehmen
50 PRINT"Welche Murmel wollen Sie beweg
en ";
60 INPUT A
70 IF A=99 THEN GOTO220
80 IF A<11 OR A>77 OR A(A)<>79 THEN GOT
350
90 PRINT A;" nach ";
100 INPUT B
110 IF B<11 OR B>77 OR A(B)<>L THEN GOT
0100
120 A((A+B)/2)=L:A(A)=L:A(B)=79
130 ZUG=ZUG+1
140 ZAEHLER=0
150 FOR F=11 TO 75
160 IF A(F)=79 THEN ZAEHLER=ZAEHLER+1
170 NEXT F
180 GOSUB230
190 PRINT"Es sind ";ZAEHLER;" am Brett"
200 IF ZAEHLER<>1 THEN GOTO40
210 IF A(44)=79 THEN PRINT"Sie haben es
in nur ";ZUG;" Zuegen geschafft!":END
220 PRINT"Das Spiel ist vorbei und Sie h
aben versagt!":END
230 REM Ausdruck
240 CLS:REM Ersetzen Sie dies durch HOME
, wenn Ihr Computer es hat
250 PRINT"Geben Sie die Seitenkoordinate
zuerst ein"
260 PRINT TAB(5);"99 fuer Aufgeben"
270 PRINT"      1 2 3 4 5 6 7"
280 PRINT TAB(5);
290 FOR D=11 TO 75
300 T=10*(INT(D/10))
310 IF D-T=8 THEN D=D+2:PRINT ;T/10:PRIN

```

```

T TAB(5)::GOTO330
320 PRINT CHR$(A(D));" ";
330 NEXT D:PRINT" 7"
340 PRINT:PRINT:PRINT
350 PRINT"Bisherige Zuege: ";ZUG
360 PRINT:PRINT
370 RETURN
380 REM Initialisieren
390 CLS
400 DIM A(87)
410 L=42
420 FOR D=11 TO 75
430 T=10*(INT(D/10))
440 IF D-T=8 THEN D=D+3
450 READ A(D)
460 NEXT D
470 ZUG=0
480 RETURN
490 REM 42 ist ASC("#")
500 REM 79 ist ASC("O")
510 DATA 32,32,79,79,79,32,32
520 DATA 32,32,79,79,79,32,32
530 DATA 79,79,79,79,79,79,79
540 DATA 79,79,79,42,79,79,79
550 DATA 79,79,79,79,79,79,79
560 DATA 32,32,79,79,79,32,32
570 DATA 32,32,79,79,79

```

Zyklotron

In **Zyklotron** werfen Sie durch das Nennen einer Zahl Atome von oben in eine Maschine. Sie müssen danach versuchen, aus dem Verhalten der Atome die in der Maschine versteckten Antimateriepartikel zu finden.

Trifft das Atom so ein Antimateriepartikel, wird es entweder absorbiert oder nach rechts oder links abgelenkt. Die Auflösung des Problems wird noch dadurch erschwert, daß ein Atom durch mehrere Antimateriepartikel beeinflußt werden kann.

Es sind immer drei Antimateriepartikel in der Maschine. Sie könnten auch alle in einem Zyklotron stecken. Ihnen steht nur eine sehr begrenzte Anzahl von Versuchen zur Verfügung. Bei jedem Versuch geben Sie entweder die Nummer an, wo Sie das Atom einwerfen wollen, oder Sie sind waghalsig und raten die Position der Antimateriepartikel durch die Eingabe von 8.

So sieht der Programmablauf aus:

Sie haben 6 Sekunden
Geben Sie die Zyklotronzahl ein,
oder 8, wenn Sie zu wissen glauben,
wo die Antimaterie liegt
??

```

      2
  - - - - -
  - - - - -
  - - - - -
  - - - - -
  - - - - -
      2

```

Das Atom kam bei 2 heraus

Sie haben 5 Sekunden
Geben Sie die Zyklotronzahl ein,
oder 8, wenn Sie zu wissen glauben,
wo die Antimaterie liegt
??

```

      3
  - - - - -
  - - - - -
  - - - - -
  - - - - -
  - - - - -
      3

```

Das Atom kam bei 3 heraus

Sie haben 4 Sekunden
Geben Sie die Zyklotronzahl ein,
oder 8, wenn Sie zu wissen glauben,
wo die Antimaterie liegt
?5

```

      5
  - - - - -
  - - - - -
  - - - - -
  - - - - -
  - - - - -
  - - - - -
  
```

Das Atom wurde absorbiert

Sie haben 3 Sekunden
Geben Sie die Zyklotronzahl ein,
oder 8, wenn Sie zu wissen glauben,
wo die Antimaterie liegt
?7

```

      7
  - - - - -
  - - - - -
  - - - - -
  - - - - -
  - - - - -
  - - - - -
  
```

Das Atom kam bei 7 heraus

Sie haben 2 Sekunden
Geben Sie die Zyklotronzahl ein,
oder 8, wenn Sie zu wissen glauben,
wo die Antimaterie liegt
?1

```

  1
  - - - - -
  - - - - -
  - - - - -
  - - - - -
  - - - - -
  - - - - -
  
```

Das Atom kam bei 1 heraus

Sie haben 1 Sekunden
Geben Sie die Zyklotronzahl ein,
oder 8, wenn Sie zu wissen glauben,
wo die Antimaterie liegt
?8
Wo glauben Sie liegt
die Antimaterie?
?4,5,6
Bravo. Sie haben die Antimaterie
mit 1 Sekunden Zeitvorsprung
lokalisiert

Ihre Bestzeit bisher ist 5

Und hier das Listing für Ihr eigenes Zyklotron:

```

10 REM Zyklotron
20 CLS
30 N=0:BP=9:DIM B(7),Q(3),W(3)
40 PRINT"Druecken Sie eine Taste, wenn"
50 PRINT"Sie spielbereit sind"
60 N=N+1:IF INKEY$(10)="" THEN GOTO60
70 REM RANDOMIZE N
80 REM Antimaterie setzen
90 FOR A=1 TO 3
100 B(A)=INT(RND(1)*7)+1
110 NEXT A
120 FOR H=6 TO 1 STEP-1
130 CLS
140 PRINT:PRINT:PRINT"Sie haben ";H;" Se
  
```



```

kunden"
150 PRINT"Geben Sie die Zyklotronzahl ein,"
160 PRINT"oder 8, wenn Sie zu wissen glauben,"
170 PRINT"wo die Antimaterie liegt"
180 INPUT C
190 IF C=8 THEN GOTO520
200 GOSUB450
210 D=1
220 IF B(D)=C THEN GOTO350
230 IF D<3 THEN D=D+1:GOTO220
240 GOSUB500
250 PRINT"Das Atom kam bei ";C;" heraus"
260 FOR Z=1 TO 4000:NEXT Z
270 NEXT H
280 PRINT:PRINT:PRINT
290 PRINT"Tut mir leid, die Zeit ist vor
bei"
300 PRINT:PRINT"Die Antimaterie war bei"
310 FOR A=1 TO 3
320 PRINT B(A);
330 NEXT A
340 END
350 REM Antimateriereaktionen
360 IF RND(1)>.9 THEN GOTO420
370 IF C=1 THEN C=2:GOTO210
380 IF C=7 THEN C=6:GOTO210
390 F=INT(RND(1)*2)-1
400 C=C+F
410 GOTO210
420 REM Antimaterie absorbiert Atom
430 PRINT"Das Atom wurde absorbiert"
440 GOTO260
450 PRINT:PRINT:PRINT TAB(3+2*C);C
460 FOR Z=1 TO 5
470 PRINT TAB(5);"- - - - -"
480 NEXT Z
490 RETURN

```

```

500 PRINT TAB(3+2*C);C
510 RETURN
520 REM Ortungsversuch
530 PRINT"Wo glauben Sie liegt"
540 PRINT"die Antimaterie?"
550 INPUT Q(1),Q(2),Q(3)
560 FOR T=1 TO 3:W(T)=B(T):NEXT T
570 SR=0
580 FOR X=1 TO 3:FOR Y=1 TO 3
590 IF Q(X)=B(Y) THEN SR=SR+1:B(Y)=0:Q(X)
)=0
600 NEXT Y:NEXT X
610 GOTO650
620 FOR P=1 TO 3
630 B(P)=W(P)
640 NEXT P
650 IF SR>2 THEN GOTO700
660 IF SR>0 THEN PRINT"Sie haben ";SR;"
richtig geortet"
670 IF SR=0 THEN PRINT"Sie haben sich to
tal geirrt!"
680 FOR T=1 TO 3:B(T)=W(T):NEXT T
690 GOTO260
700 PRINT"Bravo. Sie haben die Antimater
ie"
710 PRINT"mit ";H;" Sekunden Zeitvorspru
ng"
720 PRINT"lokalisiert"
730 IF 6-H<BP THEN BP=6-H
740 PRINT:PRINT"Ihre Bestzeit bisher ist
";BP
750 FOR Z=1 TO 4000:NEXT Z
760 GOTO90

```

Vertauschereien

In **Vertauschereien** werden Sie mit den Ziffern 1 bis 9 in einer zufälligen Anordnung konfrontiert. Sie müssen diese Ziffern nun mit möglichst wenigen Zügen in die Form 123456789 bringen. Sie geben Ihre Züge als Ziffern ein. Der Computer verwendet dann diese Nummern, um das Vertauschen durchzuführen. Es funktioniert wie folgt:

Bitte warten Sie...

3

39

396

3968

39684

396845

3968451

39684517

396845172

Zug Nr.1

396845172

Wieviele soll ich vertauschen?

Zug Nr.2

327154869

Wieviele soll ich vertauschen?

Zug Nr.3

968451723

Wieviele soll ich vertauschen?

Zug Nr.4

963271548

Wieviele soll ich vertauschen?

Zug Nr.5

984517236

Wieviele soll ich vertauschen?

Zug Nr.6

984516327

Wieviele soll ich vertauschen?

Zug Nr.7

987236154

Wieviele soll ich vertauschen?

Zug Nr.8

987234516

Wieviele soll ich vertauschen?

Zug Nr.9

987615432

Wieviele soll ich vertauschen?

Zug Nr.10

987612345

Wieviele soll ich vertauschen?

Zug Nr.11

987654321

Wieviele soll ich vertauschen?

123456789

Sie haben es geschafft!

Und das in nur 11 Zuegen...

Das ist das Listing des Spieles:

```
10 REM Vertauschereien
20 CLS
30 PRINT:PRINT"Bitte warten Sie..."
40 GOSUB100
50 GOSUB230
60 IF A$="123456789" THEN GOT0400
70 M=M+1
80 GOT050
90 END
```

```

100 REM RANDOMIZE
110 M=1:X=0
120 A$=""
130 FOR T=1 TO 9
140 L=INT(RND(1)*9)+49
150 Q=1
160 IF MID$(A$,Q,1)=CHR$(L) THEN GOTO140
170 IF Q<T THEN Q=Q+1:GOTO160
180 A$=A$+CHR$(L)
190 PRINT A$
200 PRINT
210 NEXT T
220 RETURN
230 REM Ausdruck
240 CLS
250 PRINT:PRINT:PRINT
260 PRINT"Zug Nr. ";M
270 PRINT:PRINT:PRINT
280 PRINT A$
290 PRINT:PRINT"Wieviele soll ich vertauschen?"
300 IF INKEY$(10)<>" " THEN GOTO300
310 C$=INKEY$(10)
320 R=VAL(C$)
330 IF R<1 OR R>8 THEN GOTO310
340 B$=""
350 FOR T=9 TO R STEP-1
360 B$=B$+MID$(A$,T,1)
370 NEXT T
380 A$=LEFT$(A$,R-1)+B$
390 RETURN
400 PRINT:PRINT:PRINT
410 PRINTA$
420 PRINT:PRINT:PRINT
430 PRINT"Sie haben es geschafft!"
440 PRINT:PRINT"Und das in nur ";M;" Zügen..."
450 END

```

Flipper

Dieses fesselnde Spiel verlangt ein ganz schönes Maß an geistiger Flexibilität von Ihnen. Wenn Sie das Spiel beginnen, werden Sie ein Durcheinander von X und * auf den Feldern des Spielbrettes sehen. Sie müssen das Spiel mit einem X in der Mitte und acht * darum herum beenden.

Sie geben die Nummer des Feldes ein, dessen Inhalt Sie umdrehen wollen. Wird ein Eckstück umgedreht, so ändern sich auch die angrenzenden Felder in ihr Gegenteil, ein X wird zu einem * und umgekehrt. Drehen Sie ein Mittelstück um, so werden auch die beiden angrenzenden Felder in der Linie und die Mittelstücke der anderen drei Seiten umgedreht.

So sieht die Aktion aus:

```

1 2 3      * * *
4 5 6      * X X
7 8 9      X X X

```

Anzahl der X :5

Sie brauchen nur eines - im mittleren Feld

Welches wollen Sie anstossen?

Sie haben 4 angeanstossen

```

1 2 3      X * *
4 5 6      X X X
7 8 9      * X X

```

Das war Zug 1

Anzahl der X :6

Sie brauchen nur eines - im
mittleren Feld

Welches wollen Sie anstossen?

Sie haben 7 angeanstossen

1 2 3 X * *

4 5 6 * * X

7 8 9 X * X

Das war Zug 2

Anzahl der X :4

Sie brauchen nur eines - im
mittleren Feld

Welches wollen Sie anstossen?

Sie haben 8 angeanstossen

1 2 3 X * *

4 5 6 * * X

7 8 9 * X *

Das war Zug 3

Anzahl der X :3

Sie brauchen nur eines - im
mittleren Feld

Welches wollen Sie anstossen?

Sie haben 9 angeanstossen

1 2 3 X * *

4 5 6 * X *

7 8 9 * * X

Das war Zug 4

Anzahl der X :3

Sie brauchen nur eines - im
mittleren Feld

Welches wollen Sie anstossen?

Sie haben 6 angeanstossen

1 2 3 X * X

4 5 6 * X X

7 8 9 * * *

Das war Zug 5

Hier ist das Listing, damit Sie es selber machen können:

```

10 REM Flipper
20 DIM A(10),F(4)
30 Z=-1:Q=42:X=88:P=0
40 REM RANDOMIZE
50 FOR C=1 TO 9
60 A(C)=Q
70 IF INT(RND(1)+.5)=0 THEN A(C)=X
80 NEXT C
90 GOSUB260
100 Z=Z+1
110 N=0
120 FOR C=1 TO 9
130 IF A(C)=X THEN N=N+1
140 NEXT C
150 IF N=1 AND A(5)=X THEN GOTO340
160 IF Z>0 THEN PRINT:PRINT"Das war Zug ";Z
170 PRINT:PRINT"Anzahl der X :";N
180 PRINT:PRINT"Sie brauchen nur eines -
im":PRINT"mittleren Feld"
190 PRINT:PRINT"Welches wollen Sie ansto-
ssen?"
200 IF INKEY$(10)<>" " THEN GOTO200
210 A$=INKEY$(10)
220 N=VAL(A$):IF N<1 OR N>9 THEN GOTO210
230 P=N
240 GOSUB370
250 GOTO90
260 CLS:PRINT:PRINT
270 IF P<>0 THEN PRINT"Sie haben ";P;" a-
ngeanstossen"
280 PRINT:PRINT:PRINT"1 2 3",CHR$(A(1));
" ";CHR$(A(2));" ";CHR$(A(3))
290 PRINT
300 PRINT"4 5 6",CHR$(A(4));" ";CHR$(A(5))
310 PRINT
320 PRINT"7 8 9",CHR$(A(7));" ";CHR$(A(8)

```

```

));" ";CHR$(A(9))
330 RETURN
340 PRINT:PRINT
350 PRINT"Sie haben es in nur ";Z;" Zueg-
en geschafft"
360 END
370 IF A(N)=X THEN RETURN
380 IF N=1 THEN F(1)=2:F(2)=4:F(3)=5:F(4)
)=10
390 IF N=2 THEN F(1)=1:F(2)=3:F(3)=10:F(4)
)=10
400 IF N=3 THEN F(1)=2:F(2)=5:F(3)=6:F(4)
)=10
410 IF N=4 THEN F(1)=1:F(2)=7:F(3)=10:F(4)
)=10
420 IF N=5 THEN F(1)=2:F(2)=4:F(3)=8:F(4)
)=6
430 IF N=6 THEN F(1)=3:F(2)=9:F(3)=10:F(4)
)=10
440 IF N=7 THEN F(1)=4:F(2)=5:F(3)=8:F(4)
)=10
450 IF N=8 THEN F(1)=7:F(2)=9:F(3)=10:F(4)
)=10
460 IF N=9 THEN F(1)=8:F(2)=5:F(3)=6:F(4)
)=10
470 FOR G=1 TO 4
480 F=0
490 IF A(F(G))=X THEN F=1
500 IF F=1 THEN A(F(G))=Q
510 IF F=0 AND A(F(G))=Q THEN A(F(G))=X
520 NEXT G
530 A(N)=X
540 RETURN

```

Suche nach Daryl

Hier haben wir das schwierigste Programm von allen. Daryl hat sich in einem unterirdischen Höhlensystem der verwirrendsten Komplexität verirrt.

Sie müssen den Weg in das System und Daryl, der in Höhle 29 ist, finden und dann auch noch sicher mit ihm zum Ausgang gelangen. Die Höhlen sind durch ein sich nicht veränderndes, kohärentes System von Tunneln verbunden. Sie sollten versuchen, dieses Rätsel zu lösen, indem Sie eine Karte erstellen, während Sie sich ins Innere vorkämpfen. Seien Sie jedoch gewarnt, einige Höhlen liegen unterhalb anderer! Die Höhlen sind nicht schön der Reihe nach ihren Nummern zugeordnet angelegt, die Nummern sind nur da, damit Sie sich nicht darauf beziehen können.

Sie werden das Programm als schwierig empfinden, es ist aber keineswegs unmöglich. Die Bezeichnung der Himmelsrichtungen innerhalb des Programmes sollte mit Vorsicht behandelt werden. Eigentlich ist Norden nicht oben und Süden nicht unten. Die Richtungen beziehen sich auf das ganze Höhlensystem, auf den Grundriß. Norden ist dann oben, Osten auf der rechten Seite, etc. Auf jeden Fall werden Sie die Richtungsangaben, zumindest beim ersten Versuch, das Programm zu spielen, verwirren.

Geben Sie nicht leichtfertig auf. Es ist machbar. Die Anzahl der möglichen Wege, die Sie nehmen können, ist außerordentlich groß. Wenn Sie es also einmal auf einem Weg nicht geschafft haben, versuchen Sie vielleicht, einen anderen zu finden und dann die Höhle auf einem dritten zu verlassen. Veteranen des Programmes versuchen jedesmal, einen neuen Weg zu finden und dann auf einem ganz anderen Weg (mit Daryl) wieder zum Ausgang zu gelangen. Um zu beweisen, daß das Problem gelöst werden kann, zeige ich Ihnen hier einen Weg zu Daryl und einen anderen wieder heraus:

Sie sind auf Ihrem Weg, um Daryl zu finden, der in diesem Höhlensystem gefangen ist

Sie sind in Höhle 1

Von dieser Höhle führen Tunnels zu:
2 7 10

Daryl ist nördlich

Daryl-Entfernungsmesser zeigt an: 30

In welche Höhle wollen Sie gehen ?2

Sie sind auf Ihrem Weg, um Daryl zu finden, der in diesem Höhlensystem gefangen ist

Sie sind in Höhle 2

Von dieser Höhle führen Tunnels zu:
25 11 1 8

Daryl ist nordöstlich

Daryl-Entfernungsmesser zeigt an: 12

In welche Höhle wollen Sie gehen ?8

Sie sind auf Ihrem Weg, um Daryl zu finden, der in diesem Höhlensystem gefangen ist

Sie sind in Höhle 8

Von dieser Höhle führen Tunnels zu:
26 2

Daryl ist nordöstlich

Daryl-Entfernungsmesser zeigt an: 33

In welche Hoehle wollen Sie gehen ?26

Sie sind auf Ihrem Weg, um Daryl zu finden, der in diesem Hoehlensystem gefangen ist

Sie sind in Hoehle 26

Von dieser Hoehle fuehren Tunnels zu:
8 3 23 12

Daryl ist noerdoestlich

Daryl-Entfernungsmesser zeigt an: 54

In welche Hoehle wollen Sie gehen ?23

Sie sind auf Ihrem Weg, um Daryl zu finden, der in diesem Hoehlensystem gefangen ist

Sie sind in Hoehle 23

Von dieser Hoehle fuehren Tunnels zu:
18 26

Daryl ist noerdoestlich

Daryl-Entfernungsmesser zeigt an: 86

In welche Hoehle wollen Sie gehen ?18

Sie sind auf Ihrem Weg, um Daryl zu finden, der in diesem Hoehlensystem gefangen ist

Sie sind in Hoehle 18

Von dieser Hoehle fuehren Tunnels zu:
13 15 22 23

Daryl ist noerdwestlich

Daryl-Entfernungsmesser zeigt an: 54

In welche Hoehle wollen Sie gehen ?22

Sie sind auf Ihrem Weg, um Daryl zu finden, der in diesem Hoehlensystem gefangen ist

Sie sind in Hoehle 22

Von dieser Hoehle fuehren Tunnels zu:
18 29

Daryl ist noerdwestlich

Daryl-Entfernungsmesser zeigt an: 74

In welche Hoehle wollen Sie gehen ?29

Bravo, Sie haben Daryl offenbar sicher und gut gefunden....

Nun ist es Zeit, den Rueckweg
zum Ausgang mit Daryl
anzutreten....

Sie haben Daryl gefunden, und kehren
zum Ausgang zurueck

Sie sind in Hoehle 29

Von dieser Hoehle fuehren Tunnels zu:
22

Der Ausgang ist noerdwestlich

Ausgang-Entfernungsmesser zeigt an: 55

In welche Hoehle wollen Sie gehen ?22

Sie haben Daryl gefunden, und kehren
zum Ausgang zurueck

Sie sind in Hoehle 22

Von dieser Hoehle fuehren Tunnels zu:
18 29

Der Ausgang ist noerdwestlich

Ausgang-Entfernungsmesser zeigt an: 44

In welche Hoehle wollen Sie gehen ?18

Sie haben Daryl gefunden, und kehren
zum Ausgang zurueck

Sie sind in Hoehle 18

Von dieser Hoehle fuehren Tunnels zu:
13 15 22 23

Der Ausgang ist noerdwestlich

Ausgang-Entfernungsmesser zeigt an: 24

In welche Hoehle wollen Sie gehen ?15

Sie haben Daryl gefunden, und kehren
zum Ausgang zurueck

Sie sind in Hoehle 15

Von dieser Hoehle fuehren Tunnels zu:
18 6

Der Ausgang ist suedwestlich

Ausgang-Entfernungsmesser zeigt an: 15

In welche Hoehle wollen Sie gehen ?6

Sie haben Daryl gefunden, und kehren
zum Ausgang zurueck

Sie sind in Hoehle 6

Von dieser Hoehle fuehren Tunnels zu:
15 19

Der Ausgang ist oestlich

Ausgang-Entfernungsmesser zeigt an: 5

In welche Hoehle wollen Sie gehen ?19

Sie haben Daryl gefunden, und kehren
zum Ausgang zurueck

Sie sind in Hoehle 19

Von dieser Hoehle fuehren Tunnels zu:
20 6

Der Ausgang ist suedoestlich

Ausgang-Entfernungsmesser zeigt an: 46

In welche Hoehle wollen Sie gehen ?20

Sie haben Daryl gefunden, und kehren
zum Ausgang zurueck

Sie sind in Hoehle 20

Von dieser Hoehle fuehren Tunnels zu:
19 14 21

Der Ausgang ist suedoestlich

Ausgang-Entfernungsmesser zeigt an: 25

In welche Hoehle wollen Sie gehen ?21

Sie haben Daryl gefunden, und kehren
zum Ausgang zurueck

Sie sind in Hoehle 21

Von dieser Hoehle fuehren Tunnels zu:
11 25 20

Der Ausgang ist suedoestlich

Ausgang-Entfernungsmesser zeigt an: 63

In welche Hoehle wollen Sie gehen ?25

Sie haben Daryl gefunden, und kehren
zum Ausgang zurueck

Sie sind in Hoehle 25

Von dieser Hoehle fuehren Tunnels zu:
21 2

Der Ausgang ist suedoestlich

Ausgang-Entfernungsmesser zeigt an: 43

In welche Hoehle wollen Sie gehen ?2

Sie haben Daryl gefunden, und kehren
zum Ausgang zurueck

Sie sind in Hoehle 2

Von dieser Hoehle fuehren Tunnels zu:
25 11 1 8

Der Ausgang ist suedoestlich

Ausgang-Entfernungsmesser zeigt an: 22

In welche Hoehle wollen Sie gehen ?1

Sie sind ein Held!!

Sie haben es mit Daryl
zurueckgeschafft....

Gehen Sie geradewegs nach Washington,
um dort vom Praesidenten geehrt zu
werden.....
Bis spaeter, Held!

Und das ist das Listing für das letzte „Gehirnspiel“:

```
10 REM Suche nach Daryl
20 GOSUB770:REM Initialisieren
30 GOSUB480:REM Raumbeschreibung
40 GOSUB360:REM Bewegen
50 IF DA=0 AND HOEHLE=29 THEN GOSUB90:RE
EM Daryl gefunden
60 IF DA=1 AND HOEHLE=1 THEN GOTO210:RE
M Sicher daheim
70 GOTO30
```

```
80 REM *****
90 REM Daryl gefunden
100 REM *****
110 PRINT:PRINT"Bravo, Sie haben Daryl o
ffenbar"
120 PRINT"sicher und gut gefunden...."
130 FOR P=1 TO 2000:NEXT P
140 PRINT:PRINT"Nun ist es Zeit, den Rue
ckweg"
150 PRINT"zum Ausgang mit Daryl"
160 PRINT"anzutreten...."
170 FOR P=1 TO 2000:NEXT P
180 DA=1
190 RETURN
200 REM *****
210 REM Sicher daheim
220 REM *****
230 PRINT:PRINT
240 PRINT"Sie sind ein Held!!"
250 FOR P=1 TO 2000:NEXT P
260 PRINT:PRINT"Sie haben es mit Daryl"
270 PRINT"zurueckgeschafft...."
280 PRINT:PRINT
290 PRINT"Gehen Sie geradewegs nach Wash
ington,"
300 PRINT"um dort vom Praesidenten geehr
t zu"
310 PRINT"werden....."
320 FOR P=1 TO 2000:NEXT P
330 PRINT"Bis spaeter, Held!"
340 END
350 REM *****
360 REM Ziehen
370 REM *****
380 PRINT:PRINT
390 INPUT"In welche Hoehle wollen Sie ge
hen ";J
400 FLA=0
410 FOR K=1 TO 4
```

```

420 IF Z(HOEHLE,K)=J THEN FLA=1
430 NEXT K
440 IF FLA=0 THEN PRINT"Es gibt keinen T
unnel zu dieser Hoehle":FOR P=1 TO 2000:NE
XT P:GOTO390
450 HOEHLE=J
460 RETURN
470 REM *****
480 REM Raumbeschreibung
490 REM *****
500 CLS:PRINT:PRINT
510 IF DA=0 THEN PRINT"Sie sind auf Ihre
m Weg, um Daryl zu":PRINT"finden, der in d
iesem Hoehlensystem":PRINT"gefangen ist"
520 IF DA=1 THEN PRINT"Sie haben Daryl g
efunden, und kehren":PRINT"zum Ausgang zur
ueck"
530 PRINT:PRINT"Sie sind in Hoehle ";HOE
HLE
540 PRINT:PRINT"Von dieser Hoehle fuehre
n Tunnels zu:"
550 FOR Q=1 TO 4
560 IF Z(HOEHLE,Q)>0 THEN PRINT Z(HOEHLE
,Q);
570 NEXT Q
580 PRINT:PRINT
590 IF DA=1 THEN GOTO660
600 PRINT"Daryl ist ";
610 IF Z(HOEHLE,5)>4 THEN PRINT"noerd";
620 IF Z(HOEHLE,5)<4 THEN PRINT"sued";
630 IF Z(HOEHLE,6)<7 THEN PRINT"oest";
640 IF Z(HOEHLE,6)>7 THEN PRINT"west";
650 GOTO710
660 PRINT"Der Ausgang ist ";
670 IF Z(HOEHLE,5)>7 THEN PRINT"noerd";
680 IF Z(HOEHLE,5)<7 THEN PRINT"sued";
690 IF Z(HOEHLE,6)<7 THEN PRINT"oest";
700 IF Z(HOEHLE,6)>7 THEN PRINT"west";
710 PRINT"lich"

```

```

720 PRINT:PRINT
730 IF DA=0 THEN PRINT"Daryl-Entfernungs
messer zeigt an: ";10*(ABS(Z(HOEHLE,5)-4)
+ABS(Z(HOEHLE,6)-7))
740 IF DA=1 THEN PRINT"Ausgang-Entfernun
gmesser zeigt an: ";10*(ABS(Z(HOEHLE,5)-7
)+ABS(Z(HOEHLE,6)-7))
750 RETURN
760 REM *****
770 REM Initialisieren
780 REM *****
790 CLS
800 DIM Z(29,6)
810 FOR H=1 TO 29
820 READ Z(H,1),Z(H,2),Z(H,3),Z(H,4),Z(H
,5),Z(H,6)
860 NEXT H
870 DA=0:REM Noch haben Sie Daryl nicht
gefunden
880 HOEHLE=1:REM Startposition
890 RETURN
900 DATA 2,7,10,0,7,7,25,11,1,8,5,5/
910 DATA 26,0,0,0,10,1,24,9,7,0,2,10/
920 DATA 27,28,0,0,10,8,15,19,0,0,7,2/
930 DATA 4,1,0,0,5,8,26,2,0,0,7,4/
940 DATA 4,24,10,0,4,11,1,9,0,0,7,9/
950 DATA 21,2,0,0,2,7,26,28,0,0,11,4/
960 DATA 18,16,0,0,8,10,20,0,0,0,2,2/
970 DATA 18,6,0,0,6,12,13,17,0,0,9,5/
980 DATA 16,28,0,0,11,6,13,15,22,23,9,11/
990 DATA 20,6,0,0,3,1,19,14,21,0,5,2/
1000 DATA 11,25,20,0,1,4,18,29,0,0,11,11/
1010 DATA 18,26,0,0,12,1,4,9,0,0,1,12/
1020 DATA 21,2,0,0,3,4,8,3,23,12,9,3/
1030 DATA 5,0,0,0,9,7,17,5,12,0,12,7/
1040 DATA 22,0,0,0,12,12/

```

Erstellen Ihrer eigenen Spiele

Spiele erfinden

Nachdem Sie schon einige Zeit programmiert haben, werden Sie finden, daß Sie die Fähigkeiten hätten, die man braucht, um Programme zu schreiben, aber keine guten Ideen für Spielprogramme haben. Sollte das der Fall sein, dann wird Sie dieses Kapitel sehr interessieren. Ich werde einige Spiele erwähnen, die mir wie geschaffen für eine Bearbeitung zu einem Computerspiel scheinen.

Für ein zweites Buch mit Computerspielen, das gerade in Vorbereitung ist, suche ich noch neue Programme. Wenn Sie Programme geschrieben haben, von denen Sie glauben, daß sie passend sind (damit meine ich, originale Programme und nicht nur für einen bestimmten Computer geeignet), würde ich diese sehr gerne sehen, um sie vielleicht zu kaufen und in dem Buch abzudrucken. Natürlich werden Sie als Urheber dieser Programme erwähnt. Ich kann leider nur gedruckte, direkt vom Computer ausgegebene Listings in Erwägung ziehen. Schreiben Sie mir zu Händen des Herausgebers.

Aber kommen wir nun zum Schreiben der eigenen Programme. Wenn Sie nicht ein wirklich sehr bekanntes Spiel, wie z.B. Dame, in ein Programm umwandeln, brauchen Sie nicht zu beweisen, daß der Werdegang vom Spiel zum Computerprogramm auf den Zusammenstellungen hier beruht. Sie werden wahrscheinlich finden, daß es ab einem gewissen Punkt in der Entwicklung nichts oder zumindest sehr wenig mit seinen Ahnen gemein hat. Das ist sicher ein Vorteil.

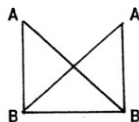
Anzupassende Spiele

Shuffle Board

Die Spieler stoßen flache, runde Spielsteine auf einem Spielfeld auf ein dreieckiges Ziel zu. Dieses Ziel besteht aus farbigen Feldern mit Nummern. Ein Spielstein, der vollkommen auf einem dieser Felder zu liegen kommt, wird mit der Punktzahl des Feldes bewertet. Es gibt sehr viele mögliche Abarten des Spiels, wie z.B. wurfpfeilartige Spiele. Haben Sie einmal ein 'Shuffleboard'-Programm erstellt, so werden Sie wahrscheinlich nur wenige Änderungen brauchen, um es in ein 10-Kegel-Bowlingsspiel zu verwandeln.

Pferdefuß

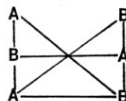
Auf einem Spielfeld, das wie folgt aussieht,



haben die Kontrahenten ihre Steine auf den Linienschnittpunkten, die mit A und B markiert sind. Nun wechseln sie sich mit dem Ziehen auf einen freien Punkt ab. Das Spiel ist zu Ende, wenn ein Spieler nicht mehr ziehen kann. Wie Sie während eines Spieles entdecken werden, ist dies trotz seiner Einfachheit ein interessantes Spiel.

Damokles

Dieses Spiel ist **Pferdefuß** ähnlich, verwendet aber ein größeres Brett und drei Steine pro Spieler. Das Ziel des Spieles ist das gleiche wie bei **Pferdefuß**. Das Brett sieht so aus:



Mu-Torere

Dieses Spiel, das von den Maoris Neuseelands kommt, wird auf einem Brett gespielt, welches das Aussehen eines achtstrahligen Sternes hat. Jeder Strahl ist mit einem Kreis in der Mitte verbunden. Dieser Kreis heißt Putahi. Die Spieler beginnen mit je vier Steinen auf den außen liegenden Punkten, die alle nebeneinander liegen. Das Ziel auch dieses Spieles ist es, wie schon bei den beiden vorangegangenen, das Ziehen des Gegenspielers unmöglich zu machen. Sie können von einem Punkt zu einem leeren benachbarten von Putahi oder zu jedem freien Punkt ziehen. Der dritte mögliche Zug ist von einem leeren Punkt zum Putahi. Das ist allerdings nur dann erlaubt, wenn zumindest einer der benachbarten Punkte durch einen Stein des Gegners besetzt ist.

Bingo

Dieses Spiel wird normalerweise mit Karten gespielt, die mit Nummern versehen sind. Die Nummern werden zufällig aus einem Topf gezogen und dann auf den Karten durchgestrichen. Der Sieger ist derjenige, der als erster fünf Nummern auf seiner Karte in jeder beliebigen Richtung durchgestrichen hat. Das Spiel kann für den Gebrauch am Computer sehr stark vereinfacht werden. Er verwendet Zahlen von 0 bis 99 und erzeugt damit zwei zufällige 'Karten', die er dann auf einem 5x5 Gitter auf den Bildschirm bringt. Danach wählt der Computer Zufallszahlen zwischen 0 und 99, besorgt das Durchstreichen und sucht die Gewinner. Der Ablauf des Programmes, zumindest der optische, kann durch das Einführen zweier Wahlvorgänge anstelle nur der einfachen Erzeugung einer Zufallszahl interessanter gestaltet werden. Diese Wahlvorgänge könnten mittels zwei sich drehender Zeiger, jeder von 0 bis 9, einer für die 'Zehner-', und eine für die 'Einerstelle', verwirklicht werden.

Würfelspiele

Wie ich schon in dem Kapitel über Würfelspiele erwähnte, sollte Ihnen das Programmieren von Würfelspielen nicht schwer fallen. Speziell dann nicht, wenn der Computer nicht zu viel 'Intelligenz' beim Spielen entwickeln muß. Es gibt eine ganze Anzahl von Würfelspielen, die Sie wahrscheinlich gerne ausprobieren und für Ihren Computer abändern wollen, darunter:

Tot umfallen

Dafür brauchen Sie fünf Würfel. Die Würfel werden vom ersten Spieler geworfen, und jeder, der 5 oder 2 zeigt, wird weggenommen. Die Summe der restlichen Augenzahlen wird aufgeschrieben. Die 2 und 5

zeigenden Würfel werden nochmals geworfen. Das setzt sich so lange fort, bis keine Würfel übrig sind. Dann endet der Wurf des jeweiligen Spielers. Auch der zweite Spieler unterwirft sich diesen Regeln. Der Sieger ist derjenige, der, nachdem beide ihren Wurf hatten, die größte Punktezahl hat. Sie können das Spiel auf einige Runden, z. B. 3 oder 5, erweitern oder aber die Anzahl der Würfel pro Versuch, unabhängig wieviele Würfel auf 2 oder 5 liegenbleiben, beschränken.

Martinetti

Für dieses Spiel brauchen Sie drei Würfel, einen Zettel für die Punkte und zwei Einsätze, einen für Sie, einen für den Computer. Auf dem Zettel ist eine Reihe mit von 1 bis 12 gekennzeichneten Feldern. Sie versuchen, die Zahlen der Reihe nach abzuheften, bevor Ihr Kontrahent es schafft. Nun wechseln sich die Spieler beim Würfeln ab. Enthält der Wurf eine 1, so kann der Einsatz des Spielers auf das Feld der 1 gelegt werden. Nach Feld 1 können die Spieler entweder die Augenzahl eines einzelnen Würfels oder die Summe jeder beliebigen Kombination heranziehen, um eine gewünschte Zahl zu erreichen. Eine Abart dieses Spieles (genannt **Everest**) ist, durch die Nummern auch wieder 'abzusteigen', nachdem man 1 bis 12 schon erreicht hat. Der Sieger von **Everest** ist derjenige, der als erster wieder im Basislager bei 1 ist.

Indianerwürfel

Dieses Spiel, ähnlich **Pokerwürfel**, wird mit fünf Würfeln gespielt. Das Ziel ist ganz einfach, die besseren 'Poker-Blätter' zu bekommen. Sie können keine zusätzlichen Karten nach dem ersten Wurf erhalten. Sie werfen die fünf Würfel auf einmal, und das ist Ihr 'Blatt'. Die Reihenfolge der Blätter ist wie folgt:

- fünf gleiche
- vier gleiche
- full house (Drilling und Paar)
- Drilling
- zwei Paare
- ein Paar
- der höchste Würfel

Bidou

Auch dieses Spiel ist **Pokerwürfel** ähnlich. Jedoch versuchen Sie hierbei, Ihre Chips loszuwerden und nicht, sie zu gewinnen. Paradoerweise ist der Sieger des Spieles derjenige, der die meisten Runden verloren hat. Sie benötigen drei Würfel für dieses Spiel und einen Vorrat an Chips, der gleichmäßig an die Spieler ausgegeben wird. Zunächst wird gesetzt und mitgezogen. Dann werfen die Spieler alter-

nierend die drei Würfel. Die Wertung der Würfe ist wie folgt (mit den 'höchsten Würfeln' zuerst):

2 1 1; 2 2 1; 4 2 1; 6 6 6; 5 5 5; 4 4 4;
 3 3 3; 2 2 2; 1 1 1; 3 3 6; 3 3 5; 3 3 4;
 3 3 2; 3 3 1; 1 1 6; 1 1 5; 1 1 4; 1 1 3;
 3 2 1; 4 3 2; 5 4 3; 6 5 4;

Ist die geworfene Kombination nicht auf der Liste, so werden die Augenzahlen zusammengezählt, und die höchste Summe gewinnt.

Barzelle (Barbooth)

Auch unter dem Namen **Barbudi** wird dieses Spiel mit 2 Würfeln gespielt. Dabei gibt es Würfe, die gewinnen, und solche, die verlieren. Ich schlage vor, mit 30 Chips zu beginnen und jedem Spieler 15 davon zu geben. Für jede Runde muß dann jeder Spieler 2 Chips bezahlen. Die Würfel werden abwechselnd geworfen. Die 'gewinnenden' Würfe sind 33, 55, 65, und 66. Die verlierenden Würfe sind 11, 22, 21 und 44. Werfen beide Spieler 'gewinnende' Punktezahlen, so gewinnt der, der in der Reihenfolge der Liste oben am weitesten links ist (d. h. 33 ist besser als 66). Werfen beide die gleichen gewinnenden Punkte, erhält die Bank das Geld. Wirft ein Spieler eine Kombination, die weder unter den gewinnenden noch unter den verlierenden ist, und der andere eine verlierende, so verliert der zweite seinen Einsatz an die Bank und der andere erhält seinen zurück.

Wirft einer eine gewinnende Augenzahl und der andere eine nicht gewertete Kombination, erhält der Sieger seine und die Chips des anderen. Wirft jedoch der eine eine gewinnende und der andere eine verlierende Kombination, so schießt die Bank dieselbe Anzahl Chips hinzu, wie insgesamt gesetzt wurde. Das Spiel wird so lange fortgesetzt, bis ein Spieler oder die Bank keine Chips mehr hat oder haben. Der Spieler mit den meisten Chips zu diesem Zeitpunkt gewinnt. Beachten Sie, daß diese Art der Gewinnverteilung nicht die gleiche ist wie in dem professionellen Spiel **Barbooth**. Es ist aber ein einfach zu computerisierendes Auszahlungsschema und schafft auch ein interessantes Spiel.

Fünftausend

Es gibt keinen Preis dafür, daß Sie erraten haben, daß der erste Spieler mit über 5000 Punkten gewinnt. Alles, was Sie zu diesem mit so vielen Spielern, wie Sie wollen, zu spielenden Spiel brauchen, sind fünf sechseckige Würfel. Die Spieler wechseln sich im Werfen der fünf Würfel ab, und gepunktet wird in der folgenden Art:

- eine 5 ist 100 wert;
- eine 2 ist 50 wert;
- drei 5en auf einen Versuch geworfen sind 1000 wert;
- Drillinge, abgesehen von 5en und 2en punkten 100mal ihren Wert (d.h. drei 1en punkten 100, drei 6en punkten 600);
- wird die Reihe 1-2-3-4-5 auf einen Versuch erzielt, so bringt das 1500 Punkte;
- 2-3-4-5-6 in einem Wurf zählen 2000 Punkte.

Beachten Sie, daß 4en und 6en in diesem Spiel außer in einer Reihe nicht punkten. Haben Sie einmal gewürfelt, so können die nicht punktenden Würfel nochmals werfen, wenn Sie wollen. Erzielen Sie mit diesem Wurf jedoch keine Punkte, so dürfen Sie diesen nicht mehr wiederholen, und der gesamte Wurf punktet 0. So lange punktende Würfel gemacht werden, können alle nicht punktenden Würfel nochmals geworfen werden.

Das Spiel belohnt langsame und vorsichtige Spieler mit ganz guten Ergebnissen. Gegen einen Spieler, der Glück hat und eine gewaltige Punktezahl bei einem Wurf erreicht, hat jener jedoch keine Gewinnaussichten.

Kartenspiele

Haben Sie einmal eine Routine ausgearbeitet, die den Computer Karten halten, mischen und austeilen läßt (eine solche Routine ist in meinem Spiel **Memory** in diesem Buch enthalten), dann haben Sie das Skelett für eine wahre Flut von Programmen. Jedes Buch über Kartenspiele kann Ihnen wahrscheinlich mehr Ideen geben, als Sie bewältigen können. Hier sind einige für den Anfang:

Trente und Quarante

Dieses aus dem Europa des 17. Jahrhunderts stammende Spiel ist auch unter dem Namen **Rouge et Noir** bekannt. In der Kasinoversion werden sechs Päckchen Karten verwendet, doch funktioniert es auch mit einem ganz gut. Die Karten werden auf 2 Reihen aufgeteilt. Diese beiden Kartenstöße werden, nachdem jede Reihe eine zusätzliche Karte erhielt, untersucht. König, Dame, Bube zählen 10, alle anderen Karten (auch das As) bringen ihre Punktezahl. Wenn die Summe einer Reihe 31e ergibt oder übersteigt, werden keine Karten mehr zu dieser Reihe hinzugefügt. Jedoch fügt der Geber Karten zu der anderen Reihe hinzu, bis auch hier 31 erreicht oder überschritten sind. Unabhängig, ob die Farben der Karten zutreffen oder nicht, heißt die erste Reihe 'schwarz' und die zweite 'rot'.

Sie können nun wetten, daß am Ende die Summe der schwarzen oder der roten Reihe näher bei 31 liegt. Sie können auch wetten, daß die erste ausgegebene Karte die Farbe des Namens der gewinnenden

Reihe hat. Alle Wetten werden gegen die Bank gemacht. Die Gewinner erhalten ihre Wette plus dem gleichen Betrag (Chancen 1:1) zurück. Wenn die Summen beider Reihen gleich 31 sind, behält die Bank die Hälfte der Einsätze und gibt die andere an die Spieler zurück.

Kartenschummeln

Dieses Spiel war gedacht, um Gesetze zu umgehen, die von einigen US-Staaten ausgegeben wurden, um das Schummel-Würfelspiel illegal zu machen. Man verwendet einen speziellen Kartenstapel, der aus den Assen, 2ern, 3ern, 4ern, 5ern und 6ern aus zwei Kartenpäckchen besteht. Das Spiel wird nach den Standardregeln von **Schummeln** gespielt (die den Regeln des in diesem Buch abgedruckten **Sieben/Elf** ähnlich sind).

In seiner einfachen Form, ohne die ziemlich schwierigen Wettkombinationen, spielt man es wie folgt:

- Eine Summe von 2, 3 oder 12 ist 'geschummelt', und der Geber verliert und muß die Karten an den nächsten Spieler zum Geben weiterreichen.
- Eine Summe von 7 oder 11 ist ein 'Natural', und der Geber gewinnt.
- Jede andere Zahl (4, 5, 6, 8, 9 oder 10) ist der 'Punkt' des Spielers, und er muß weiter Paare geben, bis er diesen 'Punkt' wieder erreicht, wodurch er gewinnt. Wird jedoch eine 7 gegeben, bevor der 'Punkt' erreicht wird, verliert der Geber.

Drei Karten Angabe

Dieses Spiel war der Vorläufer von **Poker** und, wie sein Name erraten läßt, wird es mit den Blättern von drei anstatt fünf Karten gespielt. Hier sind die gewinnbringenden Kombinationen in der Reihenfolge der höchsten zu der niedersten:

- 'Pryle', Drillinge
- 'On a bike', drei einer Farbe der Reihe nach
- 'run', drei der Reihe nach
- 'flush', drei Karten einer Farbe
- 'Paar', ein Paar
- 'Höchste Karte'

Es wird ein normales Päckchen Karten verwendet.ASSE zählen außer in 'on a bike' und 'run', wo 3, 2, As, das As, König, Dame schlägt, viel. Sie können dieses Spiel sehr leicht zu einer 'Sieben Karten Angabe' abändern. Hier würden die Spieler dann ihre sieben Karten in zwei Dreiergruppen ihrer Wahl aufteilen. Auch eine 'Neun Karten Angabe', in der drei Dreiergruppen gebildet werden, ist möglich.

As-Zwei-Joker

Dies ist ein sehr einfaches Spiel, das den Geber stark bevorteilt. Die Spieler setzen auf die Wahrscheinlichkeit, daß die nächste erscheinende Karte ein As, eine Zwei oder ein Joker ist.

Ablegen und Nehmen

Dieses Spiel verwendet eine Scheibe mit acht verschiedenen Feldern an ihrem Rand. Das Spiel beginnt, nachdem alle Spieler 10 Chips erhalten haben und 30 in einen Topf gelegt wurden. Nun wechseln sich die Spieler ab, die Scheibe zu drehen. Nach den Aufgabenstellungen, die auf den acht Feldern der Scheibe notiert sind, müssen die Spieler dann, je nachdem, welche erscheint, Chips in den Topf legen oder welche herausnehmen. Ein Spieler scheidet aus, wenn er keine Chips mehr hat. Das Spiel ist zu Ende, wenn der Topf leer ist. Die auf die Seite der Scheibe geschriebenen Anweisungen sind:

- eins ablegen
- eins nehmen
- drei ablegen
- drei nehmen
- vier ablegen
- vier nehmen
- alle ablegen
- alle nehmen

Boule

Das ist eine vereinfachte Form von Roulette, welche im 8. Jahrhundert aufkam. In europäischen Casinos kann man es oft finden. Das Rad ist in 18 Segmente geteilt, die mit 1 bis 9 zweimal numeriert sind. Sie können auf die gewinnende Nummer setzen, ob sie schwarz oder rot, gerade oder ungerade, hoch oder niedrig sein wird. Ausgenommen der 5, die normalerweise gelb ist, wechseln sich die roten und die schwarzen Nummern ab. Daher gibt es z.B. sowohl eine rote 6 wie auch eine schwarze, ebenso eine rote 9 und eine schwarze, etc. Eine 'hohe' Nummer ist größer als 5, eine 'niedrige' kleiner als 5. Wetten Sie erfolgreich auf eine einzelne Zahl, so werden Sie 7 zu 1 ausbezahlt. Die Charakteristiken einer Zahl (wie rot oder ungerade) bringen eine Auszahlung von 1 zu 1.

Glossar

Anweisung – ein Element einer Programmiersprache, das dem Computer die Durchführung einer speziellen Aufgabe aufträgt. Eine Anweisung in Assemblersprache wäre z.B. ADD, die, wie Sie sicher erraten haben, dem Computer eine Addition befiehlt.

Akkumulator – Teil der Computerlogik, der das unmittelbare Ergebnis einer Rechnung speichert.

Adresse – eine Nummer, die auf den Speicherplatz hinweist, in dem die Information abgespeichert ist.

Algorithmus – die Folge der Schritte, um ein Problem zu lösen.

Alphanumerisch – gewöhnlich dazu verwendet, eine Tastatur zu beschreiben. Es bedeutet, daß sie alphabetische und numerische Zahlen umfaßt. Ein numerische Tastatur, als Gegensatz, hat nur Tasten mit den Zahlen von 1 bis 9.

APL – steht für 'Automatic Programming Language'. Eine Computersprache, die von Iverson zu Beginn der 60er Jahre entwickelt wurde. Sie unterstützt eine große Anzahl von Benutzer- und Datenstrukturen. Sie verwendet eine Anzahl von nicht standardisierten Zeichen.

Application Software – Dies sind auf spezielle Aufgaben zugeschnittene Programme, wie z.B. Textverarbeitung oder Arbeiten mit Postlisten und Briefen.

ASCII – bedeutet 'American Standard Code for Information Interchange'. Es ist ein fast weltweit verbreiteter Code für Buchstaben, Ziffern und Zeichen.

Assembler – ist ein Programm, das ein anderes in Assemblersprache (ein Computerprogramm, in dem ein einzelner Befehl, wie z.B. ADD, in eine für den Computer verständliche Form gebracht wird) geschriebenes Programm in eine Form bringt, mit der der Computer arbeiten kann.

BASIC – bedeutet 'Beginner's All-Purpose Symbolic Instruction Code'. Das ist die meistverbreitete mit Mikrocomputern verwendete Sprache. Sie ist sehr leicht zu erlernen, viele ihrer Befehle sind dem Englischen sehr angelehnt.

Batch (Block) – eine Gruppe von Aktionen, die von dem Computer ohne Unterbrechung in einem Block bearbeitet werden.

Baud – eine Maßeinheit für die Geschwindigkeit, mit der Daten übertragen werden. Sie steht für die Anzahl der Bits (ein Informationssignal) pro Sekunde.

Benchmark – ein Test, um das Verhalten eines Computers zu messen. Das Ergebnis wird dann mit einem auf einem anderen Computer gemachten Test verglichen.

binär – ein Zahlensystem, in dem es nur zwei Symbole gibt, 0 und 1 (im Gegensatz zum normalen Dezimalsystem, in dem es 10 Symbole gibt: 1, 2, 3, 4, 5, 6, 7, 8, 9 und 0). Ihr Computer arbeitet „binär“.

Boole'sche Algebra – jene Algebra der Entscheidungsfindung und Logik, die von dem Engländer George Boole, einem Mathematiker, erfunden wurde und im Herzen Ihres Computers die Entscheidungsfindung steuert.

Bootstrap – ein Programm, das sofort nach dem Einschalten Ihres Computers eingelesen wird. Es versetzt den Computer in ein Stadium, in dem er andere Programme akzeptiert und versteht.

Buffer – ein Speicher, der die Eingabe von einem Eingabegerät, wie der Tastatur, aufnimmt, und dann mit der vom Computer vorgeschriebenen Geschwindigkeit weitergibt.

Bug – ein Fehler im Programm.

Bus – eine Gruppe elektrischer Verbindungen zwischen einem Computer und peripheren Geräten oder einem anderen Computer.

Byte – die kleinste Gruppe von Bits (siehe bit), die ein Computerwort bildet. Allgemein wird ein Computer beschrieben als '8 Bit' oder '16 Bit'. Das bedeutet, daß ein Wort aus einer Kombination von entweder 8 oder 16 Nullen und Einsen besteht.

Central Processing Unit (CPU, Zentraleinheit) – das Herz des Computers, in dem logische, arithmetische und kontrollierende Funktionen ausgeführt werden.

Character Code – die Zahl in ASCII (siehe ASCII), die auf ein bestimmtes Zeichen verweist, so wie 32 für die Leerstelle und 65 für den Buchstaben 'A'.

Cobol – steht für 'Common Business Orientated Language', eine dem Englischen sehr nahe Standardprogrammiersprache, die hauptsächlich im Geschäftsleben verwendet wird.

Compiler – ein Programm, das ein in einer höheren Programmiersprache (dem Menschen angepaßt) geschriebenes Programm in die Maschinensprache, die der Computer unmittelbar versteht, übersetzt.

Concatenate – summieren (zwei Strings zu addieren ist auch unter 'concatenate' bekannt).

CP/M – steht für 'Control Program for Microcomputers', ein fast weltweit verbreitetes System auf Disketten, das von Digital Research, Pacific Grove, Kalifornien, entwickelt und verbreitet wurde.

Datei – eine Gruppe von zusammenhängenden Sätzen, die gemeinsam verarbeitet werden, wie eine Inventurdatei oder eine Studenten-datei.

Daten – der Oberbegriff für alle vom Computer bearbeitete Information.

Database – eine Zusammenstellung von Daten, die so organisiert ist, daß sie schnellen Zugriff durch den Computer erlaubt.

Debug – ein Programm von Fehlern (bugs) befreien.

Disk (Diskette) – ein magnetisierbares Speichermedium (weitere Beschreibung als 'hard disk', 'floppy disk' oder nur 'floppy'). Es wird zur Abspeicherung von Computerdaten und Programmen verwendet. Bis zu einem gewissen Grad sehen Disketten wie Schallplatten aus und haben auch eine ähnliche Funktion. Gewöhnlich haben sie Durchmesser von 20, 13, oder 8 cm. Kleinere 'Mikrodisketten' für einige Systeme kann man auch finden.

Dokumentation – die geschriebenen Anweisungen und Erklärungen, die das Programm begleiten.

DOS – steht für 'Disk Operating System' (und wird allgemein als 'doss' ausgesprochen). Es ist ein vielseitiges Programm, das dem Computer die Kontrolle über ein Diskettensystem ermöglicht.

Dot-Matrix Drucker – ist ein Drucker, der die Buchstaben und Zeichen durch Punkte auf Acht-mal-Acht- oder Sieben-mal-Sieben-Gittern formt.

Double-density – Adjektiv, das zur Beschreibung von Disketten, die mit einer speziellen Technik aufgenommen wurden, dient. Diese Technik verdoppelt, wie schon aus dem Namen ersichtlich, den auf der Diskette zur Verfügung stehenden Speicherplatz.

Dynamic memory (Flüchtiger Speicher) – ein Computerspeicher, der immer wieder neu geladen werden muß, damit sein Inhalt nicht verlorengeht.

EPROM – steht für 'Erasable Programmable Read Only Memory'. Es enthält die Computerinformation in einer halb-permanenten Form. Um seinen Inhalt zu löschen, muß es längere Zeit ultraviolettem Licht ausgesetzt werden.

Error messages – stellt Information vom Computer für den Bediener dar. Manchmal besteht sie nur aus wenigen Ziffern, aber meist doch in Form einer Phrase (wie 'Out of memory'). Sie weist auf den Fehler oder Irrtum im Programm oder in der Bedienung hin, der den Computer in der Programmausführung unterbrochen hat.

Echtzeit – wenn ein Computer einen Vorgang zur selben Zeit wie in der realen Welt bearbeitet, so sagt man, der Vorgang geschieht in Echtzeit. Ein Beispiel wäre ein Programm, das die Entwicklung einer Bakterienkolonie mit der gleichen Geschwindigkeit zeigt, wie sich diese Bakterienkolonie wirklich entwickeln würde. Viele Spiele, die eine Reaktion in Echtzeit verlangen, sind entwickelt worden. Die meisten 'Arkade'-Spiele zählen zu ihnen.

Feld – Eine Zusammenstellung von Symbolen, die eine bestimmte Gruppe ergeben, wie z.B. ein Identifizierungscode, ein Name, ein Datum... Ein Feld ist meist ein Teil eines Records (Satzes).

Firmware – Die festen Teile eines Computers heißen 'Hardware', die Programme in maschinenlesbarer Form auf Diskette oder Kassette werden 'Software' genannt. Die Programme allerdings, die im Computer fest verdrahtet sind, heißen 'Firmware'. Firmware kann durch Software bis zu einem gewissen Grad unter bestimmten Umständen geändert werden.

Flag – ist ein Indikator innerhalb eines Programmes. Der Zustand des 'flag' (d.h. der Wert, den er hat) gibt Aufschluß über bestimmte Ereignisse.

Floppy disk – siehe Disk.

Flußdiagramm – ein geschriebener Plan der Programmstruktur und des Programmablaufes. Es verwendet verschiedene Figuren, so z.B. Rechtecke mit schiefen Seiten für die Computeraktionen, eine Raute für Computerentscheidungen,... für die verschiedenen Programmteile. Meistens wird zunächst ein Flußdiagramm geschrieben, bevor noch eine Zeile in den Computer eingegeben wurde.

FORTRAN – eine höhere Computersprache, meist für Forschungsarbeiten verwendet (von FORMula TRANslation).

Gate – ein Computerteil, der die Entscheidungen trifft. Er erlaubt einem Stromkreis, entweder in einer oder in der anderen Richtung zu fließen, je nachdem, ob die Bedingungen erfüllt sind.

GIGO – Abkürzung für 'garbage in garbage out'. Es bedeutet, daß, wenn Mist oder falsche Daten in einen Computer eingegeben werden, auch das Resultat Mist oder falsch sein muß.

Global – eine Reihe von Bedingungen, die das ganze Programm betreffen, heißt global, zum Unterschied von lokal.

Hardware – die festen Teile eines Computers (siehe 'Software' und 'Firmware').

Hexadezimal – ein Zählsystem, das von Maschinencodeprogrammierern sehr gern verwendet wird, da es mit der Zahlenspeichermethode

des Computers sehr eng verwandt ist. Es basiert auf der Zahl 16 im Gegensatz zu unserem 'normalen' Dezimalsystem, das auf der Zahl 10 basiert.

Hex pad – eine Tastatur, ähnlich der eines Taschenrechners, die für die direkte Eingabe von Hexadezimalzahlen vorgesehen ist.

Höhere Programmiersprachen – Programmiersprachen, die der Sprache Englisch sehr ähnlich sind. Niedere Sprachen sind der Sprache näher, die der Computer versteht. Weil höhere Programmiersprachen zuerst in eine computerverständliche Form übersetzt werden müssen, sind sie langsamer als ihre niedrigen Gegenstücke.

Input – jede Information, die während des Programmlaufes in ein Programm eingegeben wird.

I/O – steht für Input/Output-Gerät, eines Behelfes, dessen sich der Computer bedient, um mit der Außenwelt in Verbindung zu treten.

Interpreter – übersetzt die höhere Programmiersprache (die der Mensch verstehen kann) in eine Form, die der Computer versteht.

Joystick – ein Analoginstrument, das Signale in den Computer eingibt, die der Position des Joysticks entsprechen. Hauptsächlich wird es für Spielprogramme verwendet.

Kilobyte – eine Einheit der 'Speichermessung'; ein Kilobyte (meist mit K abgekürzt) entspricht 1024 Byte (siehe Byte).

Maschinensprache – der Schritt unterhalb der niederen Programmiersprache; die Sprache, die der Computer direkt versteht.

Mainframe (Hauptcomputer) – der Name für riesige Computer, wie z.B. die IBM 307. Computer werden auch in Minicomputer oder Mikrocomputer eingeteilt (das wäre so etwas, wie Sie haben).

Maus – ein Kontrollgerät, etwas kleiner als eine Zigarettenschachtel, das über den Tisch gerollt wird, um einen Lichtpunkt auf dem Bildschirm zu bewegen, mit dem man in dem gerade laufenden Programm unter verschiedenen Möglichkeiten wählen und Entscheidungen treffen kann. Mäuse funktionieren entweder dadurch, daß sie die Bewegungen ihrer Räder oder das Muster eines Gitters, auf dem sie bewegt werden, ablesen.

Mikroprozessor – der Chip, der im Herzen Ihres Computers liegt. Er macht die 'Denkarbeit'.

Modem – steht für MOdulator/DEModulator. Es ist eine Maschine, die die Telefontelekommunikation zwischen zwei Computern ermöglicht.

Monitor – (a) ein abgewandelter Fernsehschirm, der als Anzeigegerät für den Computer dient, er enthält keinen Programmabstimmungs-

oder Empfangsteil; (b) die Information innerhalb eines Computers, die ihm das Verstehen und Ausführen der Programmbefehle ermöglicht.

Motherboard – ein Gerät, das meist außerhalb des eigentlichen Computers ist. Es besitzt Steckkontakte, damit man zusätzliche Schaltungen an den Computer anschließen kann. Dadurch erhält man Eigenschaften, die nicht zur Standardausrüstung des Computers zählen (wie z.B. hochauflösende Graphikgeräte oder eine automatische Kontrolle).

Niedere Programmiersprache – eine Sprache, die der im Computer verwendeten sehr ähnlich ist (siehe höhere Programmiersprache).

Netzwerk – eine Gruppe von Computern, die miteinander arbeiten.

Numerische Tastatur – eine Vorrichtung in erster Linie für die Eingabe von numerischen Informationen in den Computer, ähnlich einem Taschenrechner.

Oktal – ein auf acht basierendes Zahlensystem (verwendet die Ziffern 1, 2, 3, 4, 5, 6, 7 und 0).

On-Line – Vorrichtung, die unter der direkten Kontrolle des Computers steht.

Operating System – Das ist das „große Chefprogramm“ oder eine Anzahl von Programmen, die innerhalb des Computers seine Operationen kontrollieren. So ruft es z.B. Routinen auf, wenn diese benötigt werden, oder setzt Prioritäten.

Output (Ausgabe) – alle Daten, die vom Computer während seiner Arbeit erstellt werden, ob diese Daten nun auf dem Bildschirm, dem Drucker ausgegeben werden, oder interne Verwendung finden, ist nebensächlich.

Pascal – eine höhere Computersprache, von Niklaus Wirth in den späten 60er Jahren entwickelt. Er befürwortet diszipliniertes, strukturiertes Programmieren.

Port – ein „Ausgabe-“ oder „Eingabeloch“ am Computer, durch das Daten transportiert werden.

Programm – die Reihe der Anweisungen, die der Computer ausführt, um die vorher bestimmte Aufgabe zu lösen.

PILOT – eine höhere Programmiersprache, meist für die Entwicklung von erzieherischen Computerprogrammen verwendet.

RAM – steht für 'Random Access Memory'. Das ist der Speicher, der im Computer das aktuelle Programm beinhaltet. Der Inhalt von RAM kann, der Inhalt vom ROM (Read Only Memory) kann nicht unter Software-Kontrolle geändert werden.

Refresh – der Inhalt von dynamischen Speichern (siehe Speicher) muß von Zeit zu Zeit einen Energieschub erhalten, damit er nicht verlorengeht. Das Signal, das den Speicher an seinen Inhalt erinnert, heißt 'refresh'-Signal.

Register – ein Ort in einem Computerspeicher, der Daten enthält.

Reset – ein Signal, das den Computer in den Zustand zurückführt, in dem er war, als er eingeschaltet wurde.

ROM – siehe RAM

RS-232 – ein Standardinterface (von der Electronic Industries Association definiert), das ein Modem und ähnliche Terminalgeräte mit einem Computer verbindet.

Speicher – jener Teil oder jene Teile des Computers, die dazu verwendet werden, um Informationen und Programme aufzubewahren, die gerade bearbeitet werden. Außerdem sind dort auch noch die Anweisungen aufbewahrt, wie die Ansprüche des Programmes erfüllt werden können. Der Speicher ist fix in Ihrem Computer. Prinzipiell gibt es zwei Arten von Speichern (siehe RAM und ROM).

S-100-Bus – auch dies ist ein Standardface (siehe RS-232), es besteht aus 100 parallelen Kommunikationswegen, die dazu verwendet werden, Anschlüsse innerhalb eines Mikrocomputers miteinander zu verbinden.

SNOBOL – eine höhere Programmiersprache, die, von den Bell Laboratories entwickelt, Mustererkennung und Stringmanipulation verwendet.

Software – das Programm, dem der Computer folgt (siehe Firmware).

Stack – der letzte Punkt einer Reihe von Ereignissen, die auf einer 'last in, first out'-Basis bearbeitet werden.

Subroutine – der Teil eines Programmes oder eine Gruppe von Programmanweisungen, die innerhalb eines Programmes öfter abgerufen werden.

Syntax – wie in der menschlichen Sprache ist die Syntax auch in der Computersprache die strukturelle Regel, die die Verwendung der Sprache steuert.

System Software – Teile des Programmes, die administrative Aufgaben haben oder beim Schreiben anderer Programme helfen. Sie sind allerdings nicht direkt an der Erreichung des Endzieles des Programmes beteiligt.

Thermischer Drucker – ein Gerät, das den Output eines Computers auf wärmeempfindliches Papier druckt. Obwohl diese Drucker leiser

als andere sind, ist der Ausdruck nicht immer leicht zu lesen. Außerdem ist die Aufbewahrung des Papiers problematisch.

Time-Sharing – dieser Ausdruck wird in bezug auf viele Benutzer an unabhängigen Terminals verwendet, die aber nur einen Computer verwenden. Dieser teilt seine Zeit zwischen den Benutzern so auf, daß es jedem so erscheint, als hätte er die 'volle Aufmerksamkeit' des Computers.

Turnkey-System – ein Computersystem (hauptsächlich in der Geschäftswelt verwendet), das bei Lieferung nur noch des 'Drehens eines Schlüssels' bedarf, um es abzurufen.

Volatile memory (Flüchtiger Speicher) – ein Speicher, der seinen Inhalt verliert, wenn die Stromzufuhr abgestellt wird (siehe memory, refresh, ROM, RAM).

Word processor (Textverarbeitung) – ein zugeteilter Computer (oder ein Computer, auf dem ein Textverarbeitungsprogramm läuft), der an eine 'intelligente Schreibmaschine' angeschlossen ist. Daher steht eine große Zahl von Anpassungs- und Korrekturmöglichkeiten zur Verfügung.

Zeilendrucker – ein Drucker, der eine komplette Zeile von Zeichen auf einmal druckt.

Konversionen

BBC Micro:

Die vorliegenden Programme wurden auf einem BBC Micro erstellt. Um aber die Kompatibilität mit anderen BASIC-Dialekten zu erhalten, wurde in den Programnteilen, die den Drucker steuern, der Befehl "LPRINT" verwendet. Um diese Programnteile auch auf Ihrem Computer einzusetzen, sollten Sie die "LPRINT"-Befehle durch "PRINT"-Anweisungen ersetzen. Vor dem Programmteil für den Drucker sollten Sie den Befehl "VDU2" einbauen, am Ende "VDU3"), um den Drucker wieder auszuschalten. Um die Verträglichkeit mit möglichst vielen Druckern zu erhalten, wurde auf den Einsatz von Procedures und Intervariablen verzichtet.

VC 20:

Fast alle Programme benötigen zusätzlichen Speicher. Durch die geringe Anzahl der Zeichen pro Zeile benötigt das Ausdruckformat einige Adaptionen.

Die "RANDOMIZE"-Befehle sollten ohne "REM"-Befehle eingegeben werden. Sie können generell versuchen, Speicherplatz durch Weglassen der REM-Befehle zu sparen.

CBM 64:

Die "RANDOMIZE"-Befehle sollten ohne REM eingegeben werden, ansonsten sind größere Veränderungen nicht notwendig.

IBM PC:

Wo "RANDOMIZE" ohne eine Variable steht, sollte es durch "RANDOMIZE VAL (WRITE\$(TIMES,2))" ersetzt werden.

Die anderen "RANDOMIZE"-Befehle sollten ohne REM eingegeben werden. Die Funktion RND(1) muß durch RND ersetzt werden.

Sinclair ZX 81/TS 1000:

Bei manchen Programmen – wenn bei Feldern die rechte Spalte oder Zeile verwendet wird – muß das Feld um eins größer dimensioniert und die Ordnungszahl selbst um eins erhöht werden.

DIM A(8,8) wird zu DIM A(9,9)

A(Q,R) wird zu A(Q+1, R+1)

Die Verarbeitung von Zeichenketten erfordert ebenso einige Veränderungen:

MID\$(A\$,x,y) wird zu A\$(X TO Y)
RIGHT\$(A\$,x) wird zu A\$(LEN A\$-X TO)
LEFT\$(A\$,x) wird zu A\$(1 TO X)
Variablenzuweisungen müssen mit dem Kommando "LET" erfolgen.
"RANDOMIZE" sollte ohne REM eingegeben werden.
Mehrfachanweisungen müssen in einzelne Zeilen geteilt werden.
Die Funktion RND(1) muß durch RND ersetzt werden.
Durch die geringe Anzahl von Zeichen pro Zeile benötigt der Ausdruck möglicherweise einige Änderungen. Für alle Programme ist die 16K-Speichererweiterung notwendig.

Sinclair ZX Spectrum:

Die "REM RANDOMIZE"-Befehle müssen durch RAND ersetzt werden.
Bei Variablenfeldern gilt dasselbe wie für den ZX 81/TS 1000, ebenso, was ich für die Zeichenkettenbehandlung geschrieben habe.
Durch die geringere Anzahl von Zeichen pro Zeile benötigt das Ausdruckformat möglicherweise Adaption.
RND(1) muß durch RND ersetzt werden.

Dragon 32/64:

RANDOMIZE sollte ohne REM eingegeben werden.
Änderungen größeren Umfangs sind ansonsten nicht erforderlich.