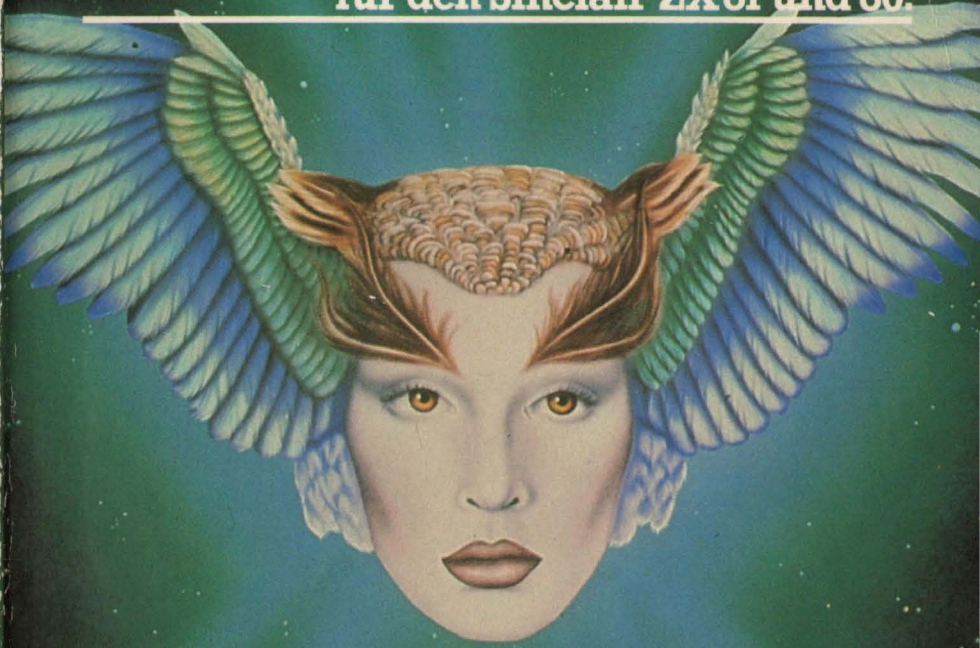
Tim Hartnell

Entdecken Sie die unendlichen Dimensionen Ihres ZX 81

Sämtliche Funktionen, über 100 Super-Programme für den Sinclair ZX 81 und 80.



ZX POWER Cooperation


ZX
POWER
Cooperation

Tim Hartnell

Entdecken Sie die unendlichen Dimensionen Ihres ZX 81

Sämtliche Funktionen,
über 100 Super-Programme
für den Sinclair ZX 81 und 80.

Verlag Cooperation, München

Titel der englischen Originalausgabe
GETTING ACQUAINTED WITH YOUR ZX81
Aus dem Englischen übertragen von Th. Guss.

Für Jürgen
He was never known
to make a foolish move.
Bob Dylan

1. Auflage 1982

Umschlaggrafik: Fred-Jürgen Rogner, London
Umschlagentwurf: Agentur Cooperation, München
Herstellung: Helmut Mayer
Satz: Fotosatz Kretschmann, Bad Aibling

ISBN 3-88945-000-8

Printed in Germany

****

Einführung

****

Dieses kleine Buch wurde geschrieben, um Sie mit interessanten und lohnenden Programmen zu versorgen, vom allerersten Tag an, an dem Sie Ihren Computer bekommen. Die Programme reichen von einfachen Spielen, die wenig Geschicklichkeit erfordern bis zu sehr komplexen »intelligenten« Programmen wie Dame oder Mühle. Es gibt auch einige Programme, die die immensen mathematischen Fähigkeiten des ZX81 demonstrieren, beispielsweise Programme, die Sinuskurven plotten, Daten sortieren, die Wurzeln von Quadratgleichungen ausarbeiten und berechnen, wieviel Zinsen Sie für einen Kredit bezahlen müssen.
Die Programme sind nicht nur für sich selbst wertvoll, sondern demonstrieren auch bestimmte Funktionen des ZX81. Ich glaube, der beste und unterhaltsamste Weg, sich mit Computern zu befassen ist, mit ihnen zu spielen, Funktionen auszuprobieren, die Programmlistings zu untersuchen und zu entdecken, wie und warum der Computer in diesem Programm so und nicht anders arbeitet. In weiterer Folge verändern und verbessern Sie die Programme, sodaß sie Ihre eigenen werden.
Alle Programme in diesem Buch laufen ohne Veränderung des ZX81. Viele von ihnen laufen auf einem mit neuem ROM ausgestatteten ZX80 genauso wie aufgelistet. Andere brauchen die Eingabe eines Strings (in der Form INPUT A $, manchmal auch gefolgt von einem CLS-Befehl als nächste Zeile), um den Ablauf lang genug anzuhalten, um zu sehen, was geschieht. In anderen kann zu diesem Zweck eine PAUSE-Funktion benutzt werden. Sie werden sehen, daß es einfach ist, die notwendigen Änderungen durchzuführen. Wenige (sehr wenige) der Programme laufen nicht mit dem neuen ROM.
Sie werden manchmal in Programmen das sehr eigenartige Symbol # bemerken. Versuchen Sie NICHT, dieses einzugeben. Es steht für einen einzelnen Zwischenraum, wenn dieser Zwischenraum unbedingt notwendig und nicht aus dem Zusammenhang ersichtlich ist. Der benötigte Speicher für jedes Programm ist entweder angegeben (wenn 1 K angegeben ist, be-

deutet das, daß dieses Programm in einen ZX81 ohne jede Erweiterung paßt) oder nicht (wenn Sie voraussetzen können, daß dieses Programm mehr als 1 K braucht, obwohl viele Programme gekürzt werden können und so in eine 1 K-Maschine passen).
Wie Sie wissen, können Sie inverse Buchstaben (das bedeutet weiß auf schwarzem Hintergrund) direkt von der Tastatur aus eingeben. Bei den Programmen in diesem Buch können Sie jeden PRINT-Gegenstand, den Sie wollen, invertieren. Aber dort, wo ich glaube, daß es besonders notwendig ist, habe ich die entsprechenden Buchstaben unterstrichen.

5∅ PRINT »SIE HABEN GEWONNEN«

Sie werden bemerken, daß inverse PRINT-Zeilen manchmal leichter zu lesen sind, wenn sie einen invertierten Zwischenraum an Anfang und Ende der Zeichenkette setzen.

Tim Hartnell, Oktober 1981, London

WICHTIGER HINWEIS!

Die Programmlistings in diesem Buch wurden wegen der besseren Lesbarkeit mit einem Normalpapierdrucker ausgedruckt. Nur die Grafikzeichen wurden in die Listings einmontiert.

Das Zeichen # in den Programmen, wie auch im Text beschrieben, steht für 1 Zwischenraum.

Unterstrichene Angaben in den Programmen bedeuten inverse Buchstaben (weiß auf schwarzem Hintergrund); Sie können jeden PRINT-Gegenstand in diesem Buch invertieren.

Eine Null wird in diesem Buch so dargestellt: ∅ und das Multiplikationszeichen sieht so aus: *

****

Fehlercodes

****

Die Fehlercodes Ø—9 kommen sowohl beim ZX8O als auch beim ZX8l vor, die Fehlercodes A—F nur beim ZX8l (und beim ZX8O mit neuen ROM).

Codebedeutungen

CODE Ø ... erfolgreiche Beendigung
CODE 1 ... NEXT ohne FOR
CODE 2 ... Variable wurde nicht definiert
CODE 3 ... Der Index befindet sich außerhalb des erlaubten Bereiches
CODE 4 ... Speicherplatz vollständig belegt
CODE 5 ... Bildschirm voll
CODE 6 ... Arithmetischer Überlauffehler
CODE 7 ... Return ohne GOSUB.
CODE 8 ... INPUT wurde versucht als Kommando einzugeben, das ist verboten.
CODE 9 ... Stop-Befehl ausgeführt
CODE A ... Nicht erlaubtes Argument für bestimmte Funktionen
CODE B ... Zahl außerhalb eines zulässigen Bereiches
CODE C ... Der Text des Argumentes von VAL ergibt keinen gültigen numerischen Ausdruck.
CODE D ... Programm wurde mit BREAK unterbrochen
CODE E ... wird nicht verwendet
CODE F ... Sie haben eine Leerkette als Programmname benützt, das ist nicht zulässig.

****

»Erste Schritte«

****

Beginnen wir, indem wir den ZX81 anstecken, um den Cursor (das ist das weiße K in einem kleinen schwarzen Kästchen) in die linke untere Ecke Ihres Fernsehers zu bekommen.
Geben Sie das folgende Programm ein, das eine Serie von Zufallszahlen zwischen 1 und 10 generiert. Beachten Sie, daß das # einen Zwischenraum in einem PRINT-Gegenstand bedeutet. Wir werden dieses Symbol in Programmen verwenden, wenn der Zwischenraum notwendig und nicht aus dem Zusammenhang ersichtlich ist.

```
10 SLOW
20 PRINT INT (RND*10)+1;"#";
30 RUN
```

Dieses Programm ergibt einen Bildschirm voll mit zufällig erzeugten Zahlen zwischen 1 und 10. So wie sie sind, haben diese Zahlen keinen großen Nutzen, bekommen ihn aber, wenn sie als Teil eines Spieles verwendet werden. Das folgende Programm verwendet den Zufallszahlengenerator, um Zahlen zwischen 1 und 6 für das Spiel »Russisches Roulette« zu erzeugen. Beachten Sie, daß Sie nur 1 K Speicher für dieses Spiel brauchen, sodaß Sie kein Zusatzmodul anstecken müssen. Auch der Standard-ZX81 kann sich dieses Programm merken.
Das Prinzip von Russischem Roulette ist einfach. Sie haben eine Pistole mit 6 Kammern, doch nur eine enthält eine Patrone. Sie halten die Pistole an Ihren Kopf, ziehen am Abzug und ... entweder »Peng« oder »Klick«. Geben Sie das Programm ein, lassen Sie es einige Male laufen, nehmen Sie dann wieder das Buch zur Hand für eine Erklärung der Dinge, die Sie aus diesem Programm lernen können.

```
  10 PRINT "WIE HEISSEN SIE?"
  30 INPUT A$
  50 PRINT "WOLLEN SIE SPIELEN,"
;A$;"?"
  60 INPUT B$
```

```
 70 CLS
 80 IF B$="NEIN" THEN GOTO 270
 90 LET J=0
100 PRINT "DREUCKEN SIE N/L"
110 INPUT C$
120 CLS
130 LET J=J+1
140 LET G=INT (RND*6)+1
170 IF G<6 THEN PRINT "KLICK"
190 IF G=6 THEN GOTO 220
200 IF J=10 THEN GOTO 250
210 GOTO 100
220 CLS
230 PRINT "BUMM...";
240 GOTO 230
250 PRINT A$;"#UEBERLEBT#";
260 GOTO 250
270 CLS
280 PRINT "FEIGLING...";
290 GOTO 280
```

Es gibt eine unglaubliche Anzahl von nützlichen Dingen, die Sie aus diesem ersten Programm lernen können.
Zunächst sehen Sie, daß jede Zeile mit einer Zeilennummer beginnt. Beim ZX81 können diese Zeilennummern zwischen 1 und 9999 liegen, aber Sie werden herausfinden, daß es am besten ist, wenn Sie mit Vielfachen von 10 arbeiten, weil Sie so auch später noch Zeilen einfügen können.
Zusätzliche Zeilen sortieren sich selbst in die richtige Reihenfolge und der Computer arbeitet alle Zeilen jeweils von der niedrigsten zur höchsten Nummer ab, solange nicht das Programm befiehlt, zu einer anderen Zeile zu springen. Nach der ersten Zeilennummer haben wir das Wort PRINT, wahrscheinlich die meistbenützte Anweisung in der Computersprache BASIC, der Sprache Ihres ZX81. An das Wort PRINT schließen Sie das »Wort« an, von dem Sie wollen, daß es der ZX81 ausdruckt. In diesem Fall sind es die Worte »IHR NAME«.
Versuchen Sie vor die ersten Anführungszeichen ein Kommazeichen zu setzen, sodaß die Zeile 10 nun lautet: 10 PRINT, »IHR NAME?«.
Lassen Sie das Programm wieder laufen und Sie werden bemerken, daß der ZX81 die Worte über den Bildschirm bewegt hat. Der Gebrauch des Kommas ist ideal, um PRINT-Gegenstände in Kolumnen anzuordnen (genauso ist es ideal, um langweilige Zahlentafeln auszudrucken). Der Computer stoppte, als er zur

Zeile 3Ø (INPUT A $) kam. Dieses A $ (ein Buchstabe, gefolgt von einem Dollar-Zeichen) heißt in der Computersprache BASIC String und kann jede Kombination von Buchstaben, Zeichen und Zahlen sein. Für den Anfang werden wir bei Buchstaben bleiben. Der Computer stoppte also und wartete auf eine Eingabe — Ihren Namen. Nachdem Sie Ihren Namen eingetippt und NEWLINE gedrückt hatten, sodaß es der Computer akzeptieren konnte, verstand der Computer, was A $ bedeuten sollte und verwendete es, wie Sie ja sicher auch bemerkt haben, das nächstemal, als A $ im Programm vorkam, um Ihren Namen auszudrucken.

Als nächstes fragte Sie der Computer, ob Sie spielen wollen (Zeile 5Ø) und akzeptierte einen zweiten String (B $) in Zeile 6Ø als Ihre Antwort. Zeile 7Ø (CLS) löschte den Bildschirm, aber erlaubte dem ZX81 nicht, den Inhalt von B $ zu vergessen.

Der ZX81 kann wie alle Computer Entscheidungen treffen und danach handeln. In diesem Fall kontrolliert er in Zeile 6Ø Ihre Antwort (B $) und springt im Falle, daß die Antwort NEIN ist (d.h. B $ = »NEIN«) zu Zeile 27Ø, die den Bildschirm löscht.

Anschließend geht der ZX zur Zeile 28Ø, wo das Wort »Hühnchen« ausgedruckt wird. Zeile 29Ø läßt den ZX81 zu Zeile 28Ø zurückspringen, immer und immer wieder, so lange, bis der Bildschirm voll ist und das Programm stoppt.

Sie werden sehen, daß das Wort »Hühnchen« auf dem Bildschirm ein recht attraktives Muster ergibt und wenn Sie etwas experimentieren wollen, nachdem Sie diesen Absatz gelesen haben, verwenden Sie eine ewige Schleife und einige graphische Symbole in einem PRINT-Befehl, um den Bildschirm mit Mustern zu füllen.

Aber wieder zurück zum Russischen Roulette. Wenn Sie auf die Frage, ob Sie spielen wollen, nicht mit NEIN antworten, fährt der Computer zu Zeile 9Ø, wo der Variablen J der Wert Ø zugewiesen wird.

Im SINCLAIR-BASIC kann jedem Buchstaben oder jeder Buchstabenkombination oder jeder Zahlenkombination, der ein Buchstabe vorangestellt wird, ein Wert zugewiesen werden. Zeile 9Ø könnte also z.B. lauten LET SUM = Ø oder LET P Q = Ø. Sie befolgen also nun die Anweisung, die mit Zeile 1ØØ ausgedruckt wird (Drücken Sie NEWLINE) und der Bildschirm wird jetzt wieder gelöscht (Zeile 12Ø). Zeile 13Ø addiert 1 zum Wert von J, sodaß J nun den Wert 1 hat. Das

nächstemal, wenn das Programm durchlaufen wird, wird wieder 1 hinzugezählt, also wird J den Wert 2 haben usw.
Zeile 14Ø (LET G=INT (RND *6)+1) verwendet natürlich den Zufallszahlengenerator.
Zeile 17Ø trifft eine andere Entscheidung. Die Art dieser Anweisung wird IF/THEN-Anweisung genannt, dies wegen ihrer Form IF (wenn) ein Ereignis eintrifft, THEN tue das und das (wenn der Reifen flach ist, dann pumpe ihn auf). Die Bedingung kann sehr vielfältig sein, von IF X=96 über IF A $=»Frosch« bis zu komplizierteren Bedingungen wie IF A—B=2 * E—Y/4 THEN...
In diesem Fall überprüft die IF/THEN-Anweisung, ob der Zufallszahlengenerator eine Zahl kleiner als 6 generiert hat und für den Fall, daß dies eintritt, wird KLIK ausgedruckt. Für den Fall, daß das nicht zutrifft, sucht der ZX81 das komplette Programm ab, Zeile für Zeile, so lange, bis er etwas findet, was er tun kann. Wenn G=6 ist (Zeile 19Ø), wird die Zeile 27Ø bearbeitet, die eine ähnliche Methode benutzt, wie wir sie vorher besprochen haben, als es darum ging, das HÜHNCHEN auszudrucken. In diesem Fall wird das fatale Wort PENG ausgedruckt.
Unter der Voraussetzung, daß G kleiner als 6 ist, wird der Computer in Zeile 17Ø KLIK ausdrucken, Zeile 19Ø ignorieren und dann zu Zeile 2ØØ kommen. Wenn er dort ankommt, wird er überprüfen, welcher Wert J zugeordnet wurde. Wenn J gleich 1Ø ist, ist das Spiel gewonnen, also wird der Computer zu Zeile 25Ø gehen, um Ihnen das mitzuteilen. Wenn C noch nicht gleich 1Ø ist, ignoriert der Computer die Zeile 2ØØ und findet dann in Zeile 21Ø eine Anweisung, die er ausführen kann und geht dann zurück zu Zeile 1ØØ. Nachdem Sie diese etwas verwirrende Erklärung einige Male gelesen haben und sich selbst einige Male mit NEWLINE erschossen haben, würde ich Sie bitten (bevor Sie schummeln und lesen, wie ich es gemacht habe), einige Zeilen hinzuzufügen oder zu ändern, sodaß das Programm den Spieler zu einem neuen Spiel einlädt, wenn er das erste Spiel überlebt hat, ohne daß RUN eingegeben werden muß.
Nachdem Sie Ihre eigene Version dieser Änderung gemacht haben — und es gibt eine große Anzahl an Möglichkeiten — lesen Sie bitte weiter und sehen Sie, wie ich es gemacht habe. Die wahrscheinlich einfachste Möglichkeit ist, folgende Zeile hinzuzufügen:

```
255 PRINT "NOCHEINMAL?"
```

Zeile 26Ø müssen Sie dann in GOTO 6Ø ändern. Wenn Sie dies tun, verwenden Sie die Routine der Zeilen 6Ø, 7Ø und 8Ø zweimal. Sie können das Programm auch anders geändert haben. Es ist gleichgültig, welche Änderungen Sie gemacht haben, solange sie das gewünschte Ende bewirken. Es gibt eine andere Möglichkeit, mit der man den gleichen Effekt erzielen kann: Der Einsatz einer Subroutine.
Immer dann, wenn der Computer zu einem GOSUB (GOtoSUBroutine) kommt, springt er zur angegebenen Zeile und macht dort so lange weiter, bis er zum Kommando RETURN kommt. Der Computer geht dann zur Zeile NACH dem GOSUB-Kommando zurück. Machen Sie die folgenden Änderungen bei Ihrem Russischen Roulette-Programm und der Sachverhalt wird Ihnen klarer werden.

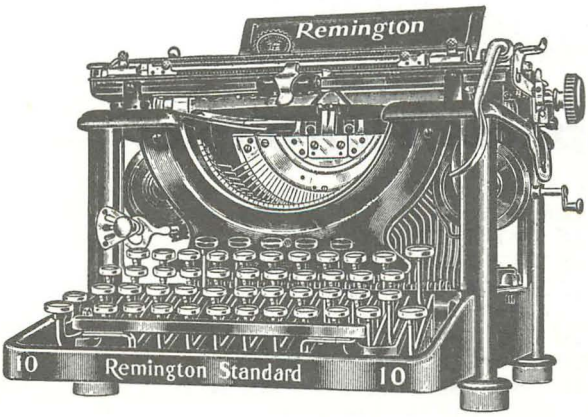
```
300 INPUT B$
310 CLS
320 IF B$="NEIN" THEN GOTO 270
330 RETURN
```

Wenn Sie das Programm ablaufen lassen, wird der Computer jedesmal zur Subroutine geschickt, wenn er zur Zeile 6Ø oder 26Ø kommt.

Hier ist ein anderes Programm mit dem Namen BEAT THE DEALER, in dem eine Subroutine (mit Start bei Zeile 3ØØ) über Ihr Geld buchführt und eine zweite (mit Start bei Zeile 4ØØ) zwei Würfel rollen läßt.
BEAT THE DEALER beginnt damit, daß der ZX81 zwei Würfel wirft. Die Summe der beiden Würfel wird gespeichert. Als nächstes wirft der Mensch die Würfel. Wenn die Gesamtsumme des Menschen größer ist als die des ZX81, gewinnt der Mensch 5.5Ø Pfund und der Computer verliert die gleiche Menge. Für den Fall, daß der Punktestand des Menschen der gleiche ist oder kleiner als der des ZX81, bekommt der Computer die 5.5Ø Pfund und der Mensch verliert diesen Betrag. Nach einigen Runden wird Ihnen das klar werden und wenn Sie das Programm in Aktion gesehen haben, werden Sie viel besser verstehen, was eine Subroutine tut.

```
10 LET M=30
20 LET N=M
30 LET Z=5.5
40 GOSUB 300
50 GOSUB 400
```

```
 60 LET C=A+B
 65 SCROLL
 70 PRINT "ICH:";A;"#";B,C
 80 INPUT A$
 90 GOSUB 400
100 LET D=A+B
105 SCROLL
110 PRINT "SIE:";A;"#";B,D
120 IF D>C THEN GOTO 160
130 LET M=M-Z
140 LET N=N+Z
150 GOTO 40
160 LET M=M+Z
170 LET N=N-Z
180 GOTO 40
300 SCROLL
310 SCROLL
320 PRINT "SIE $";M,"ICH $";N
330 INPUT A$
340 RETURN
400 LET A=INT (RND*6)+1
410 LET B=INT (RND*6)+1
420 RETURN
```



****

Ernsthafte Anwendungen

****

SINCLAIR Research hat dem ZX81 viele nützliche Funktionen gegeben, die auf dem ZX80 unmöglich waren. Dieses Kapitel betrachtet einige der neuen Möglichkeiten.
Hier ist eine Anzahl von kurzen Programmen — geschrieben von Tony Baker — die die Graphik-Fähigkeiten des ZX81 demonstrieren. Lassen Sie diese im SLOW-MODE laufen.

Oval

```
  10 FOR A=1 TO 100
  20 LET B=PI*A/50
  30 PRINT AT 9*COS B+10,14*SIN
B+15;"#"
  40 NEXT A
```

Überlappende Spiralen

```
  10 FOR A=1 TO 400
  20 LET B=PI*A/50
  30 LET C=(400-A)/400
  40 PLOT (20.5*COS B+30)*C,(20*
SIN B+20)*C
  50 NEXT A
```

Spirale, Plot/Unplot

```
   5 FOR D=1 TO 2
  10 FOR X=1 TO 400
  15 IF D=1 THEN LET A=X
  17 IF D=2 THEN LET A=400-X
  20 LET C=(400-A)/400
  25 LET B=PI*A/50
  30 IF D=1 THEN PLOT (20.5*COS
B)*C+30,(20*SIN B)*C+20
  35 IF D=2 THEN UNPLOT (20.5*CO
S B)*C+30,(20*SIN B)*C+20
  40 NEXT X
  50 NEXT D
```

Sinuskurven

```
10 FOR A=2 TO 120
20 LET B=A*PI/30
30 PLOT A/2,SIN B*20+20
40 NEXT A
```

Doppelsinuskurve

```
 5 FOR C=1 TO 2
10 FOR A=2 TO 120
15 IF C=1 THEN LET B=A*PI/60
20 IF C=2 THEN LET B=-A*PI/60
25 PLOT A/2,SIN B*20+20
30 NEXT A
35 NEXT C
```

Sin/Cos

```
10 FOR A=2 TO 120
20 LET B=A*PI/60
30 PLOT A/2,SIN B*20+20
40 PLOT A/2,COS B*20+20
50 NEXT A
```

Fallende Sinuskurven

```
 5 DIM S(4)
10 FOR Z=1 TO 8
15 FOR A=1 TO 120
20 LET S(1)=SIN (A*PI/60)*20+2
0
25 LET S(2)=COS (A*PI/60)*20+2
0
30 LET S(3)=40-S(1)
35 LET S(4)=40-S(2)
40 LET J=Z-INT (Z/4)*4+1
45 LET K=Z+3-INT ((Z+2)/4)*4
50 UNPLOT A/2,S(J)
60 PLOT A/2,S(K)
65 NEXT A
70 NEXT Z
```

Plot-Demonstration

Sie können einige dieser Programme zusammenhängen, wenn Sie genug Speicherplatz haben, und so ein effektvolles Demonstrationsprogramm erhalten.

```
  10 FOR A=1 TO 100
  20 LET B=PI*A/50
  30 PRINT AT 9*COS B+10,14*SIN
B+15;"#"
  40 NEXT A
  50 PAUSE 200
  60 CLS
  70 FOR A=1 TO 400
  80 LET B=PI*A/50
  85 LET C=(400-A)/400
  90 PLOT (20.5*COS B+30)*C,(20*
SIN B+20)*C
 100 NEXT A
 110 PAUSE 200
 120 CLS
 130 FOR A=2 TO 120
 140 LET B=A*PI/30
 150 PLOT A/2,SIN B*20+20
 160 NEXT A
 170 PAUSE 200
 180 CLS
 190 FOR C=1 TO 2
 200 FOR A=2 TO 120
 210 IF C=1 THEN LET B=A*PI/60
 220 IF C=2 THEN LET B=-A*PI/60
 230 PLOT A/2,SIN B*20+20
 240 NEXT A
 250 NEXT C
```

Wenn Sie eine ununterbroche Demonstration wollen, fügen Sie folgende Zeilen hinzu:

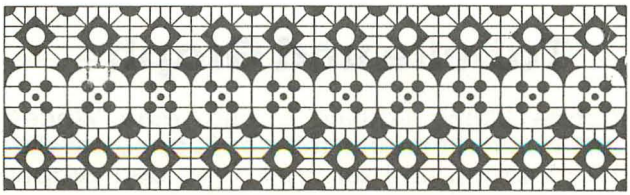
```
 260 PAUSE 200
 270 CLS
 280 RUN
```

Multiplikationsquiz

Der ZX81 kann benutzt werden, um Zahlenquizzes zu erzeugen und die Antworten des Benützers zu überprüfen.

```
  10 LET Z=0
  20 GOSUB 460
  30 PRINT TAB 4;"MULTIPLIKATION
STEST"
  40 GOSUB 460
  50 PRINT "SCHWIERIGKEITSGRAD (
1-10)?"
  60 INPUT A
  70 IF A<1 OR A>10 THEN GOTO 60
 100 GOSUB 460
 110 PRINT "WIE VIELE AUFGABEN?"
```

```
 120 INPUT B
 130 IF B<1 THEN GOTO 120
 140 CLS
 150 FOR G=1 TO B
 160 LET C=A*INT (RND*10)+1
 170 LET D=A*INT (RND*10)+1
 180 LET E=C*D
 190 GOSUB 460
 200 PRINT "AUFGABE NR.";G
 210 GOSUB 460
 220 PRINT "WIEVIEL GIBT#";C;"#M
 230 INPUT F
 240 PRINT "####";F
 250 GOSUB 460
 260 IF F=E THEN GOTO 420
 270 PRINT "FALSCH.DAS ERGEBNIS
IST#";E
 280 GOSUB 460
 290 PRINT TAB 8;"IHR STAND IST#
";Z
 300 PRINT TAB 9;"RICHTIG VON#";
G
 310 PRINT "DRUECKEN SIE N/L#";
 320 IF NOT G=B THEN PRINT "UM W
EITERZU-###MACHEN"
 330 INPUT A$
 340 CLS
 350 NEXT G
 360 FOR K=1 TO 3
 370 GOSUB 460
 380 NEXT K
 390 PRINT "ENDE DES TESTS.IHR S
TAND IST"
 400 PRINT TAB 8;INT (Z*100/(G-1
));"#PROZENT"
 410 STOP
 420 LET Z=Z+1
 440 PRINT "RICHTIG.DAS ERGEBNIS
 IST#";E
 450 GOTO 280
 460 FOR S=1 TO 3
 470 PRINT
 480 NEXT S
 490 RETURN
```



Quadrate

(Demonstration von SCROLL)

Das nächste Programm zeigt die SCROLL-Funktion in der Anwendung:

```
 10 LET K=0
 20 GOSUB 220
 30 PRINT TAB 8;"QUADRATE"
 40 GOSUB 220
 50 PRINT TAB 6;"BEGINNZAHL?"
 60 INPUT A
 70 GOSUB 220
 80 PRINT TAB 6;"ENDZAHL?"
 90 INPUT B
100 GOSUB 220
110 PRINT "DRUECKEN SIE N/L FUE
R DIE","TABELLE"
120 INPUT A$
130 CLS
140 FOR X=A TO B STEP (1 AND A<
B)-(1 AND A>B)
150 LET K=K+1
160 IF K>22 THEN SCROLL
170 PRINT TAB 4;X;"#QUADRIERT I
ST#";X*X
180 NEXT X
185 IF K>20 THEN SCROLL
190 PRINT
195 IF K>20 THEN SCROLL
200 PRINT TAB 8;"ENDE DER TABEL
LE"
210 STOP
220 FOR C=1 TO 3
230 PRINT
240 NEXT C
250 RETURN
```

Variable K fungiert als Zählvariable, um das SCROLLen zu starten, wenn der Bildschirm beinahe voll ist. Sie können diese Methode benutzen, wenn die Programmausgabe sequentiell erfolgt und länger als eine Bildschirmseite voll ist, sodaß die Gefahr eines Programmabsturzes besteht.

Tabelle und Graph

Die nächste kleine Routine erzeugt eine Werte-Tabelle in Übereinstimmung mit der Formel, die Sie in Zeile 9Ø plazieren und zeichnet dann den Graphen.

Sie können leicht auf den Ausdruck verzichten, wenn Sie die Zeilen 13Ø bis 16Ø löschen. Sie werden auch mit dem verwendeten Maßstab experimentieren müssen, um den effektvollsten Ausdruck Ihrer Formel zu erhalten.

```
 10 PRINT "NIEDRIGSTER WERT (X)
UM ZU","PLOTTEN?"
 20 INPUT A
 30 PRINT A,"HOECHSTER?"
 40 INPUT B
 50 CLS
 60 DIM C(B)
 70 FOR D=A TO B
 80 SCROLL
 90 LET C(D)=    IHRE FORMEL
100 PRINT D,C(D)
110 NEXT D
120 PAUSE 300
130 CLS
140 FOR D=A TO B
150 PLOT D,C(D)
160 NEXT D
```

Testen Sie dieses Programm mit A=1 und B=6Ø unter der Verwendung der folgenden Formel:

```
LET C(D)=20*COS D
LET C(D)=D*D/100
LET C(D)=5*SQR D
```

Artithmetischer Durchschnitt

Dieses Programm akzeptiert eine Serie von Zahlen und arbeitet dann ihren Durchschnitt aus. Es verwendet die Formel:

$$\bar{A} = \frac{1}{N} \sum_{i=1}^{N} a$$

```
 10 LET C=0
 20 PRINT "WIE VIELE ZAHLEN WOL
LEN SIE","EINGEBEN?"
 30 INPUT A
 35 SCROLL
 40 FOR B=1 TO A
 50 SCROLL
```

```
 60 PRINT "ZAHL#";B;"#?"
 70 INPUT D
 80 SCROLL
 90 PRINT D,"NOCH#";A-B
100 LET C=C+D
110 NEXT B
120 CLS
130 PRINT "GESAMT WAR#";C
140 PRINT
150 PRINT "DURCHSCHNITT IST:";(
INT (C/A)*100)/100
```

Zeile 15Ø sorgt dafür, daß der Durchschnitt auf höchstens zwei Dezimalstellen berechnet wird. Ändern Sie die 1ØØ auf 1Ø für eine Dezimalstelle, auf 1ØØØ für drei usw.

Data Sort

Das Programm sortiert bis zu 1ØØ Zahlen in abfallende Reihenfolge. Geben Sie die Zahlen (positiv und/oder negativ) nacheinander ein, wie verlangt. Wenn Sie wollen, daß sie der ZX81 von der höchsten bis zur niedrigsten Zahl sortiert, geben Sie Ø ein.

```
  5 DIM Q(110)
 10 LET Z=0
 20 LET Y=10
 30 SCROLL
 35 PRINT "GEBEN SIE ZAHL#";Z+1
;"#EIN:";
 40 INPUT Q(Y)
 50 IF Q(Y)=0 THEN GOTO 100
 60 LET Z=Z+1
 70 PRINT Q(Y)
 80 LET Y=Y+1
 90 GOTO 30
100 CLS
110 FOR X=10 TO Z+8
120 FOR N=X+1 TO Z+9
130 IF Q(X)>Q(N) THEN GOTO 170
140 LET E=Q(X)
150 LET Q(X)=Q(N)
160 LET Q(N)=E
170 NEXT N
180 NEXT X
190 FOR N=1 TO Z
200 SCROLL
210 PRINT N,Q(N+9)
220 NEXT N
```

Quadratische Gleichung

$$ax^2+bx+c=\emptyset \ (a\neq\emptyset)$$

$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

Beide Wurzeln werden ausgedruckt, wenn es zwei echte Wurzeln gibt, eine Wurzel, wenn die beiden Wurzeln gleich sind und sowohl der echte als auch der imaginäre Teil von imaginären Wurzeln. Es ist leicht am Ausdruck zu sehen, welche Wurzeln ausgedruckt werden.

```
  5 PRINT "GEBEN SIE A EIN:"
 10 INPUT X
 15 PRINT X,"GEBEN SIE B EIN:"
 20 INPUT Y
 25 PRINT Y,"GEBEN SIE C EIN:"
 30 INPUT Z
 35 CLS
 40 LET Y=-Y/2/X
 45 LET Q=Y*Y-Z/X
 50 IF Q=0 THEN GOTO 200
 55 IF Q>0 THEN GOTO 300
 60 LET E=SQR (-Q)
 65 PRINT "IMAGINAERE WURZELN:"
 70 PRINT """ECHTE"" WURZEL:";Y
 75 PRINT "IMAGINAERE WURZEL:";
E
 80 STOP
200 PRINT "GLEICHE WURZELN:";Y
210 STOP
300 PRINT "ECHTE WURZELN:"
310 PRINT ,Y+SQR Q
320 PRINT ,Y-SQR Q
```

Standardabweichung, Durchschnitt, Varianz

Das Varianzprogramm verwendet die folgenden Formeln:

$$\text{Durchschnitt} = \frac{\sum x}{n} = \bar{x}$$

$$\text{Varianz} = \frac{\sum x^2}{n} \quad \bar{x}^2 = V$$

$$\text{Standartabweichung} = \sqrt{V}$$

Um dieses Programm laufen zu lassen, geben Sie zuerst die Zahl und dann die Häufigkeit ein. Falls Sie einen Fehler machen, geben Sie zuerst den Buchstaben E, dann die Positionsnummer, bei der der Fehler gemacht wurde, ein. Sie können die Zeile dann neu eintippen. Sind alle Daten eingegeben, tippen Sie den Buchstaben Q, um die Berechnung zu starten.

```
  10 PRINT "EINGABE###ZAHL####FR
EQUENZ"
  20 PRINT "
"
  30 LET E=RND
  40 LET Q=RND
  50 DIM X(50)
  60 DIM F(50)
  70 LET A=0
  80 LET B=0
  90 LET A=A+1
 100 LET B=B+1
 110 IF B>19 THEN SCROLL
 120 PRINT A;
 130 INPUT X
 140 IF X=E THEN GOTO 210
 150 IF X=Q THEN GOTO 260
 160 LET X(A)=X
 170 PRINT TAB 10;X;
 180 INPUT F(A)
 190 PRINT TAB 21;F(A)
 200 GOTO 90
 210 LET B=B+1
 220 IF B>19 THEN SCROLL
 230 PRINT TAB 0;"EINGABE?"
 240 INPUT A
```

```
250 GOTO 100
260 LET N=0
270 LET A=0
280 LET B=0
290 FOR Z=1 TO 50
300 LET N=N+F(Z)
310 LET A=A+F(Z)*X(Z)
320 LET B=B+F(Z)*X(Z)*X(Z)
330 NEXT Z
340 CLS
350 PRINT AT 5,0;"DURCHSCHNITT:
",,,A/N
360 PRINT ,,"VARIANZ:",,,B/N-A*
A/(N*N)
370 PRINT ,,"STANDARDABWEICHUNG
:",,SQR (B/N-A*A/(N*N))
```

Wenn Sie das Programm etwas aufmotzen wollen, um zu zeigen, daß es tatsächlich etwas tut, fügen Sie die folgenden Zeilen hinzu:

```
295 PRINT AT 21,10;"ICH DENKE "
305 PRINT AT 21,10;"ICH DENKE "
315 PRINT AT 21,10;"ICH DENKE "
325 PRINT AT 21,10;"ICH DENKE "
```

—Lösung von Gleichungen—

Das folgende Programm kann benutzt werden, um **jede** Gleichung zu lösen, in der x = Ø ist. Das Programm druckt auch die Lösungen, auf die es während der Berechnung stößt und die Abweichung aus. Es ist faszinierend, dem Programm bei der Arbeit zuzusehen. Die Lösung ist auf 9 Dezimalstellen richtig.

Wenn Sie beispielsweise das x der Gleichung x*x—sin x = Ø finden wollen, würden Sie nach der Aufforderung des Computers in Zeile 3Ø x * x—sin x eingeben. Wenn Sie wissen, daß die Antwort, die Sie erhalten, zwischen Ø,8 und Ø,9 liegt, geben Sie diese beiden Zahlen ein. Wenn Sie keine Vorstellung von der Lösung haben, geben Sie einfach zwei **verschiedene** zufällig gewählte Zahlen ein und überlassen Sie den Rest dem ZX81.

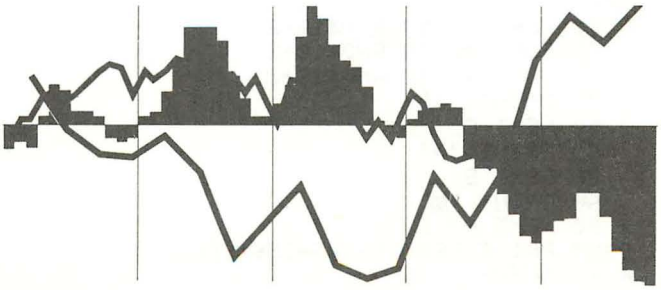
Wenn Sie die Quadratwurzel einer Zahl finden wollen, geben Sie x * 3 minus der Zahl ein (z.B. x * 3—2,79, wenn Sie die dritte Wurzel von 2,79 wollen). Um Quadratwurzeln zu bekommen, geben Sie x * 2 minus der Zahl ein.

```
  10 PRINT "VERVOLLSTAENDIGEN SI
E DIESE","GLEICHUNG ALS FUNKTION
 VON X:"
  20 PRINT "0=";
  30 INPUT A$
  40 PRINT A$
  45 PRINT
  50 PRINT "BITTE GEBEN SIE ZWEI
 UNGLEICHE##ZAHLEN EIN,VON DENEN
 SIE"
  55 PRINT "GLAUBEN,DASS DIE LOE
SUNG DAZWI-#SCHEN LIEGT."
  60 INPUT A
  65 PRINT
  70 PRINT A,
  80 LET X=A
  90 LET F=VAL A$
 100 IF F<>0 THEN GOTO 130
 110 PRINT AT 13,3;"DAS ERGEBNIS
 IST",A
 120 STOP
 130 INPUT B
 140 PRINT B
 150 PRINT AT 10,5;"ERGEBNIS:"
 160 PRINT AT 11,5;"FEHLER:"
 170 LET X=B
 180 LET G=VAL A$
 190 PRINT AT 10,14;A
 200 PRINT AT 11,12;ABS (G-F)
 210 IF ABS (F-G)>1E-9 AND G<>0
THEN GOTO 240
 220 LET A=B
 230 GOTO 110
 240 LET X=(G*A-F*B)/(G-F)
 245 LET F=G
 250 LET A=B
 260 LET B=X
 270 GOTO 180
```

Zinsenberechnung

Der ZX81 kann auch für praktische Aufgaben eingesetzt werden. Das folgende Programm arbeitet aus, wieviel Zinsen für einen Kredit bezahlt werden müssen. Es akzeptiert die Länge der Kreditdauer in Tagen, aber wenn Sie herausfinden wollen, wie hoch die Zinsen bei einer Kreditdauer von mehreren Jahren sind, geben Sie die Anzahl der Jahre (z.B. 45) multipliziert mit 365 ein (das bedeutet, wenn der ZX81 nach der Länge der Kreditdauer fragt, geben Sie für 25 Jahre 25 * 365 ein). Sie können das Programm aber selbstverständlich auch so ändern, daß es die Eingabe in Jahren statt in Tagen akzeptiert.

```
  10 PRINT "PROZENTUELLE VERZINS
UNGSRATE?"
  20 INPUT V
  30 PRINT V
  40 PRINT "STAMMKAPITAL?"
  50 INPUT S
  60 PRINT S
  70 PRINT "DAUER DES KREDITES I
N TAGEN?"
  80 INPUT T
  90 CLS
 100 LET V=INT (S*T/36500*V+.9)
 110 PRINT "ZINSEN:";V
 120 PRINT
 130 PRINT "GESAMT (KAPITAL+ZINS
EN):";S+V
```



****

Was tut man mit 1 K?

****

Viele ZX81 Anwender, die Erfahrung mit der Arbeit auf einem ZX80 haben, fühlen sich beengt durch den kleinen Speicher, den sie mit einer 1 K Maschine zur Verfügung haben. Durch eine Vielzahl von Gründen sind 1 K auf dem ZX81 nicht so effizient wie die gleiche Speichermenge auf dem ZX80, unter anderem:

* Systemvariablen brauchen 125 Bytes auf dem ZX81 und nur 40 beim ZX80.
* Zeilennummern brauchen 3 Bytes beim alten ROM (2 für die Zeilenzahl, eine für das Ende der Zeile) und brauchen 5 Bytes beim ZX81.
* Zahlen sind die großen Platzverschwender beim ZX81. Eine Zahl braucht die Anzahl ihrer Ziffern plus 1 Byte. Wie Sie sehen werden, ist es hinsichtlich des Speicherplatzes billiger, LET P=1 zu sagen und dann P statt 1 zu verwenden, wenn Sie mehr als drei Mal die Ziffer 1 in Ihrem Programm haben. LET Q=P+P ist billiger als LET Q=2.

Sie können feststellen, wie viele Bytes Sie bei einem Programm verbraucht haben, indem Sie die folgende Zeile direkt eingeben:

```
PRINT PEEK 16396+256*PEEK 16397-
16509
```

Es ist auch nützlich, die Zeile als Zeile 9999 einzugeben, sodaß Sie jederzeit die Anzahl der verbrauchten Bytes durch GOTO 9999 feststellen können.

Geben Sie das folgende Programm ein und lassen Sie es laufen:

```
  10 FOR A=1 TO RND*10+2
  20 FOR B=1 TO RND*10+2
  30 FOR C=1 TO RND*10+2
  40 PRINT "*";
  50 NEXT C
  60 PRINT "#";
  70 NEXT B
  80 PRINT "▒";
  90 NEXT A
9999 PRINT PEEK 16396+256*PEEK 1
6397-16509
```

Sie werden feststellen, daß es 2Ø6 Bytes benötigt. Nun geben Sie folgende Zeilen dazu

```
5 LET D=10
```

und ändern die 1Ø in den Zeilen 1Ø, 2Ø und 3Ø in D und lassen Sie das Programm wieder laufen. Dieses Mal werden Sie feststellen, daß das Programm nur 2Ø1 Bytes braucht, obwohl Sie eine Zeile hinzugefügt haben. Nun, diese Ersparnis ist nicht sehr dramatisch, aber sie beweist, daß Speicher auf diese Weise gespart werden kann. Und das kann wichtig sein, wenn Sie an die Grenzen des Speicherplatzes vorstoßen.

Als nächstes fügen Sie hinzu:

```
3 LET E=PI/PI
4 LET F=E+E
```

Dies weist E dem Wert 1 und F den Wert 2 zu und trotz des Umweges ist es das Rationellste.

Gehen Sie noch einmal durch Ihr Programm und ändern Sie alle Einser in E's und alle Zweier in S. Lassen Sie es wieder laufen und Sie sollten feststellen, daß das Programm jetzt nur noch 187 Bytes braucht, nun schon eine ganze Menge weniger als die 2Ø6, mit denen wir begonnen haben, obwohl drei zusätzliche Zeilen hinzugefügt wurden.
Trotz der Tatsache, daß nur 1 K eine Einschränkung bedeutet, können sehr lohnende Programme in sogar weniger als dieser Speichermenge geschrieben werden. Zum Beispiel das folgende Programm, das die Quadrat- und Kubikwurzeln jeder angegebenen Zahl berechnet — es braucht nur 154 Bytes inklusive der Zeile 9999.

```
  10 PRINT "ZAHL?"
  20 INPUT Z
  30 PRINT Z
  35 PRINT "WURZEL?"
  40 INPUT W
  50 CLS
  60 PRINT "DIE#";W;"#TE WURZEL
VON#";Z
  70 PRINT "IST#";Z**(1/W)
9999 PRINT PEEK 16396+256*PEEK 1
6397-16509
```

Um das Programm zu verwenden, geben Sie die gewünschte Zahl (beispielsweise 1ØØØ) und dann, wenn Sie dazu aufgefordert werden, die gewünschte Wurzel ein (z. B. 3 für die Kubikwurzel).

Weil jede Zahl in einem ZX-Programm Speicherplatz verbraucht, ist es sinnvoll, von Zeit zu Zeit eine Zeile fallen zu lassen. Viele Programme enden mit Zeilen wie diesen:

```
IF X>20 THEN PRINT "YOU WIN"
IF X>20 THEN STOP
```

Ein einfacher Weg, dies ohne die zweite Zeile zu erreichen ist, einen Buchstaben, der noch nicht für eine Variable benutzt wurde, gleich nach die PRINT-Zeile zu stellen. Dies stoppt das Programm (mit einer Fehlermeldung). Hier ist ein Beispielprogramm, das diesen Sachverhalt klar machen sollte:

```
10 INPUT A$
20 PRINT A$
30 IF A$="S" THEN PRINT "ENDE"
40 IF A$="S" THEN STOP
50 GOTO 10
```

Diese Version braucht ..... Bytes (arbeiten Sie es selbst aus mit der Hilfe von Zeile 9999). Nun geben Sie diese Version ein und überprüfen die Länge:

```
10 INPUT A$
20 PRINT A$
30 IF A$="S" THEN PRINT "ENDE"
;W
50 GOTO 10
```

Diese Version braucht ..... Bytes.

Als Übung geben Sie das folgende Programm ein und lassen Sie es laufen, sodaß Sie sehen, was es tut (Sie geben zuerst einen Einzelbuchstaben ein, dann NEWLINE, dann eine Zahl, dann NEWLINE, dann eine andere Zahl) und versuchen Sie dann, es so umzuschreiben, daß es weniger Speicherplatz verbraucht.

```
10 INPUT A$
20 IF A$="X" THEN STOP
35 INPUT A
40 INPUT B
```

```
  50 FOR C=1 TO A
  60 FOR D=1 TO B
  70 IF RND>.6 THEN GOTO 100
  80 PRINT "#";
  90 GOTO 110
 100 PRINT A$;
 110 NEXT D
 120 PRINT
 130 NEXT C
 140 GOTO 10
9999 PRINT PEEK 16396+256*PEEK 1
6397-16509
```

Lassen Sie es laufen und Sie werden feststellen, daß es 218 Bytes benötigt. Nachdem Sie das Programm gekürzt haben, blättern Sie in diesem Buch weiter und sehen Sie, wie ich es gelöst habe:

```
  5 LET Z=100
 10 INPUT A$
 20 IF A$="X" THEN STOP
 35 INPUT A
 40 INPUT B
 50 FOR C=Z/Z TO A
 60 FOR D=Z/Z TO B
 70 IF RND>.6 THEN GOTO Z
 80 PRINT "#";
 90 GOTO Z+SQR Z
100 PRINT A$;
110 NEXT D
120 PRINT
130 NEXT C
140 GOTO Z/PI
9999 PRINT PEEK 16396+256*PEEK 1
6397-16509
```

Diese Version braucht 209 Bytes (mit der üblichen Zeile 9999 am Ende der beiden Programme). Versuchen Sie, es weiter zu kürzen.

Im Buch »Probleme zur Computerlösung« von Donald D. Spencer wird ein Problem wie dieses gestellt:

> »Wenn fünf Vogelpaare je drei Eier ausbrüten, dann sterben, die verbleibenden 15 Vögel sich paaren und jeder der 15 Paare drei Vögel ausbrütet und dann stirbt usw. Wie viele Vögel wird es nach 5 Jahren geben?«

Schreiben Sie ein Programm, um dieses Problem zu lösen und kommen Sie dann wieder zum Buch zurück, um zu sehen, wie ich es gelöst habe.

Im folgenden Beispiel habe ich die Namen der Variablen voll ausgeschrieben, d. h., ich habe eine Variable mit Namen »Vögel« verwendet und die Anzahl der Vögel zugewiesen, ebenso wie der Variablen »Jahr« die Anzahl der Jahre. Arbeiten Sie aus, wieviele Bytes Ihr Programm braucht, dann geben Sie meines ein, berechnen die Anzahl der Bytes und dann kürzen Sie das Programm, indem Sie die Variablennamen auf Einzelbuchstaben kürzen. Auf diese Weise werden Sie eine ganze Menge an Speicher sparen.

```
  5 LET JAHR=0
 10 LET VOEGEL=10
 20 LET VOEGEL=INT (VOEGEL/2*3)
 30 LET JAHR=JAHR+1
 40 SCROLL
 50 PRINT "JAHR#";JAHR;"###VOEG
EL#";VOEGEL
 60 FOR P=1 TO 60
 70 NEXT P
 80 GOTO 20
```

Unser nächstes Programmbeispiel, das bedeutend schwerer zu kürzen ist, heißt »Jumping Jackrabbit«. Der Hase (ein invertierter Zwischenraum) beginnt im Mittelpunkt eines 5 x 5-Gitters, das auf der Außenseite eines Zylinders gezeichnet wird, sodaß die rechte und die linke Seite, nicht aber der obere und untere Rand gleich sind. Je ein Koyote, tötlicher Feind des Hasen, wartet am oberen und unteren Rand, um den Hasen, der nicht sehr hell ist, zu fressen. Stößt der Hase am oberen oder unteren Rand an, stoppt das Programm mit einer Fehlermeldung, um zu zeigen, daß der Hase gefressen wurde. Die Sprünge des Hasen sind zufällig, ein Kästchen pro Sprung. Sie werden entdecken, daß dies ein sehr faszinierendes Display ergibt, das Sie zu anderen Programmen inspirieren kann.

```
  5 DIM A(25)
  7 LET D=0
 10 FOR A=1 TO 25
 20 LET A(A)=27
 30 NEXT A
 40 LET Z=13
 50 LET A(Z)=128
 60 PRINT AT 8,0;
 70 FOR A=1 TO 25
 80 PRINT CHR$ A(A);
100 IF 5*INT (A/5)=A THEN PRINT
```

```
110 NEXT A
120 LET D=D+1
130 PRINT ,D
140 LET C=INT (RND*5)+1
150 IF C=2 OR C=3 THEN GOTO 140
160 IF RND>.5 THEN LET C=-C
170 LET A(Z)=27
180 LET Z=Z+C
190 GOTO 50
```

Sie werden entdecken, daß der Hase sehr selten, wenn überhaupt, mehr als fünfzehn Sprünge überlebt.
Zum Schluß findet sich hier ein einfaches FRUIT-MASCHINE-Programm, das unter Zuhilfenahme vieler platzsparender Tricks, die wir besprochen haben, geschrieben wurde. Geben Sie es ein, lassen Sie es laufen und wenn Sie es einmal in Aktion gesehen haben, versuchen Sie, es so umzuschreiben, daß es noch platzsparender wird.

```
 10 LET K=3
 20 LET T=2
 30 LET M=30
 40 LET Z=M/M
 50 LET A=M-M
 60 LET B=A
 70 LET C=A
 80 PRINT "SIE HABEN $";M
 90 INPUT A$
100 CLS
110 LET M=M-T
120 FOR D=Z TO K
130 GOTO 160+INT (RND*K)*10
160 PRINT "##";
162 LET A=A+Z
164 GOTO 184
170 PRINT "*#";
172 LET B=B+Z
174 GOTO 184
180 PRINT "**";
182 LET C=C+Z
184 NEXT D
190 IF A=Z AND B=Z AND C=Z THEN
LET M=M+T*T
200 IF A=K OR B=K OR C=K THEN L
ET M=M+K*T
210 IF M<Z THEN PRINT "SIE SIND
PLEITE";W
220 GOTO 50
```

****

Plotten und Drucken

****

Der ZX81 (und der ZX80 mit neuem ROM) hat sehr nützliche Fähigkeiten zum Plotten und Drucken. Die nächsten vier Programme zeigen diese Möglichkeiten im Gebrauch.

Pattern-Master

Dieses Programm erzeugt ein sich ständig änderndes Muster und verwendet dazu ein Zeichen, das Sie auswählen. Das Programm ist relativ befriedigend, wenn Sie PAUSE benutzen, wird aber wirklich interessant, wenn Sie die Pause-Zeile löschen und das Programm im SLOW-MODE laufen lassen.

```
  10 PRINT AT 5,8;"PATTERN MASTE
R"
  20 PRINT AT 10,0;"WELCHES SYMB
OL SOLL ICH IM","MUSTER VERWENDE
N"
  30 INPUT A$
  40 CLS
  50 FOR Z=1 TO 500
  60 LET A=INT (RND*11)+1
  70 LET B=INT (RND*16)+1
  80 PRINT AT A,B;A$
  90 PAUSE 30
 100 PRINT AT 22-A,B;A$
 110 PAUSE 30
 120 PRINT AT A,32-B;A$
 130 PAUSE 30
 140 PRINT AT 22-A,32-B;A$
 150 PAUSE 100
 160 LET A=INT (RND*11)+1
 170 LET B=INT (RND*16)+1
 180 PRINT AT A,B;"#"
 190 PAUSE 5
 200 PRINT AT 22-A,B;"#"
 210 PAUSE 5
 220 PRINT AT A,32-B;"#"
 230 PAUSE 5
 240 PRINT AT 22-A,32-B;"#"
 250 PAUSE 5
 260 IF RND>.4 THEN GOTO 160
 270 NEXT Z
```

```
280 PAUSE 400
290 CLS
300 RUN
```

Capriccio

»Capriccio: Musik, ein lebhaftes Stück, frei komponiert...« (Collins English Dictionary). Capriccio ist ein ähnliches Programm wie PATTERN-MASTER mit dem Unterschied, daß es die Muster auf einem feinerem Netz (62x4Ø) erzeugt und die PLOT und UNPLOT-Funktionen statt PRINT AT verwendet.

Machen Sie die Schleife in Zeile 2ØØ länger, wenn Sie das Programm etwas verlangsamen wollen.

```
 10 SLOW
 20 RAND
 30 LET A=INT (RND*62)
 40 LET B=INT (RND*40)
 50 IF RND>.25 THEN GOTO 150
 60 PLOT A,B
 70 GOSUB 200
 80 PLOT A,40-B
 90 GOSUB 200
100 PLOT 62-A,B
110 GOSUB 200
120 PLOT 62-A,40-B
130 GOSUB 200
140 GOTO 30
150 UNPLOT A,B
160 UNPLOT A,40-B
170 UNPLOT 62-A,B
180 UNPLOT 62-A,40-B
190 GOTO 30
200 FOR N=1 TO 5
210 NEXT N
220 RETURN
```

Schmetterling

SCHMETTERLING ist eine einfachere Version des obigen Programms und paßt in 1 K.

```
10 LET C=62
20 LET D=40
30 LET A=C*RND
40 LET B=D*RND
50 PLOT A,B
60 PLOT A,D-B
70 PLOT C-A,B
80 PLOT C-A,D-B
90 RUN
```

Wie Sie sehen, gibt es keine Möglichkeit, eines der kleinen schwarzen Vierecke wieder zu löschen, sodaß der Bildschirm eventuell eine massive schwarze Fläche wird.

Star-Map

STAR-MAP erzeugt eine Galaxie auf Ihren TV-Schirm und ist ein effektives Display-Programm, wenn es einige Zeit lang laufen gelassen wird. Das Programm zeichnet ein schwarzes Rechteck und beginnt dann mittels PRINT AT und UNPRINT AT Sterne in das schwarze Nichts zu zeichnen.

```
 10 SLOW
 20 FOR A=1 TO 10
 30 FOR B=1 TO 32
 40 PRINT "#";
 50 NEXT B
 60 NEXT A
 70 PRINT AT RND*9,RND*31;"*"
 80 PRINT AT RND*9,RND*31;"#"
 90 PRINT AT RND*9,RND*31;"#"
100 PRINT AT RND*9,RND*31;"#"
110 PRINT AT 1,1;"#"
120 PRINT AT 1,1;"*"
130 GOTO 70
```

Wir haben uns einige Zeit mit »ernsthaften« Anwendungen befaßt und werfen nun einen Blick auf die Spielmöglichkeiten des ZX81.

Geiger

GEIGER arbeitet sowohl in SLOW und FAST-MODE. Das abgedruckte Listing ist für FAST gedacht, aber es braucht nur kleine Änderungen, um den bestmöglichen Gebrauch von SLOW zu machen. Sie suchen eine kleine Menge Plutonium, die unter einer festen schwarzen Fläche versteckt ist. Sie müssen lernen, die Ausgabe Ihres Geigerzählers zu interpretieren, um das Plutonium zu finden. Die Anzeige Ihres Geigerzählers wird von Zeile 29Ø bestimmt, die Sie einfach ändern können, wenn Sie eine andere Form der Anzeige bevorzugen. Wenn Sie das Programm wirklich schwierig gestalten wollen, können Sie die Ausgabe des Geigerzählers etwas unberechenbarer machen, indem Sie einen Zufallsfaktor einführen.

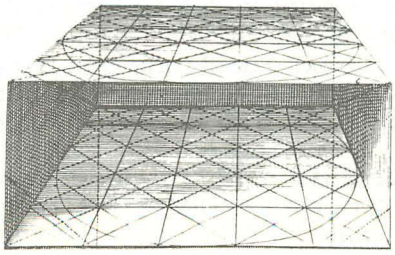
```
 10 LET S=0
 20 LET A=2+INT (RND*8)
 30 LET B=1+INT (RND*9)
 40 PRINT
 50 PRINT
 60 FOR X=1 TO 10
 70 FOR Y=1 TO 10
110 PRINT AT X,Y;"#"
120 NEXT Y
130 NEXT X
140 LET C=1
150 LET D=1
170 PRINT AT C,D;"X"
175 LET S=S+1
180 PAUSE 40
190 PRINT AT C,D;"+"
200 IF INKEY$="7" THEN LET C=C-
1
210 IF INKEY$="5" THEN LET D=D-
1
220 IF INKEY$="6" THEN LET C=C+
1
230 IF INKEY$="8" THEN LET D=D+
1
235 IF INKEY$="S" THEN STOP
240 IF C<1 THEN LET C=1
250 IF C>10 THEN LET C=10
260 IF D<1 THEN LET D=1
270 IF D>10 THEN LET D=10
280 IF A=C AND B=D THEN GOTO 90
0
290 PRINT AT 11,0;"GEIGERZAEHLE
R MISST:";ABS (B-D)*7.29+C+A
300 GOTO 170
```

```
 900 PRINT AT 13,3;"SIE HABEN DA
S PLUTONIUM IN",S;"#VERSUCHEN GE
FUNDEN"
 910 PRINT "####################
#############"
 920 PRINT AT C,D;"P"
 930 PAUSE 500
 940 CLS
 950 RUN
```

Space Boy

```
 100 PRINT
 110 PRINT TAB 4;"SPACE BOY"
 120 PRINT
 130 PRINT
 140 PRINT "BEI DIESEM SPIEL MUE
SSEN SIE DEN"
 150 PRINT "WANDERNDEN SPACE BOY
 SO STEUERN,"
 160 PRINT "DASS SIE ALLE GEGENS
TAENDE,DIE"
 170 PRINT "SICH IM OBERTEIL DER
 KAPSEL"
 180 PRINT "BEWEGEN,VERNICHTEN."
 190 PRINT
 200 PRINT
 210 PRINT "TASTE X BEWEGT SIE N
ACH LINKS,"
 220 PRINT "TASTE M NACH RECHTS.
"
 230 PRINT
 235 PRINT "SIE HALTEN DEN SPACE
 BOY IN"
 237 PRINT "BEWEGUNG,WENN SIE IH
N MIT DER"
 238 PRINT "BEWEGLICHEN PLATTFOR
M ABSTOSSEN."
 240 PRINT
 250 PRINT "DRUECKEN SIE N/L,WEN
N SIE"
 260 PRINT "SPIELBEREIT SIND"
 270 INPUT A$
 275 CLS
5010 FOR A=1 TO 100
5020 LET B=PI*A/50
5030 PRINT AT 9*COS B+12,14*SIN
B+17;"#"
5050 NEXT A
5060 CLEAR
```

```
5062 PRINT AT 5,11;"▗▄▛▀▀▀▀▀▜▄▖
▝"
5065 PRINT AT 6,11;"▐▀▀▀▀▀▀▀▀▀▀
▌"
5070 PRINT AT 7,10;"▐▀▀▀▀▀▀▀▀▀▀
▀▘"
5075 PRINT AT 6,11;"▐▀▀▀▀▀▀▀▀▀▀
▌"
5451 LET C=10
5452 LET D=10
5453 LET PUNKTE=0
5455 LET A=19
5456 LET B=21
5457 LET E=1
5459 LET K=5
5510 LET B=B-RND*5+RND*5
5520 IF INKEY$="X" THEN LET D=D-
1
5525 IF D<8 THEN LET D=23
5530 IF INKEY$="M" THEN LET D=D+
1
5535 IF D>23 THEN LET D=23
5540 IF B>21 THEN LET B=21
5550 IF B<9 THEN LET B=9
5555 LET C=C+E
5600 PRINT AT A,B;"▗▄▖"
5710 PRINT AT C,D+1;"#";K
5720 PRINT AT C+1,D+1;"▐▀▌"
5721 IF K=0 THEN GOTO 6010
5740 PRINT AT A,B;"###"
5750 PRINT AT C,D+2;"#"
5760 PRINT AT C+1,D+1;"   "
5770 IF C=18 AND ABS (D-B)>2 THE
N LET K=K+1
5775 IF C+1=19 THEN LET E=-1
5785 IF C=5 THEN LET E=1
5800 LET PUNKTE=PUNKTE+1
6000 GOTO 5510
6010 PRINT AT 10,4;"##SIE HABEN#
";PUNKTE;"#PUNKTE##"
6015 PAUSE 400
6020 CLS
6030 RUN 240
```



****

Arrays

****

ARRAYS und der Gebrauch des DIM (Dimension) Statements können verwirrend sein. DIM wird gebraucht, wenn Sie eine Liste aufstellen wollen. Sie möchten beispielsweise die Liste der Zahlen 1 bis 15 mit dem Namen A in der Form LET A (1)=1, LET A (2)=2 usw. bis LET A (15)=15 speichern.

Um das zu tun informieren Sie den ZX81, daß Sie ein Array. brauchen, um diese Elemente aufzunehmen (1, 2 usw. bis 15), indem Sie eingeben:

```
10 DIM A(15)
```

Die Zahl innerhalb der Klammern heißt Subscript und ein Element der Form F (2) oder K (8) heißt subskribierte Variable. Hier ist ein Programm, daß das DIM-Statement demonstrieren soll:

```
10 DIM A(10)
20 FOR B=1 TO 10
30 LET A(B)=2*B
40 PRINT "A(";B;")=";A(B)
50 NEXT B
```

Wenn Sie dieses Programm laufen lassen, dann bekommen Sie:

A (1)=2
A (2)=4
A (1 Ø)=2 Ø
usw. usw. ...

Ändern Sie die Zeile 1 Ø auf DIM A (5) und versuchen Sie, das Programm wieder laufen zu lassen. Diesmal werden Sie nur bis zu A (5)=1 Ø kommen. Das Programm stoppt bei diesem Punkt (und gibt die Fehlermeldung 3/3 Ø ab, das bedeutet, daß die benötigte subskribierte Variable, z.B. B=6 bis B=1 Ø außerhalb des Bereiches war). Ändern Sie Zeile 1 Ø jetzt auf DIM A (2 Ø) und lassen Sie das Programm wieder laufen. Sie werden keinen Unterschied zu DIM A (1 Ø) bemerken. Sie ändern nichts (außer, daß Sie mehr Speicherplatz

verbrauchen), wenn sie ein größeres Array einrichten, als Sie tatsächlich benötigen. Nun fügen Sie zu Ihrem Programm die folgenden Zeilen dazu:

```
 10 DIM A(10)
 60 INPUT B$
 70 FOR Z=1 TO 15
 80 LET A(Z)=3*Z
 90 PRINT "A(";Z;")=";A(Z)
100 NEXT Z
```

Wenn Sie das laufen lassen, werden Sie das gleiche Ergebnis bis A (1 Ø)=2 Ø bekommen, gefolgt vom Symbol, das Sie zur Eingabe eines Strings auffordert (drücken Sie in diesem Fall NEWLINE). Der ZX81 wird dann anzeigen:

A (1)=3 bis zu ...
A (1 Ø)=3 Ø

Der Fehlercode 3/8 Ø wurde angezeigt, weil das Array in Zeile 1 Ø kleiner definiert wurde als der Bedarf in Zeile 8 Ø. Das heißt, Zeile 7 Ø machte das nächste Z=F und Zeile 8 Ø verlangte deshalb eine subskribierte Variable mit Namen A (11), die es natürlich nicht finden konnte, weil das Array nur Platz für 1 Ø Elemente hatte.

Pferderennen

Das nächste Programm, PERDERENNEN, verwendet ein Array (definiert in der ersten Zeile), um die Positionen von vier kleinen Pferden zu speichern (die inversen Buchstaben A, B, C und D), die auf einer Rennstrecke auf Ihrem Bildschirm (oder wenigstens auf einem sehr groben Abbild einer Rennstrecke) hinauf- und hinuntergaloppieren. Nachdem am Ende der Endstand angezeigt wurde, fährt die Rennstrecke weg und ein neues Rennen beginnt.

```
 10 DIM A(4)
 20 SLOW
 30 FOR A=1 TO 22
 40 FOR C=1 TO 9
 50 PRINT "▒";
 60 NEXT C
 70 FOR B=1 TO 15
 80 PRINT "#";
 90 NEXT B
100 FOR C=1 TO 8
110 PRINT "▒";
```

```
 120 NEXT C
 130 NEXT A
 140 LET A=12
 150 LET B=15
 160 LET C=18
 170 LET D=21
 180 GOTO 230
 190 PRINT AT ABS (22-A(1)),A;"#
"
 200 PRINT AT ABS (22-A(2)),B;"#
"
 210 PRINT AT ABS (22-A(3)),C;"#
"
 220 PRINT AT ABS (22-A(4)),D;"#
"
 230 FOR G=1 TO 4
 240 LET A(G)=1+A(G)+INT (RND*2)
 250 NEXT G
 260 PRINT AT ABS (22-A(1)),A;"A
"
 270 PRINT AT ABS (22-A(2)),B;"B
"
 280 PRINT AT ABS (22-A(3)),C;"C
"
 290 PRINT AT ABS (22-A(4)),D;"D
"
 300 FOR N=1 TO 10
 310 NEXT N
 320 IF A(1)<42 AND A(2)<42 AND
A(3)<42 AND A(4)<42 THEN GOTO 19
0
 330 PRINT AT 3,2;"PUNKTE:A-";A(
1);"#B-";A(2);"#C-";A(3);"#D-";A
(4)
 340 FOR N=1 TO 100
 350 NEXT N
 360 CLS
 370 RUN
```

Alien Imploders

```
   5 LET U=0
  10 GOSUB 1000
  90 FOR K=1 TO 70
 100 PRINT AT 3,0;A$;AT X,Y-1;"#
A#"
 200 LET A$=A$(2 TO )+A$(1)
 300 IF INKEY$="5" THEN LET Y=Y-
1
 350 IF INKEY$="8" THEN LET Y=Y+
1
```

```
 360 IF INKEY$="1" THEN GOSUB 50
00
 500 NEXT K
 510 PRINT AT 12,9;"ENDE DER RUN
DE"
 520 IF Q>U THEN LET U=Q
 530 IF U<>0 THEN PRINT AT 1,0;"
HOECHSTER PUNKTESTAND:";U
 540 FOR D=1 TO 60
 550 NEXT D
 555 PRINT AT 12,9;"**BITTE WART
EN**"
 560 LET Z=RND**RND
 570 PRINT AT 12,9;"##############
####"
 700 GOTO 10
1000 LET A$="##X###X###X###X###X##M#
#M##M###M###H"
1020 LET S=0
1030 LET X=18
1040 LET Y=16
1050 PRINT AT X,0;"###############
####################"
1060 PRINT AT 0,7;"###############
####"
1500 RETURN
5000 IF A$(Y)<>"#" THEN LET S=S+
1
5002 IF A$(Y)="H" THEN LET S=S+2
5003 IF A$(Y)="M" THEN LET S=S+1
5005 LET A$(Y)="#"
5008 FOR A=18 TO 1 STEP -3
5010 PRINT AT A,Y;"."
5020 PRINT AT A,Y;"#"
5030 NEXT A
5040 LET Q=S*6453+INT (RND*3)
5050 IF Q>5000 THEN PRINT AT 0,7
;"PUNKTE:";Q
5070 IF Y>2 AND Y<30 THEN LET Y=
Y+INT (RND*3)-INT (RND*3)
5500 RETURN
```

Der ZX81 kann auch n-dimensionale Array aufstellen (wobei n ein positiver Integer sein muß). Ein zweidimensionales Array wird definiert mit einer Anweisung wie DIM A (1 Ø, 1 Ø), was 1 ØØ Variablen ergibt. Genauer gesagt 1 Øxl Ø subskribierte Variable von 1,1 bis 1,1 Ø.

Versteckspiel in 3-D

In diesem Programm müssen Sie den ZX81 finden, der sich in einem dreidimensionalen Array versteckt. Um ihn zu finden, stellen Sie sich vor, daß der ZX81 sich in einem Würfel mit den Maßen 10x10x10 versteckt. Sie geben drei Koordinaten ein (jeweils eine gefolgt von NEWLINE); der Computer gibt Ihnen dann eine Antwort, die Ihnen erlaubt, Ihren Tip zu modifizieren. Sie werden bald lernen, wie Sie das Feedback des Computers interpretieren und Ihren Tip ändern müssen.

```
  10 DIM A(10,10,10)
  20 LET A=INT (RND*10)+1
  30 LET B=INT (RND*10)+1
  40 LET C=INT (RND*10)+1
  50 LET A(A,B,C)=1
  60 FOR Z=1 TO 10
  70 PRINT "WO VERSTECKE ICH MIC
H?"
  80 PRINT "VERSUCH NR.";Z
  90 INPUT D
 100 INPUT E
 110 INPUT F
 120 PRINT D;"##";E;"##";F
 130 IF A(D,E,F)=1 THEN GOTO 340
 140 PRINT "DANEBEN"
 150 PRINT "HIER IST EINE HILFE:
"
 160 FOR G=1 TO ABS (A-D)
 170 PRINT "#";
 180 NEXT G
 190 PRINT
 200 FOR G=1 TO ABS (B-E)
 210 PRINT "▒";
 220 NEXT G
 230 PRINT
 240 FOR G=1 TO ABS (C-F)
 250 PRINT "▮";
 260 NEXT G
 270 PRINT
 280 PRINT "DRUECKEN SIE N/L FUE
R DEN","NAECHSTEN VERSUCH"
 290 INPUT A$
 300 CLS
 310 NEXT Z
 320 PRINT "DAS WAR SCHLECHT"
 330 GOTO 350
 340 PRINT "SIE HABEN MICH IN#";
Z;"VERSUCHEN","GEFUNDEN"
 350 PRINT "ICH WAR AUF#";A;"##"
;B;"##";C
```

```
360 PAUSE 200
370 CLS
380 RUN
```

Blaster-Mind 1

Es gibt auch Zeichenarrays. Im nächsten Programm, BLASTER-MIND, werden zwei Zeichenarrays (A $ und B $) verwendet. A $ speichert den 4-Zahlencode, den Sie aufdecken müssen, und akzeptiert Ihren Tip. Das Programm vergleicht dann Ihren Tip mit seinem Code Element für Element. Das Prinzip des Spieles ist einfach: Der Computer erstellt einen String, der vier Zahlen beinhaltet. Keine Zahl ist doppelt und ∅ wird nicht verwendet. Sie geben Ihren Tip als String B $ ein und drücken dann NEWLINE. Für eine richtige Zahl in der richtigen Position bekommen Sie einen schwarzen Punkt. Für eine richtige Zahl in der falschen Position einen weißen. Sie haben zehn Versuche, die Zahlenkombination herauszufinden. Nach jedem Versuch druckt der ZX81 Ihren Tip aus und gibt dann den Punktestand der schwarzen (als inverse Zahl) und der weißen Punkte (als normale Zahl) an. Verwenden Sie eine Zahl in einem Versuch nicht zweimal.

```
  10 DIM A$(4)
  20 DIM B$(4)
  30 LET A$(1)=STR$ (INT (RND*9)
+1)
  40 FOR A=2 TO 4
  50 LET A$(A)=STR$ (INT (RND*9)
+1)
  60 FOR B=1 TO A-1
  70 IF A$(B)=A$(A) THEN GOTO 50
  80 NEXT B
  90 NEXT A
 100 LET C$=A$
 110 FOR E=1 TO 10
 120 SCROLL
 130 PRINT "GEBEN SIE VERSUCH NR
.";E;"#EIN"
 140 INPUT B$
 150 LET B=0
 160 LET W=0
 170 IF B$=A$ THEN GOTO 360
 180 FOR C=1 TO 4
 190 IF A$(C)=B$(C) THEN LET B=B
 200 IF A$(C)=B$(C) THEN LET A$(
C)="0"
 210 NEXT C
```

```
 220 FOR C=1 TO 4
 230 IF A$(C)="0" THEN GOTO 280
 240 FOR D=1 TO 4
 250 IF A$(C)<>B$(D) THEN GOTO 2
70
 260 LET W=W+1
 270 NEXT D
 280 NEXT C
 290 LET A$=C$
 300 SCROLL
 310 PRINT ,B$;"#";CHR$ (B+156);
"#";W
 320 NEXT E
 330 SCROLL
 340 PRINT "DIE ZEIT IAT UM"
 350 GOTO 380
 360 SCROLL
 370 PRINT "SIE HATTEN RECHT IN#
";E
 372 SCROLL
 375 PRINT "VERSUCHEN"
 380 SCROLL
 390 PRINT "DER CODE WAR#";A$
 400 PAUSE 200
 410 CLS
 420 RUN
```

Blaster-Mind 2

Hier ist eine andere Version des Spieles, die nur drei Zahlen verwendet und in 1 K paßt. Verwendet werden die Zahlen Ø bis 9, aber Ø ist niemals die erste Ziffer. Wie in der vierziffrigen Version wird keine Ziffer wiederholt. Es gibt weniger »Konversation« in dieser Version, um das Programm in 1 K zu bekommen, aber Sie werden es immer noch unterhaltsam finden (und bedeutend einfacher als die Vierziffern-Version).

```
  10 DIM A$(3)
  20 DIM B$(3)
  30 LET A$=STR$ (INT (RND*900)+
100)
  35 IF A$(1)=A$(2) OR A$(1)=A$(
3) OR A$(2)=A$(3) THEN GOTO 30
  40 FOR C=1 TO 10
  50 INPUT B$
  60 IF A$=B$ THEN GOTO 270
  70 LET C$=A$
  80 LET B=156
  90 LET W=0
 100 FOR D=1 TO 3
```

```
 110 IF A$(D)<>B$(D) THEN GOTO 1
40
 120 LET B=B+1
 130 LET A$(D)="Z"
 140 NEXT D
 150 FOR D=1 TO 3
 160 FOR E=1 TO 3
 170 IF B$(D)<>A$(E) THEN GOTO 2
00
 180 LET W=W+1
 190 LET A$(D)="Z"
 200 NEXT E
 210 NEXT D
 220 LET A$=C$
 230 PRINT B$;"##";CHR$ B;"##";W
 240 NEXT C
 250 PRINT A$
 260 STOP
 270 PRINT A$;"##SIE SCHAFFTEN E
S IN#";C;"#ZUEGEN"
```

Die zwei vorangegangenen Programme haben Zeichenarrays benutzt. Um passende Zeichenkettenarrays (wo jedes Element der Liste länger ist als nur ein Zeichen) zu bekommen, müssen Sie nicht nur die Anzahl der Worte in der Liste dimensionieren, sondern auch die Länge des längsten Wortes, das Sie in das Array schreiben wollen.

Das nächste Programm — SCHERE/STEIN/PAPIER — zeigt dies in der Praxis. Zeile 2Ø (DIM A $ (3,8)) stellt ein Array von drei Worten auf, wobei jedes Wort bis zu 8 Buchstaben lang sein kann.

Schere/Stein/Papier

Es ist egal, ob ein Wort kürzer oder gleich lang als die zweite Zahl in der Dimensionsanweisung (die 8 in Zeile 2Ø) ist, solange das Wort nicht mehr Buchstaben hat als die dimensionierte Anzahl. Jedes Wort kann auch durch Eingabe nur der ersten Zahl bestimmt werden. Der ZX81 setzt voraus, daß sie jeden Buchstaben des Wortes wollen.

Wir geben zuerst den ersten Teil des Programms ein, um dies zu verdeutlichen:

```
10 RAND
20 DIM A$(3,6)
30 LET A$(1)="STEIN"
```

```
 40 LET A$(2)="SCHERE"
 50 LET A$(3)="PAPIER"
 60 SCROLL
 70 PRINT A$(INT (RND*3)+1)
 80 GOTO 60
```

Lassen Sie dies laufen und Sie werden eine endlose, aus den Worten Schere/Stein/Papier gebildete Reihe auf dem Bildschirm bekommen. Stoppen Sie das Programm mit BREAK und geben Sie den Rest des Programmes ein (was einige Zeilen der obigen Demonstration löscht).

```
 60 LET P=0
 70 LET C=0
 80 INPUT A
 90 CLS
100 LET B=INT (RND*3)+1
110 PRINT AT 7,10;"SIE",A$(A);T
AB 11;P
120 PRINT AT 14,10;"ICH",A$(B);
TAB 11;C
130 PRINT AT 3,12;
140 IF A=B THEN GOTO 190
150 IF A=1 AND B=2 OR A=2 AND B
=3 OR A=3 AND B=1 THEN GOTO 210
160 PRINT "ICH GEWINNE"
170 LET C=C+1
180 GOTO 80
190 PRINT "UNENTSCHIEDEN"
200 GOTO 80
210 PRINT "SIE GEWINNEN"
220 LET P=P+1
230 GOTO 80
```

Sie können dieses Spiel so lange spielen, wie Sie wollen und halten es an, indem Sie irgendeinen Buchstaben mit Ausnahme von A, B, C oder P eingeben. Um zu spielen, erinnern Sie sich daran, daß der Stein die Schere schlägt (weil ein Stein eine Schere schartig macht), die Schere Papier schlägt (weil Scheren Papier schneiden können), und Papier den Stein schlägt (weil ein Stein in Papier eingewickelt werden kann).
Wie Sie wahrscheinlich schon beim Eingeben des Programmes bemerkt haben, erfordert dieses Spiel keinerlei Geschicklichkeit, weder vom Computer noch von Ihnen. T speichert den Stand des Spielers und der Punktestand des Computers wird in der Variablen C gehalten. Beachten Sie den Gebrauch von TAB in den Zeilen 11Ø und 12Ø. Diese ermöglichen es, mit nur einer Programmzeile zwei Schriftzeilen auszudrucken. Das

Handbuch des ZX81 erklärt das (und den Gebrauch von PRINT AT) auf den Seiten 115 und 116.
War bei diesem Spiel weder von Ihnen noch vom Computer besondere Geschicklichkeit erforderlich, verlangt das nächste Geschicklichkeit von beiden, allerdings bedeutend mehr vom ZX81.

Codebreaker

Dies ist das ideale Spiel, wenn Sie es müde sind, den Zahlencode des ZX81 zu erraten. Hier muß der ZX81 Ihre Codes herausfinden. Das Programm erlaubt dem Computer, einen Code von N Zahlen aufzudecken, wobei die verwendeten Zahlen von 1 bis L reichen. Im Programmlisting habe ich N auf 4 und L auf 6 gesetzt, aber Sie können dies selbstverständlich ändern. Der ZX81 findet auch Kombinationen heraus, bei denen eine Zahl mehrmals vorkommt. Er druckt seinen Tip aus, dann erscheint das Wort »Schwarz«. Hier geben Sie die Anzahl der Ziffern ein, die richtig und auf der richtigen Stelle sind. Der Computer druckt dann »Weiß« und Sie teilen ihm mit, wie viele Zahlen er richtig, aber auf dem falschen Platz hat. Das Programm ist sehr langsam, deshalb lassen Sie es immer im FAST-MODE laufen.

```
  1 LET N=4
  2 LET M=6
  3 LET L=M
  5 DIM G(N+1)
 10 DIM M(M*N)
 20 DIM B(M)
 30 DIM D(N)
 40 DIM W(M)
 50 DIM H(N)
 60 FOR Z=1 TO N
 80 LET G(Z)=INT (RND*L)+1
 90 NEXT Z
100 FOR G=1 TO M
110 FOR Z=1 TO N
120 PRINT G(Z);
130 LET M(N*G+Z-N)=G(Z)
140 NEXT Z
150 PRINT "#SCHWARZ=";
160 INPUT B(G)
170 PRINT B(G)
180 IF B(G)=N THEN GOTO 220
190 PRINT
200 PRINT "CODE IN#";G;"#VERSUC
HEN ERRATEN"
210 STOP
```

```
220 PRINT "#WEISS=";
230 INPUT W(G)
240 PRINT W(G)
250 LET G(Z)=G(Z)+1
260 FOR Z=1 TO N
270 IF G(Z)-L-1 THEN GOTO 300
280 LET G(Z)=1
290 LET G(Z+1)=G(Z+1)+1
300 NEXT Z
320 FOR Z=1 TO G
330 FOR J=1 TO N
340 LET D(J)=M(N*Z+J-N)
350 LET H(J)=G(J)
360 NEXT J
370 LET B=0
380 LET W=0
390 FOR J=1 TO N
400 IF D(J)=0 OR H(J)=0 OR D(J)
-H(J) THEN GOTO 440
410 LET D(J)=0
420 LET H(J)=0
430 LET B=B+1
440 NEXT J
450 IF B-B(Z) THEN GOTO 250
460 FOR J=1 TO N
470 FOR K=1 TO N
480 IF D(J)=0 OR H(K)=0 OR D(J)
-H(K) THEN GOTO 520
490 LET D(J)=0
500 LET H(K)=0
510 LET W=W+1
520 NEXT K
530 NEXT J
540 IF W-W(Z) THEN GOTO 250
550 NEXT Z
560 NEXT G
```

Mondlandung 1, 2 und 3

Wir machen nun eine weitere Lernpause und werfen einen Blick auf drei Versionen eines sehr populären Programmes — Mondlandung.
Das Raumschiff startet in einer zufälligen Höhe über dem Mond (135∅ bis 1549) mit einer zufällig Benzinmenge (8∅ bis 178.5) und mit einer zufälligen gewählten Geschwindigkeit (∅ bis 19). Wenn Sie Glück haben, bekommen Sie eine Ausgangsposition, die es leicht macht, zu landen. Wenn nicht, brauchen Sie alle Geschicklichkeit eines Astronauten, um eine erfolgreiche Landung durchzuführen. Wenn Sie nach der Aufforderung des Computers zuviel Schub eingeben (Zeile

224Ø), haben Sie das große Vergnügen zu explodieren (Zeile 23Ø5). Die gewählten Variablennamen machen es einfach, sie durch das Programm zu verfolgen. Höhe ist H, Benzin ist B, Geschwindigkeit ist G, Schub ist S und Dauer ist D. Wenn Sie selbst Programme schreiben, ist es eine gute Idee, derartige Variablennamen zu verwenden (wenn Sie Speicherplatz verbrauchen wollen, können Sie die Bezeichnungen sogar ausschreiben) weil es einfacher ist, herauszufinden, was das Programm tut, wenn Sie es nach einigen Wochen wieder laufen lassen. Beachten Sie, daß die Zeilen 21Ø1 und 21Ø6 den Effekt haben, daß die Werte von B, H und G auf eine Dezimalstelle gekürzt werden, bevor sie ausgedruckt werden.

```
2005 LET R=0
2010 LET H=1350+INT (RND*200)
2020 LET B=80+INT (RND*400)/2
2030 LET G=INT (RND*20)
2035 PRINT
2040 PRINT "DRUECKEN SIE N/L UM
DIE LANDUNG","ZU BEGINNEN"
2045 INPUT U$
2047 CLS
2048 LET S=1
2050 LET D=0
2060 LET K1=1
2090 CLS
2095 FOR A=0 TO D
2100 PRINT AT 16-H/100,K1;"***"
2101 LET B=INT (B*10)
2102 LET B=B/10
2103 LET H=INT (H*10)
2104 LET H=H/10
2105 LET G=INT (G*10)
2106 LET G=G/10
2107 IF R>0 OR H<1 THEN LET H=0
2108 IF R>0 OR H<1 THEN LET A=D
2109 PRINT AT 0,0;"HOE:";H;"##GE
S:";G;"##BE:";B;"####"
2112 LET G=G+17-S/3
2115 LET K1=K1+RND
2116 IF K1>26 THEN LET K1=20
2120 LET H=H-G
2125 IF H>1600 THEN GOTO 2305
2130 LET B=B-S/5
2140 IF 2*(INT (A/2))=A THEN PRI
NT AT 16-H/100,K1;"▗▄▖"
2145 IF 2*(INT (A/2))<>A THEN PR
INT AT 16-H/100,K1;"▐▀▌"
2190 PRINT AT 16,0;"ZEIT:";D-A;"
▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀"
```

```
2220 IF H<100 AND B>0 AND G<10 T
HEN LET R=1
2222 IF H<100 AND G>9 THEN LET R
=2
2223 IF B<1 THEN LET R=3
2224 IF R>0 THEN GOTO 2280
2228 NEXT A
2230 PRINT AT 19,0;"SCHUB?##";
2235 INPUT S
2250 PRINT S,"DAUER (1-11)?"
2260 INPUT D
2265 IF D>10 THEN GOTO 2260
2270 GOTO 2090
2280 IF R=1 THEN PRINT "ERFOLGRE
ICHE LANDUNG"
2290 IF R=2 THEN PRINT "BRUCHLAN
DUNG",,"SIE HABEN EINEN#";INT (G
*4.5);"#METER","TIEFEN MONDKRATE
R GEBOHRT"
2300 IF R=3 THEN PRINT "SIE HABE
N KEINEN SPRIT MEHR","DANN GUTE
NACHT..."
2305 IF H>1600 THEN PRINT "IHRE
FAEHRE IST EXPLODIERT"
2310 PAUSE 400
2320 CLS
2330 RUN
```

Beachten Sie, daß alle Zeilennummern in diesem Programm zwischen 2.ØØØ und 2.33Ø liegen. Dies deshalb, damit Sie, wenn Sie wollen, dieses Programm mit anderen zu einem großen Programm kombinieren können und so beide Programme auf einmal laufen können. Wenn Sie dies tun, müssen Sie ein »Menü« hinzufügen, das die Programmnamen der zwei (oder mehreren) Programme mit den entsprechenden GOTO s angibt.

Das obige Mondlandungsprogramm hat einen Nachteil. Es verwendet falsche Formeln, die mehr wegen ihres Effekts im Spiel ausgewählt wurden als wegen ihrer Beziehung zur Wirklichkeit.

Das nächste Landungsprogramm verwendet echte Gleichungen, aber hat kein nennenswertes Display Nachdem Sie es geschafft haben, die Raumfähre im oberen Programm sicher zu landen, geben Sie das folgende Programm ein, und nachdem Sie einmal erfolgreich landen können, arbeiten Sie Ihr eigenes Display aus. Die verwendeten Variablen sind A (Geschwindig-

keit), B (Höhe), C (Treibstoff) und T (Schub). Die benutzten Formeln sind:
$H = H + Vt + {}^1/_2 at^2$
$V = V + at$
Eine Eingabe von 5 setzt den Schub der Anziehungskraft gleich.

```
  10 LET A=-20-INT (RND*60)
  20 LET B=450+INT (RND*50)
  30 LET C=120+INT (RND*30)
  40 PRINT AT 5,0;"HOEHE#";B;"#G
ESCHW.";A;"##","TANK#";C;"###"
  50 IF C<1 THEN GOTO 180
  60 PRINT AT 10,8;"WIE VIEL SCH
UB?"
  70 INPUT S
  80 PRINT AT 10,8;"############
###"
  90 IF S>C THEN LET S=0
 100 LET C=C-S
 110 LET B=B+A+(S-5)/2
 120 LET A=A+(S-5)/2
 130 IF B>0 THEN GOTO 40
 140 IF ABS B<5 AND ABS A<5 THEN
 PRINT AT 5,0;"SIE SIND SICHER G
ELANDET"
 150 IF ABS B<5 AND ABS A<5 THEN
 STOP
 160 PRINT AT 5,0;"HOEHE#";B;"#T
ANK#";C;"#####","###########"
 170 GOTO 190
 180 PRINT AT 5,0;"SIE HABEN KEI
NEN TREIBSTOFF MEHR"
 190 PRINT AT 10,0;"SIE SCHLUGEN
 MIT#";ABS A;"#KM/H AUF"
```

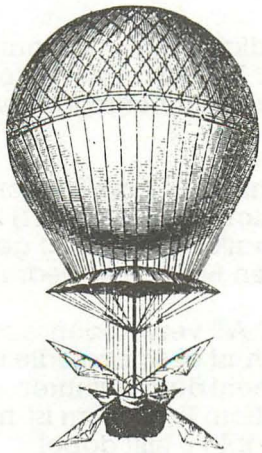
Das letzte Mondlandungsprogramm hat ein Display und paßt in 1 K. Es verwendet eine modifizierte Form der Formeln, die im vorhergegangenen Programm benutzt wurden.

In diesem Programm ist K die Position des Raumschiffes auf dem Bildschirm. S ist Treibstoff, H ist Höhe und Z sorgt dafür, daß das »alte« Raumschiff gelöscht wird, bevor das neue auf den Bildschirm gedruckt wird.

Wenn Sie PRINT AT verwenden, um bewegliche Graphik zu erzeugen, ist es sinnvoll, die Lösung erst im letztmöglichen Moment durchzuführen, um die Zeit, auf der das Objekt nicht am Bildschirm ist, möglichst gering zu halten. Die Variable Z hilft dabei.

```
  10 LET K=3
  20 LET B=300
  25 LET G=15
  30 LET H=500
  40 LET Z=H
  50 PRINT AT 13,15;"▀▀▀▀"
  60 PRINT AT 6,25;"T#";B;"###";
TAB 25;"H#";H;"###";TAB 25;"G#";
G;"###"
  70 PRINT AT (600-Z)/50,K;"##"
  80 LET K=K+RND
  90 PRINT AT (600-H)/50,K;"▄▀▄"
 100 LET Z=H
 110 INPUT S
 120 LET G=G+5-S
 130 LET H=H-G
 140 LET B=B-ABS S
 145 IF B<1 THEN GOTO 190
 150 IF H>0 THEN GOTO 60
 160 IF G>10 THEN GOTO 190
 170 PRINT B-G;"#PUNKTE FUER DIE
SE LANDUNG"
 180 GOTO 170
 190 PRINT "#AUFSCHLAG MIT#";G;
 200 GOTO 190
```

Ich habe drei Versionen dieses Programms aufgenommen, um zu zeigen, wie verschiedene Methoden das Display, den Einfluß des Spielers und die Chance auf Erfolg verändern können. Nachdem Sie nun drei, etwas verschiedene Programme in Aktion gesehen haben, schreiben Sie Ihr eigenes Mondlandungsprogramm.



****

Programme schreiben

****

Die meisten Programmierbücher empfehlen Ihnen, mit dem Zeichnen eines »FLOWCHART« zu beginnen — einer netten Kombination von Kreisen, diamantenförmigen und schrägen Rechtecken, die den Weg und die Operationen zeigen, denen der Computer folgen muß, um das Programm auszuführen. In der Theorie ist das wunderbar, aber in der Praxis, besonders dann, wenn man mit einem Computer wie dem ZX81 arbeitet, sind die Zeit und der damit verbundene Aufwand es nicht wert.

Trotzdem ist es grundlegend wichtig, exakt zu wissen, was der Computer tun soll, bevor man beginnt, ein neues Programm zu entwickeln. Sogar dann, wenn Sie noch nicht wissen, wie Sie dies auf dem ZX81 erreichen. Manchmal kann eine grobe Art Flowchart — nur die wichtigsten Schritte, die der ZX81 braucht, verbunden durch Linien und Bogen — helfen, Ihre Gedanken auf das Wichigste zu konzentrieren. Das ist auch eine gute Methode, mögliche Probleme gar nicht erst aufkommen zu lassen, wie beispielsweise eine ewige Schleife, oder nicht genau zu wissen, wie die Aufgabenstellung für den Computer lautet.
Ich arbeite normalerweise den Kern eines Programmes auf Papier aus und gebe ihn dann in den ZX81 ein, wobei ich bei Zeile 1ØØ beginne, um genug Platz davor zu lassen für die Belegung von Variablen, Aufstellung von Arrays und dergleichen mehr.
Wenn Sie an einem sehr einfachen Spiel arbeiten oder ein Programm von einem Buch oder einer Zeitschrift adaptieren, können Sie genauso gut direkt am ZX81 arbeiten. Aber haben Sie ein Notizbuch griffbereit, um sich Dinge, wie beispielsweise, daß der String A $ für den Namen des Spielers verwendet wird, notieren zu können. Wenn Sie eine 16 K Speichererweiterung haben, lassen Sie trotz der Anwesenheit des Zusatzspeichers keine schlechten Programmiergewohnheiten einreißen. Es ist sehr einfach, eine lange und schwerfällige Kette von IF/THEN-Befehlen aufzustellen, die eigentlich durch einen IF/THEN-Befehl und ein GOSUB er-

setzt werden sollten. Wenn Sie noch Speicherplatz übrig haben, ist es manchmal eine zu große Mühe, das Programm aufzuräumen. GOTO-Befehle entstehen meist in einer Vielzahl von Situationen, die sich meist ergeben, weil Sie nicht rechtzeitig daran gedacht haben, wieviel Zeilen Sie brauchen würden. Wenn Sie ein Listing untersuchen, achten Sie besonders auf GOTO-Befehle, die nicht mit einer Bedingung verknüpft sind.

Das Beste, was Sie einem längeren Programm geben können, ist das Element der Überraschung. Wenn Sie Situationen einbauen können, die nicht jedesmal entstehen, wenn das Spiel gespielt wird, haben Sie sichergestellt, daß es für wesentlich längere Zeit interessant bleibt, als dies der Fall wäre, wenn eine Spielsituation bei jedem Problemlauf ausgelöst wird.

Sie können langsam laufende Spiele bis zu einem gewissen Grad beschleunigen, indem Sie oft gebrauchte Subroutinen an den Anfang des Programmes stellen. Das Programm wird etwas schneller laufen, weil der Computer jedesmal, wenn er zu einem GOTO oder GOSUB Befehl kommt, das gesamte Listing durchlaufen muß. Wenn die Subroutine nun am Anfang des Listings ist, muß der ZX81 beispielsweise nicht jedesmal die Spielanleitung durchforsten, wenn er nach einer Subroutine sucht. Sie können als erste Zeile des Programms einen GOTO Befehl vorsehen, mit dem der Computer dann über die Subroutinen springt. Variable, die zu Programmbeginn einen Wert zugewiesen bekommen sollen, können auch am Ende des Programms aufgelistet werden, wenn danach ein GOTO-Befehl folgt, der den Computer zum eigentlichen Programmbeginn zurückführt.

Eine andere Möglichkeit, Programme zu verbessern und sie für längere Zeit interessant zu machen, ist, die Auswahl zwischen verschiedenen Schwierigkeitsgraden anzubieten. Achten Sie aber darauf, daß sich der Schwierigkeitsgrad tatsächlich erhöht und daß auch auf höchster Spielstufe ein durchschnittlicher Punktestand erreichbar ist.

Sie können das Spiel noch interessanter machen, indem Sie Punkte oder ähnliche Bewertungen vergeben, die in Beziehung zu der Geschwindigkeit oder der Geschicklichkeit des Spielers stehen. Eine andere Mög-

lichkeit ist, dem Spieler einen Titel zu geben (wie Flottenkapitän oder Inkompetenter Trottel), der abhängig von seiner Leistung ist. Punkte und Titel garantieren, daß der Spieler am Spiel für eine längere Zeit interessiert bleibt, und daß er sein Bestes versuchen wird, um einen Höchstpunktestand oder eine Höchstbewertung zu bekommen.

Denken Sie an diese Punkte, wenn Sie die Programme in diesem Buch verbessern, oder wenn Sie Ihre eigenen Programme schreiben.

Noughts and Crosses

Wahrscheinlich werden Sie wissen, wie dieses Spiel, (das teilweise auch als OXO bekannt ist) gespielt wird. Abwechselnd setzt jeder Spieler ein O oder ein X auf eine Stelle in einem 3x3-Gitter und versucht, drei seiner Steine in eine Reihe (horizontal, vertikal oder diagonal) zu bekommen. In dieser Version — die gerade noch in 1 K paßt — beginnt immer der Computer und belegt das oberste linke Feld. Trotz der Kürze des Programms ist der ZX81 unschlagbar, obwohl Sie von Zeit zu Zeit ein Unentschieden erreichen können, wenn Sie ein guter Spieler sind. Um einen Stein auf ein Feld zu setzten, berühren Sie einfach die entsprechende Zahl; sofort wird das Feld mit einem Ihrer Steine belegt. Die Antwort des ZX81 kommt genauso schnell.
Der ZX81 weiß nicht, ob er gewonnen hat, da das Programm mit einer entsprechenden Routine 1 K gesprengt hätte. Wenn Sie eine 16 K-Speichererweiterung haben, könnten Sie versuchen, eine »Ich-habe-gewonnen«- und »Es-ist-ein-Unentschieden«-Routine einzubauen.
Auf den ersten Blick scheint es einfach, das Noughts-and-Crosses-Spiel zu programmieren, aber das ist nicht der Fall, und es ist schwieriger, ein verletzbares Programm zu schreiben, als eines, das praktisch unschlagbar ist.
Nach dem Buch »COMPUTERS, THEIR IMPACT AND USE« von Robert E. LYNCH und Ray RICE gibt es 362.88Ø verschiedene Partien von Noughts-and-Crosses. Dieses Programm spielt 4Ø.32Ø davon.

```
 10 PRINT "
 20 PRINT "  X  |  2  |  3"
 30 PRINT "
 40 PRINT "-----------------"
 50 PRINT "
 60 PRINT "  4  |  5  |  6"
 70 PRINT "
 80 PRINT "-----------------"
 90 PRINT "
100 PRINT "  7  |  8  |  9"
110 PRINT "
120 GOSUB 200
130 LET A=T-1
140 LET B=T
150 LET T=CODE "
                    "(8*A+B)
```

```
                    "(8*A+B)
160 LET X=61
170 GOSUB 260
180 GOSUB 200
190 GOTO 140
200 LET X=52
210 IF INKEY$<>"" THEN GOTO 210
220 IF INKEY$="" THEN GOTO 220
230 LET T=CODE INKEY$-29
240 PRINT AT CODE "          "(T)
CODE "          "(T);CHR$ X
250 RETURN
```

Wenn Sie dieses Programm auf einem ZX8O mit neuem ROM laufen lassen, ändern Sie die folgenden Zeilen:

```
210 PAUSE 4E4
220 POKE 16437,-1
```

Rollerball

Ein Ball wird über den Bildschirm gegen eine Wand geschleudert. Seine Position bestimmt Ihren Punktestand. Lassen Sie das Programm laufen, um herauszufinden, wie man es spielt, und arbeiten Sie dann die Strategie aus, um den Höchstpunktestand zu erreichen.

```
 10 PRINT TAB 7;"ROLLER-BALL"
 20 LET H=3
 30 LET F=10*(RND*5)+1
 40 FOR G=1 TO 10
 50 PRINT "DAS IST VERUCH NR.";
G
 60 LET F=F+3*RND
 70 PRINT AT 11,0;"WOHIN WOLLEN
SIE ROLLEN (1-100)?"
```

```
  80 IF G>1 THEN PRINT AT 18,0;"
LETZTER VERSUCH WAR AUF#";D
  90 LET C=1
 100 INPUT D
 110 IF D<1 OR D>100 THEN GOTO 1
00
 120 CLS
 130 FOR E=1 TO 18
 140 PRINT AT 0,0;"11111??9??111
1????***????111111"
 150 PRINT AT 18-E,C;CHR$ (43+E)
 170 IF E<18 THEN PRINT AT 18-E,
C;"#"
 180 LET C=C+D/F
 190 IF C>31 THEN PRINT AT 13,10
;"QUERSCHLAEGER"
 200 IF C>31 THEN LET C=1
 210 NEXT E
 220 IF C<6 OR C>9 AND C<16 OR C
>25 THEN LET K=1
 230 IF C>5 AND C<10 OR C>15 AND
 C<19 OR C>22 AND C<26 THEN LET
K=9
 240 IF C>18 AND C<23 THEN LET K
=50
 250 PRINT "IHRE PUNKTE:";K
 260 LET H=H+K
 270 IF G>1 THEN PRINT "GESAMTPU
NKTE:";H;"#AUS#";G;"#WUERFEN"
 280 NEXT G
```

High-Flyer

Wenn Sie beginnen wollen, Spiele aus diesem Buch zu verbessern, dann beginnen Sie mit dem nächsten Programm. Wie Sie bemerken werden, wenn Sie das Programm laufen lassen, hat der Spieler praktisch nichts zu tun. Nachdem Sie einmal RUN, gefolgt von NEWLINE, gedrückt haben, brauchen Sie sich nur zurücklehnen und das Spiel zu beobachten.
Sie können eine »Drücken Sie NEWLINE für das nächste Spiel« — Routine hinzufügen, oder, wenn Sie sich besonders kreativ fühlen, die Möglichkeit, eine oder mehrere Walzen anzuhalten. Wenn Ihnen HIGH FLYER zu schnell läuft, fügen Sie eine Schleife FOR N=1 TO 50, NEXT N, hinzu. Sie müssen allerdings herausfinden, wo diese Schleife einzubauen ist.

```
  20 PRINT TAB 5;"$$ HIGH-FLYER
$$"
  30 PRINT
```

```
  40 PRINT "ZEIT FUER DIE FRUIT-
MACHINE,"
  50 PRINT "ICH HOFFE,SIE HABEN
GUTE","REAKTIONEN"
  60 PAUSE 200
  70 LET M=(INT (RND*20)+10)/2
  80 CLS
  90 PRINT "SIE HABEN ZU BEGINN
$";M
 100 LET C=0
 110 LET D=0
 120 LET E=0
 130 LET F=0
 140 PRINT
 150 PRINT
 160 PRINT
 170 PRINT TAB 5;
 180 FOR Z=1 TO 4
 190 LET B=INT (RND*4)+1
 200 IF B=2 THEN GOTO 260
 210 IF B=3 THEN GOTO 290
 220 IF B=4 THEN GOTO 320
 230 LET C=C+1
 240 PRINT "#$$$$#";
 250 GOTO 340
 260 LET D=D+1
 270 PRINT "#■■■■#";
 280 GOTO 340
 290 LET E=E+1
 300 PRINT "#▒▒▒▒▒#";
 310 GOTO 340
 320 LET F=F+1
 330 PRINT "#****#";
 340 PAUSE 50
 350 NEXT Z
 360 PRINT
 370 PRINT
 380 IF C=1 AND D=1 AND E=1 AND
F=1 THEN GOTO 450
 390 IF C=4 OR D=4 OR E=4 OR F=4
 THEN GOTO 450
 400 IF C=3 OR D=3 OR E=3 OR F=3
 THEN GOTO 500
 410 LET S=(INT (RND*4)+1)/2
 420 PRINT "SIE VERLIEREN $";S
 430 LET M=M-S
 440 GOTO 550
 450 PRINT "$ JACKPOT $"
 460 PRINT
 470 LET M=M+10
 480 PRINT "SIE GEWINNEN $10"
 490 GOTO 550
 500 LET M=M+2
 510 PRINT
```

```
 520 PRINT "3 VON EINER SORTE"
 530 PRINT
 540 PRINT "SIE GEWINNEN $2"
 550 IF M<1 THEN GOTO 640
 560 IF M>25+RND*30 THEN GOTO 67
0
 570 PRINT
 580 PRINT "IHR GUTHABEN:$";M
 590 PRINT
 600 PRINT TAB 8;"BITTE WARTEN"
 610 PAUSE 200
 620 CLS
 630 GOTO 100
 640 PRINT "SPIELENDE,LEICHTSINN
IGER","SIE SIND PLEITE"
 660 GOTO 680
 670 PRINT "SIE HABEN DIE BANK M
IT $";M,"GESPRENGT"
 680 PAUSE 400
 690 CLS
 700 RUN
```

Kuchenkampf

Der Zufallszahlengenerator bäckt ein Backrohr voll Kuchen, die Sie und der ZX81 abwechselnd essen. Es gibt ein Limit, wie viele Kuchen auf einmal gegessen werden dürfen, und der Verlierer ist derjenige, der den letzten Kuchen ißt. Es ist recht leicht, ein Programm zu schreiben, das dieses Spiel niemals verliert, aber ein derartiges Programm wird rasch langweilig, wenn nicht ärgerlich. Deshalb hat dieses Programm eine eingebaute Schwäche. Trotzdem wird es mehr als die Hälfte aller Spiele gewinnen.

```
   1 PRINT TAB 4;"KUCHENKAMPF"
   2 SLOW
   3 LET M=0
   4 LET E=0
   5 LET Z=10+INT (RND*20)
   6 LET H=3+INT (RND*3)
   7 IF M=0 THEN PRINT "ES SIND
NOCH#";Z;"#KUCHEN DA","SIE KOENN
EN MAX.";H;"#KUECHEN ESSEN"
   8 IF E>0 AND M=0 THEN PRINT "
SIE HABEN#";E;"#GENOMMEN","ICH H
ABE#";Q;"#GENOMMEN"
   9 FOR K=1 TO Z
  10 PRINT "@@@@#";
  11 IF RND>.6 THEN PRINT
```

```
  12 NEXT K
  13 PRINT
  14 PRINT
  15 IF M=1 THEN PRINT TAB 4;"SI
E GEWINNEN"
  16 IF M=2 THEN PRINT TAB 8;"IC
H GEWINNE"
  17 IF M>0 THEN GOTO 15
  18 PRINT
  19 PRINT "WIE VIELE WOLLEN SIE
 NEHMEN?"
  20 INPUT E
  21 IF E<1 OR E>H THEN GOTO 20
  22 LET Z=Z-E
  23 IF Z>0 THEN GOTO 26
  24 LET M=2
  25 GOTO 34
  26 LET Q=Z-1-INT ((Z-1)/(H+1))
*(H+1)
  27 IF Q=0 AND Z<>1 THEN LET Q=
INT (RND*H)+1
  28 IF Q>=Z THEN GOTO 27
  29 IF Z>E+2 AND RND>.4 THEN LE
T Q=Q+INT (RND*2)-INT (RND*2)
  30 IF Q>H OR Q<0 THEN GOTO 26
  31 IF Q=0 THEN LET Q=1
  32 LET Z=Z-Q
  33 IF Z=0 THEN LET M=1
  34 CLS
  35 GOTO 7
```

Hier ist eine etwas einfachere Version des gleichen Programms, die in 1 K paßt und die dem Spieler eine etwas bessere Gewinnchance gibt. Nach der ersten Fassung finden Sie eine Änderung, die Ihnen eine sehr gute Gewinnchance gibt.

Timo Nimbo

```
  10 LET M=0
  20 LET E=0
  30 LET Z=15+INT (RND*10)
  40 IF 2*INT (Z/2)=Z THEN GOTO
30
  50 LET H=3
  60 IF E>0 THEN PRINT AT 7,0;"S
IE HABEN#";E;"#GENOMMEN,","ICH#"
;Q
  70 FOR K=1 TO Z
  80 PRINT K;"# ■";
  90 IF RND>.6 THEN PRINT
```

```
100 NEXT K
110 INPUT E
115 IF E>H THEN GOTO 110
120 LET Z=Z-E
130 IF Z=0 THEN PRINT "ICH GEWI
NNE"
140 IF Z=0 THEN STOP
150 LET Q=Z-1-INT ((Z-1)/(H+1))
*(H+1)+INT (RND*2)
160 IF Q>Z OR Q=0 OR Q=4 THEN G
170 LET Z=Z-Q
180 IF Z=0 THEN PRINT "SIE GEWI
NNEN"
190 IF Z=0 THEN STOP
200 CLS
210 GOTO 60
```

Es gibt keinen Unterschied zwischen den beiden Programmen, was die beim Entfernen erlaubte Höchstanzahl betrifft — jeweils drei. Um sich selbst eine bedeutend bessere Gewinnchance einzuräumen, machen Sie im obigen Programm folgende Änderungen:

```
150 LET Q=INT (RND*3)+1
160 IF Q>Z THEN GOTO 150
```

Deducto

Ein sehr kurzes 1 K Programm, in dem Sie 1Ø Versuche haben, um eine, vom ZX81 gewählte Zahl zwischen 1 und 2ØØ zu finden. An den Tip, den Sie in Zeile 9Ø bekommen, müssen Sie sich erst ein bißchen gewöhnen, aber nachdem Sie einmal drei Spiele gespielt haben, sollten Sie in der Lage sein, den Hinweis ohne Schwierigkeiten zu verstehen. Beachten Sie, wie SCROLL verwendet wird, um einen Bildschirmüberlauf zu verhindern. SCROLL muß im Programm vor jedem PRINT-Befehl verwendet werden. Versuchen Sie einfach, SCROLL einmal auszulassen und warten Sie ab, was geschieht.

```
10 LET A=INT (RND*200)+1
20 IF RND>RND THEN GOTO 10
30 FOR B=1 TO 10
40 SCROLL
50 PRINT "VERSUCH NR.";B
60 INPUT C
70 SCROLL
80 IF C=A THEN GOTO 110
```

```
  90 PRINT C;"#FALSCH. HILFE:";S
QR (ABS (A-C))
 100 NEXT B
 110 SCROLL
 120 PRINT A
 130 IF C<>A THEN STOP
 135 SCROLL
 140 PRINT "SIE HABEN ES IN#";B;
"#VERSUCHEN"
 150 SCROLL
 160 PRINT "GESCHAFFT"
```

Unglaublich verschwindendes RAM

Entfernen Sie das Display von Geiger, ändern Sie das Programm ein wenig, und Sie haben ein 1 K Spiel, in dem Sie versuchen, ein fehlendes RAM Chip in einem 15x15 Gitter, unter Verwendung Ihrer Intelligenz und eines patentierten RAM Detektors, zu finden.

```
  10 LET A=INT (RND*15)+1
  20 LET B=INT (RND*15)+1
  30 LET C=1
  40 SCROLL
  50 SCROLL
  60 PRINT "SUCHE#";C
  70 INPUT D
  80 INPUT E
  90 IF A=D AND B=E THEN GOTO 16
0
 100 SCROLL
 110 PRINT "####";D;"##";E;"##ES
 IST NICHT DA"
 120 SCROLL
 130 PRINT "DETEKTOR MISST:";(A-
D)+(B-E)/10
 140 LET C=C+1
 150 GOTO 40
 160 SCROLL
 170 PRINT "SIE HABEN ES GEFUNDE
N"
 180 SCROLL
 190 PRINT "SIE ERREICHTEN#";300
-9*C;"#PUNKTE"
 200 GOTO 160
```

In diesem Programm ist das RAM auf A und B versteckt und C zählt die Anzahl Ihrer Versuche. Zeile 13Ø: Der RAM Detektor gibt Ihnen sehr nützliche Informationen.

Nach zwei oder drei Spielen sollten Sie seine Anzeige verstanden haben. Beachten Sie, daß — im Gegensatz zu einigen anderen »Gittersuchspielen« (wie z. B. 3 D-Jagd) — die Anzahl Ihrer Versuche nicht beschränkt ist. Trotzdem sollten Sie versuchen, das RAM in der kürzest möglichen Zeit zu finden. Ihre Bewertung (Zeile 19Ø) steht in direkter Beziehung zu der Anzahl der von Ihnen gemachten Versuche, um das Ding zu finden.

Chemin de fer

Bakkarat wurde ungefähr 154O unter der Regierung von Karl VIII. von Frankreich nach Italien gebracht. Nach der Meinung von Geschichtsforschern ist es sehr unwahrscheinlich, daß Karl es auf einem ZX81 gespielt hat. Das italienische Spiel hieß Bakkarat und dieses Spiel — Chemin de Fer — ist ein entfernter Verwandter dieses alten Klassikers. ZX81-Chemin de Fer basiert auf einer Würfelversion des Casinospiels, das üblicherweise mit Karten gespielt wird. Sie und der ZX81 (die Bank) werfen jeder fünf Würfel. Wenn irgendein Würfel auf zwei oder fünf fällt, muß er erneut geworfen werden. Sie addieren die Augenzahl der Würfel, die nicht auf zwei oder fünf gefallen sind und addieren sie zur Gesamtsumme der erneut geworfenen Würfel.
Fällt ein Würfel das zweite Mal auf zwei oder fünf, zählt er als Null. Das Ziel des Spielers ist es, so nah wie möglich an die Zahl neun oder an eine zweiziffrige Zahl mit der Endziffer neun heranzukommen. Das Programm kürzt eine zweiziffrige Nummer automatisch auf ihre Endziffer. Nach neun gewonnenen Spielen (Unentschieden nicht gezählt) stoppt das Programm.

```
 10 LET B1=0
 20 LET P1=0
 30 GOTO 340
 40 LET D=0
 50 LET C=0
 60 FOR G=1 TO 5
 70 LET A=INT (RND*6)+1
 80 IF A=2 OR A=5 THEN LET C=C+
1
 90 IF A=2 OR A=5 THEN LET A=0
100 PRINT A;"#";
110 LET D=D+A
120 NEXT G
130 PRINT
140 PRINT D;"##";
```

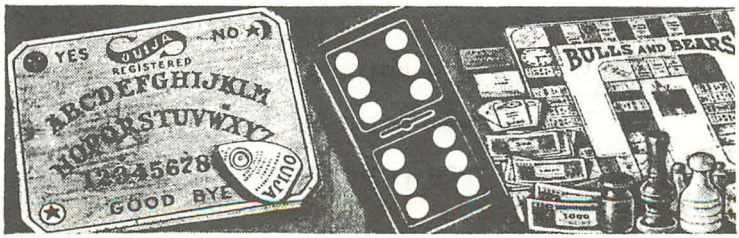
```
 150 IF D>9 THEN LET D=D-10
 160 PRINT D
 170 IF D>9 THEN LET D=D-10
 180 PRINT "GESAMT DER ERSTEN WU
RFES:";D
 190 IF D=9 THEN PRINT ,"LA GRAN
DE"
 200 IF D=8 THEN PRINT ,"LA PETI
TE"
 210 IF D=7 THEN PRINT ,"NATURAL
"
 220 IF C=0 OR D=7 OR D=8 OR D=9
 THEN RETURN
 230 PRINT "ES MUSS EINE#";C;"#N
OCHEINMAL","GEWORFEN WERDEN"
 240 FOR A=1 TO C
 250 LET E=INT (RND*6)+1
 260 IF E=2 OR E=5 THEN LET E=0
 270 LET D=D+E
 280 NEXT A
 290 PRINT D;"##";
 300 IF D>9 THEN LET D=D-10
 310 PRINT D
 320 IF D>9 THEN LET D=D-10
 330 RETURN
 340 PRINT "BANK"
 350 GOSUB 40
 360 PRINT "ENDGESAMT:";D
 370 INPUT A$
 380 LET J=D
 390 PRINT "SPIELER"
 400 GOSUB 40
 405 PRINT "ENDGESAMT:";D
 410 INPUT A$
 420 PRINT "BANK","SPIELER"
 430 PRINT J,D
 440 IF J=D THEN PRINT "UNENTSCH
IEDEN"
 450 IF J=D THEN GOTO 510
 460 IF J>D THEN PRINT "BANK";
 470 IF J>D THEN LET B1=B1+1
 480 IF J<D THEN PRINT "SPIELER"
;
 490 IF J<D THEN LET P1=P1+1
 500 PRINT "#GEWINNT"
 510 PRINT "GESAMT"
 520 PRINT B1,P1
 530 IF B1+P1=9 THEN STOP
 540 INPUT A$
 550 CLS
 560 GOTO 340
```

****

Aufbau einer Programmbibliothek

****

SAVEn Sie jedes Programm, bevor Sie es laufen lassen, nur für den Fall, daß Sie einen Programmfehler eingebaut haben, der das Programm zusammenbrechen und verschwinden läßt.
Widerstehen Sie der Versuchung, alle Ihre Programme auf eine Cassette zu speichern — ein Programm nach dem anderen! Die Frustration, die Sie erleben werden, während Sie warten, bis der ZX81 ein bestimmtes Programm gefunden hat (unter der Voraussetzung, daß Sie in der Lage sind, sich an den Codenamen, den sie dem Programm gegeben haben, zu erinnern), ist das Geld, das Sie sich bei den Cassetten ersparen, nicht wert.
Gehen Sie in einen Computershop und erstehen Sie ein Paket C 12 Cassetten. Nehmen Sie immer nur ein Programm auf jede Cassette auf; mindestens zweimal auf jeder Seite. Für den Fall, daß etwas mit Ihren Cassetten geschieht und daß es schwierig oder unmöglich ist, ein Programm zu laden.
Schreiben Sie den Programmnamen und den Codenamen, unter dem Sie das Programm gespeichert haben, auf die Cassette und auf die Einlagekarte. Sie werden sehen, dies erleichtert es, das Programm zu finden, das Sie wollen, und der Aufbau Ihrer Bibliothek wird Ihnen ein gewisses Gefühl der Befriedigung verleihen, wenn Sie die vielen Programme Seite an Seite in Ihrem Bücherschrank stehen sehen.



Cricket

Ich habe in diesem Programm England gegen Australien spielen lassen, aber es gibt keinen Grund, warum Sie dies nicht durch Teams Ihrer Wahl ersetzen sollten. Sie können das Programm leicht umschreiben, so daß es, wenn es das erste Mal läuft, um die Eingabe der beiden Teams ersucht. Wenn Sie sich entscheiden, dieses Programm in SLOW laufen zu lassen und die PAUSE-Befehle löschen, dann fügen Sie eine Schleife FOR N=1 TO 30, NEXT N, zwischen den Zeilen 1170 und 1180 ein.

```
1000 PRINT AT 3,9;"KRICKET"
1020 GOTO 1340
1030 PAUSE 100
1040 LET B=B+1
1050 IF B=7 THEN LET C=C+1
1060 IF B=7 THEN LET B=1
1070 CLS
1080 LET E=INT (RND*7)+1
1120 IF E>5 THEN GOSUB 1205
1135 IF A=2 THEN PRINT AT 11,5;
AUSTRALIEN:";A(4)
1137 PRINT AT 9,5;"ENGLAND:";A(3
)
1140 IF RND>.5 THEN LET E=0
1150 PRINT AT 13,5;"SPIELER#";D
1155 PRINT AT 15,5;"UEBER-";C;"#
##BALL#";CHR$ (B+156);"#"
1157 PRINT AT 17,5;"LAEUFT VON B
ALL#";B;"-";E
1170 LET A(A+2)=A(A+2)+E
1180 GOTO 1030
1205 LET P=INT (RND*3)+1
1207 CLS
1210 FOR Y=1 TO 6
1211 PRINT AT 12,9;"SPIELER#";D;
"#OUT"
1212 PAUSE 10
1213 PRINT AT 12,9;"SPIELER#";D;
"#OUT"
1214 PRINT AT 16,9;"**********
******"
1217 PAUSE 10
1220 IF P=1 THEN PRINT AT 16,11;
"#AUS DEM FELD#"
1230 IF P=2 THEN PRINT AT 16,11;
"#GEFANGEN#"
1240 IF P=3 THEN PRINT AT 16,11;
"GEWORFEN"
1245 NEXT Y
```

```
1250 LET D=D+1
1255 PAUSE 200
1257 CLS
1260 IF D=11 THEN GOSUB 1420
1270 IF A<>3 THEN RETURN
1280 CLS
1290 PRINT TAB 4;"SPIELENDE"
1295 PRINT
1297 IF A(3)>A(4) THEN PRINT "EN
GLAND#";
1310 IF A(4)>A(3) THEN PRINT "AU
STRALIEN#";
1320 PRINT "GEWINNT UM#";ABS (A(
3)-A(4));"#LAEUFEN"
1330 PAUSE 700
1334 RUN
1340 DIM A(4)
1350 LET A=1
1360 LET B=0
1370 LET C=1
1380 LET D=1
1390 LET A(3)=0
1400 LET A(4)=0
1410 GOTO 1030
1420 LET A=A+1
1425 CLS
1430 LET B=1
1435 IF A=2 THEN PRINT AT 13,4;"
ENGLAND AUS AUF#";A(3)
1437 PAUSE 200
1440 LET C=1
1450 LET D=1
1460 CLS
1470 RETURN
```

Digitaluhr

Dieses kurze Programm macht effektvollen Gebrauch von PAUSE und ist für den Ablauf im FAST-MODE geschrieben. Sollten Sie den reibungslosen Ablauf von SLOW bevorzugen, können Sie Zeile 17Ø löschen und durch eine kleine Schleife ersetzen, die dem ZX81 auch im SLOW-MODE erlaubt, die Zeit vernünftig anzuzeigen. Diese Uhr funktioniert nicht sehr genau, aber es ist eine gute Demonstration des ZX81.

```
10 GOSUB 200
60 PRINT AT 5,4;H;":";
70 IF M<10 THEN PRINT "0";
80 PRINT M;":";
90 IF S<10 THEN PRINT "0";
```

```
100 PRINT S
110 LET S=S+1
120 IF S=60 THEN LET M=M+1
130 IF S=60 THEN LET S=0
140 IF M=60 THEN LET S=S+1
150 IF M=60 THEN LET M=0
160 IF H=13 THEN LET S=1
170 PAUSE 33
190 GOTO 30
200 PRINT
210 PRINT
220 PRINT TAB 4;"DIGITALUHR"
230 PRINT
240 PRINT "WIE SPAET IST ES?"
250 PRINT
260 PRINT "STUNDEN?"
270 INPUT H
280 IF H>12 THEN GOTO 270
285 PRINT H
290 PRINT "MINUTEN?"
300 INPUT M
310 IF M>59 THEN GOTO 300
315 PRINT M
320 PRINT "UND SCHLIESSLICH,SEK
UNDEN?"
330 INPUT S
340 IF S>59 THEN GOTO 330
350 CLS
360 RETURN
```

Das nächste kleine Programm druckt eine Anzahl von inversen Zwischenräumen auf den Bildschirm. Ihre Positionen hängen davon ab, welche Taste (5, 6, 7 oder 8) Sie drücken. Die Schleife FOR N=1 TO 1Ø, NEXT N, wird verwendet, um das Ganze etwas zu verlangsamen. Ohne sie geht Ihnen das Ganze wahrscheinlich zu schnell, um es noch vernünftig steuern zu können. Trotzdem können Sie diese Schleife entfernen, verlängern oder verkürzen, wenn Sie wollen.

```
 10 SLOW
 20 GOSUB 180
 30 PRINT AT Y,X;CHR$ K
 40 LET P$=A$
 50 LET A$=INKEY$
 60 IF A$="" THEN LET A$=P$
 70 IF A$="7" THEN LET Y=Y-1
 80 IF A$="6" THEN LET Y=Y+1
 90 IF A$="5" THEN LET X=X-1
100 IF A$="8" THEN LET X=X+1
110 IF X<2 THEN LET X=2
120 IF X>31 THEN LET X=31
130 IF Y<2 THEN LET Y=2
```

```
140 IF Y>20 THEN LET Y=20
150 FOR N=1 TO 10
160 NEXT N
170 GOTO 30
180 LET X=16
190 LET Y=7
200 LET K=128
210 LET A$=""
220 RETURN
```

Die Subroutine (von Zeile 18Ø an) plaziert den ersten Punkt (auf 7,16) und legt fest, daß CHR $ 128 ausgedruckt wird. Auf alle Fälle können Sie die Startposition und das Aussehen der Zeichen verändern (oder durch den Zufallsgenerator bestimmen lassen, wenn Ihnen das besser gefällt).

Back-Stabber

Das Programm erklärt sich selbst und ist lustig zu spielen.

```
  10 LET D=20
  20 FOR F=1 TO 30
  30 PRINT AT 10,10;"BITTE WARTE
N"
  40 PRINT AT 10,10;"BITTE WARTE
N"
  50 NEXT F
  70 CLS
  80 LET A=INT (RND*10)+1
  90 LET B=INT (RND*10)+1
 100 IF NOT ABS (B-A)>1 THEN GOT
O 90
 110 LET C=INT (RND*10)+1
 120 IF C=A OR C=B THEN GOTO 110
 130 PRINT "MEINE ERSTE ZAHL IST
#";A
 140 PRINT TAB 8;"MEINE ZWEITE#"
;B
 150 PRINT
 160 PRINT TAB 4;"SIE HABEN $";D
 170 PRINT
 180 PRINT "WIEVIEL WETTEN SIE D
ARAUF,DASS"
 190 PRINT "MEINE NAECHSTE ZAHL
ZWISCHEN"
 195 PRINT A;"#UND#";B;"#LIEGT?"
 200 INPUT E
 210 IF E>D THEN GOTO 200
```

```
 220 IF E<.01 THEN PRINT "GAUNER
"
 230 FOR F=1 TO 50
 240 PRINT AT 10,10;"ICH DENKE"
 250 PRINT AT 10,10;"ICH DENKE"
 260 NEXT F
 270 PRINT AT 10,10;"##########"
 280 PRINT
 290 PRINT "MEINE ZAHL IST#";C
 300 IF E<.01 THEN GOTO 20
 310 IF C>A AND C<B OR C<A AND C
>B THEN PRINT "BRAVO,SIE GEWINNE
N $";2*E
 320 IF C>A AND C<B OR C<A AND C
<B THEN LET D=D+2*E
 330 IF C>A AND C<B OR C<A AND C
<B THEN GOTO 20
 340 PRINT "LEIDER VERLIEREN SIE
 $";E-.5
 350 LET D=D-E+.5
 360 IF D>.5 THEN GOTO 20
 370 PRINT "SIE SIND PLEITE"
```

Zombies

Ein massives schwarzes Rechteck erscheint auf dem Bildschirm. Zuerst erscheinen Sümpfe (graue Vierecke), dann die Zombies (inverse Sterne), dann Sie (ein inverses A). Plötzlich gehen die Zombies auf Sie zu. Es ist egal, wohin Sie gehen, die Zombies bewegen sich auf Sie zu. Ihre einzige Fluchtmöglichkeit ist, sie in die Sümpfe zu locken. Sie können die Sümpfe nicht überqueren, aber wenigstens können Sie sie sehen. Die Zombies sind blind, und darin liegt Ihre einzige Hoffnung.

Die Anfangsposition wird zufällig aufgestellt, mit der Ausnahme, daß die Subroutine von Zeile 1ØØØ an etwas beeinflußte Zufallszahlen produziert, um alles ungefähr an den richtigen Platz zu stellen (zum Beispiel Sie ungefähr in die Mitte, eine großzügige Anzahl Sümpfe und die Zombies außerhalb der Sümpfe).

Sie bewegen sich in die Richtungen der kleinen Richtungspfeile auf den Tasten 5, 6, 7 und 8, indem Sie die entsprechende Taste einmal drücken. Um diagonal zu ziehen, verwenden Sie die Shifttaste und dann 5, 6, 7 oder 8. Sie werden sich 45° gegen den Uhrzeigersinn bewegen, wenn Sie den Pfeil und Shift gleichzeitig drücken. Sie können immer nur einen Zug auf

einmal machen und müssen warten, bis alle Zombies sich ebenfalls bewegt haben, bevor Sie Ihren nächsten Zug machen können. Wenn Sie ein Zombie erwischt hat, geht das Programm in eine ewige Schleife, und Sie müssen BREAK drücken, um es zu stoppen.

```
  2 RAND
  5 DIM X(21)
  6 DIM Y(21)
 10 PRINT "                              
    "
 20 FOR I=1 TO 20
 30 PRINT "                              
    "
 40 NEXT I
 50 PRINT "                              
    "
 70 FOR I=1 TO 42
 80 LET A=15*INT (I/21)+136
 90 GOSUB 1000
100 IF INT (I/21)=1 THEN LET X(
I-20)=X
110 IF INT (I/21)=1 THEN LET Y(
I-20)=Y
120 NEXT I
130 LET U=X
140 LET V=Y
160 IF INKEY$="" THEN GOTO 160
170 LET B=CODE INKEY$
200 IF (B=33 OR B=112 OR B=114)
AND U>1 THEN LET U=U-1
210 IF (B=34 OR B=113 OR B=114)
AND V<20 THEN LET V=V+1
220 IF (B=35 OR B=112 OR B=115)
AND V>1 THEN LET V=V-1
230 IF (B=36 OR B=113 OR B=115)
AND U<20 THEN LET U=U+1
234 PRINT AT Y,X+4;"#"
235 LET X=U
236 LET Y=V
240 PRINT AT Y,X+4;"A"
250 FOR I=1 TO 21
260 IF X(I)=0 THEN GOTO 350
270 PRINT AT Y(I),X(I)+4;"#"
280 LET S=0
290 IF X(I)-U THEN LET S=(X(I)-
U)/ABS (X(I)-U)
300 LET T=0
310 IF Y(I)-V THEN LET T=(Y(I)-
V)/ABS (Y(I)-V)
320 LET X(I)=X(I)-S
330 LET Y(I)=Y(I)-T
331 PRINT AT Y(I),X(I)+4;
```

```
 332 LET A$=CHR$ PEEK (PEEK 1639
8+256*PEEK 16399)
 333 IF A$="▒" THEN LET X(I)=0
 334 IF A$="A" THEN GOTO 370
 340 IF X(I) THEN PRINT "*"
 350 NEXT I
 360 GOTO 130
 370 PRINT AT Y,X+4;"*"
 380 PRINT AT Y,X+4;"*"
 390 GOTO 370
1000 LET X=10+SGN (RND-.5)*INT (
RND*(I/3 AND I<>42)+1)
1010 IF X<1 OR X>20 THEN GOTO 10
00
1020 LET Y=10+SGN (RND-.5)*INT (
RND*(I/3 AND I<>42)+1)
1030 IF Y<1 OR Y>20 THEN GOTO 10
20
1040 PRINT AT Y,X+4;CHR$ A
1050 RETURN
```

Biorhythmus

Die Wissenschaft der Biorhythmen basiert auf dem Glauben, daß jeder von uns von drei Zyklen bestimmt wird, die im Moment der Geburt starten und sich bis zum Tode wiederholen.

Physisch	Der dreiundzwanzigtägige körperliche Zyklus steht in Beziehung zur Ausdauer, Aggressivität und physischer Stärke
Emotionell	Der achtundzwanzigtägige emotionelle Zyklus bezieht sich auf Emotionen, Optimismus—Pessimismus, Frustrationen, Stimmung und Laune
Geistig	Der dreiunddreißigtägige geistige Rhythmus steht in Beziehung zu Logik, Denken, Folgern, Urteilen, Leichtigkeit des Ausdrucks und Verstand

Das Programm läßt Sie Ihr Geburtsdatum, dann das Datum des Tages, der Sie interessiert, eingeben und arbeitet den schlimmsten Punkt jedes der drei Zyklen in Beziehung zu dem von Ihnen angegebenen Tag aus.

```
 20 GOTO 210
 30 LET T=0
 40 IF B-3>=0 THEN LET T=2
 50 IF T=2 THEN LET B=B+1
 60 IF T=2 THEN GOTO 90
 70 LET A=A-1
```

```
  80 LET B=B+13
  90 LET E=INT (365.25*A)+INT (3
0.6*B)+C
 100 RETURN
 110 LET T=0
 120 LET F=G-INT (G/D)*D
 130 IF F>D/2 THEN LET T=2
 140 IF T=2 THEN LET H=D-F
 150 IF T=2 THEN GOTO 180
 160 IF F=0 OR D/2=F THEN PRINT
"HEUTE"
 170 IF F=0 OR D/2=F THEN RETURN

 180 LET H=D/2-F
 190 PRINT "NACH#";H;"#TAGEN"
 200 RETURN
 210 LET Y$="DER SCHLECHTESTE TA
G IST "
 220 LET X$=" ZYKLUS"
 230 CLS
 240 PRINT "IHR GEBURTSJAHR (Z.B
. 1967)?"
 250 INPUT A
 260 PRINT "GEBURTSMONAT (Z.B. 1
0)?"
 270 INPUT B
 280 PRINT "GEBURTSTAG (Z.B. 23)
?"
 290 INPUT C
 300 CLS
 310 GOSUB 30
 320 LET J=E
 330 PRINT "UM IHREN BIORHYTHMUS
 FUER EINEN"
 335 PRINT "TAG ZU ERRECHNEN,GEB
EN SIE DAS"
 340 PRINT "DATUM EIN,DAS SIE IN
TERESSIERT."
 350 PRINT
 360 PRINT "JAHR ?"
 370 INPUT A
 375 LET Z=A
 380 IF A<1900 OR A>1999 THEN GO
TO 370
 390 PRINT "MONAT ?"
 400 INPUT B
 405 LET Q=B
 410 PRINT "TAG ?"
 420 INPUT C
 421 GOSUB 8000
 422 LET P=C
 425 GOSUB 30
 430 CLS
 440 LET G=E-J
```

```
 450 PRINT "BIORHYTHMUS FUER ";P
;"/";Q;"/";
 452 IF Z-1900<10 THEN PRINT "0"
;
 455 PRINT Z-1900
 460 PRINT
 470 PRINT
 480 LET D=23
 490 PRINT "PHYSISCHER";X$
 500 PRINT Y$
 510 GOSUB 110
 520 PRINT
 530 LET D=28
 540 PRINT "EMOTIONELLER";X$
 550 PRINT Y$
 560 GOSUB 110
 570 PRINT
 580 LET D=33
 590 PRINT "MENTALER";X$
 600 PRINT Y$
 610 GOSUB 110
 620 PRINT
 630 PRINT "GEBEN SIE J EIN,UM F
UER EIN","WEITERES DATUM DEN BIO
RHYTHMUS"
 635 PRINT "ZU BERECHNEN"
 640 INPUT U$
 645 CLS
 650 IF U$="J" THEN GOTO 330
8000 CLS
8010 FOR I=1 TO 96
8015 PRINT "     *";
8040 NEXT I
8050 RETURN
```

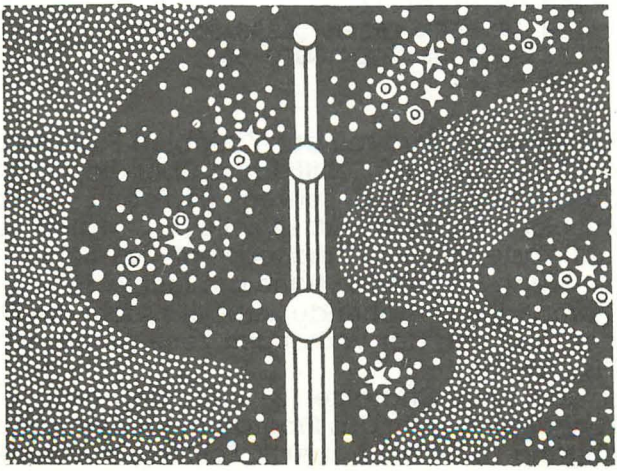
Swarm

In diesem kurzen »Arcade«-Spiel steuern Sie ein Raumschiff im Weltraum und werden plötzlich in einem Astroidenschwarm gefangen. Sie müssen Ihr Schiff zwischen den Astroiden durchsteuern (die in diesem Fall wie Sterne aussehen).
Es gibt keinen Bewertungsmechanismus in der ersten Version dieses Spieles, daher weiß der ZX81 es auch nicht, wenn Sie mit einem Stern zusammengeknallt sind. Wie Sie später sehen werden, verlangsamen Routinen, die Derartiges feststellen oder Punkte vergeben, das ganze Programm.
Mit der »Z«-Taste fahren Sie nach links, mit der »M«-Taste nach rechts. Nachdem Sie das Spiel einmal ge-

spielt haben und bevor Sie das Listing der zweiten Version dieses Spiels lesen, könnten Sie versuchen, eine »Entdeckungs-Routine« einzubauen. Das Programm, wie angegeben, läuft in 1 K.

```
 10 LET A=10
 20 LET B=15
 30 LET C=20
 40 PRINT AT C,RND*30;"*"
 50 PRINT AT A,B;"##"
 60 SCROLL
 70 IF INKEY$="Z" THEN LET B=B-
1
 80 IF INKEY$="M" THEN LET B=B+
1
 90 PRINT AT A,B;"▝▘"
100 GOTO 40
```

Das ganze Geheimnis dieses Programmes liegt im Gebrauch von SCROLL. Ihr Schiff (ausgedruckt in Zeile 9Ø und gelöscht in Zeile 5Ø) steht still in der Mitte des Bildschirmes, während die Sterne vorbeiflitzen.
Wenn Sie keine Entdeckungs-Routine einbauen wollen, könnten Sie vielleicht eine kleine Änderung vornehmen, die Ihr Schiff davor schützt, zu weit nach links oder rechts zu fahren (und möglicherweise einen »B«-Fehler zu erzeugen). Sie würden zwei Zeilen brauchen, die sich zwischen den gegenwärtigen Zeilen 8Ø und 9Ø befinden sollten. Wenn Sie sich nicht sicher sind, wie Sie das durchführen sollen, werfen Sie einen Blick auf die Zeilen 24Ø und 27Ø im Geigerprogramm.



Sternensturm

SWARM 2, das wir Sternensturm genannt haben, paßt noch immer in 1 K, obwohl es
a) eine Routine, die entdeckt, ob Sie etwas berührt haben
b) eine Routine, die Ihr Schiff davor schützt, über die Seiten zu fahren und
c) einen Bewertungsmechanismus
eingebaut hat.
Die Taste »M« ist Ihre einzige Steuerungsmöglichkeit. Tun Sie nichts, bewegt sich das Schiff nach links. Drücken Sie »M«, bewegt es sich nach rechts. Das Programm stoppt, wenn Sie einen Stern berührt haben, und statt des Sternes erscheint Ihr Punktestand. Jeder Punktestand über 3ØØ ist gut. Lassen Sie uns Ihren besten Punktestand wissen (kein Schummeln, bitte), und wir werden die Besten in der nächsten Ausgabe dieses Buches erwähnen.

```
 10 LET F=PI-PI
 20 LET A=VAL "10"
 30 LET B=VAL "15"
 40 LET C=A+A
 50 PRINT AT C,RND*30;"*"
 60 PRINT AT A,B;"#"
 70 LET F=F+PI/PI
 80 SCROLL
 90 IF B>2 THEN LET B=B-1
100 IF INKEY$="M" AND B<28 THEN
LET B=B+2
110 PRINT AT A,B;"V"
120 PRINT AT 11,B;
130 IF PEEK (PEEK 16398+256*PEE
K 16399)<>23 THEN GOTO 50
140 PRINT F
```

Ihr Punktestand wird mit der Variablen F abgestoppt. Zeile 9Ø bewahrt das Schiff davor, über die linke Seite hinauszuschießen und bewegt es Schritt für Schritt nach links, wenn dort Platz ist. Zeile 1ØØ wartet, bis Sie die »M«-Taste drücken und bewegt das Schiff nach rechts, wenn Sie nicht zu nahe am rechten Rand sind. Die kompliziert aussehende Zeile 13Ø sucht die PRINT-Position (die von der vorherigen Zeile gesetzt wurde) im Display-File.
Wird dort kein Charakter 23 gefunden (was einen Stern bedeutet), geht das Programm für den nächsten Durchgang zu Zeile 5Ø zurück. Sollten Sie einen Stern

erwischt haben, geht das Programm zu Zeile 140 weiter, um Ihren Punktestand auszudrucken.

Sternenschwarm

Die letzte Version ist mehr oder weniger das obige Programm mit inversen Graphiken. Sie werden bemerken, daß das Programm langsamer läuft als die zwei vorhergegangenen Versionen, da ständig ein kompletter schwarzer Streifen am unteren Rand des Bildschirmes gezeichnet werden muß.

```
  5 REM *FUER MEHR ALS 1K*
 10 FAST
 20 LET F=0
 30 FOR I=1 TO 704
 50 PRINT "#";
 70 NEXT I
 80 LET A=10
 90 LET B=15
100 LET C=20
110 SLOW
120 PRINT AT C,RND*30;"*"
130 PRINT AT A,B;"#"
140 LET F=F+1
150 SCROLL
160 PRINT AT C,0;"###############
####################"
170 IF B>4 THEN LET B=B-1
180 IF INKEY$="M" AND B<28 THEN
LET B=B+3*RND
190 PRINT AT A,B;"V"
200 PRINT AT 11,B;
210 IF PEEK (PEEK 16398+256*PEE
K 16399)<>151 THEN GOTO 120
220 PRINT "*";F;"*"
```

Rennfahrer

Mit der gleichen Idee kann ein sehr effektvolles Spiel geschrieben werden, bei dem Sie Ihren Rennwagen (ein inverses V) über eine zufällig gekrümmte Piste lenken müssen. Nachdem Sie das Programm laufen gelassen haben, finden Sie sicher viele andere effektvolle Möglichkeiten, Kapital aus der SCROLL-Funktion zu schlagen. Obwohl der Aufbau der RENNFAHRER- und SWARM-Programme sehr ähnlich ist, sieht dieses 1 K-Programm ganz anders aus.

Nachdem Sie NEWLINE gedrückt haben, warten Sie, bis die Rennstrecke gezeichnet ist und zu Ihrem Auto gelangt. Sie müssen dann Ihr Auto innerhalb der beiden schwarzen Begrenzungsstücke halten. Wenn Sie schon etwas Übung haben, verkleinern Sie die Breite der Straße (Zeile 6Ø).

```
 10 LET K=2
 20 LET A=10
 30 LET B=A
 40 LET C=A+A
 50 LET D=6
 60 PRINT AT C,D;"          "
 70 PRINT AT A,B;"▒"
 80 SCROLL
 90 IF INKEY$="M" THEN LET B=B+
1
100 IF INKEY$="Z" THEN LET B=B-
1
110 PRINT AT A,B;"V"
120 IF D>17 OR D<7 THEN LET K=-
K
130 LET D=D+RND*K
140 GOTO 60
```

Sie steuern Ihr Vehikel wie in SWARM mit den beiden Tasten »Z« und »M«, wobei Sie mit »Z« nach links und »M« nach rechts fahren. Wenn Sie die Begrenzung der Piste berühren, verschwindet sie an diesem Punkt.
Zeile 13Ø bewegt die Rennstrecke zufällig von Seite zu Seite, und Zeile 12Ø sorgt dafür, daß sich die Piste wieder in die Mitte zurückbewegt, wenn Sie zu nah an den Bildschirmrand kommt.
Zeile 7Ø löscht das Auto, das in Zeile 11Ø gedruckt wird. Es gibt keinen Grund, warum Ihr Auto ein inverses V sein sollte, also ändern Sie es, z.B. auf den Anfangsbuchstaben Ihres Vornames oder auf was Sie auf was immer Sie wollen. Wenn Sie das Auto statt mit dem Graphikzeichen von A mit dem Graphikzeichen von H löschen, werden Sie sehen, daß das Auto eine Spur hinterläßt, sodaß Sie sehen können, welche Strecke Sie zurückgelegt haben.
Das Auto startet auf 1Ø,1Ø (A, B) und mit INKEY $ (Zeilen 9Ø und 1ØØ) wird der Wert von B geändert (und Ihre Position auf dem Bildschirm nach links oder rechts verschoben). Wie in SWARM ändert sich Ihre Höhe (A) nicht, aber die SCROLL-Funktion (Zeile 8Ø) schickt Ihnen die Szenerie entgegen.

Rennfahrer 2

Rennfahrer 2 setzt Sie mit einer zusätzlichen Schwierigkeit wieder auf die Piste. Wenn Sie die Seiten der Rennstrecke berühren, stoppt das Programm. Diese Version paßt in 1 K.

```
  10 LET A=10
  20 LET B=A
  30 LET C=A+A
  40 LET D=9
  50 PRINT AT C,D;"▓▓▓▓▓▓▓▓"
  60 PRINT AT A,B;"▓"
  70 SCROLL
  80 IF INKEY$="Z" THEN LET B=B-
1
  90 IF INKEY$="M" THEN LET B=B+
1
 100 PRINT AT A,B;"H"
 110 IF D<17 THEN LET D=D+2*RND
 115 IF D>7 THEN LET D=D-2*RND
 120 PRINT AT 11,B;
 130 IF PEEK (PEEK 16398+256*PEE
K 16399)=128 THEN STOP
 140 GOTO 50
```

Rennfahrer 3 führt eine neue Schwierigkeit ein — andere Autos sind auf der Straße, die in die entgegengesetzte Richtung fahren.

```
   5 LET F=PI
  10 LET B=10
  20 LET A=B
  30 LET C=B+B
  40 LET D=B-B/B
  50 PRINT AT C,D;"#      #";AT A
,B;"#"
  60 IF RND>.6 THEN GOTO 100
  70 PRINT AT C,D+D/D+F*RND;"H"
 100 SCROLL
 110 LET B=B-(INKEY$="Z")+(INKEY
$="M")
 130 PRINT AT A,B;"Y"
 140 IF D<B+F THEN LET D=D+PI*RN
D
 150 IF D>F THEN LET D=D-PI*RND
 155 PRINT AT A+PI/PI,B;
 160 LET L=PEEK (PEEK 16398+256*
PEEK 16399)
 170 IF L=128 OR L=173 THEN STOP

 180 GOTO 50
```

Graffiti

Dieses Programm druckt die Buchstaben, die Sie ihm über die Tastatur eingeben, in vergrößerter Form aus. Sieben Buchstaben hoch und fünf Buchstaben breit. Sie berühren einfach die entsprechende Taste, und der Buchstabe erscheint auf dem Bildschirm. Drücken Sie NEWLINE, um einen Abstand zu bekommen und halten Sie die Taste des Buchstabens, der nach dem Abstand folgen soll, etwas länger gedrückt.

```
 10 LET X=1
 20 LET Y=1
 30 IF INKEY$="" THEN GOTO 30
 40 LET L=CODE INKEY$
 45 IF L=118 THEN LET L=0
 50 FOR Z=0 TO 3
 60 LET F=PEEK (2*Z+8*L+7680)
 70 LET G=PEEK (2*Z+8*L+7681)
 80 LET H=512
 90 FOR J=0 TO 3
100 LET H=H/4
110 LET A=INT (2*F/H-INT (F/H)*
2)
120 LET B=INT (F/H-INT (F/2/H)*
2)
130 LET C=INT (2*G/H-INT (G/H)*
2)
140 LET D=INT (G/H-INT (G/2/H)*
2)
150 PRINT AT X+Z,Y+J;CHR$ ABS (
2*A+B-135*C+4*D)
160 NEXT J
170 NEXT Z
180 LET Y=Y+4
190 IF Y<29 THEN GOTO 30
200 LET X=X+4
210 LET Y=0
220 GOTO 30
```

Sie können dieses Programm mit BREAK anhalten. Auf dem ZX81 läuft dieses Programm im FAST MODE, wenn Sie folgende Zeilen ändern bzw. hinzufügen.

```
 30 PAUSE 4E4
 35 POKE 16437,255
```

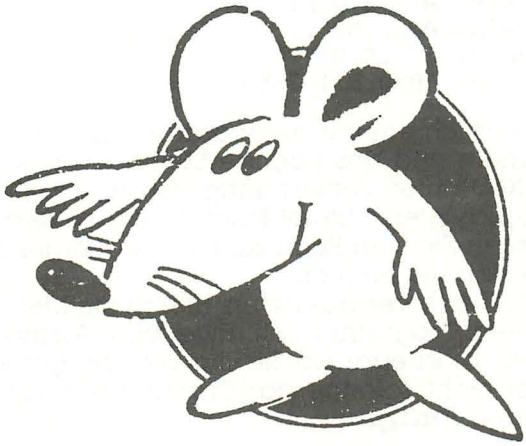
Mikromaus

Dieses Programm ist eine gute Demonstration des ZX81, und es ist faszinierend, es zu beobachten. Eine Maus (ein inverser Stern) startet von einer Zufallsposition in der Nähe der oberen linken Ecke des Bildschirmes. Sie möchte in die untere rechte Ecke kommen. Die Maus muß, um in die Ecke zu gelangen,einige zufällig verteilte Hindernisse umgehen. Wenn die Maus das Ziel erreicht hat, wird die Anzahl der von ihr benötigten Züge ausgedruckt. Die Hindernisse ändern ihre Position, die Maus blinkt einige Male, und dann beginnt das Spiel von neuem. Außer ganz zu Beginn kann sich die Maus nicht in eine Falle bringen, aus der sie sich nicht wieder befreien kann. Also haben Sie Geduld, egal, wie lange es zu dauern scheint.

```
  1 REM MICROMAUS 16K
  5 LET G=16398
  6 LET H=G+1
 10 PRINT AT 0,0;"################
####################"
 20 PRINT AT 21,0;"##############
####################"
 30 FOR A=1 TO 20
 35 PRINT AT 2+RND*18,1+RND*27;
"##"
 36 PRINT AT 2+RND*18,1+RND*27;
"##"
 37 PRINT AT 2+RND*18,1+RND*27;
"##"
 42 PRINT AT A,0;"#"
 43 PRINT AT 3+RND*15,2+RND*22;
"# #"
 44 PRINT AT 2+RND*18,1+RND*27;
"##"
 45 PRINT AT 2+RND*18,2+RND*24;
"#"
 46 PRINT AT 3+RND*15,2+RND*27;"▒"
 47 PRINT AT 2+RND*18,2+RND*24;
"#▒▒"
 50 PRINT AT A,31;"#"
 60 NEXT A
 61 FOR Z=1 TO 13
 62 PRINT AT 20,30;"#"
 63 PRINT AT 20,30;"*"
 64 PRINT AT 20,30;"*"
 65 PRINT AT 20,30;"#"
 66 PRINT AT 20,30;"*"
 67 PRINT AT 20,30;"#"
 68 NEXT Z
```

```
  70 LET A=INT (RND*8)+1
  75 LET Q=0
  80 LET B=INT (RND*15)+1
  85 PRINT AT 20,30;"#"
  90 LET E=A
  95 LET Q=Q+1
 100 LET F=B
 101 IF A=20 AND B=30 THEN GOTO
2000
 105 LET T=0
 106 IF RND>.2476 THEN GOTO 120
 110 LET Y=INT (RND*7)+1
 111 IF Y=1 THEN GOTO 120
 112 IF Y=6 THEN GOTO 169
 113 IF Y=3 THEN GOTO 200
 114 IF Y=4 THEN GOTO 250
 115 IF Y=5 THEN GOTO 290
 116 IF Y=2 THEN GOTO 154
 117 IF Y=7 THEN GOTO 330
 120 PRINT AT A+1,B;
 130 IF PEEK (PEEK G+256*PEEK H)
=0 THEN LET T=1
 140 IF T=1 THEN LET A=A+1
 150 IF T=1 THEN GOTO 1000
 152 IF RND>.2 THEN GOTO 169
 154 IF A=0 OR B=30 THEN GOTO 16
9
 155 PRINT AT A-1,B+1;
 156 IF PEEK (PEEK G+256*PEEK H)
=0 THEN LET T=1
 157 IF T=1 THEN LET B=B+1
 158 IF T=1 THEN LET A=A-1
 159 IF T=1 THEN GOTO 1000
 165 IF RND<.2 THEN GOTO 110
 169 PRINT AT A,B+1;
 170 IF PEEK (PEEK G+256*PEEK H)
=0 THEN LET T=1
 180 IF T=1 THEN LET B=B+1
 190 IF T=1 THEN GOTO 1000
 195 IF RND<.6 THEN GOTO 290
 200 PRINT AT A+1,B+1;
 210 IF PEEK (PEEK G+256*PEEK H)
=0 THEN LET T=1
 220 IF T=1 THEN LET A=A+1
 230 IF T=1 THEN LET B=B+1
 240 IF T=1 THEN GOTO 1000
 245 IF RND<.1 THEN GOTO 110
 250 PRINT AT A-1,B;
 260 IF PEEK (PEEK G+256*PEEK H)
=0 THEN LET T=1
 270 IF T=1 AND A>0 THEN LET A=A
-1
 280 IF T=1 THEN GOTO 1000
 290 PRINT AT A,B-1;
```

```
 300 IF PEEK (PEEK G+256*PEEK H)
=0 THEN LET T=1
 310 IF T=1 AND B>0 THEN LET B=B
-1
 320 IF T=1 THEN GOTO 1000
 330 IF B=0 OR A=0 THEN GOTO 110
 340 PRINT AT A-1,B-1;
 350 IF PEEK (PEEK G+256*PEEK H)
=0 THEN LET T=1
 360 IF T=1 THEN LET A=A-1
 370 IF T=1 THEN LET B=B-1
 380 IF T=1 THEN GOTO 1000
 390 GOTO 110
1000 PRINT AT E,F;"#"
1010 PRINT AT A,B;"*"
1020 GOTO 90
2000 PRINT AT 0,15;">";Q;"<"
2010 FOR N=1 TO 50
2020 NEXT N
2030 PRINT AT 0,15;"#######"
2040 GOTO 30
```



In POGO (das wunderbar in 1 K läuft und großartig erweitert werden kann, wenn Sie mehr Speicher haben) trudeln zwei Buchstaben Ihrer Wahl von rechts nach links über den Bildschirm. Zuerst geben Sie die beiden Zeilen ein. Inverse Zeichen sehen besonders effektvoll aus. Sie können zum Beispiel den Anfangsbuchstaben Ihres Namens als A $ und den eines Ihrer Freunde als B $ eingeben. Nachdem Sie das Programm einige Male laufen gelassen haben, und bevor Sie nachsehen, wie ich das gelöst habe, versuchen Sie, das Programm unter Verwendung von Zahlen und Buchstaben-Arrays umzuschreiben.

```
 10 INPUT A$
 20 INPUT B$
 30 LET A=31
 40 LET B=A
 50 LET A=A-RND
 60 PRINT AT 7,A;A$
 70 LET B=B-RND
 80 PRINT AT 14,B;B$
 90 PRINT AT 7,A;"."
100 PRINT AT 14,B;"."
110 IF A>1 AND B>1 THEN GOTO 50
```

Das ist es, was ich ein kurzes Programm nenne. Bevor Sie schummeln und das nächste Listing lesen, schreiben Sie POGO unter Verwendung von Arrays um, so daß das Display gleich bleibt. Nachdem Sie dies getan haben, kehren Sie zum Buch zurück und vergleichen Sie Ihre Version mit meiner.
Es ist nicht sehr schwer, das zu tun. Hier ist die Methode, wie ich mein Programm mit Hilfe von Arrays geändert habe. Es ist egal, ob Sie das gleiche gemacht haben oder nicht, solange das Display genauso aussieht wie ohne Arrays.

```
 10 DIM A$(2)
 20 DIM A(2)
 30 FOR B=1 TO 2
 40 INPUT A$(B)
 50 LET A(B)=31
 60 NEXT B
 70 FOR B=1 TO 2
 80 PRINT AT B*7,A(B);"."
 90 LET A(B)=A(B)-RND
100 PRINT AT B*7,A(B);A$(B)
110 NEXT B
120 IF A(1)>1 AND A(2)>1 THEN G
OTO 70
```

Dieses Programm ergibt ein Display, das beinahe ident. zu der ersten Version ist. Es hat einen großen Vorteil: Flexibilität. Wenn Sie das Programm erweitern wollen, so daß sich mehr als zwei Objekte über den Bildschirm bewegen, müssen Sie nur einige Änderungen machen. Das Ändern der ersten Version wäre viel schwieriger gewesen. Geben Sie das folgende Programm ein (oder ändern Sie nach Ihrer eigenen Methode), um vier kleine POGOs zu bekommen.

```
 10 DIM A$(4)
 20 DIM A(4)
 30 FOR B=1 TO 4
 40 INPUT A$(B)
 50 LET A(B)=31
 60 NEXT B
 70 FOR B=1 TO 4
 80 PRINT AT B*4,A(B);"."
 90 LET A(B)=A(B)-RND
100 PRINT AT B*4,A(B);A$(B)
110 NEXT B
120 IF A(1)>1 AND A(2)>1 AND A(
3)>1 AND A(4)>1 THEN GOTO 70
```

Zu unserer letzten Version ändern Sie die Vierer im obigen Programm auf Achter (mit Ausnahme der PRINT ATs, die auf PRINT AT 2 * B, A (B); geändert werden sollten) und ändern Sie Zeile 12Ø auf ein einfaches GOTO 7Ø.

Lassen Sie das laufen, und Sie werden sehen, daß eine Welle durch die POGOs läuft, wenn sie sich bewegen. Aber es gibt etwas Neues. Wenn Sie die linke Seite des Bildschirmes erreichen, stoppen sie nicht, wie das bei den vorhergegangenen Versionen der Fall war, sondern laufen entlang der punktierten Linien zurück. Das Programm wird angehalten, wenn eines der Zeichen — der Gewinner — tatsächlich vom Bildschirm springt.

Word-Processor

(unter der Verwendung des Druckers)

Dieses Programm erlaubt Ihnen, Ihren ZX-Drucker mit dem ZX81 zu koppeln und ihn mehr oder weniger als Word-Processor zu verwenden. Das Programm setzt den Text auf Block, wenn eine Zeile überfüllt ist, erlaubt Ihnen einen neuen Absatz zu beginnen, wann Sie wollen, Text zu ändern oder das Ende des Textes anzuzeigen. Wenn der Bildschirm voll ist, druckt ihn das Programm automatisch am Drucker aus und löscht ihn anschließend, um auf Ihre nächste Eingabe zu warten.
Wenn Sie im Moment noch keinen Printer haben, werden Sie dieses Programm trotzdem genießen (zu beobachten, wie es den Text justiert, ist sehr interessant) und einige Zeilen können hinzugefügt werden, um zu demonstrieren, daß der Drucker arbeitet (oder arbeiten würde, wenn Sie einen hätten).

```
  1 SLOW
 10 LET L=0
 20 LET C$=""
 30 PRINT AT L,0;C$
 40 INPUT B$
 50 IF B$="." THEN GOTO 280
 60 IF B$="/" THEN GOTO 330
 70 IF B$="0" THEN GOTO 360
 75 IF B$="S" THEN STOP
 80 IF LEN C$+LEN B$+1>32 THEN
GOTO 91
 85 LET C$=C$+B$+"#"
 90 GOTO 30
 91 LET C$=C$(1 TO LEN C$-1)
 95 FOR I=2+INT (RND*LEN C$/3)
TO LEN C$-INT (RND*LEN C$/3)
 97 IF C$(I)<>"#" THEN GOTO 150
100 LET A$=C$(1 TO I-1)+"#"+C$(
I TO )
110 IF LEN A$=32 THEN GOTO 160
150 NEXT I
160 LET C$=A$
180 PRINT AT L,0;A$
190 IF LEN C$<32 THEN GOTO 93
275 LET C$=B$+"#"
280 LET L=L+1
290 IF L=21 THEN GOTO 330
295 IF B$="." THEN GOTO 20
300 GOTO 30
330 COPY
```

Wenn Sie noch keinen Drucker haben, löschen Sie Zeile 33Ø und fügen Sie die nächsten beiden Zeilen ein. Haben Sie einen Drucker, lassen Sie Zeile 33Ø wie sie war (also COPY) und ignorieren Sie 331.
Demonstrationszeilen für Gebrauch ohne Drucker:

```
330 PRINT AT 0,0;"ICH KOPIERE"
331 INPUT U$
```

Zurück zum »echten« Programm:

```
340 CLS
350 GOTO 10
360 LET X=0
370 LET Y=0
380 GOSUB 1000
400 IF INKEY$="" THEN GOTO 400
410 LET A=CODE INKEY$
415 IF A=118 THEN LET A=0
420 IF A<112 OR A>127 THEN GOTO
500
430 IF A=112 THEN LET Y=Y-1
440 IF A=113 THEN LET Y=Y+1
450 IF A=114 THEN LET X=X-1
460 IF A=115 THEN LET X=X+1
470 IF A-121 THEN GOSUB 1000
480 IF A=121 THEN GOTO 40
490 GOTO 390
500 PRINT AT Y,X;CHR$ A
510 LET X=X+1
520 IF X<32 THEN GOTO 550
530 LET X=0
540 LET Y=Y+1
550 GOSUB 1000
560 GOTO 390
1000 PRINT AT Y,X;
1010 LET W=PEEK (PEEK 16398+256*
PEEK 16399)+128
1030 IF W>255 THEN LET W=W-256
1040 PRINT CHR$ W
1050 RETURN
```

Mit diesem Programm geben Sie Ihren Text ein und die Zwischenräume werden automatisch so bemessen, daß jede Zeile 32 Zeichen lang ist. Wenn der Bildschirm voll ist, wird er automatisch am Drucker kopiert, und der Bildschirm löscht sich, sodaß Sie einen neuen Text eingeben können.

Spezialeingaben sind:
»." bedeutet Endes des Absatzes
»/« bedeutet Ende des Textes
»Ø« bedeutet Ändern des Textes

NEWLINE bedeutet einen Zwischenraum, und nachdem der Text geändert wurde, erlaubt Ihnen SHIFT NEWLINE weitere Texteingabe.
Wenn Sie Text ändern müssen, geben Sie zuerst Ø ein und werden dann das erste Zeichen Ihres Textes invertiert sehen. Mit den Tasten 5, 6, 7 und 8 (indem Sie sie einfach niederhalten) verschieben Sie die inverse Blase in der Richtung der Pfeile auf den Tasten. Wenn Sie zu dem Buchstaben kommen, den Sie ändern wollen, geben Sie einfach den neuen Buchstaben ein.

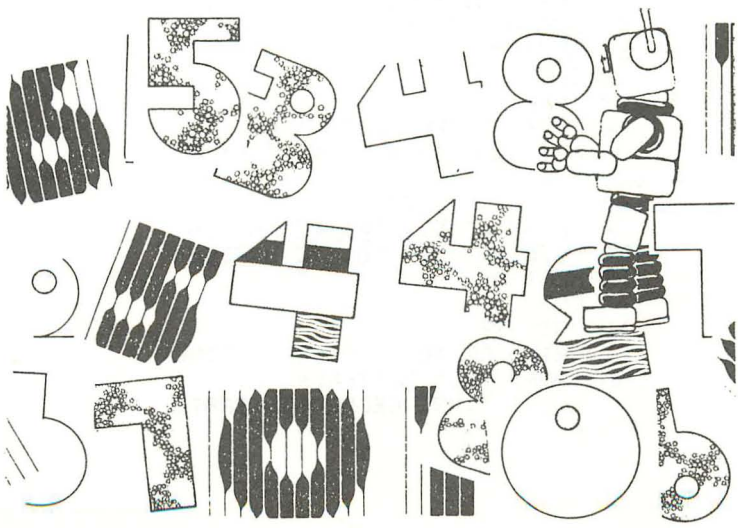
Echo Chamber

Dieses Spiel werden Sie nie gewinnen. Der ZX81 generiert eine ständig wachsende Zahl (bis zu 3Ø Ziffern lang), Ziffer für Ziffer, und Sie müssen sie wiederholen, nachdem Sie sie nur für eine kurze Zeit gesehen haben.
Beachten Sie, daß dieses Programm die Funktion STR$ verwendet (in der Subroutine 1ØØØ—1Ø2Ø), die eine numerische Variable in einen String ändert. Die Subroutine zeigt auch, wie die Verknüpfung von Strings funktioniert.

```
  5 FAST
 10 LET A$=""
 20 LET A=0
 30 PRINT AT 10,0;"ALSO,HIER IS
T TESTZAHL NR.";A+1
 40 PAUSE 100
 50 CLS
 60 GOSUB 1000
 70 PRINT AT 10,0;"DAS MUESSEN
SIE SICH MERKEN:"
 80 PRINT AT 11,31-A;A$
 90 PAUSE 30+5*A
100 CLS
110 PRINT AT 10,0;"WIE LAUTETE
DIE ZAHL?"
120 INPUT B$
130 IF B$=A$ THEN PRINT AT 10,0
;"SIE HABEN RECHT,BITTE WARTEN S
IE"
140 IF B$=A$ AND A=31 THEN GOTO
210
145 IF B$<>A$ THEN GOTO 180
150 PAUSE 100
160 CLS
170 GOTO 30
```

```
 180 PRINT AT 10,0;"DAS WAR LEID
ER FALSCH"
 190 PRINT "MEINE ZAHL WAR#";A$
 200 PRINT "IHRE ZAHL WAR#";B$
 210 PRINT "SIE ERREICHTEN#";3*A
;"#PUNKTE"
 220 PRINT AT 15,0;"WOLLEN SIE N
OCHEINMAL SPIELEN?"
 230 INPUT C$
 240 CLS
 250 IF CODE C$=47 THEN RUN
 260 FOR J=0 TO 21
 270 PRINT AT J,4;"AUF WIEDERSEH
EN"
 280 PAUSE 7
 290 NEXT J
 300 CLS
 310 STOP
1000 LET A$=A$+STR$ INT (RND*10)
1010 LET A=A+1
1020 RETURN
```

Beachten Sie den Gebrauch von CODE in Zeile 25Ø, um die Antwort des Spielers zu checken. Dies erlaubt dem Spieler »JA«, »JEIN« oder »JUCHU« einzugeben, und der ZX81 wird es als »JA« interpretieren.



Spieglein, Spieglein

SPIEGLEIN, SPIEGLEIN ist ähnlich wie ECHO CHAMBER mit dem Unterschied, daß dieses Programm Ihnen statt einer ständig wachsenden Zahl eine Serie von fünfziffrigen Zahlen gibt.
Je länger das Spiel fortschreitet, desto kürzer wird die Zeit, während der Sie die Zahl sehen und desto kürzer wird die Zeit, die Sie sich zwischen den einzelnen Zahlen erholen können.
Bei dem Programm BACK STABBER gibt es zwei Routinen (Zeile 2Ø—5Ø und Zeile 23Ø—26Ø), die zwischen den einzelnen Runden des Spiels blinkende Worte auf dem Bildschirm schreiben. SPIEGLEIN, SPIEGLEIN verwendet eine ähnliche Technik (Zeile 1ØØ—13Ø) für die Pause zwischen den aufeinanderfolgenden Gedächtnistests, und es ist diese Routine, die jedes Mal etwas kürzer läuft.
Sie bekommen eine Punktewertung (Zeile 17Ø), die siebenmal so groß ist, wie die Anzahl Ihrer richtigen Antworten.
Werfen Sie einen Blick auf Zeile 16Ø. Dort wird TAB gebraucht, um mit einer Programmzeile zwei Bildschirmzeilen mit Text in einer bestimmten Position zu füllen. Ihr ZX81-Handbuch erklärt dies auf Seite 115.

```
 10 LET A=0
 20 LET B=INT (RND*90000)+10000
 30 LET A=A+1
 40 PRINT AT 10,6;"TEST#";A,B
 50 FOR N=1 TO 40-2*A
 55 NEXT N
 60 CLS
 70 INPUT C
 80 IF B<>C THEN GOTO 160
 90 PRINT AT 10,2;C;"#IST RICHT
IG"
100 FOR D=1 TO 25-A
110 PRINT AT 5,4;"BITTE WARTEN"
120 PRINT AT 5,4;"BITTE WARTEN"
130 NEXT D
140 CLS
150 GOTO 20
160 PRINT AT 10,0;"FALSCH,MEINE
ZAHL WAR#";B;",","IHRE#";C
170 PRINT "IHRE GEDAECHTNISRATE
:";7*A
```

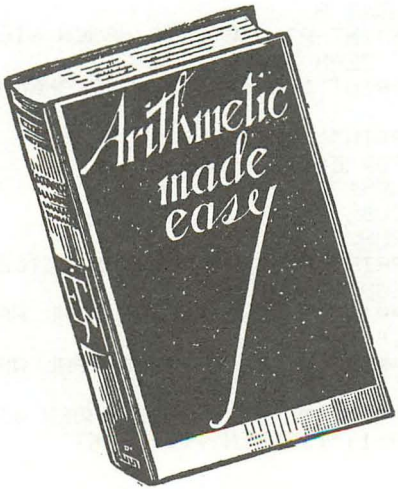
Hier ist eine längere Version dieses Spiels.

```
  10 LET B=40
  20 FOR A=1 TO 10
  40 PRINT AT 5,3;"VERSUCH NR.";
A
  50 PRINT AT 8,4;"DAS IST IHRE
ZAHL:"
  60 LET C=10000+INT (RND*90000)
  70 PRINT AT 10,0;"DIE ZAHL,DIE
 SIE SICH MERKEN"
  80 PRINT AT 12,0;"MUESSEN IST#
#";C
  90 PRINT AT 14,4;"KONZENTRIERE
N SIE SICH"
 100 FOR E=1 TO B-3*A
 110 NEXT E
 115 CLS
 120 PRINT AT 10,4;"NUN,WIE LAUT
ETE DIE ZAHL?"
 130 INPUT D
 140 CLS
 150 IF D<>C THEN GOTO 280
 160 PRINT AT 10,4;"RICHTIG,BRAV
O"
 170 FOR E=1 TO B-3*A
 180 PRINT AT 15,4;"BITTE WARTEN
"
 190 PRINT AT 15,4;"BITTE WARTEN
"
 200 NEXT E
 205 CLS
 210 NEXT A
 220 PRINT AT 8,0;"SIE HABEN SIC
H ALLE ZEHN ZAHLEN"
 230 PRINT "GEMERKT UND BEKOMMEN
 EIN"
 240 PRINT "WEITERES SPIEL"
 250 FOR E=1 TO 100
 255 NEXT E
 260 CLS
 270 RUN
 280 PRINT AT 6,4;"DAS WAR LEIDE
R FALSCH"
 290 PRINT AT 8,0;"MEINE ZAHL WA
R#";C;","
 300 PRINT AT 10,0;"IHRE ZAHL WA
R#";D
 310 PRINT AT 12,0;"SIE HABEN SI
CH#";A-1;"#ZAHLEN","GEMERKT"
```

Nicomachus

NICOMACHUS war, wie uns erzählt wurde, ein Mathematiker und Autor, der um das Jahr 1ØØ ein faszinierendes arithmetisches Buch schrieb. In diesem befindet sich das mathematische Puzzle, das im folgenden Programm verwendet wurde. Sie denken an eine Zahl zwischen 1 und 1ØØ, teilen sie nacheinander durch 3, 5 und 7 und geben jeweils den Rest jeder Division ein. Zeile 8Ø rekonstruiert dann Ihre ursprüngliche Zahl aus diesen Informationen.

```
 10 PRINT "DENKEN SIE SICH EINE
ZAHL"
 15 PRINT "ZWISCHEN 1 UND 100"
 20 PRINT "/3"
 30 INPUT A
 40 PRINT "/5"
 50 INPUT B
 60 PRINT "/7"
 70 INPUT C
 80 LET Z=70*A+21*B+15*C
 90 FOR I=1 TO 6
100 IF Z>105 THEN LET Z=Z-105
110 NEXT I
120 PRINT Z
```



****

Zufallszahlen

****

Die Zufallszahlen, die mit der RND-Funktion Ihres ZX81 generiert werden, sind nicht wirklich zufällig, obwohl sie für die meisten Zwecke ausreichen. Der Computer verwendet in Wirklichkeit eine lange Liste von Zahlen, so lang, daß es scheint, als wären sie zufällig. Die RAND-Funktion bestimmt, bei welcher Zahl der Liste der ZX81 beginnt. Diese Zahl steht in Beziehung zu der Anzahl der Einzelbilder, die Ihr Fernsehgerät seit dem Einschalten abgetastet hat. RAND n (wobei n eine Zahl ist) läßt den ZX81 bei einer Position beginnen, die in Beziehung zu n steht. Haben Sie beispielsweise RAND 7 am Beginn eines Programmes, das Zufallszahlen verwendet, eingegeben, so gibt der ZX81 jedes Mal die gleiche Zahlenfolge aus, wenn Sie das Programm ablaufen lassen. Es gibt einige Algorithmen, die Zufallszahlen generieren, und vielleicht wollen Sie die eine oder andere in einem Ihrer Programme verwenden. Diese Zufallszahlen müssen gesät werden (und das ist der Grund, warum beide Programme »Minuten nach der Stunde?« fragen, um eine Ausgangszahl zu bekommen). Es gibt keinen Grund, warum Sie nicht den Zufallszahlengenerator des ZX81 verwenden sollten, um Ausgangszahlen für diese Algorithmen zu bekommen. Zu diesem Zweck löschen Sie die Frage und ändern die INPUT-Zeile auf LET A=RND im ersten und LET X=RND im zweiten Programm.

```
  10 SLOW
  20 PRINT "MINUTEN NACH DER STU
NDE?"
  30 INPUT A
  40 LET B=A/2
  50 FOR J=1 TO 13
  60 LET B=ABS (439147+A+B)
  70 LET C=EXP 8+1
  80 LET D=23*B
  90 LET B=INT (D-INT (D/C)*C)
 100 NEXT J
 110 PRINT B
 120 GOTO 50
```

Dieses Programm generiert Zahlen mittels der Kongruenzmethode und verwendet die Gleichung:

$$x_{n+1} = 23x_n - \text{int}\left(\frac{23x_n}{10^8+1}\right) \times (10^8+1)$$

Das zweite Programm ist flexibler. Es kann leicht so eingestellt werden, daß es Zahlen in einem gewünschten Bereich generiert (auch das erste Programm kann so geändert werden, aber die Änderungen sind komplexer). Hier wurde das zweite Programm geschrieben, um einen einzelnen Würfel zu simulieren, d.h., es generiert Zahlen zwischen 1 und 6. Ändern Sie die eingeklammerte Ziffer in Zeile 70, um die höchstmögliche Zahl zu ändern.

```
 10 PRINT "MINUTEN NACH DER STU
NDE?"
 20 INPUT X
 30 LET D=6
 40 FOR J=1 TO D
 50 LET X=(X+PI)**5
 60 LET X=X-INT X
 70 LET D=INT (X*6)+1
 80 NEXT J
 90 PRINT D
100 GOTO 40
```

Das nächste Programm zeichnet eine graphische Darstellung der von der RND-Funktion generierten Zahl. Lassen Sie es einige Male wie aufgelistet laufen und fügen Sie dann 10 RAND 6 hinzu, um zu sehen, welchen Effekt das hat.

```
 20 FOR A=1 TO 20
 30 LET B=INT (RND*20)+1
 40 IF B<10 THEN PRINT "#";
 50 PRINT B;"#";
 60 FOR C=1 TO B
 70 PRINT ">";
 80 NEXT C
 90 PRINT
100 NEXT A
```

Wenn Sie den Ausdruck etwas verlangsamen wollen, fügen Sie hinzu:

```
75 LET K=RND*RND
```

Sie sollten sich an diese nützliche Technik erinnern, wenn Sie ein Programm etwas verlangsamen wollen. Eine zweite Möglichkeit, die allerdings zwei Zeilen braucht, ist eine Schleife hinzuzufügen, wie z.B. die folgende:

```
75 FOR N=1 TO 200
77 NEXT N
```

Die RND-Funktion wird in vielen Spielen verwendet, um den Wurf eines Würfels zu simulieren. Wenn sie richtig arbeitet, sollte sie (wie ein richtiger Würfel) jede Zahl zwischen 1 und 6 gleich oft produzieren. Realistischer gesagt, je öfter Sie den Würfel werfen, desto gleichmäßiger sollte die Verteilung der Zahlen zwischen 1 und 6 sein.

Das nächste Programm wirft einen Würfel so oft Sie wünschen und führt Buch über die gefallenen Zahlen. Wenn Sie das Programm laufen lassen, werden Sie um die Eingabe einer Zahl gebeten — die Anzahl der Würfe.

Wenn das Programm läuft, zeigt es Ihnen das Ergebnis des gegenwärtigen Wurfes (unter dem blinkenden schwarzen Quadrat) und um den wievielten Wurf es sich handelt. Darunter sehen Sie eine Tabelle, aus der Sie entnehmen können, wie oft jede Zahl gefallen ist.

Wenn Sie dies für eine größere Wurfanzahl ausprobieren wollen, lassen Sie das Programm in FAST laufen. ZX80-Besitzer mit neuem ROM und ZX81-Besitzer brauchen die Zeile 105 PAUSE 50. Ebenfalls sollte die Zeile 50 gelöscht werden. Das Programm ist ziemlich langsam durch die große Menge an Text, die bei jedem Durchlauf ausgedruckt werden muß. Wenn Sie eine sehr große Anzahl an Würfen austesten wollen, wie z.B. 5000, entfernen Sie die Zeile 30 und geben Sie sie als Zeile 120 neu ein. Löschen Sie die Zeilen 80 bis 100 und ändern Sie sie in Zeilen 130 bis 150. Löschen Sie die Zeilen 50 und 70 komplett und lassen Sie das Programm in FAST laufen.

```
 10 INPUT D
 20 DIM A(6)
 30 PRINT AT 10,4;"1###2###3###
4###5###6"
 40 FOR C=1 TO D
 50 PRINT AT 3,10;"*"
 60 LET A=INT (RND*6)+1
 70 PRINT AT 3,10;A,C
 75 LET A(A)=A(A)+1
 80 FOR B=1 TO 6
 90 PRINT AT 12,4*B;A(B)
100 NEXT B
110 NEXT C
```

Das nächste Programm, das auf dem eben aufgelisteten basiert, ist etwas interessanter. Wenn die Wahrscheinlichkeit, daß ein einzelner Würfel auf eine Zahl zwischen 1 und 6 fällt, in der Theorie 1 : 6 ist, ist die Wahrscheinlichkeit von bestimmten Gesamtsummen, die durch Zusammenzählen der Würfelaugen von zwei Würfeln entstehen, von der angenommenen Gesamtsumme abhängig. Das bedeutet, daß es z.B. nur eine Möglichkeit gibt, um die Zahl 2 zu erreichen (1+1), sie aber die Gesamtsumme 4 durch 2+2, 1+3 oder 3+1 erreichen können. Es gibt sechs Möglichkeiten, um eine Gesamtsumme von 7 zu erreichen.
Dieses Programm simuliert also den Wurf zweier Würfel. Nachdem Sie es einmal in Aktion gesehen haben und verstehen, was geschieht, fügen Sie eine ROUTINE hinzu, die für jede Möglichkeit den Prozentanteil an den Gesamtwürfen angibt. Wenn alles in Ordnung ist, sollten Ihre Prozentsätze an die theoretische Verteilung herankommen.

Gesamtsumme der Würfelaugen	Anzahl der Möglichkeiten, wie diese erreicht werden kann	Wahrscheinlichkeit	Prozentsatz
2	1	1/36	2,77%
3	2	2/36, 1/18	5,55%
4	3	3/36, 1/12	8,33%
5	4	4/36, 1/9	11,11%
6	5	5/36	13,88%
7	6	6/36, 1/6	16,66%
8	5	5/36	13,88%
9	4	4/36, 1/9	11,11%
10	3	3/36, 1/12	8,33%
11	2	2/36, 1/18	5,55%
12	1	1/36	2,77%

Im folgenden Programm sorgt Zeile 100 dafür, daß ein an sich schon sehr langsames Programm nicht noch langsamer dadurch wird, daß es immer wieder Zahlen ausdrucken muß, die sich nicht ändern.

```
 10 INPUT D
 20 DIM A(12)
 30 FOR C=1 TO D
 40 LET A1=INT (RND*6)+1
 50 LET A2=INT (RND*6)+1
 60 PRINT AT 3,2;"######";A1,A2
 70 PRINT AT 3,2;C
 80 LET A3=A1+A2
 90 LET A(A3)=A(A3)+1
100 FOR B=2 TO A3
110 PRINT AT B+3,0,"▒"
120 PRINT AT B+3,5;B,A(B)
130 NEXT B
140 NEXT C
```

Nun haben wir ein ordentliches Stück Zeit verwendet, um Würfel mit dem Zufallszahlen-Generator nachzuahmen. Ich glaube, daß wir das jetzt in einigen Programmen verwenden sollten.

Chuck-a-luck

Das erste Programm, auf das wir einen Blick werfen werden, heißt CHUCK-A-LUCK. In seiner ersten Inkarnation in England hatte es den sehr unwahrscheinlichen Namen Sweath-Cloth und als es in den frühen Jahren des 19. Jahrhunderts nach Amerika exportiert wurde, war es zuerst als Sweat bekannt. Durch die Jahre hindurch änderte es seinen Namen von Chucker-Luck, Chuck-Luck auf Chuck-A-Luck oder nur Chuck. Heutzutage wird es wegen der Ausrüstung, die man in der Nicht-ZX81-Version verwendet, meist The Bird Cage genannt.
Der Vogelkäfig ist ein geschlossener Drahtkäfig mit drei Würfeln. Die Spieler wetten auf die Wahrscheinlichkeit, daß eine bestimmte Zahl kommt. Wenn sie beispielsweise Ihr Geld auf 6 setzen und einer der drei Würfel fällt auf 6, bekommen Sie Ihr Geld zurück. Wenn alle drei Würfel 6 zeigen, bekommen Sie den dreifachen Einsatz. Ein wirklich einfaches Spiel, das aber bei Fans Leidenschaft erregt.
Das Programm tut die meiste Arbeit für Sie. Wenn es fragt »Höhe Ihres Einsatzes«, dann geben Sie den Betrag, den Sie verwetten wollen, ein. Sie wissen jederzeit, wieviel Geld Sie übrig haben (Sie starten mit 3Ø) und können Ihr gesamtes Geld verwetten. Sie verlieren automatisch diese Menge, wenn das Spiel beginnt (fällt ein Würfel auf Ihre Zahl, bekommen Sie Ihr Geld wieder zurück, ein Gewinn wird es erst bei zwei oder drei). Die nächste Frage des Computers ist »Zahl« und er erwartet die Eingabe einer Zahl zwischen 1 und 6. Das Spiel erklärt sich selbst, während es läuft.

```
10 LET M=30
```

M ist der Geldbetrag, mit dem Sie starten und später Ihr Restguthaben.

```
20 GOSUB 280
```

Die Subroutine 28Ø druckt Ihr Vermögen auf dem Bildschirm aus und beinhaltet eine Warteschleife (29Ø und 3ØØ), auf die in Zeile 22Ø separat zugegriffen wird.

```
30 PRINT AT 6,4;"IHR EINSATZ?#
";
40 INPUT A
```

```
 50 IF A>M THEN GOTO 40
 60 PRINT "$";A
 70 LET M=M-A
 80 PRINT AT 10,4;"ZAHL?#";
 90 INPUT B
100 IF B<>INT B OR B<1 OR B>6 T
HEN GOTO 80
110 PRINT B
```

Der nächste Teil rollt die Würfel und ändert Ihr Guthaben.

```
120 FOR C=1 TO 3
130 LET W=0
```

W gibt an, ob Sie verloren oder gewonnen haben und ist der Zahlmeister.

```
140 GOSUB 290
150 LET D=INT (RND*6+1)
160 PRINT AT 11+2*C,0;"WUERFEL#
";CHR$ (C+156);"#FIEL AUF#";CHR$
(D+156)
170 IF D=B THEN LET W=A
180 IF D=B THEN PRINT ,"GEWINN
$";W
190 LET M=M+W
```

Die Subroutine ändert die Gesamtsumme, wenn Sie gewonnen haben.

```
200 GOSUB 280
210 NEXT C
```

Es gibt eine kurze Pause, bevor das Spiel weitergeht.

```
220 GOSUB 290
230 CLS
```

Wenn Sie noch Geld haben, geht der Computer zu Zeile 2Ø zurück.

```
240 IF M>0 THEN GOTO 20
```

Wehe, wenn Sie bankrott sind...

```
250 PRINT AT 10,3;"SPIELENDE,SI
E SIND PLEITE"
260 PRINT AT 10,3;"SPIELENDE,SI
E SIND PLEITE"
270 GOTO 250
```

Die Subroutine ändert Ihr Konto und verursacht eine kleine Pause.

```
280 PRINT AT 2,6;"GUTHABEN $";M
;"##"
290 FOR N=1 TO 50
300 NEXT N
310 RETURN
```

Siebenter Himmel

In seiner Jugend war dieses Würfelspiel als »Unter oder über Sieben« bekannt. Es schien Spielern sehr attraktiv zu sein, denn es sah so aus, als wäre der Vorteil auf Seiten des Spielers. Wie Sie entdecken werden, wenn Sie mit Geld, das nur im ZX81 Variablen-Speicher existiert, spielen, zieht der Spieler den kürzeren. Wenn der Zufalls-Generator perfekt arbeitet und Sie dieses Spiel lange Zeit hindurch spielen, sollten Ihre Verluste die Gewinne mit 16,2% überwiegen. Nachdem Sie nun gewarnt worden sind, gehen Sie in den Siebenten Himmel.
Wieder einmal haben wir M verwendet, um Ihr Geld zu speichern.

```
20 LET M=30
```

Und wieder einmal haben wir eine Subroutine, um Ihr wachsendes (?) Guthaben anzuzeigen.

```
30 GOSUB 320
```

Der nächste Teil ersucht Sie, Ihre Wette auf eine der drei Möglichkeiten zu plazieren: Der Würfel landet mit einer Gesamtsumme unter 7 (A); gleich 7(B); oder über 7 (C). Sie treffen Ihre Wahl, indem Sie entweder A, B oder C eingeben.

```
  40 PRINT AT 4,3;"GEBEN SIE IHR
E WETTE AB"
  50 PRINT AT 10,0;"(A) UNTER 7#
#(B) 7##(C) UEBER 7"
```

Die nächste Zeile erklärt die möglichen Gewinne — Geld zurück (A oder C) oder 4 : 1 (B), ausgedruckt als fünffacher Einsatz, sodaß es aussieht, als ob Sie mehr bekämen ... heimtückisch!

```
  60 PRINT AT 11,5;"GLEICH######
5 ZU 1#####GLEICH"
  70 INPUT A$
```

Wenn Sie das Spiel stoppen wollen, geben Sie S ein.

```
  80 IF A$="S" THEN STOP
```

Nun geben Sie die Höhe Ihrer Wette ein.

```
  90 PRINT AT 16,5;"BETRAG?"
 100 INPUT A
 110 IF A<1 OR A>6 THEN GOTO 100
 120 PRINT AT 16,5;"GUT,$";A;"##
"
 130 LET B=INT (RND*6)+1
 140 PRINT AT 10,3;B;AT 10,10;
 150 LET C=INT (RND*6)+1
 160 PRINT C;TAB 6;
 170 LET D=C+B
 180 PRINT D
```

Diese Sektion des Programmes arbeitet aus, wie gut oder schlecht Sie gewettet haben.

```
 190 IF D=7 AND A$="B" THEN LET
W=4*A
 200 IF D<7 AND A$="A" THEN LET
W=A
 210 IF D>7 AND A$="C" THEN LET
W=A
 220 IF ((A$="A" OR A$="C") AND
D=7) OR (A$="B" AND D<>7) THEN L
ET W=-A
 230 LET M=M+W
 240 IF W>0 THEN PRINT AT 16,4;"
SIE GEWINNEN $";W
 250 IF W<0 THEN PRINT AT 16,4;"
SIE VERLIEREN $";-W
 260 GOTO 320
 270 FOR N=1 TO 60
 280 NEXT N
 290 IF M<1 THEN STOP
 300 CLS
 310 GOTO 30
 320 PRINT AT 2,4;"GUTHABEN $";M
;"##"
 330 RETURN
```

Dice-Jack

DICE-JACK ist die 1 K-Würfelversion von Black Jack. Sie werfen abwechselnd einen Würfel, addieren die Augenzahlen und versuchen, so nah wie möglich an (oder gleich auf) 21 zu kommen, aber auf keinen Fall darüberhinaus. Wenn weder Sie noch der Computer gewinnen, beginnt ein neues Spiel. Wenn Sie beide »sprengen« (d.h., Sie überschreiten 21) oder unentschieden spielen, dann stoppt das Programm.
Die PRINT-Anweisungen sind sehr kurz, um das Programm in 1 K zu bekommen, also werde ich Ihnen erklären, was auf der Mattscheibe geschieht.
Wenn Sie das Programm laufen lassen, sehen Sie in der unteren linken Ecke die Aufforderung, eine Zahl einzugeben. Wenn Sie 1 eingeben, wird der Würfel geworfen und die Augenzahl zu Ihrem Stand addiert. Wenn Sie mit Ihrem Stand schon glücklich sind und Sie keinen neuen Wurf mehr machen wollen, geben Sie 2 ein und der Computer würfelt. Ihr Gesamtergebnis ist die oberste Zeile, das des Computers die darunterstehende.
Die Zeilen 26Ø bis 28Ø sollen das ganze Spiel nur etwas verlangsamen. H ist die Gesamtsumme des Menschen und C (klarerweise) ist die des Computers. Zeile 9Ø ist die wichtigste des Spieles. Diese Zeile bestimmt die gesamte Strategie des Computers und ist deshalb die Zeile, die dem ZX81 die Entscheidung ermöglicht, einen neuen Wurf zu machen oder nicht.
Die Zeilen 17Ø bis 19Ø arbeiten aus, wer das Spiel gewonnen hat.
Wenn Sie mehr als 1 K haben, könnten Sie das Spiel etwas aufmotzen, indem Sie eine Spielanleitung und passende Aufforderungen an die Spieler einbauen.

```
 10 LET H=0
 20 LET C=H
 30 INPUT A
 40 IF A=2 THEN GOTO 90
 50 LET H=H+INT (RND*6)+1
 60 GOSUB 260
 70 PRINT AT 4,0;H
 80 GOTO 30
 90 IF C>H AND C<22 OR C>21 OR
H>21 OR H=21 AND C=21 THEN GOTO
140
 100 LET C=C+INT (RND*6)+1
 110 GOSUB 260
 120 PRINT AT 8,0;C
 130 GOTO 90
```

```
 140 PRINT AT 11,0;
 150 GOSUB 260
 160 GOSUB 260
 170 IF H=C OR H>21 AND C>21 THE
N GOTO 240
 180 IF (C>H OR H>21) AND C<22 T
HEN PRINT "ICH HABE";
 190 IF (H>C OR C>21) AND H<22 T
HEN PRINT "SIE HABEN";
 200 PRINT "#GEWONNEN"
 210 PAUSE 200
 220 CLS
 230 RUN
 240 PRINT "UNENTSCHIEDEN"
 250 STOP
 260 FOR E=1 TO 40
 270 NEXT E
 280 RETURN
```

Beachten Sie, daß die PAUSE zweimal hintereinander aufgerufen wird (Zeile 15Ø und 16Ø). Dies soll den Eindruck erwecken, daß der ZX81 ausarbeitet, wer gewonnen hat. Beachtenswert ist auch Zeile 14Ø, die Platz spart, indem sie das gleiche PRINT AT zweimal verwendet. Das wird in den Zeilen 18Ø und 19Ø offensichtlich.
Wenn Sie mehr als 1 K haben, könnten Sie überlegen, außer der Spielanleitung und der Aufforderung an die Spieler auch das Ergebnis jedes Wurfes und die wachsende Gesamtsumme auszudrucken.

Barbudi

Das ist ein einfaches Kartenspiel, das auf dem Würfelspiel BARBUDI basiert. Der Spieler zieht zwei Karten aus einem Stoß, der nur aus Neunern, Zehnern, Buben, Damen, Königen oder Assen besteht. Bestimmte Kombinationen (die Sie kennenlernen werden, während Sie sich mit dem Programm beschäftigen) sind Gewinner und andere Kombinationen sind Verlierer.
Der Geldbetrag, den Sie pro Runde gewinnen oder verlieren können, steigert sich laufend während des Spiels. Das Geld, das Sie gewinnen, kommt aus dem Speicher des ZX81 (Variable C, bestimmt in Zeile 3Ø) und das Geld, das Sie verlieren (Ihre Geldvariable ist H, Zeile 4Ø), geht in den Speicher des ZX81. Das Spiel läuft, bis einer der beiden Beträge kleiner ist als 1.
Beachten Sie, daß der String A$ (aufgestellt in Zeile 2Ø) nicht dimensioniert ist, aber daß einzelne Elemen-

te aus dem STRING herausgenommen und verwendet werden können.

```
10 RAND
20 LET A$="9ZBDKA"
30 LET C=30
40 LET H=30
50 LET A=0
60 LET A=A+1
70 PRINT
80 PRINT "DRUECKEN SIE N/L FUE
R BLATT#";A
90 LET G=0
```

Wie Sie aus der Subroutine, die in Zeile 25Ø beginnt, ersehen können, zeigt G Gewinn oder Verlust an.

```
100 INPUT B$
110 IF B$>"" THEN STOP
```

Zeile 11Ø erlaubt es Ihnen, das Programm jederzeit anzuhalten, indem Sie vor NEWLINE ein Zeichen eingeben.

```
120 CLS
130 GOSUB 250
150 IF G=1 THEN PRINT "SIE HABE
N GEWONNEN"
160 IF G=2 THEN PRINT "ICH HABE
GEWONNEN"
```

Der nächste Teil händigt Ihnen das Geld aus (oder nimmt es Ihnen weg).

```
170 IF G=1 THEN LET H=H+A/2
180 IF G=1 THEN LET C=C-A/2
190 IF G=2 THEN LET H=H-A/2
200 IF G=2 THEN LET C=C+A/2
210 IF C<1 OR H<1 THEN STOP
220 PRINT
230 PRINT "SIE $";H,"ICH $";C
240 GOTO 60
```

Diese Subroutine verteilt die Karten mittels Zufallszahlen und zieht dann das der jeweiligen Nummer entsprechende Element aus dem String in Zeile 2Ø heraus.

```
250 LET D=INT (RND*6)+1
260 LET E=INT (RND*6)+1
270 PRINT AT 3,10;"▀▀▀▀▀"
280 PRINT AT 4,10;"▌ ";A$(D);"#"
;A$(E);" ▐"
```

```
290 PRINT AT 5,10;"▌";A$(D);"#"
;A$(E);"▐"
300 PRINT AT 6,10;"▙▄▄▄▄▟"
310 PRINT AT 6,14;
320 LET F=D+E
330 IF F=5 OR F>8 THEN LET G=1
340 IF F<4 OR F=7 THEN LET G=2
350 RETURN
```

Die Zeilen 33Ø und 34Ø bestimmen, welche Kombinationen gewinnen und welche verlieren. Das Spiel ist leicht zugunsten des Spielers geschrieben. Nachdem Sie einmal die gewinnenden und verlierenden Kartenpaare kennen, könnten Sie diese beiden Zeilen etwas ändern, sodaß der Vorteil auf Seiten des armen kleinen ZX81 ist, oder um das Spiel 1ØØ%-ig fair zu machen.

Blackjack

Dieses BLACKJACK-Programm verwendet die gleiche Idee, Strings zu verwenden wie BARBUDI. Verwendet wird ein undefinierter String (A$, Zeile 6Ø), der die Karten hält, und ein definierter String (C$, in den Zeilen 1Ø bis 5Ø), der die passenden Karten bestimmt. C$ muß definiert werden, weil seine Elemente mehr als ein Zeichen lang sind.
Wie das Programm hier aufgelistet ist, agiert es als Dealer. Es gibt sich zuerst selbst eine Karte und gibt dann Ihnen Ihre erste Karte. Von da an können Sie entscheiden, weitere Karten zu nehmen oder nicht, also Ihren Gesamtwert beizubehalten. Ziel des Spieles ist es, eine Gesamtsumme von 21 (die BLACKJACK genannt wird) zu bekommen, oder so nah wie möglich an 21 heranzukommen, 21 aber nicht zu überschreiten. Sollte Ihre Gesamtsumme höher als 21 sein, sind Sie »gesprengt«. Die Zahlenkarten zählen entsprechend ihrer Zahlen (z.B. zählt der Herzzweier 2), Bilderkarten (Bube, Dame, König) zählen als Zehn und das As zählt als Elf, außer es würde Sie sprengen, wenn es zu Ihrer Gesamtsumme hinzugezählt würde. In diesem Fall zählt es als Eins.
Das Programm arbeitet automatisch die Gesamtsumme für beide Spieler aus, und ändert ein As von 11 auf 1, wenn es notwendig ist.
Wenn Sie mit dem Programm vertraut sind, werden Sie es wahrscheinlich so umschreiben wollen, daß der

ZX81 sowohl Ihr als auch sein eigenes Geld verwaltet und Wetten annimmt.

```
 10 DIM C$(4,5)
 20 LET C$(1)="HERZ"
 30 LET C$(2)="KARO"
 40 LET C$(3)="PIK"
 50 LET C$(4)="TREFF"
 60 LET A$="23456789ZBDKA"
 70 GOTO 220
 80 LET A=INT (RND*13)+1
 90 LET E=A
100 FOR N=1 TO 35
110 NEXT N
120 LET Y=INT (RND*4)+1
130 LET R=Y
140 PRINT "MEINE KARTE IST#";A$
(A);"#";C$(Y)
150 LET A=A+1
160 IF A>9 AND A<14 THEN LET A=
10
170 IF A=14 THEN LET A=11
180 IF A=11 AND Z+A>21 THEN LET
 A=1
190 LET Z=Z+A
200 PRINT "MEIN GESAMTSTAND IST
#";Z
210 RETURN
220 LET H=0
230 LET Z=0
240 GOSUB 80
250 PRINT
260 PRINT "DRUECKEN SIE N/L FUE
R KARTE 1"
270 INPUT Z$
280 LET B=INT (RND*13)+1
290 LET Y=INT (RND*4)+1
300 PRINT AT 4,6;A$(B);"#";C$(Y
)
310 LET B=B+1
320 IF B>9 AND B<14 THEN LET B=
10
330 IF B=14 THEN LET B=11
340 IF B=11 AND H+B>21 THEN LET
 B=1
350 LET H=H+B
360 PRINT AT 5,0;"IHR GESAMTSTA
ND IST#";H
370 PRINT AT 6,1;"GEBEN SIE ""A
"" FUER NOCH EINE"
380 PRINT "KARTE EIN ODER N/L U
M ZU STEHEN"
390 INPUT B$
400 IF B$="A" THEN GOTO 280
```

```
 410 CLS
 420 IF H=21 THEN PRINT ,"BLACKJ
ACK"
 430 PRINT
 440 PRINT "GUT,SIE STEHEN AUF#"
;H
 450 IF H>21 THEN PRINT ,"GESPRE
NGT"
 460 IF H>21 THEN GOTO 660
 470 PRINT
 480 PRINT "MEINE ERSTE KARTE WA
R#";A$(E);"#";C$(R)
 490 PRINT "UND MEIN GESAMTSTAND
 WAR#";Z
 500 PRINT
 510 PRINT AT 6,0;"ICH GEBE JETZ
T"
 520 GOSUB 80
 530 IF Z>21 THEN PRINT ,"GESPRE
NGT"
 540 IF Z=21 THEN PRINT ,"BLACKJ
ACK"
 550 IF Z<18 AND H<22 THEN GOTO
520
 560 FOR N=1 TO 35
 570 NEXT N
 580 CLS
 590 PRINT "SIE STANDEN AUF#";H
 600 PRINT
 610 IF Z<22 THEN PRINT "ICH STE
HE AUF#";Z
 620 PRINT
 630 IF H=21 THEN PRINT "SIE HAB
EN EINEN BLACKJACK"
 640 IF H=21 AND Z=21 THEN PRINT
 "UND ICH AUCH"
 650 IF Z=21 AND H<>21 THEN PRIN
T "ICH HABE EINE BLACKJACK"
 660 IF (Z>H OR H>21) AND Z<22 T
HEN PRINT "ICH GEWINNE"
 670 IF (H>Z OR Z>21) AND H<22 T
HEN PRINT "SIE GEWINNEN"
 680 IF H<22 AND H=Z THEN PRINT
"UNENTSCHIEDEN"
 690 IF H>21 AND Z>21 THEN PRINT
 "WIR SIND BEIDE GESPRENGT"
 700 FOR N=1 TO 50
 710 NEXT N
 720 CLS
 730 RUN
```

****

Informationen in Strings speichern

****

Blastermind

In den Programmen BARBUDI und BLACKJACK haben wir bestimmte Informationen in Form von Zeichen mittels Strings gespeichert.
Viele BASICs haben eine READ/DATA-Möglichkeit, aber beim ZX81 wurde darauf zugunsten der Druckersteuerungsbefehle verzichtet. Also müssen wir einen Ersatz finden. REMbefehle sind für diesen Zweck nützlich, aber Strings können genauso gut benutzt werden und stellen sich als flexibler heraus.
In der nächsten kleinen Routine beinhalten die Elemente der Strings M$ und N$ Informationen in Zusammenhang mit ihrer Position. Ich erkläre Ihnen nicht, was diese Routine auf dem Bildschirm ausdruckt, und ich fordere Sie heraus, es zu erraten. Geben Sie das Programm jetzt ein und lassen Sie es laufen.

```
  10 LET M$="▒▒▒▒▒▒??????"+CHR$
11+"£$:"
  20 LET N$="▘▀▀▄█▗▀▀▀▄█▗▛▖▀▘"
  30 FOR A=1 TO 16
  40 PRINT AT CODE M$(A),CODE N$
(A);"#"
  50 NEXT A
```

Beachten Sie, daß in keinem der beiden Strings Zwischenräume vorkommen. Nachdem Sie das Programm laufen gelassen haben, ändern Sie Zeile 4∅ auf:

```
  40 PLOT CODE M$(A),CODE N$(A)
```

Jetzt bekommen Sie etwas anderes, oder? Dieses Programm arbeitet so, weil die Schleife von Zeile 3∅ jeden String Element für Element durchkämmt und es in eine Zahl ändert (den CODE des Zeichens); diese

Information wird dann für PRINT AT oder PLOT gebraucht.
Das eigenartige Ding in N$ (+CHR$ 11+) steht deswegen, weil Zeichen 11 — das Anführungszeichen — nicht direkt in den String eingegeben werden kann.
In unserem nächsten Programm — Mühle — werden Strings verwendet, um die Beziehung zwischen den einzelnen Positionen zu speichern, sodaß der ZX81 entscheiden kann, wohin er ziehen soll (A$); ebenfalls in dem STRING gespeichert ist die Information, um das Spielfeld auszudrucken (B$ und C$).

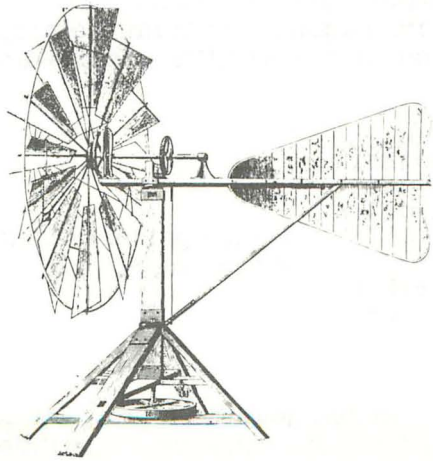
Mühle

Der ZX81 spielt bei Mühle ein starkes Angriffsspiel und wird Sie wahrscheinlich bei zwei bis drei von fünf Spielen schlagen, bis Sie seine Strategie kennen (das Spiel unterscheidet sich etwas von dem Mühlespiel, das wir gewohnt sind). Sie haben zwölf Steine, die Sie abwechselnd mit dem ZX81 auf dem Spielbrett plazieren. Das Ziel ist es, drei Steine in eine Reihe zu bekommen, horizontal, vertikal oder diagonal. Beachten Sie, daß Sie bei diesem Programm wählen können, ob Sie oder der ZX81 beginnen sollen.
Wenn Sie das Programm kürzen wollen, können Sie diese Möglichkeit leicht löschen. In den Strings gibt es keine Zwischenräume.

```
  10 LET A$="
$ 11+CHR$ 11+"
  20 LET B$="
£???£"
  30 LET C$="
::£"
  40 DIM L$(16,3)
  50 FOR Z=1 TO 16
  60 LET L$(Z)=A$(3*Z-2 TO 3*Z)
  70 NEXT Z
  80 PRINT "WOLLEN SIE BEGINNEN?
"
  90 INPUT A$
 100 CLS
 110 FOR Z=1 TO 24
 115 PRINT AT CODE B$(Z),CODE C$
(Z);Z
 120 NEXT Z
 130 DIM M$(24)
 140 IF CODE A$=47 THEN GOTO 190
```

```
 150 PRINT TAB 20;
 160 LET M=INT (RND*24+1)
 170 LET M$(M)="■ "
 180 LET K=189
 185 GOSUB 1000
 190 INPUT M
 200 LET M$(M)="■ "
 210 LET K=180
 215 GOSUB 1000
 220 DIM S$(16)
 230 FOR Z=1 TO 16
 240 LET S$(Z)=CHR$ (CODE M$(COD
E L$(Z,1))+CODE M$(CODE L$(Z,2))
+CODE M$(CODE L$(Z,3)))
 250 IF S$(Z)<>"■" THEN GOTO 310
 280 PRINT AT 11,11;"SIE";TAB 11
;"###";TAB 11;"GEWINNEN"
 300 STOP
 310 IF S$(Z)="▒" THEN GOTO 730
 320 NEXT Z
 330 FOR Z=1 TO 16
 340 IF S$(Z)=" ■" THEN GOTO 730
 350 NEXT Z
 360 FOR Z=1 TO 15
 370 FOR J=Z+1 TO 16
 380 IF S$(Z)="▄ " AND S$(J)="▄ "
THEN GOSUB 830
 390 NEXT J
 400 NEXT Z
 410 FOR Z=1 TO 15
 420 FOR J=Z+1 TO 16
 430 IF S$(Z)="■ " AND S$(J)="■ "
THEN GOSUB 830
 440 NEXT J
 450 NEXT Z
 460 FOR Z=1 TO 16
 470 FOR J=1 TO 16
 480 IF S$(Z)="▄" AND S$(J)="#"
THEN GOTO 620
 490 NEXT J
 500 NEXT Z
 510 LET A=0
 520 LET A=A+1
 530 IF A=24 THEN STOP
 540 LET M=M+1
 550 IF A/8=INT (A/8) THEN LET M
=M+8
 560 IF M>25 THEN LET M=M-24
 570 IF M=9 THEN LET M=1
 580 IF M=17 THEN LET M=9
 590 IF M=25 THEN LET M=17
 600 IF M$(M)<>"#" THEN GOSUB 52
0
 610 GOTO 170
```

```
 620 LET A$=""
 630 FOR K=1 TO 3
 640 FOR L=1 TO 3
 650 IF L$(Z,K)=L$(J,L) THEN LET
 A$=L$(Z,K)
 660 NEXT L
 670 NEXT K
 680 IF A$="" THEN GOTO 490
 690 FOR K=1 TO 3
 700 IF A$=L$(J,K) THEN NEXT K
 710 LET M=CODE L$(J,K)
 720 GOTO 170
 730 FOR K=1 TO 3
 740 IF M$(CODE L$(Z,K))<>"#" TH
EN NEXT K
 750 LET M=CODE L$(Z,K)
 760 IF S$(Z)<>"▒" THEN GOTO 170
 770 LET K=189
 780 GOSUB 1000
 800 PRINT AT 11,11;"ICH";TAB 11
 "###";TAB 11;"GEWINNE"
 820 STOP
 830 FOR K=1 TO 3
 840 FOR L=1 TO 3
 850 IF L$(Z,K)=L$(J,L) AND M$(C
ODE L$(Z,K))="#" THEN GOTO 750
 860 NEXT L
 870 NEXT K
 880 RETURN
1000 PRINT AT CODE B$(M),CODE C$
(M);CHR$ K;"#"
1010 RETURN
```



****

Peek und Poke

****

Die Befehle PEEK und POKE scheinen komplizierter als sie sind. Die folgende Erklärung ist etwas vereinfacht und unkomplett, aber sollte Ihnen genug sagen, um diese beiden Befehle anwenden und etwas darüber nachdenken zu können.
Stellen Sie sich vor, daß der denkende Teil des ZX81 nicht größer als eine Zündholzschachtel ist. Jede dieser Zündholzschachteln hat eine Nummer, sodaß wir jederzeit in die Zündholzschachtel hineinsehen können (PEEK), um zu sehen, was sich darin befindet, oder wir können etwas hineinlegen (POKE).
In der komplizierten Welt der Computer wird die Zahl, die auf der Seite der Zündholzschachtel geschrieben steht, Adresse genannt, wenn wir also sehen wollen, was sich in einer Zündholzschachtel befindet, dann PEEKEN wir in die Adresse. Wenn Sie etwas in die Schachtel hineinlegen wollen, dann POKEN Sie etwas in die Adresse.
Weil der ZX81 mit Zahlen arbeitet, ist das, was sich in der Zündholzschachtel befindet (d.h. in oder auf jeder Adresse), eine Zahl. Wir können Zahlen hinein POKEN und wenn wir eine Adresse PEEKEN, dann bekommen wir eine Zahl.
Jeder Teil des Programmes hat eine Adresse und wir können Teile des Programmes ändern, indem wir Dinge in die entsprechende Adresse POKEN.
Versuchen Sie das folgende Programm einzugeben, lassen Sie es laufen und betrachten Sie dann das REM-Statement in Zeile 1Ø.

```
10 REM AAA
20 LET A=16514
30 LET B=16515
40 LET C=16516
50 INPUT D
60 POKE A,D
70 POKE B,D-1
80 POKE C,D+1
90 LIST
```

D muß größer als 2 und kleiner als 6Ø sein.

Nachdem Sie das laufen gelassen haben, werden Sie feststellen, daß sich das AAA in Zeile 1Ø geändert hat,

weil die Zeilen 6∅ bis 8∅ Neuinformationen in diese Adressen GEPOKT haben (16514, 16515, 16516).
Adresse 16514 ist die erste Adresse nach dem Wort REM in der ersten Zeile eines Programmes. Es hat einen großen Vorteil, Zahlen in REM-Statement zu speichern — sie sind nicht verloren, wenn RUN oder CLEAR benutzt wird, während normal belegte Variablen (so wie LET X=3) in diesem Fall verloren sind.
Im nächsten Programm, bei dem Sie eine Zahl, die sich der ZX81 ausgedacht hat, erraten müssen, ist die Geheimnummer des ZX81 in der Adresse 16514 gespeichert. Ihr Tip wird in 16515 und die Anzahl der Versuche, die Sie gebraucht haben, in 16516 behalten.
Geben Sie es ein, lassen Sie es einige Male laufen und versuchen Sie zu verstehen, wie das Programm arbeitet.

```
  10 REM AAA
  20 POKE 16514,INT (RND*59)+2
  30 POKE 16516,0
  40 LET A=16514
  50 LET B=16515
  60 LET C=16516
  70 POKE C,PEEK C+1
  80 PRINT "GEBEN SIE VERUCH NR.
";PEEK C;"#EIN"
  90 INPUT D
 100 POKE B,D
 110 IF PEEK B<PEEK A THEN GOTO
170
 120 IF PEEK B<PEEK A·THEN PRINT
 PEEK B;"#IST ZU KLEIN"
 130 IF PEEK B>PEEK A THEN PRINT
 PEEK B;"#IST ZU GROSS"
 140 IF PEEK C<11 THEN RUN 40
 150 PRINT "DIE ZEIT IST VORBEI"
 160 GOTO 180
 170 PRINT "SIE SCHAFFTEN ES IN#
";PEEK C;"#VERSUCHEN"
 180 PRINT "ICH DACHTE AN#";PEEK
 A
```

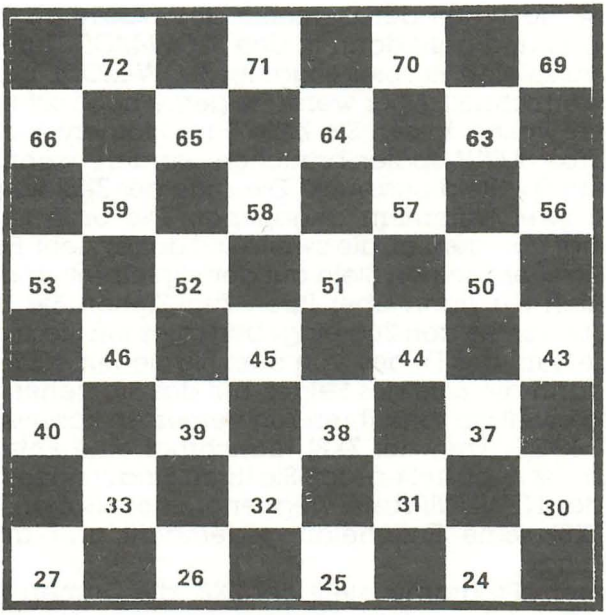
Dame

Das nächste Kapitel erlaubt es Ihnen, ein komplettes Damespiel zu entwickeln. Der wichtigste Grund, warum es ins Buch aufgenommen wurde, ist, daß ich daran eine Methode, Spielbretter für Brettspiele so zu numerieren, daß der Computer leicht damit arbeiten kann, demonstriere. Ein ähnliches Numerierungssystem kann als Kern eines Schach- oder Mühleprogramms verwendet werden. Ich ersuche Sie dringend, den Text sorgfältig durchzuarbeiten, und die Programme wie aufgelistet einzugeben. Wenn Sie dies tun, werden Sie eine Menge lernen, das Sie gebrauchen können, wenn Sie Ihre eigenen Brettspiele schreiben. Sie haben wenig davon, wenn Sie nur das endgültige Dameprogramm ohne die vorhergegangenen Schritte eingeben.

Das Numerierungssystem

Die normale Art, ein Damebrett zu numerien, ist, die weißen Quadrate von 1 bis 32 abzuzählen (eigentlich würden die schwarzen Quadrate gezählt, aber hier werden die weißen verwendet, weil das im Zusammenhang mit dem Computer einfacher ist). Doch diese Numerierungsmethode erzeugt ein Problem, wenn wir versuchen, einen Zug durch die Differenz zwischen zwei diagonal aneinanderliegenden Quadraten zu definieren. In der einen Richtung kann der Unterschied zwischen den Quadraten drei oder vier sein, und in der anderen Richtung vier oder fünf. Es gibt also keine »freien« Zahlen, die anzeigen, wo der Rand des Spielbrettes beginnt. Es schrieb ein Mann mit Namen A.L. Samuels um 1960 einen Artikel für Scientific American (siehe Strachey, Christopher »Systems Analysis and Programming« in Teilen von Scientific American, W.H. Freeman and Co., San Francisco, 1971), in dem er ein cleveres Numerierungssystem erfand, wobei die Differenz zwischen diagonal aufeinanderstoßenden Quadraten jeweils vier und fünf (oder minus vier und minus fünf) ist. Es erlaubte auch, Feldern Nummern zuzuordnen, die sich nicht mehr auf dem Spielplan befanden. Ich habe dieses Numerierungssystem etwas geändert, um es für den ZX81 passender zu machen und in meinem System ist die Differenz zwischen Quadraten jeweils sechs oder sieben (oder minus sechs und minus sieben). Mein

System stellt ganz einfach ein Array mit 82 Elementen auf und ordnet bestimmten Elementen des Arrays bestimmte Quadrate des Spielfeldes zu. Alle anderen faßt der Computer als vom Brett entfernt auf.
In diesem Programm ordnet der ZX81 jedem Feld die Zahl 9, einem leeren Quadrat die Zahl ∅, einem Spielstein des Computers den Wert 1 (einer Dame des Computers den Wert 2), einem Spielstein des Spielers minus 1 und einer Dame des Spielers minus 2 zu. Das hört sich etwas kompliziert an, aber halten Sie mit mir durch und alles wird (hoffentlich) klar werden.



Hier ist mein numeriertes Spielfeld. Sie können sehen, daß wenn Sie von der oberen rechten Ecke (69) zum Quadrat schräg darunter ziehen, die Differenz zwischen den beiden Quadraten minus 6 beträgt. Wählen Sie nun irgend ein anderes Quadrat auf dem Spielfeld, von dem Sie hinunter und nach links ziehen können, und Sie werden sehen, daß sich jeweils eine Differenz von minus 6 zwischen den beiden Quadraten ergibt. Der Computer kann relativ leicht mit derartig vorhersehbaren Ergebnissen arbeiten. Fahren Sie in die andere Richtung, also hinauf und nach links, und Sie werden zwischen den beiden Quadraten eine Differenz von minus 7 feststellen. In der ersten

Version von Dame werden wir wirklich auf dem hier abgedruckten Spielfeld spielen, also sollten Sie besser anfangen, eine größere Anzahl von kleinen Knöpfen oder ähnlichen als Spielsteine verwendbaren Dingen zu suchen.

Das Programm besteht aus zwei Teilen. Der erste stellt das Spielfeld auf (die Subroutine, die bei Zeile 9ØØØ beginnt) und der zweite (1Ø bis 37Ø) spielt das Spiel. Ihre Spielsteine beginnen im unteren Teil des gedruckten Spielfeldes (auf den niedrigen Zahlen) und die des ZX81 im oberen Teil. Sie plazieren Ihre Steine und die des ZX81 auf dem Spielfeld und drücken RUN. Der ZX81 ordnet die Werte der Elemente dem Array im FAST-MODE zu und geht dann in den SLOW-MODE, um das Spiel zu spielen. Er spielt sogar im SLOW-MODE bemerkenswert schnell. Aber wenn Sie gerne noch schneller spielen wollen, lassen Sie Zeile 5 im Moment aus, obwohl Sie SLOW später brauchen werden, wenn der ZX81 ein Spielfeld ausdruckt. Die Züge des ZX81 werden durch zwei Nummern angegeben. Die erste ist das Quadrat von dem er, die zweite auf das er zieht. Bewegen Sie zuerst seinen Stein auf dem Spielbrett und entscheiden Sie dann über Ihren Zug. Ziehen Sie Ihren Stein, bevor Sie den Zug eingeben (dies tun Sie, indem Sie die Zahl des Feldes, von dem Sie ziehen, NEWLINE und dann die Zahl des Feldes, auf das Sie ziehen, eingeben), weil Sie sonst Ihren Zug vergessen könnten. Im FAST-MODE kann der ZX81 manchmal eine Entscheidung so schnell treffen, daß Sie Ihren Finger noch nicht von der NEWLINE-Taste weggenommen haben, ehe der ZX81 seine Entscheidung gemacht und ausgedruckt hat.

In diesem Programm stellt der ZX81 die Damen automatisch auf (seine Damen werden in Zeile 4Ø, Ihre in Zeile 1ØØ aufgestellt) und er verwendet seine Damen mit dem größtmöglichen Effekt trotz der Tatsache, daß die einzige Strategie für den Gebrauch der Damen in Zeile 65 (die seine Damen auf den Feldern 24 bis 53 ziehen läßt) stehen. Beachten Sie, daß in diesem Programm mehrfache Sprünge weder vom ZX81 noch von Ihnen vorgesehen sind. Das Programm wurde absichtlich unvollständig gelassen, sodaß Sie daran arbeiten können, nachdem Sie das Material, das in diesem Kapitel präsentiert wurde, verstanden haben. Am Ende finden Sie eine Anzahl von Vorschlägen für Verbesserungen, darunter auch den Einbau von Mehrfachsprüngen.

Hier ist der erste Teil des Programmes, der Teil, der die

Grundposition aufstellt. Er erzeugt ein Array (DIM A (82)) und füllt es dann mit Zahlen, die die Spielsteine (1 und minus 1), leere Spielfelder (Ø) und Spielfelder außerhalb des Spielfeldes (9) repräsentieren.

```
    1 GOSUB 9000
    5 SLOW
   10 STOP
 9000 FAST
 9010 DIM A(82)
 9020 DIM X(2)
 9030 LET X(1)=-6
 9040 LET X(2)=-7
 9050 FOR Z=1 TO 82
 9060 LET A(Z)=9
 9070 IF Z<73 AND Z>55 AND NOT (Z
 =67 OR Z=68 OR Z=60 OR Z=61 OR Z
 =62) THEN LET A(Z)=1
 9080 IF Z<54 AND Z>42 AND NOT (Z
 =47 OR Z=48 OR Z=49) THEN LET A(
 Z)=0
 9090 IF Z<41 AND Z>23 AND NOT (Z
 =34 OR Z=35 OR Z=36 OR Z=28 OR Z
 =29) THEN LET A(Z)=-1
 9100 NEXT Z
 9110 LET A$="MEIN ZUG#"
 9120 LET B$="IHR ZUG?"
 9130 RETURN
```

Überprüfen Sie, ob dieses Programm richtig arbeitet, bevor Sie weitergehen. Fügen Sie 8ØØØ STOP hinzu und lassen Sie das Programm dann laufen. Wenn das Programm stoppt, lassen Sie den ZX81 folgendes ausdrucken, um sicher sein zu können, daß er die Werte richtig zugeordnet hat. Die richtige Antwort steht in Klammern nach dem Befehl.

PRINT A (23)	(9)	PRINT A (38)	(-1)
PRINT A (54)	(9)	PRINT A (51)	(Ø)
PRINT A (64)	(1)	PRINT A (73)	(9)

Wenn das zufriedenstellend funktioniert, SAVEn Sie es einige Male (nur für den Fall, daß Sie das Programm später verlieren) und geben Sie dann den folgenden Teil ein, der das eigentliche Spiel spielt.

```
   10 LET Q=0
   20 FOR Z=24 TO 72
   30 IF NOT (A(Z)=1 OR A(Z)=2) T
 HEN GOTO 100
   40 IF A(Z)=1 AND Z>23 AND Z<28
  THEN LET A(Z)=2
```

```
  50 FOR X=1 TO 2
  60 IF A(Z+X(X))<0 AND A(Z+2*X(
X))=0 THEN LET Q=X(X)
  65 IF Z>55 THEN GOTO 80
  70 IF A(Z)=2 AND A(Z-X(X))<0 A
ND A(Z-2*X(X))=0 THEN LET Q=-X(X
)
  80 IF NOT Q=0 THEN GOTO 120
  90 NEXT X
 100 IF A(Z)=-1 AND Z>68 AND Z<7
3 THEN LET A(Z)=-2
 105 NEXT Z
 110 IF Q=0 THEN GOTO 160
 120 LET A(Z+Q)=0
 130 LET A(Z+2*Q)=A(Z)
 140 LET A(Z)=0
 150 PRINT A$;Z,Z+2*Q
 155 GOTO 320
 160 LET Y=0
 170 LET Z=24+INT (RND*48)
 180 LET Y=Y+1
 190 IF Y<100 AND NOT (A(Z)=1 OR
 A(Z)=2) THEN GOTO 170
 200 FOR X=1 TO 2
 210 IF A(Z+X(X))=0 THEN LET Q=X
(X)
 220 IF A(Z)=2 AND A(Z-X(X))=0 T
HEN LET Q=-X(X)
 230 IF NOT Q=0 THEN GOTO 290
 240 NEXT X
 260 IF Y<100 THEN GOTO 170
 270 PRINT "SIE HABEN GEWONNEN"
 280 STOP
 290 LET A(Z+Q)=A(Z)
 300 LET A(Z)=0
 310 PRINT A$;Z,Z+Q
 320 PRINT B$
 325 INPUT A
 330 INPUT B
 335 CLS
 340 LET A(B)=A(A)
 350 LET A(A)=0
 360 IF ABS (A-B)>7 THEN LET A(A
+(INT ((B-A)/2)))=0
 370 GOTO 10
```

Spielen Sie eine komplette Partie mit dem ZX81 und verwenden Sie kleine Knöpfe auf dem gedruckten Spielplan, kommen Sie dann wieder zum Buch zurück, und wir werden das Programm Zeile für Zeile durchgehen, um zu erklären, wie es arbeitet.
Zeile 10 stellt die Variable Q als Indikator (oder »Flag«) auf, um zu zeigen, daß noch kein Zug gemacht wurde.

Wenn eine Zugentscheidung gefällt wurde, ändert sich Q auf einen Wert, der nicht Ø ist. Zeile 2Ø startet die Schleife, die jedes Quadrat des Spielbretts überprüft, um zu kontrollieren, ob Beute gemacht werden kann. Zeile 3Ø checkt, ob das überprüfte Quadrat (genaugenommen das Element des Array) ein ZX81-Spielstein (1) oder eine ZX81-Dame (2) ist. Trifft das nicht zu, verschwendet der Computer keine Zeit darauf, die Logikbefehle zu durchlaufen und springt auf Zeile 1ØØ. Bei Zeile 1ØØ überprüft er, ob ein Spielstein des Spielers (minus 1) auf der Damelinie steht, und wenn das der Fall ist, ändert er den Spielstein in eine Dame (minus 2).
Wenn der Computer entdeckt, daß der Stein über den er nachdenkt, einer seiner eigenen ist, überprüft er (in Zeile 4Ø), ob er nicht in eine Dame (also von 1 auf 2) geändert werden sollte. Als nächstes geht er durch die X-Schleife, prüft nach, ob das Feld darunter und links (minus 6) kleiner als Ø ist (ist es das, beinhaltet das Quadrat einen Stein des Spielers) und ob das Quadrat in der doppelten Entfernung (also minus 12) ein leeres Quadrat ist. Wenn diese beiden Bedingungen wahr sind, setzt er den Beuteindikator (Q) auf minus 6. Zeile 65 läßt den Stein, wenn er eine Dame ist, nicht weiter hinauswandern, als auf Feld 55 (wie schon vorher besprochen, als wir über die Strategie der Dame sprachen).
In Zeile 7Ø macht er eine ähnliche Kontrolle wie in Zeile 6Ø, mit dem Unterschied, daß er dieses Mal überprüft, ob die Dame (2) einen Beutezug nach rückwärts machen kann. Sie werden bemerken, wenn Sie auf den numerierten Spielplan sehen, daß die legalen Damezüge des ZX81 plus 6 und plus 7 sind, also sucht diese Zeile einen Stein des Spielers auf dem Feld plus 6 und überprüft, ob das Quadrat darunter (plus 12) frei ist. Trifft das zu, wird der Beuteindikator auf plus sechs gesetzt (tatsächlich wird er auf minus 7 (auf X) gesetzt, was aber den gleichen Effekt hat).
Wenn der ZX81 in Zeile 8Ø Q auf Null findet, weiß er, daß er nicht die Entscheidung getroffen hat, einen Stein zu schlagen. Sollte Q nicht Null sein, springt er auf Zeile 12Ø, um nach der Entscheidung zu handeln. Bei Zeile 9Ø ackert er mit dem zweiten Wert (minus 7) von X (X) durch die X-Schleife. Zeile 1Ø5 führt die Suche durch die Z-Schleife weiter und überprüft jedes Quadrat, um sicher zu sein, daß kein möglicher Fang ausgelassen wurde.
Wenn Q immer noch Null ist, wenn der ZX81 zu Zeile 11Ø kommt, er also weiß, daß es keinen möglichen Fang

auf dem Spielfeld gibt, geht er zu Zeile 16Ø um einen zufälligen legalen Zug zu finden. Sollte Q nicht gleich Null sein, machen die Zeilen 12Ø, 13Ø und 14Ø einen Beutezug und die Zeile 15Ø zeigt ihn an. Nachdem der ZX81 seinen Zug gemacht hat, geht er zur Zeile 32Ø, um auf den Zug des Spielers zu warten.
Wenn kein legaler Zug gemacht werden konnte, sucht der Computer nach einem zufälligen Zug. Der Indikator Y (Zeile 16Ø) zählt die Anzahl der gemachten Versuche, um einen legalen Zug zu finden (nach jedem Versuch wird in Zeile 18Ø 1 zu Y addiert). Wenn nach 99 Versuchen noch immer kein Zug gefunden wurde, (Zeile 26Ø) gibt sich der ZX81 klugerweise geschlagen. Zeile 19Ø kontrolliert, ob auf dem von ihm gewählten Feld einer seiner eigenen Steine steht; wenn nicht und Y kleiner als 1ØØ ist, geht er zurück, um mit dem Zufallsgenerator ein neues Feld auszusuchen. Nachdem er einen eigenen Stein gefunden hat, führt er den X-Schleifentest (Zeilen 2ØØ bis 24Ø) durch, um zu sehen, ob er einen legalen Zug durchführen kann, wobei er wieder den Indikator Q verwendet, um zu überprüfen, ob bereits ein Zug gemacht wurde. Die Zeilen 27Ø und 28Ø bekennen, wie schon erwähnt, die Niederlage. Die Zeilen 29Ø bis 31Ø führen den Zug durch und drucken ihn für den Spieler aus.
Der Spieler macht dann seinen Zug (Zeilen 325 und 33Ø), und der ZX81 handelt danach (Zeilen 34Ø und 36Ø), wobei er die etwas komplizierte Zeile 36Ø verwendet, um einen seiner Steine zu entfernen, wenn der Zug des Spielers größer als 7 ist. Er weiß, daß ein Zug zu einem angrenzenden Feld entweder 6, 7, minus 6 oder minus 7 sein muß, und ein Zug, der im Wert größer ist, nur deswegen entstehen kann, weil der Spieler über einen Stein des ZX81 gesprungen ist (oder geschummelt hat). Der ZX81 nimmt nie an, daß der Spieler schummelt (weil er das selbst nicht kann). Also verwendet er die Zeile 36Ø, um auszuarbeiten, welches Quadrat zwischen dem Quadrat von dem der Spieler gezogen ist, und dem Quadrat auf das er zog, liegt, um seinen eigenen Stein zu löschen.
Nachdem Sie den Vorgang verstanden haben, können Sie beginnen, das Programm zu verbessern. Ich werde Ihnen eine Möglichkeit angeben, ein Schachbrett auszudrucken (obwohl Sie noch immer auf den numerierten Spielplan Bezug nehmen müssen, um Ihre Züge einzugeben), die anderen Verbesserungen jedoch liegen bei Ihnen. Nachdem Sie den Ausdruck des Spielfeldes unter Kontrolle haben, könnten Sie ver-

suchen, die folgenden Punkte hinzufügen, um das Programm in ein echtes Damespiel zu ändern:

1. Fügen Sie mehrfache Sprünge hinzu (das ist relativ einfach, weil jeder Mehrfachsprung eine voraussehbare arithmetische Änderung des Startfeldes ist. Das Array ist groß genug, um Sie in den meisten Fällen davor zu bewahren, während der Suche nach mehrfachen Sprüngen herauszuspringen.
   Mehrfache Sprünge des Spielers sind sehr einfach. Bringen Sie den ZX81 einfach dazu, »Ist das ein mehrfacher Sprung?« zu fragen.
   Wenn ein Stein geschlagen wird (Zeile 36Ø) und die Antwort »Ja« ist, lassen Sie ihn zu Zeile 325 zurückspringen, sodaß der zweite Zug eingegeben werden kann.
2. Fügen Sie eine Gesamtwertung hinzu, sodaß der Computer nach jeden Zug ausdruckt, wieviele Steine jeder von Ihnen geschlagen hat. Der ZX81 könnte sich dann auch als Sieger deklarieren, wenn er, sagen wir, 9 Ihrer Steine geschlagen hat.
3. Erfinden Sie eine Möglichkeit, die Züge direkt einzugeben, statt auf dem numerierten Spielplan nachzusehen.

Es gibt wahrscheinlich noch weitere Verbesserungen, die gemacht werden können, aber ich glaube, daß diese die wichtigsten sind.

—Hinzufügen eines Displays—

Die letzte Verbesserung, die ich Ihnen für das Programm gebe, ist eine kleine von Tony Baker entwickelte Routine, die es ermöglicht, ein Spielbrett auszudrucken.

Löschen Sie Zeile 32Ø, ändern Sie 31Ø und 15Ø auf GOSUB 5ØØØ und geben Sie dann das Folgende ein:

```
5000 FOR M=24 TO 72
5010 IF A(M)=1 THEN LET A(M)=61
5020 IF A(M)=-1 THEN LET A(M)=52
5030 IF A(M)=2 THEN LET A(M)=48
5040 IF A(M)=-2 THEN LET A(M)=13
5050 NEXT M
5060 PRINT AT 1,0;A$;Z,Z+Q
5070 PRINT AT 3,11;B$
5080 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
5090 FOR K=0 TO 3
5100 FOR J=0 TO 3
5110 PRINT "#";CHR$ A(72-J-13*K)
;
5120 NEXT J
5130 PRINT "▒"
5140 FOR J=0 TO 3
5150 PRINT CHR$ A(66-J-13*K);" "
;
5160 NEXT J
5170 PRINT "▒"
5180 NEXT K
5190 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
5200 FOR M=24 TO 72
5210 IF A(M)=61 THEN LET A(M)=1
5220 IF A(M)=52 THEN LET A(M)=-1
5230 IF A(M)=48 THEN LET A(M)=2
5240 IF A(M)=13 THEN LET A(M)=-2
5250 NEXT M
5260 RETURN
```

Nachdem alles an seinem Platz ist, werden Sie sehen, daß sich die Spielsteine tatsächlich bewegen. Ignorieren Sie die Zeile »Mein Zug...«, wenn Sie die gleiche Feldnummer zweimal angibt. Warten Sie, bis sie sich geändert hat und beobachten Sie dann die Bewegung des Steins des Computers. Wenn der Computer einen Stein schlägt, sieht das besonders gut aus. Warten Sie, bis das inverse L in der unteren linken Ecke erscheint und geben Sie dann Ihren Zug ein.
Beachten Sie, daß die zwei Schleifen (5ØØØ bis 5Ø5Ø und 52ØØ bis 525Ø) bestimmen, welches Zeichen

welchen Stein bedeutet. Selbstverständlich können Sie dies ändern, wenn Sie wollen. Beachten Sie, daß, genau wie im »echten« Damespiel eine Dame erst nach dem Zug, mit dem sie zur Damelinie bewegt wurde, eine Dame wird.

Life 1 und 2

Bevor wir zu unserem endgültigen Dameprogramm gehen, möchte ich gerne zeigen, wie ein numeriertes Gitter, das die gleiche zahlenmäßige Beziehung zwischen aneinanderanstoßenden Quadraten hat, verwendet werden kann, um das Programm LIFE zu produzieren.
LIFE wurde von John Conway von der Cambridge University im Jahre 1970 erfunden. Das Spiel, das die Geburt, das Wachsen und den Tod einer Zellenkolonie simuliert, erzeugt faszinierende Effekte.
Jede Zelle entsteht, überlebt oder stirbt nach folgenden von Conway aufgestellten Regeln:

* Jede Zelle des Gitters (das in unserem Fall auf der Außenseite eines Zylinders gezeichnet ist) hat acht mögliche Nachbarn.
* Jede Zelle mit zwei oder drei Nachbarn überlebt bis zur nächsten Generation.
* Gibt es drei und nur drei benachbarte Zellen, entsteht eine neue Zelle.
* Jede Zelle mit vier oder mehr Nachbarn stirbt wegen Überbevölkerung.

```
  1 REM 16K LIFE MIT PLOT
  5 LET G=0
  7 RAND
 10 DIM A(10,10)
 20 DIM B(10,10)
 30 FOR X=2 TO 9
 40 FOR Y=2 TO 9
 50 IF RND>.55 THEN LET A(X,Y)=
1
 60 LET B(X,Y)=A(X,Y)
 70 NEXT Y
 80 NEXT X
 90 GOSUB 1000
 95 LET G=G+1
100 FOR X=2 TO 9
110 FOR Y=2 TO 9
120 LET C=0
```

```
 130 IF A(X-1,Y-1)=1 THEN LET C=
C+1
 140 IF A(X-1,Y)=1 THEN LET C=C+
1
 150 IF A(X-1,Y+1)=1 THEN LET C=
C+1
 160 IF A(X,Y-1)=1 THEN LET C=C+
1
 170 IF A(X,Y+1)=1 THEN LET C=C+
1
 180 IF A(X+1,Y-1)=1 THEN LET C=
C+1
 190 IF A(X+1,Y)=1 THEN LET C=C+
1
 200 IF A(X+1,Y+1)=1 THEN LET C=
C+1
 210 IF A(X,Y)=1 AND C<>3 AND C<
>2 THEN LET B(X,Y)=0
 220 IF A(X,Y)=0 AND C=3 THEN LE
T B(X,Y)=1
 230 NEXT Y
 240 NEXT X
 250 GOTO 90
1000 PRINT AT 8,0;"GENERATION#";
G
1001 FOR X=1 TO 10
1002 FOR Y=1 TO 10
1004 IF A(X,Y)=1 THEN UNPLOT X,Y
+11
1006 IF A(X,Y)=1 THEN PLOT X+11,
Y+11
1010 IF A(X,Y)=0 THEN PLOT X,Y+1
1
1012 IF A(X,Y)=0 THEN UNPLOT X+1
1,Y+11
1015 LET A(X,Y)=B(X,Y)
1020 IF A(X,Y)=1 THEN UNPLOT X,Y
1025 IF A(X,Y)=1 THEN PLOT X+11,
Y
1030 IF A(X,Y)=0 THEN PLOT X,Y
1035 IF A(X,Y)=0 THEN UNPLOT X+1
1,Y
1040 NEXT Y
1050 NEXT X
1060 RETURN
```

```
   1 REM LIFE MIT PRINT AT
   5 LET G=0
   7 RAND
  10 DIM A(10,10)
  20 DIM B(10,10)
  30 FOR X=2 TO 9
  40 FOR Y=2 TO 9
  50 IF RND>.45 THEN LET A(X,Y)=
1
  60 LET B(X,Y)=A(X,Y)
  70 NEXT Y
  80 NEXT X
  90 GOSUB 1000
  95 LET G=G+1
 100 FOR X=2 TO 9
 110 FOR Y=2 TO 9
 120 LET C=0
 130 IF A(X-1,Y-1)=1 THEN LET C=
C+1
 140 IF A(X-1,Y)=1 THEN LET C=C+
1
 150 IF A(X-1,Y+1)=1 THEN LET C=
C+1
 160 IF A(X,Y-1)=1 THEN LET C=C+
1
 170 IF A(X,Y+1)=1 THEN LET C=C+
1
 180 IF A(X+1,Y-1)=1 THEN LET C=
C+1
 190 IF A(X+1,Y)=1 THEN LET C=C+
1
 200 IF A(X+1,Y+1)=1 THEN LET C=
C+1
 210 IF A(X,Y)=1 AND C<>3 AND C<
>2 THEN LET B(X,Y)=0
 220 IF A(X,Y)=0 AND C=3 THEN LE
T B(X,Y)=1
 230 NEXT Y
 240 NEXT X
 250 GOTO 90
1000 PRINT AT 4,13;"#GENERATION#
";G
1001 FOR X=1 TO 10
1002 FOR Y=1 TO 10
1015 LET A(X,Y)=B(X,Y)
1020 IF A(X,Y)=1 THEN PRINT AT X
,Y;"O"
1030 IF A(X,Y)=0 THEN PRINT AT X
,Y;"#"
1040 NEXT Y
1050 NEXT X
1060 RETURN
```

Dame

Es ist Zeit, wieder zu unserem Damespiel zurückzukehren, und diesesmal ist es ein »richiges« Damespiel mit einer einfachen Methode, Züge einzugeben, der Möglichkeit zu mehrfachen Sprüngen, einer Gesamtwertung und einigen anderen Features.
Dieses Programm ist im SLOW-MODE sehr eindrucksvoll, besonders in den frühen Stufen des Spiels. Trotzdem kann der ZX81, wenn weniger Steine auf dem Brett sind, und der ZX81 länger überlegen muß, bevor er sich für einen Zug entscheidet, sehr langsam werden.
Deshalb könnten Sie es bevorzugen, das gesamte Programm im FAST-MODE laufen zu lassen. Auf jeden Fall sollten Sie im Anfangsstadium, während Sie Ihr Programm testen, auf FAST schalten, einfach, um Zeit zu sparen.

```
   5 REM DAME-16K
  10 FAST
  20 RAND
  30 GOSUB 9000
  50 LET ZUG=0
  60 GOSUB 7000
  70 SLOW
1000 IF C$(1)="K" THEN GOTO 1010
1005 PRINT AT 16,3;"IHR LETZTER
ZUG WAR NACH ";C$
1010 LET ZUG=0
1015 PRINT AT 17,4;"NEUER ZUG (B
UCHST.,ZAHL)?"
1020 INPUT B$
1030 PRINT AT 17,14;"VON ";B$;"
NACH?   "
1040 INPUT C$
1050 PRINT AT 17,26;" ";C$
1055 PRINT "    BITTE WARTEN"
1060 LET Z(1)=CODE B$*10+CODE B$
(2)
1070 LET Z(2)=CODE C$*10+CODE C$
(2)
1080 FOR I=1 TO 2
1090 IF Z(I)=410 THEN LET B(I)=7
2
1100 IF Z(I)=412 THEN LET B(I)=7
1
1110 IF Z(I)=414 THEN LET B(I)=7
0
1120 IF Z(I)=416 THEN LET B(I)=6
9
1130 IF Z(I)=419 THEN LET B(I)=6
6
```

```
1140 IF Z(I)=421 THEN LET B(I)=6
5
1150 IF Z(I)=423 THEN LET B(I)=6
4
1160 IF Z(I)=425 THEN LET B(I)=6
3
1170 IF Z(I)=430 THEN LET B(I)=5
9
1180 IF Z(I)=432 THEN LET B(I)=5
8
1190 IF Z(I)=434 THEN LET B(I)=5
7
1200 IF Z(I)=436 THEN LET B(I)=5
6
1210 IF Z(I)=439 THEN LET B(I)=5
3
1220 IF Z(I)=441 THEN LET B(I)=5
2
1230 IF Z(I)=443 THEN LET B(I)=5
1
1240 IF Z(I)=445 THEN LET B(I)=5
0
1250 IF Z(I)=450 THEN LET B(I)=4
6
1260 IF Z(I)=452 THEN LET B(I)=4
5
1270 IF Z(I)=454 THEN LET B(I)=4
4
1280 IF Z(I)=456 THEN LET B(I)=4
3
1290 IF Z(I)=459 THEN LET B(I)=4
0
1300 IF Z(I)=461 THEN LET B(I)=3
9
1310 IF Z(I)=463 THEN LET B(I)=3
8
1320 IF Z(I)=465 THEN LET B(I)=3
7
1330 IF Z(I)=470 THEN LET B(I)=3
3
1340 IF Z(I)=472 THEN LET B(I)=3
2
1350 IF Z(I)=474 THEN LET B(I)=3
1
1360 IF Z(I)=476 THEN LET B(I)=3
0
1370 IF Z(I)=479 THEN LET B(I)=2
7
1380 IF Z(I)=481 THEN LET B(I)=2
6
1390 IF Z(I)=483 THEN LET B(I)=2
5
1400 IF Z(I)=485 THEN LET B(I)=2
4
1410 NEXT I
1420 LET ZUG=0
```

```
1430 LET A(B(2))=A(B(1))
1440 LET A(B(1))=B
1450 IF ABS (B(1)-B(2))>7 THEN L
ET ZUG=1
1460 IF ABS (B(1)-B(2))>7 THEN L
ET A((B(1)+B(2))/2)=B
1465 FAST
1470 GOSUB 7000
2000 REM SPRUNG
2002 LET Q=0
2004 FOR Z=69 TO 72
2005 IF A(Z)=H THEN LET A(Z)=W
2007 NEXT Z
2010 FOR Z=24 TO 72
2020 IF A(Z)<>C AND A(Z)<>K THEN
 GOTO 2110
2030 IF A(Z)=C AND Z>23 AND Z<28
 THEN LET A(Z)=K
2040 GOSUB 8000
2080 IF Q<>0 THEN GOTO 2125
2110 NEXT Z
2120 GOTO 2240
2125 LET A(Z+2*Q)=A(Z)
2130 LET S=S+1
2135 LET A(Z+Q)=B
2145 LET A(Z)=B
2150 GOSUB 7000
2155 LET Z=Z+2*Q
2160 LET Q=0
2170 GOSUB 8000
2190 IF Q<>0 THEN GOTO 2125
2210 GOTO 50
2235 REM ERRECHNETER ZUG
2240 FOR Z=24 TO 72
2250 IF A(Z)=9 OR A(Z)=B OR A(Z)
=H OR A(Z)=W THEN GOTO 2305
2260 FOR D=1 TO 4
2265 LET U=X(D)
2270 IF A(Z+U)=H AND D>3 THEN GO
TO 2290
2280 IF (A(Z+U)=H OR A(Z+U)=W) A
ND A(Z-U)=B AND A(Z-2*U)=K THEN
LET Q=Z-2*U
2290 IF Q=0 AND (A(Z+U)=H OR A(Z
+U)=W) AND A(Z-U)=B AND (A(Z-2*U
)=K OR A(Z-2*U)=C) THEN LET Q=Z-
2*U
2300 IF Q=0 THEN NEXT D
2305 IF Q=0 THEN NEXT Z
2310 IF Q=0 THEN GOTO 2348
2320 LET A(Q+X(D))=A(Q)
2330 LET A(Q)=B
2340 GOSUB 7000
2345 GOTO 50
2347 REM ZUFAELLIGER ZUG
```

```
2348 FAST
2349 LET IT=0
2350 DIM I(49)
2355 FOR J=1 TO 49
2360 LET I(J)=J+23
2365 NEXT J
2370 LET N=49
2373 LET Y=1+INT (RND*N)
2375 LET Z=I(Y)
2380 IF A(Z)=9 OR A(Z)=B OR A(Z)
=H OR A(Z)=W THEN GOTO 2430
2390 FOR D=1 TO 4
2395 LET U=X(D)
2400 IF A(Z)=C AND D>2 THEN GOTO
 2430
2404 IF IT=1 THEN GOTO 2410
2406 IF A(Z+U)=B AND A(Z+2*U)<>H
 AND A(Z+2*U)<>W AND A(Z+13*SGN
U)<>H AND A(Z+13*SGN U)<>W AND A
(Z-1+2*(ABS (D-2.5)>1))<>H AND A
(Z-1+2*(ABS (D-2.5)>1))<>W THEN
LET Q=U
2408 GOTO 2420
2410 IF A(Z+U)=B THEN LET Q=U
2420 IF Q<>0 THEN GOTO 2460
2425 NEXT D
2430 LET I(Y)=I(N)
2433 LET N=N-1
2436 IF N>0 THEN GOTO 2373
2438 LET IT=IT+1
2439 IF IT=1 THEN GOTO 2355
2440 LET E$="VERLOREN"
2450 GOSUB 7000
2460 LET A(Z+Q)=A(Z)
2470 LET A(Z)=B
2490 GOTO 50
7000 REM AUSDRUCK SPIELBRETT
7010 PRINT AT 16,0;"
                    "
7020 PRINT AT 17,0;"

      "
7030 IF ZUG=1 THEN LET T=T+1
7090 PRINT AT 2,4;"COMPUTER ";CH
R$ (S+156),"  MENSCH ";CHR$ (T+1
56)
7100 PRINT AT 4,8;[redacted]
7110 PRINT AT 5,8;"[redacted] ";CHR$ A(7
2);" ";CHR$ A(71);" ";CHR$ A(70)
;" ";CHR$ A(69);"[redacted]"
7120 PRINT AT 6,8;"[redacted]";CHR$ A(66
);" ";CHR$ A(65);" ";CHR$ A(64);
" ";CHR$ A(63);"[redacted] "
7130 PRINT AT 7,8;"[redacted] ";CHR$ A(5
9);" ";CHR$ A(58);" ";CHR$ A(57)
```

```
;" ";CHR$ A(56);"■C"
7140 PRINT AT 8,8;"D■";CHR$ A(53
);" ";CHR$ A(52);" ";CHR$ A(51);
" ";CHR$ A(50);" ■D"
7150 PRINT AT 9,8;"E■ ";CHR$ A(4
6);" ";CHR$ A(45);" ";CHR$ A(44)
;" ";CHR$ A(43);"■E"
7160 PRINT AT 10,8;"F■";CHR$ A(4
0);" ";CHR$ A(39);" ";CHR$ A(38)
;" ";CHR$ A(37);" ■F"
7170 PRINT AT 11,8;"G■ ";CHR$ A(
33);" ";CHR$ A(32);" ";CHR$ A(31
);" ";CHR$ A(30);"■G"
7180 PRINT AT 12,8;"H■";CHR$ A(2
7);" ";CHR$ A(26);" ";CHR$ A(25)
;" ";CHR$ A(24);" ■H"
7190 PRINT AT 13,8;"■12345675■
"
7210 IF E$="VERLOREN" THEN PRINT
 AT 17,7;"ICH GEBE DAS SPIEL AUF
"
7220 IF E$="VERLOREN" THEN STOP
7230 IF S>=12 THEN PRINT AT 17,1
2;"ICH HABE GEWONNEN"
7240 IF T>=12 THEN PRINT AT 17,1
0;"SIE HABEN GEWONNEN"
7250 IF T>=12 OR S>=12 THEN STOP

7260 LET U$=""
7265 IF ZUG=1 THEN SLOW
7270 IF ZUG=1 THEN PRINT AT 15,0
;"KOENNEN SIE NOCHEINMAL","SPRIN
GEN (J/N)?"
7280 IF ZUG=1 THEN INPUT U$
7290 PRINT AT 15,0;"
          ","                    "
7300 LET ZUG=0
7305 IF U$<>"J" THEN FAST
7310 IF U$="J" THEN GOTO 1000
7320 RETURN
8000 FOR D=1 TO 4
8005 LET U=X(D)
8010 IF A(Z)<>K AND D>2 THEN GOT
O 8050
8020 IF (A(Z+U)=H OR A(Z+U)=W) A
ND A(Z+2*U)=B THEN LET Q=U
8030 IF Q<>0 THEN GOTO 8050
8040 NEXT D
8050 RETURN
9000 DIM A(100)
9005 LET Y=0
9010 DIM X(4)
9020 LET X(1)=-6
9030 LET X(2)=-7
9040 LET X(3)=6
```

```
9050 LET X(4)=7
9060 LET H=189
9070 LET C=180
9080 LET W=188
9090 LET K=176
9100 LET B=128
9105 LET Q=0
9110 FOR Z=1 TO 100
9120 LET A(Z)=9
9130 IF Z<73 AND Z>55 AND Z<>67
AND Z<>68 AND Z<>60 AND Z<>61 AN
D Z<>62 THEN LET A(Z)=C
9140 IF Z<54 AND Z>42 AND Z<>47
AND Z<>48 AND Z<>49 THEN LET A(Z
)=B
9150 IF Z<41 AND Z>23 AND Z<>34
AND Z<>35 AND Z<>36 AND Z<>28 AN
D Z<>29 THEN LET A(Z)=H
9160 NEXT Z
9170 LET E$=""
9180 LET S=0
9190 LET T=0
9210 DIM B$(2)
9220 DIM C$(2)
9225 LET C$(1)="K"
9230 DIM Z(2)
9240 DIM B(2)
9300 PRINT AT 10,4;"WOLLEN SIE B
EGINNEN (J/N)?"
9305 SLOW
9310 INPUT J$
9315 FAST
9320 CLS
9330 IF J$="J" THEN RETURN
9335 LET ZUG=0
9340 GOSUB 7000
9350 LET Z=57+INT (RND*3)
9360 LET Q=-7+INT (RND*2)
9370 LET A(Z+Q)=C
9380 LET A(Z)=B
9390 RETURN
```

Wenn Sie die Arbeit, das DAME-Programm einzugeben, ermattet hat, können Sie sich nun mit einigen von Alastair Gourlay — einem Mitglied des ZX-Anwender-Clubs in Schottland — geschriebenen 1K Spielen erholen.

Türme von Hanoi

In diesem Spiel müssen Sie versuchen, fünf Ringe von Turm eins (Spitze) auf Turm drei (Fuß) zu transferieren. Wenn Sie einmal versuchen sollten, einen Ring auf einen kleineren zu legen, verlieren Sie das Spiel.
Am Bildschirm wird ständig die Anzahl der Züge, die Sie gemacht haben, angezeigt.

```
 10 DIM A$(3,6)
 20 DIM A(3)
 30 LET T=0
 40 LET A$(1)="54321-"
 50 LET A$(2)="------"
 60 LET A$(3)=A$(2)
 70 LET A(1)=5
 80 PRINT "ZUG NR.";T
 90 PRINT
100 FOR A=1 TO 3
110 PRINT A$(A)
120 PRINT
130 NEXT A
140 IF A$(3)="54321-" THEN GOTO
290
150 PRINT "VON"
160 INPUT A
170 PRINT A
180 PRINT "NACH"
190 INPUT B
200 LET T=T+1
210 CLS
220 LET A(B)=A(B)+1
230 LET A$(B,A(B))=A$(A,A(A))
240 LET A$(A,A(A))="-"
250 LET A(A)=A(A)-1
260 IF A(B)<2 THEN GOTO 80
270 IF A$(B,A(B))>A$(B,A(B)-1)
THEN GOTO 350
280 GOTO 80
290 PRINT
300 PRINT "SIE HABEN ES IN#";T;
"#ZUEGEN"
310 PRINT
320 PRINT "GESCHAFFT"
330 PAUSE 4E4
340 RUN
350 PRINT
360 PRINT "SIE WOLLEN MICH BETR
UEGEN"
370 PRINT "SIE HABEN VERLOREN"
380 PRINT
390 GOTO 330
```

Blockout

In diesem 1K Programm verwenden Sie die Tasten 5 und 8, um den schwarzen Hindernissen auszuweichen und das Ziel auf der rechten Seite zu erreichen. Wenn Sie sich bewegen, hinterlassen Sie eine Spur. Wenn Sie sich einsperren, können Sie zurückgehen, und das Spiel so vielleicht retten.
Wenn Sie etwas berühren oder über den Rand fahren, werden Sie auf eine Startposition auf der linken Seite zurückgesetzt; soferne Sie nicht über den rechten Rand hinausgekommen sind (wo die Welt endet). Zwei Versionen sind angegeben. Nachdem Sie das langsame Spiel beherrschen, probieren Sie die schwierigere schnelle Version aus.

```
 20 LET B=INT (RND*8)*10+1
 30 LET C=INT (RND*8)*10+9
 40 FOR A=1 TO 8
 50 PRINT "                "
 60 NEXT A
 70 LET E=0
 80 LET D=PEEK 16396+256*PEEK 1
6397
 90 POKE C+D,151
100 POKE B+D,149
110 LET A$=INKEY$
```

Für das schnelle Spiel verwenden Sie:

```
120 IF A$="" THEN GOTO 140
```

Für das langsame Spiel:

```
120 IF A$="" THEN GOTO 110
```

Folgende Zeilen brauchen Sie für beide Versionen:

```
130 LET E=VAL A$
140 POKE B+D,21
150 LET B=B-(E=5)+(E=6)*10-(E=7
)*10+(E=8)
160 IF B>79 THEN LET B=71
170 IF B<1 OR PEEK (D+B)=128 TH
EN LET B=B-1
180 IF PEEK (B+D)=118 THEN LET
B=B-1
190 IF PEEK (B+D)=151 THEN STOP
```

Für ein noch schwierigeres Spiel fügen Sie folgendes hinzu:

```
200 FOR A=1 TO 2
210 LET E=INT (RND*80)+1
220 IF PEEK (E+D)<>8 THEN GOTO
100
230 POKE E+D,128
240 NEXT A
250 GOTO 100
```

Salvador

Mit diesem Programm können Sie hübsche Bilder auf den Bildschirm zeichnen; es ist flexibler, als das ARTIST Programm.
Um den Cursor über den Bildschirm zu führen, zeichnend oder weglöschend, verwenden Sie die Tasten 5, 6, 7 und 8. Wenn Sie auf »O« drücken, startet der Cursor mit dem Zeichnen, wenn Sie auf »Ø« drücken, verwandelt er sich in einen Radiergummi.

```
 10 LET X=0
 20 LET Y=0
 30 LET B=1
 40 IF INKEY$<>"" THEN GOTO 40
 50 LET A$=INKEY$
 60 IF A$="0" THEN LET B=-B
 70 UNPLOT X,Y
 80 PLOT X,Y
 90 IF B=1 THEN UNPLOT X,Y
100 IF A$="" THEN GOTO 50
110 LET X=X-(VAL A$=5)+(VAL A$=
8)
120 LET Y=Y-(VAL A$=6)+(VAL A$=
7)
130 IF X<0 THEN LET X=0
140 IF X>41 THEN LET X=41
150 IF Y<0 THEN LET Y=0
160 IF Y>33 THEN LET Y=33
170 PLOT X,Y
180 GOTO 50
```

Bandit

Dieses Spiel kostet Sie schwindelerregende $ 5 pro Runde. Sie starten mit $ 50. Sie gewinnen mit den Kombinationen

* die ersten beiden Symbole sind gleich
* das erste und das letzte Symbol sind gleich
* alle drei Symbole sind gleich

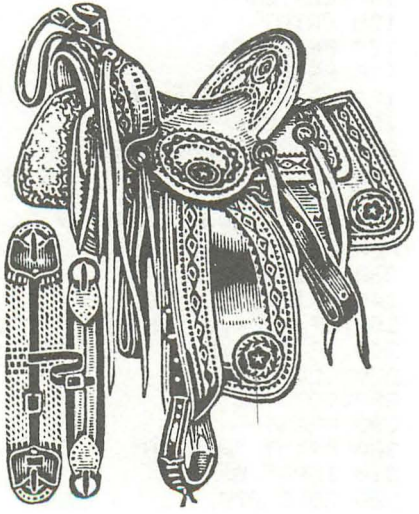
einen Geldbetrag, der vom jeweiligen Symbol (dem Codezeichen 1 bis 7) abhängt.
Wenn »HALT« erscheint, geben Sie »J« ein, um ein Rad anzuhalten, anderenfalls geben Sie einen Zwischenraum und dann NEWLINE ein, um das nächste Spiel zu starten.

```
 10 LET C=50
 20 DIM A(3)
 30 DIM B$(3)
 40 LET A$="▒▒▒▒▒▒▒▒▒▒▒▒▒"
 50 PRINT "#BANDIT#"
 60 PRINT A$
 70 PRINT A$
 80 PRINT "▒";
 90 FOR X=1 TO 3
100 IF B$(X)="J" THEN GOTO 120
110 LET A(X)=INT (RND*7)+1
120 PRINT CHR$ (X);"▒";
130 NEXT X
140 LET B$=""
150 PRINT
160 PRINT A$
170 PRINT A$
180 PRINT "########"
190 LET C=C-5
200 IF C<=0 THEN STOP
210 IF A(1)=A(2) OR A(1)=A(3) T
HEN LET C=C+A(1)*5
220 IF A(1)=A(2) AND A(2)=A(3)
THEN LET C=C+A(1)*8
230 PRINT
240 PRINT ": $";C
250 IF RND<.3 THEN GOTO 290
260 IF INKEY$="" THEN GOTO 260
270 CLS
280 GOTO 50
290 PRINT
300 PRINT "HALTEN"
310 INPUT B$
320 GOTO 270
```

Dodge City

In diesem Spiel verwenden Sie die Tasten 1 und 4, um den schwarzen Hindernissen auszuweichen, die auf Sie zukommen.

```
  10 POKE 16418,12
  20 FOR X=1 TO 12
  30 PRINT "▗";TAB 20;"▗"
  40 NEXT X
  50 LET S=0
  60 LET Q=10
  70 LET Q=Q+(INKEY$="4")-(INKEY
$="1")
  80 SCROLL
  90 LET P=PEEK 16396+256*PEEK 1
6397+Q
 100 IF PEEK P<>0 THEN GOTO 150
 110 POKE P,10
 120 PRINT AT 11,0;"▗";TAB (INT
(RND*18)+1);"■";TAB 20;"▗"
 130 LET S=S+1
 140 GOTO 70
 150 PRINT S
```



****

Umwandeln von ZX80-Programmen für den ZX81

****

Im allgemeinen ist es sehr einfach, Programme, die für den ZX80 mit altem ROM geschrieben wurden, so zu ändern, daß sie auf einer Maschine mit neuem ROM laufen. Programme, die PEEK und POKE beinhalten — besonders in REM-Statements — können allerdings einige Probleme verursachen.
In vielen Fällen werden Sie feststellen, daß die Programme auf einer Maschine mit neuem ROM viel besser laufen. Das heißt, sie sind wirkungsvoller, wenn sie dem Anwender Anweisungen geben, haben bewegliche Grafiken oder eine bessere Bildschirmausgabe. Allerdings ist das neue ROM mit Platz nicht so großzügig, wie es das alte ROM war. Viele 1K Programme passen mit dem neuen ROM nicht mehr in 1K, weil die Systemvariablen des neuen ROMs viel mehr von den ursprünglichen 1K verbrauchen, als sie das bei der 4K ROM Maschine tun.

Bewegtes Display

Wenn Sie ein Programm für das alte ROM in eine Maschine mit neuem ROM eingeben, löschen Sie eine Routine, die bewegte Grafiken ermöglicht, völlig.
Ändern Sie die Zeitziffer, die üblicherweise in der GOSUB-Zeile POKE 16414,n (wobei n die Zeitvariable ist) angegeben wird.
In anderen Programmen ist die Zeit als LET T=n definiert. Wenn Sie PAUSE verwenden, setzen Sie diese auf n (also PAUSE n), um das effektvollste Display zu erhalten. Erinnern Sie sich daran, daß PAUSE 50 das Display für eine Sekunde anhält, PAUSE 25 für eine halbe, PAUSE 100 für zwei Sekunden und so weiter.
Als Regel, welche Ziffer Sie hinter das Wort PAUSE stellen sollen, können Sie sich merken, daß je höher die im Programm für das alte ROM verwendete Zahl ist, (bis zu und inbegriffen 254), desto kürzer ist die Zeit, während der das Display angehalten wird.

Zufallszahlen

Hinsichtlich des Speicherplatzes ist es beim neuen ROM teurer, eine Zufallszahl zu erzeugen, als beim alten.

Hier sind die zwei Versionen:

ALTES ROM	**NEUES ROM**
LET J=RND(6)	LET J=INT (RND*6)+1

Etwas Speicher wird durch die Tatsache gespart, daß INT und RND beim neuen ROM durch einen einzigen Tastendruck eingegeben werden können und so nur je ein Byte besetzen. Trotzdem frißt das Multiplikationszeichen und die Addition (die dafür sorgt, daß Sie nicht ∅ als Teil Ihrer Zufallszahlenreihe bekommen) Speicherplatz auf.
Wenn Sie in einem Programm viele Zufallszahlen innerhalb verschiedener Bereiche generieren müssen, könnte es sich lohnen, eine Subroutine der Sorte LET J= INT (RND*K)+1 einzubauen, und K jedesmal vor dem GOSUB zu belegen.
Natürlich braucht dies länger, als die Zeile öfter im Programm einzubauen, aber wenn die Rechenzeit nicht vorrangig wichtig ist (und das ist kaum der Fall, außer Sie haben bewegte Grafiken), werden Sie bemerken, daß dies Ihnen Zeit beim Programmieren spart und in einigen Programmen — wenn Sie in der Subroutine mehr tun, als nur Zufallszahlen zu generieren — auch Speicherplatz.
PRINT RND gibt Ihnen eine Zufallszahlensequenz zwischen null und eins an, wenn es in einer ewigen Schleife so angeordnet wird:

```
10 PRINT RND;"#";
20 PAUSE 40
30 RUN
```

Diese Sequenz kann sehr nützlich sein. Zum Beispiel kann die Zeile für das alte ROM

IF RND(2)=1 THEN ...

leicht auf die fast identische Zeile des neuen ROM

IF RND=.5 THEN ...

geändert werden. Sie können dies auch verwenden, um Entscheidungen auf statistischen Grundlagen zu treffen.
Wenn Sie also wollen, daß ein bestimmter Programmzweig ungefähr jedes dritte Mal verfolgt wird, können Sie einfach sagen: IF RND=.34 THEN ...

Es gibt eine nette kleine Routine, die die Zeile POKE Y*33+X+1+PEEK (16396)+PEEK (16397)*256,n verwendet, um das Zeichen n beim alten ROM-ZX80 auf die Position Y,X zu setzen (Y ist die Anzahl der Zeilen von oben gezählt, X die Anzahl der Zeichen von der linken Seite des Displays aus. Die Maschinen mit neuem ROM tun dies automatisch, was auch den Vorteil hat, daß mehr als ein Zeichen auf eine Position gesetzt werden kann. Ich werde das erklären:
Beim neuen ROM verwenden Sie PRINT AT in der folgenden Weise: Sie müssen zwei Koordinaten, Y (Zeilen von oben gezählt) und X (Zeichen von links) angeben. Diese werden in einer Zeile wie dieser verwendet, die das Wort »ENDE« ungefähr in die Mitte des Bildschirmes setzt.

```
10 PRINT AT 10,14;"ENDE"
```

Sie trennen die zwei Koordinaten mit einem Komma und setzen einen Strichpunkt nach der zweiten Koordinate, vor die Worte, die Sie ausgedruckt haben möchten.
Die zwei Koordinaten können auch während des Programmablaufes ausgearbeitet werden, also ist die Zeile PRINT AT B,A/3; »ENDE« gültig. Die Funktion PRINT AT führt automatisch die Integerfunktion durch, streicht also alle Kommastellen einer Variablen.
Wenn Sie also etwas an einer bestimmten Stelle ausdrucken wollen, oder das Gefühl haben, daß ein Programm durch die Verwendung von PRINT AT verbessert werden kann (und es kann eine Menge von leeren PRINT-Zeilen oder Schleifen, die Zwischenräume auf den Bildschirm drucken, ersparen), sollten Sie es verwenden. Wenn Sie ein Objekt über den Bildschirm bewegen wollen, müssen Sie einen PAUSE n - Befehl nach den ersten PRINT AT-Befehl setzen und — nach der Pause — ein zweites PRINT AT hinzufügen, das die vorher bedruckten Positionen wieder löscht. Hier ist ein einfaches Beispiel:

```
10 LET A=0
20 LET B=0
30 PRINT AT A,B;"X"
40 PAUSE 30
50 PRINT AT A,B;"#"
60 LET A=A+RND
70 LET B=B+RND
80 IF A>18 THEN LET A=0
```

```
 90 IF B>18 THEN LET B=0
100 GOTO 30
```

Das bewegt ein X zufällig über den Bildschirm (mehr oder weniger diagonal). Wenn Sie mit einem ZX81 oder einem neuen ROM arbeiten, können Sie das lange POKE Y*33 ...usw. weglassen, und durch ein einfaches PRINT AT ersetzen.
Das neue ROM hat auch eine TAB-Funktion, die ein PRINT-Statement an jedem beliebigen Punkt ausdrucken kann, ohne daß Sie eine Schleife mit leeren Zwischenräumen verwenden müssen.

ALTES ROM

```
10 FOR A=1 TO 10
20 PRINT "#";
30 NEXT A
40 PRINT "ENDE"
```

NEUES ROM

```
10 PRINT TAB 10;"ENDE"
```

Beachten Sie, daß nach der Zahl ein Strichpunkt erforderlich ist, bevor die Zeichen ausgedruckt werden. TAB kann durch einen einzigen Tastendruck eingegeben werden.

Grafiken

Auf Geräten mit neuem ROM sind alle Grafiksymbole (darunter inverse Grafiksymbole, inverse Zahlen und Buchstaben, sogar ein inverser Zwischenraum) direkt von der Tastatur einzugeben. Das spart Ihnen die Verwendung von CHR$ (n) (sollten Sie es trotzdem brauchen, wird es Ihnen gefallen, da CHR$ mit einem einzigen Tastendruck eingegeben werden kann).
Inverse Zeichen können verwendet werden, um Programme etwas aufzumotzen; beispielsweise können Anleitungen oder Aufforderungen an den Spieler in inversen Buchstaben ausgegeben werden.
Um andere Programme umzuschreiben, verwenden Sie die folgende Tabelle, in der zuerst die Position des alten und dann des neuen ROM angegeben ist:

Shift Q	..... Grafikzeichen von	5
Shift W	.....	6
Shift E	.....	1
Shift R	.....	2
Shift T	.....	D
Shift A	.....	A
Shift S	.....	T
Shift D	.....	4
Shift F	.....	3
Shift G	.....	S

REM-Statements, TL$

Beim alten ROM ist die erste Adresse nach dem Wort REM 16427. Die äquivalente Adresse des neuen ROMs ist 16514. Sie müssen besonders sorgfältig umrechnen, wenn Sie Programme umschreiben, die auf in REM-Statements gespeicherten Daten aufbauen. Die TL$-Funktion des alten ROM gibt es beim neuen nicht mehr, sodaß Antworten des Benutzers, die aus zwei oder mehr Zeichen bestehen, und die der Computer bearbeitet, indem er Buchstabe für Buchstabe abtrennt, durch Anweisungen ersetzt werden müssen, die es dem Benutzer erlauben, die Information Buchstabe für Buchstabe einzugeben (und dem Computer, damit zu arbeiten). Die String-Arrays des neuen ROM können als eine Art READ/DATA-Funktion verwendet werden. Wenn Sie die Antworten des Benutzers zwei Zeichen lang lassen wollen, ist LET A$=A$ (2 TO) beim neuen ROM das Entsprechende zu LET A$=TL$ (A$).

Als generelle Regel geben Sie die Funktion INT (durch einfachen Tastendruck beim neuen ROM erreichbar) immer vor einer Division ein. Wenn also im Programm für das alte ROM LET F=A/16 steht, sollte die Version für das neue ROM LET F= INT (A/16) sein. Das ist nicht notwendig, wenn Sie das Ergebnis der Rechnung für PRINT AT oder TAB brauchen, da dann die INT-Funktion automatisch durchgeführt wird.

****

Umwandlung der Systemvariablen:

****

ALTES ROM	NEUES ROM
16384	16384
16385	16385
16386	16391
16387	16392
16388	existiert nicht
16389	existiert nicht
16390	16394
16391	16395
16392	16400
16393	16401
16394	16404
16395	16405
16396	16396
16397	16397
16398	existiert nicht
16399	existiert nicht
16400	16412
16401	16413
16402	16418
16403	16419
16404	16420
16405	16408
16406	16409
16407	16427
16408	16428
16409	16429
16410	16432
16411	16433
16412	16434
16413	16435
16414	16436
16415	16437
16416	existiert nicht
16417	existiert nicht
16418	existiert nicht
16419	existiert nicht
16420	existiert nicht
16421	existiert nicht

16422
16423
16424
16425
16426
16427
16428
16429
16430
16431
16432
16433
16434
16435
16436
16437
16438
16439
16440
16441
16442
16443
16444
16445
16446
16447
16448
16449
16450
16451
16452
16453
16454
16455
16456
16457
16458
16459
16460
16406
16407
16509
16510
16513
16514
16515
16516
16517
16518
16519
16520
16521
16522
16523
16524
16525
16526
16527
16528
16529
16530
16531
16532
16533
16534
16535
16536
16537
16538
16539
16540
16541
16542
16543
16544
16545
16546
16547

Diese Tabelle wurde in erster Linie entwickelt, um in REM-Statements POKEn und PEEKen zu können.

Inhalt

Dieses Buch wurde in der Absicht verfaßt, jedem ZX-Fan etwas zu geben. Dem Anfänger wird anhand von unterhaltsamen Programmen jeder Befehl des ZX 81 anschaulich erläutert. Der Fachmann findet eine Vielzahl an interessanten Programmen mit originellen Problemlösungen. Die Programme reichen von einfachen Spielen bis zu komplizierten Maschinencode-Programmen. »ZX-Power« beinhaltet auch ein Kapitel über die Umwandlung von ZX 80 Programmen für den ZX 81. Die englische Computerzeitschrift »Computing Today« schrieb über dieses Buch: »The book has something for everyone«. Auf jeden Fall finden Sie in diesem Buch eine Menge Ideen und Tips, die Sie in eigenen Programmen verwenden können — und sollten, wenn Sie aus Ihrem ZX das Meiste herausholen wollen.

ZX POWER
Cooperation

ISBN 3-88945-000-8

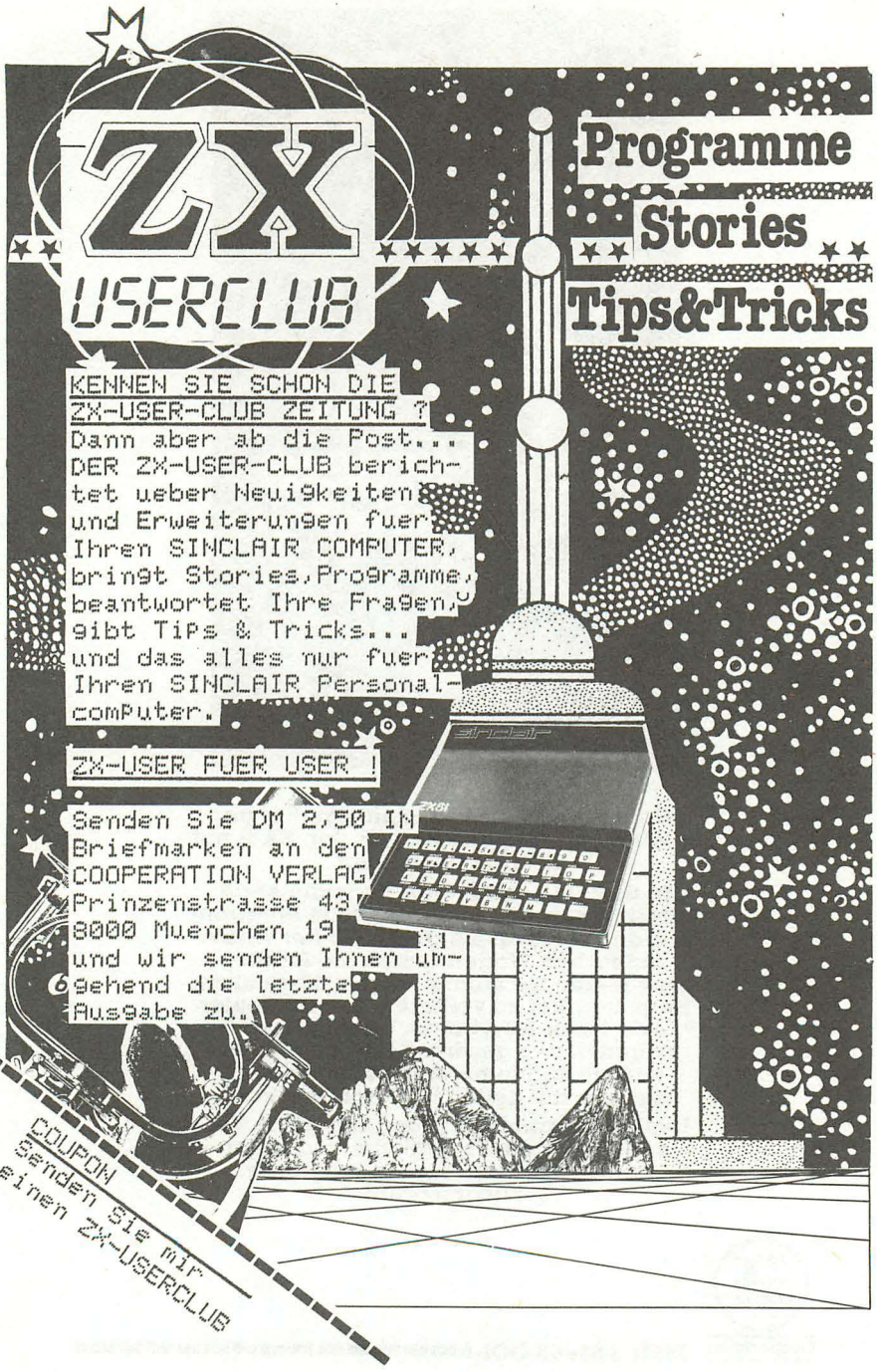
Unglaublich, was Alastair Gourlay an Spielen und Programmen für den ZX 81 mit 1K-RAM präsentiert.

In diesem Buch finden Sie spannende Unterhaltung und viele wertvolle Programmiertips. Lenken Sie geschickt bei einem römischen Wagenrennen die Pferde, spekulieren Sie einmal an der Börse ohne Haus und Hof zu verlieren, der Computer verrät Ihren Biorythmus, oder spielen Sie 17 und 4, wer gewinnt? Und noch viele weitere Spiele nicht nur zum Spielen, sondern mit denen man spielen kann.

Lernen Sie spielend programmieren. Verändern Sie die Programme und schon erhalten Sie Ihr eigenes Programm. Faszinierend!

ZX POWER
Cooperation

ISBN 3-88945-001-6


ZX
USERCLUB
Programme
Stories
Tips&Tricks
KENNEN SIE SCHON DIE ZX-USER-CLUB ZEITUNG ?
Dann aber ab die Post... DER ZX-USER-CLUB berichtet ueber Neui9keiten und Erweiterun9en fuer Ihren SINCLAIR COMPUTER, brin9t Stories, Pro9ramme, beantwortet Ihre Fra9en, 9ibt Tips & Tricks... und das alles nur fuer Ihren SINCLAIR Personalcomputer.
ZX-USER FUER USER !
Senden Sie DM 2,50 IN Briefmarken an den COOPERATION VERLAG Prinzenstrasse 43 8000 Muenchen 19 und wir senden Ihnen umgehend die letzte Aus9abe zu.
COUPON
Senden Sie mir einen ZX-USERCLUB

ZX-POWER

Dieses Buch wurde in der Absicht verfaßt, jedem ZX-Fan etwas zu geben. Dem Anfänger wird anhand von unterhaltsamen Programmen jeder Befehl des ZX 81 anschaulich erläutert. Der Fachmann findet eine Vielzahl an interessanten Programmen mit originellen Problemlösungen. Die Programme reichen von einfachen Spielen bis zu komplizierten Maschinencode-Programmen. »ZX-Power« beinhaltet auch ein Kapitel über die Umwandlung von ZX 80 Programmen für den ZX 81. Die englische Computerzeitschrift »Computing Today« schrieb über dieses Buch: »The book has something for everyone«. Auf jeden Fall finden Sie in diesem Buch eine Menge Ideen und Tips, die Sie in eigenen Programmen verwenden können — und sollten, wenn Sie aus Ihrem ZX das Meiste herausholen wollen.

Cooperation

ISBN 3-88945-000-8