

Adventure-Programmiertechnik

Es ist soweit. Im nun folgenden Abschnitt erfahren Sie endlich, wie man sie programmiert: Super-Adventures, wie es sie bis jetzt noch nicht gibt. Sie erhalten alle notwendigen Grundlagen, um professionelle Adventures zu erstellen.

In meinem ersten Adventure-Kurs habe ich bereits ausführlich beschrieben, wie man eine Adventure-Idee aufgreift, diese ausarbeitet, Pläne anfertigt etc. Wir wollen uns deshalb hier gar nicht mehr erst lange mit solchen Grundlagen aufhalten, sondern gleich zur Sache kommen. Auf den nächsten Seiten programmieren wir gemeinsam die ersten Fortsetzungsszenen von Gordon Saga II. An dieser Stelle muß ich auch gleich noch einmal ein paar Worte zur Gordon Saga I verlieren. Ich möchte mich bei allen begeisterten Adventurern bedanken, die im Spiel mit mühevoller Kleinstarbeit meine Telefonnummer herausgefunden und mich dann angerufen haben. Ich erhielt zahlreiche Anrufe aus allen Teilen Deutschlands, Österreichs und der Schweiz. Bedanken möchte ich mich vor allem deshalb, weil ich bisher keinen einzigen bösen Anruf erhielt. Einige teilten mir sogar mit, daß das Spiel Fehler enthält und deshalb nicht lösbar sei. Es stimmt, es sind einige Fehler enthalten, aber ich kann versichern, daß diese Fehler in keinsten Weise zur Unlösbarkeit beziehungsweise zum Absturz des Spiels führen: es sind lediglich kleine Logikfehler, wie sie bei der Programmierung eines komplexen Adventures unvermeidbar sind. An dieser Stelle sei auch die »Incredible Four« begrüßt, die Gordon Saga I geknackt und beim Titelbild das HAPPY-SOFTWARE-Emblem durch UNHAPPY SOFTWARE ersetzt haben (PS: Wenn ich euch erwische, dreh ich euch den Hals um!). »Wo bleibt Gordon Saga, Teil II« haben mich viele gefragt. Die Programmierung von Teil II habe ich kurz vor Abschluß abgebrochen. Dies hat seinen guten Grund: Teil II ist einfach um Längen besser als Teil I in puncto Text, Grafik und Parser. Wenn jemand, der Gordon Saga I kennt, von Gordon Saga II hört, so vergleicht er unwillkürlich den zweiten mit dem ersten Teil – und dies wäre nicht gerade die beste Werbung.

Als kleinen Vorgeschmack möchte ich gemeinsam mit Ihnen nun die ersten Szenen von Gordon Saga II programmieren. Dazu ist notwendig, daß Sie wissen, wie Gordon Saga I ausgeht. Am Spielende treibt der Spieler mit einem Floß auf einem stillen unterirdischen See in einem geheimnisvollen Gebirge. Plötzlich entsteht im See ein riesiger Strudel, der den Spieler samt Floß in die Tiefe reißt – die Pforte zu einer geheimnisvollen Welt hat sich aufgetan ...

Und dies sind die ersten Szenen von Gordon Saga II: Der Spieler wacht in einer Glaskuppel am Grunde des Sees wieder auf. Von der Glaskuppel aus führt ein Tunnel in die Dunkelheit. Durch diesen Tunnel gelangt man in eine Kristallgrotte ...

Den Lageplan für diesen Teil des Spieles zeigt Bild 1.

Wir haben 4 Räume: Glaskuppel, Tunnel, Kristallgrotte und Felsplateau. Außerdem ist bei jedem Raum aufgezählt, welche Objekte sich in ihm befinden. Von Raum 3 gelangt man nicht so ohne weiteres zu Raum 4, was durch die gestrichelte Linie angedeutet ist – man braucht irgendwie das Schwert.

Wie fängt man jetzt an zu programmieren?

Nun, man muß sich erst einmal überlegen, wie man die einzelnen Programmteile – also Basic-Programm, Zeichensatzlader, Grafikbilder, relative Dateien etc. auf einer Disk unterbringt. Am besten, man entscheidet sich, eine Diskette beidseitig zu verwenden.

Auf Seite 1 bringt man das Grundprogramm, den deutschen Zeichensatz, eventuell Maschinenprogramme und Grafikbilder unter.

Das heißt, außer den Grafikbildern kommt auf Seite 1 alles, was einmal in den Rechner geladen wird und dann dort bleibt. Auf Seite 2 kommt die relative Datei, auf der alle Texte, Kommentare und Raumbeschreibungen gespeichert sind. Außerdem wird auf Seite 2 die relative Wortschatzdatei untergebracht. In der Entwicklungsphase eines Adventures eignet sich Parser 4.0 zweifellos am besten – man kann leicht neue Wörter einfügen etc. Erst ganz am Ende, wenn alle Spielszenen fertig programmiert sind – das Adventure also fertig ist –, sollte man endgültig auf Parser 5.0 umrüsten.

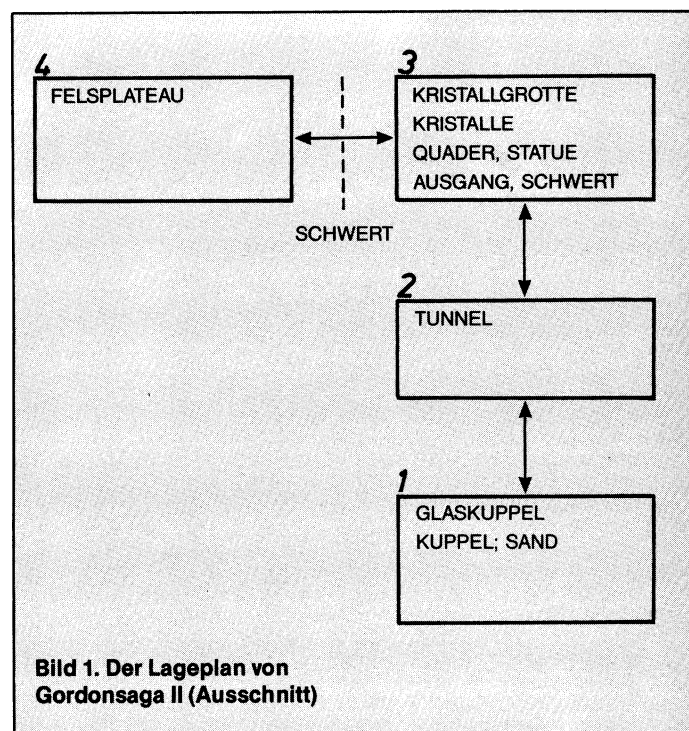
Nehmen Sie sich jetzt bitte eine gute Diskette her, die für beidseitiges Bespielen geeignet ist, und formatieren Sie beide Seiten. Dann nehmen Sie die Disk-Utility her und ändern die IDs der Disketten so, daß der ID der ersten Seite »1side«, der der zweiten Seite »2side« lautet. Dies ist notwendig, damit später innerhalb des Programmes festgestellt werden kann, ob die richtige Disk-Seite eingelegt ist. Jetzt brauchen wir den Relativ-Datei-Editor. Mit ihm eröffnen wir auf der 2. Diskseite eine relative Datei mit folgenden Angaben:

Satzlänge 41

Anzahl Sätze 3000

Filename »TEXTE«

Jetzt müssen Sie auf die zweite Diskseite noch die relative Wortschatzdatei überspielen. Dies geht nicht so ohne weiteres, da es nur sehr wenige Kopierprogramme gibt, die relative Files einzeln kopieren können. Sie können jedoch mit dem Editor eine neue Wortschatzdatei anlegen und dann mittels VERSETZEN-Befehl (siehe auch Anleitung zum Editor) die alte Wortschatzdatei in die neu angelegte übertragen. Eine andere Möglichkeit besteht darin, einfach die ganze Diskettenseite, auf der sich die Wortschatzdatei befindet, mit einem Diskettenkopierprogramm auf die zweite Seite unserer jetzigen Arbeitsdiskette zu kopieren und dann mittels Scratch



und Validate den Rest zu löschen. Diskseite 2 muß jetzt so aussehen:

```
0 "gordon saga ii" 2side
492 "texte" rel
32 "wortschatz" rel
140 blocks free.
```

Jetzt können wir mit der Arbeit beginnen. Der erste Schritt besteht darin, den Wortschatz und die Wörter zu erweitern, die für die aktuelle Spielszene, die sich in der Regel aus einigen Räumen zusammensetzt, benötigt werden. Man sollte nie zu viele Räume auf einmal behandeln.

Wortschatz erweitern

Denken Sie sich immer eine kleine Szene aus, programmieren Sie diese und arbeiten Sie erst dann weiter, wenn diese Szene einwandfrei läuft. Durch dieses Szene-für-Szene-Programmieren hält sich die Fehlersuche in engen Grenzen, und man kann sicher sein, daß das Adventure bei der Fertigstellung nur wenige Logikfehler enthält – da jede Szene für sich ausgiebig getestet wurde. Wie erweitert man den Wortschatz? Ganz einfach. Für unsere jetzige Szene müssen wir den Wortschatz um folgende Objekte ergänzen:

Glaskuppel	1
Sand	2
Tunnel	3
Kristallgrotte	4
Quader	5
Statue	6
Schwert	7
Zettel	8
Amulett	9
Grube	10
Oeffnung	11
Felsplateau	12
Kristalle	13

Der Wortschatz wird ganz einfach erweitert, indem Sie mit dem relativen Datei-Editor die neuen Wörter ans Ende des bisherigen Wortschatzes anhängen (ab Satz Nr. 198) und dann die gesamte Datei neu sortieren. Hängen Sie bitte folgende Sätze an:

198	glaskuppel	3001
199	sand	3002
200	tunnel	3003
201	kristallgrotte	3004
202	quader	3005
203	statue	3006
204	schwert	3007
205	zettel	3008
206	amulett	3009
207	grube	3010
208	oeffnung	3011
209	felsplateau	3012
210	kristalle	3013

Jetzt muß der Wortschatz neu sortiert werden. Beachten Sie bitte, daß mit der Sortierroutine des relativen Datei-Editors nur Dateien sortiert werden können, die nicht mehr als 300 Sätze haben. Übersteigt die Anzahl der Worte im Wortschatz diese Zahl, so müssen Sie zum Sortieren des Wortschatzes den Super-Sorter hernehmen, bei dem kein Limit besteht.

Nachdem der Wortschatz erweitert ist, können wir uns dem Text des Adventures zuwenden. Textadventures zeichnen

sich besonders durch lange, beschreibende Texte aus. Nicht umsonst arbeiten viele Softwarehäuser bei der Erstellung ihrer Adventures mit Schriftstellern zusammen. Alleine Adventures schreiben ist langweilig. Auch sind Programmierer meistens keine guten Texter. Vielleicht tun Sie sich hier am besten mit einem Freund zusammen, der gerne schreibt. Zu zweit ein Spiel aushecken macht schließlich auch viel mehr Spaß.

Anschließend finden Sie den Text zu den ersten Szenen von Gordon Saga II. Bitte schreiben Sie ihn in die entsprechenden Sätze der Texte-Datei:

```
1 Als Sie aus Ihrer Bewußlosigkeit
2 erwachen, finden Sie sich auf einem
3 weichen weißen Sandboden liegend
4 wieder.
5 Augenblicklich erinnern Sie sich an die
6 letzten Ereignisse vor Ihrer
7 Bewußtlosigkeit .. In Ihren Gedanken
8 setzt sich das Abbild der riesigen
9 Seeschlange zusammen, die Sie mit Hilfe
10 einer goldenen Armbrust erlegt hatten.
11 Ihre folgende Erinnerung gilt dem
12 plötzlich aufgekommenen Strudel, durch
13 den Ihr Floß samt Ihnen in die Tiefe
14 gerissen worden war.
15 Obwohl es Ihnen noch immer unbegreiflich
16 ist, wie Sie all dies überleben konnten,
17 machen Sie sich daran, Ihre jetzige
18 Umgebung näher zu untersuchen ....
19
20 GORDON SAGA II. Die Rückkehr des Helden
21 Ein Abenteuerspiel von Michael Nickles.
22 Parsersystem 4.0 - Copyright 1985
23 Auszug für Kurs im 64er-Sonderheft. *
24 Glaskuppel
25 Sie befinden sich innerhalb einer
26 Glaskuppel am Grunde eines Sees.
27 Außerhalb der Kuppel ist das Wasser so
28 dunkel, daß man nichts erkennen kann.
29 Der weiße Sand am Boden strahlt ein
30 schwaches Licht aus, das die Kuppel
31 erhellt. Ein kleiner Tunnel führt von
32 hier aus hinaus in die endlose
33 Dunkelheit. *
34 Tunnel
35 Sie befinden sich in einem zwei Meter
36 hohen und etwa drei Meter breiten
37 Tunnel. Die Wände sind aus einem Glas
38 angefertigt, das etwas trüber als das
39 Glas, aus dem die Kuppel besteht, ist.
40 Der Tunnel verläuft in nördliche
41 Richtung. Weit entfernt in nördlicher
42 Richtung sehen Sie ein helles Licht. *
43 Kristallgrotte.
44 Dies ist eine riesige Grotte, deren Wände
45 mit Kristallen, die in allen Farben des
46 Lichtspektrums glitzern, besetzt sind.
47 Der Boden besteht aus glatten, strahlend
48 weißen Marmorplatten. Trotz zahlreicher
49 Unebenheiten ähnelt diese Grotte dem
50 Inneren einer Halbkugel mit einem Radius
51 von ca. 100 Metern. Der einzige Ausgang
52 von hier ist ein in südlicher Richtung
53 verlaufender Tunnel. *
54 Felsplateau
55 Dieses Felsplateau außerhalb der Grotte
56 bietet Ihnen eine fantastische Aussicht.
```

57 Sie blicken über ein weites Land hinweg.
 58 das in Dunkelheit gehüllt ist. Am Himmel
 59 sehen Sie eine schwarze Scheibe, deren
 60 Ränder orangerot leuchten - es scheint,
 61 als sei die Dunkelheit durch eine
 62 andauernde Sonnenfinsternis bedingt.
 63 Etwa 300 Meter unterhalb des Plateaus
 64 befindet sich ein Wald mit einer Ihnen
 65 unbekannten Flora.
 66 All diese eindrucksvollen Anblicke
 67 verstärken in Ihnen das Gefühl, daß Sie
 68 sich in einer fremden Welt befinden.
 69 In der östlichen Gebirgswand ist ein
 70 Zugang zu einer Grotte. *

71 Sie graben in dem leuchtenden Sand. *

72 Es ist sinnlos, noch tiefer zu graben. *

73 Nachdem Sie ein etwa 50 cm tiefes Loch
 74 in den Sand gegraben haben, finden Sie
 75 einen Zettel und ein Amulett. *

76 Im Sandboden ist eine ca. 50 cm tiefe
 77 Grube ausgehoben - es scheint so, als ob
 78 hier vor kurzem jemand gegraben hat! *

79 Auf dem Zettel steht etwas geschrieben! *

80 "Supermike grüßt Captain Future,
 81 Section 8, Pink, Brutallica, Gommel,
 82 GCS (ich weiß, daß ihr genau 35 km
 83 Luftlinie von mir entfernt wohnt -
 84 Paßt bloß auf!!!), und natürlich alle
 85 Freunde, RAMBO und alle Girls, die mich
 86 besonders gut kennen!!" *

87 Das Amulett ist aus massivem Gold
 88 angefertigt, mit schwarzen Edelsteinen
 89 besetzt und hängt an einer goldenen
 90 Halskette. *

91 Die Grube ist völlig leer. *

92 Die Glaskuppel ist so groß, daß sich
 93 etwa 10 ausgewachsene Männer darin
 94 aufhalten können. *

95 Der Sand ist sehr grobkörnig und
 96 leuchtet seltsamerweise. *

97 Die Kristalle glitzern in allen Farben
 98 des Lichtspektrums. *

99 Die Kristalle sind fest in die Wände
 100 eingelassen und lassen sich auch mit
 101 größten Anstrengungen nicht entfernen. *

102 Auf dem Marmorboden, exakt im Zentrum
 103 des Raumes, steht ein goldener Quader,
 104 von dem eine ca. 3 Meter hohe Statue
 105 emporragt. *

106 Nur wenige Meter von der Statue entfernt
 107 steckt ein goldenes Schwert zu etwa
 108 einem Drittel seiner Gesamtlänge im
 109 Marmorboden. *

110 Das Schwert ist fest im Marmorboden
 111 verankert und läßt sich um keinen
 112 Millimeter herausziehen. *

113 Die Statue gibt die groben Umrisse eines
 114 jungen Kriegers wieder. Er hat lange
 115 Haare, ist von athletischer Körperstatur
 116 und trägt eine Rüstung. Sie fühlen sich
 117 von der Statue in einer seltsamen Art
 118 und Weise angezogen. *

119 Als Sie die Statue mit Ihrer rechten
 120 Hand berühren, durchfährt augenblicklich
 121 eine Lähmung Ihren ganzen Körper und Sie
 122 sind nicht mehr in der Lage, Ihren Arm
 123 zurückzuziehen. Unmittelbar vor Ihnen

124 Augen beginnt die Statue plötzlich ihr
 125 Äußeres zu verändern.....

126 Die Lähmung läßt schließlich nach und
 127 Sie stürzen benommen zu Boden.

128 Nachdem Sie den Schock überwunden haben,
 129 stellen Sie fest, daß nicht nur die
 130 Statue ihr Äußeres verändert hat -
 131 während die Statue Ihre Form angenommen
 132 hat, haben Sie den Körper des jungen
 133 Kriegers erhalten, den die Statue noch
 134 vor kurzem verkörpert hat. *

135 Die Statue gibt jetzt die Umrisse und
 136 Gesichtszüge eines Ihnen wohlvertrauten
 137 Menschenkörpers wieder - ein Körper, der
 138 noch vor kurzem der ihrige war! *

139 Mit der unglaublichen Kraft, die in
 140 Ihrem neuen Körper zu stecken scheint,
 141 ziehen Sie das Schwert ohne jegliche
 142 Anstrengung aus dem Marmorboden.
 143 Kaum, daß Sie es in Ihren Händen halten,
 144 beginnt der gesamte Raum um Sie herum
 145 leicht zu beben und das Schwert strahlt
 146 in gleißendem Licht.

147 Aus dem Nichts ertönt plötzlich eine
 148 Ihnen altbekannte Stimme. "In Dir
 149 erfüllt sich die alte Prophezeiung!
 150 Geh, GORDON, die Macht ist mit Dir!"

151 Als die letzten Worte des Uralten
 152 gerade verklingen, tut sich in der
 153 westlichen Wand der Grotte eine Öffnung
 154 auf und gibt den Ausgang zu einem
 155 Felsplateau frei.

156 Sie stehen verwundert da und fragen sich
 157 immer wieder, warum gerade Sie der
 158 Auserwählte sind und zu welchem Zweck
 159 der Uralte Sie in diese fremde Welt
 160 geholt hat..... *

161 In der westlichen Wand der Grotte ist
 162 eine Öffnung, die den Ausgang zu einem
 163 Felsplateau ermöglicht. *

164 die Glaskuppel
 165 der Sand
 166 der Tunnel
 167 die Kristallgrotte
 168 der Quader
 169 die Statue
 170 das Schwert
 171 der Zettel
 172 das Amulett
 173 die Grube
 174 die Öffnung
 175 das Felsplateau
 176 die Kristalle

Sie haben sicherlich bereits bemerkt, wie Adventuretexte in einer relativen Datei untergebracht werden: Jede Raumbeschreibung oder jeder Kommentar wird mit einem Sternchen abgeschlossen. Nur die Objektnamen selbst stehen allein in einer Zeile, den Geschlechtsartikel vorangestellt.

Damit ist auch der Text für die ersten Szenen auf Disk vorhanden. Jetzt benötigen wir ein Programmiergerüst, um das herum wir unser Adventure aufbauen. Dieses Gerüst finden Sie in Listing 1.

Dieses Gerüst soll Ihnen als Grundprogramm für jedes Adventure dienen. In ihm sind nahezu alle Routinen enthalten, die für die Programmierung beziehungsweise Entwicklung eines Textadventures notwendig sind.

```

0 REM ***** <109>
1 REM GORDON SAGA II - 1985 BY SM <188>
2 REM ***** <111>
3 IF I=0 THEN I=1:LOAD"D.Z.SATZ",8,1 <076>
4 POKE 648,204:POKE 53272,56:POKE 56576,0: <144>
  PRINT"<CLR>":REM D.T.SATZ AKTIVIEREN
5 OPEN 15,8,15:PRINT"<CLR,YELLOW>";CHR$(14 <204>
  );
6 SIDE=2:GOSUB 53200:PRINT"<CLR>"; <081>
15 GOSUB 60000: REM TABELLEN EINLESEN <100>
20 DIM WI(100): REM ZIMMER-MERKER <253>
30 SC=0:MO=0 : REM SCORE/VERSUCH <065>
40 SA=1:GOSUB 53600 <004>
  : REM TITELTEXT
50 ZN=1:GOSUB 54000 <067>
  : REM BESCHREIB. 1.RAUM
100 REM START ----- <206>
----- <077>
101 :
110 IF SB<>0 THEN SA=SB:SB=0:GOSUB 53600:R <133>
  EM KOMMENTAR AUSGEBEN
120 GOSUB 50000 :REM BEFEHLSEING <150>
  ABE <106>
130 : <218>
300 REM GEH-ROUTINE ----- <021>
310 IF RI=0 THEN 400 <017>
320 I=MAP(ZN,RI)
330 IF I=0 THEN PRINT"DIESE RICHTUNG IST N <119>
  ICHT MÖGLICH!":GOTO 400 <186>
340 ZN=I:GOSUB 54000 <129>
400 REM SCHAU/UNTERSUCHE ----- <034>
410 IF VE<>11 THEN 500 <137>
415 IF O1=0 THEN GOSUB 54000:GOTO 500
420 SA=OS(O1):GOSUB 53400:GOSUB 53500:OB$= <154>
  "J"+RIGHT$(I$,LEN(I$)-1) <144>
425 IF OB(O1)=-1 THEN 430
430 IF OB(O1)<>ZN THEN X$=OB$+" IST HIER N <023>
  ICHT ZU SEHEN!":GOSUB 53900:GOTO 500
440 IF OU(O1)=0 THEN X$=OB$+" HAT KEINE BE <127>
  SONDEREN MERKMALE.":GOSUB 53900
450 IF OU(O1) THEN SA=OU(O1):GOSUB 53800 <153>
500 REM NIMM ----- <042>
505 IF VE<>12 THEN 600 <134>
506 IF SP=1 THEN 550:REM NIMM ALLES <071>
507 SA=OS(O1):GOSUB 53400:GOSUB 53500:OB$= <241>
  "J"+RIGHT$(I$,LEN(I$)-1)
510 IF O1=0 THEN PRINT"DHR SATZ ENTHÄLT KE <250>
  IN OBJEKT !":GOTO 600
514 : IF OB(O1)=-1 THEN PRINT"DAS HABEN SIE <095>
  BEREITS !":GOTO 600
515 : IF OB(O1)<>ZN THEN X$=OB$+" IST HIER <245>
  NICHT ZU SEHEN!":GOSUB 53900:GOTO 600
516 IF OA(O1)=0 THEN 600 <052>
520 : IF OA(O1)=1 THEN PRINT"BIZARR!":GOTO <065>
  600
525 : IF OA(O1)=2 THEN X$=OB$+" IST VIEL ZU <121>
  SCHWER ZUM TRAGEN!":GOSUB 53900:GOTO
  600
530 : IF OA(O1)=3 THEN PRINT"GENOMMEN.":OB( <035>
  O1)=-1:GOTO 600 <018>
550 : <149>
554 PRINT:FOR I=1 TO OZ <246>
555 : IF OB(I)<>ZN THEN 580
556 SA=OS(I):GOSUB 53400:GOSUB 53500:OB$= <063>
  "J"+RIGHT$(I$,LEN(I$)-1) <230>
560 : I$=OB$+" : <106>
565 IF SQ=1 AND I=O1 THEN 580:REM AUSSER <133>
570 : IF OA(I)=1 THEN PRINT I$;"BIZARR !"
575 : IF OA(I)=2 THEN PRINT I$;"ZU SCHWER Z <131>
  UM TRAGEN !"
577 : IF OA(I)=3 THEN PRINT I$;"GENOMMEN.": <206>
  OB(I)=-1 <156>
580 NEXT I <178>
600 REM INVENTUR ----- <237>
601 IF VE<>13 THEN 700
605 PRINT"<DOWN>SIE HABEN: {SPACE,UP}" <244>
610 IC=0:FOR I=1 TO OZ <250>
611 SA=OS(I):GOSUB 53400:GOSUB 53500:OB$= <194>
  "J"+RIGHT$(I$,LEN(I$)-1):I$=OB$
612 : IF LEFT$(I$,3)="DER" THEN J$=RIGHT$(I$ <143>
  ,LEN(I$)-3):I$="DEN"+J$
615 : IF OB(I)=-1 THEN PRINT"<11RIGHT>";I$: <139>
  IC=1
620 NEXT I:IF IC=0 THEN PRINT"<11RIGHT>BOC <155>
  K AUF BIER !"
700 REM VERLIERE ----- <100>
701 IF VE<>14 THEN 800 <086>
702 IF SP=1 THEN 750:REM VERLIER ALLES <156>
703 IF O1=0 THEN PRINT"DHR SATZ ENTHÄLT KE <062>
  IN OBJEKT!":GOTO 800
705 IF OB(O1)<>-1 THEN PRINT"<DOWN>SIE HAB <063>
  EN DAS NICHT !":GOTO 800
715 OB(O1)=ZN:PRINT"OK.":GOTO 800 <137>
750 IC=0:FOR I=1 TO OZ <134>
755 : IF SQ=1 AND I=O1 THEN 780:REM AUSSER <198>
760 SA=OS(I):GOSUB 53400:GOSUB 53500:OB$= <115>
  "J"+RIGHT$(I$,LEN(I$)-1):I$=OB$+" :
770 : IF OB(I)=-1 THEN OB(I)=ZN:PRINT I$;"O <078>
  K.":IC=1
780 NEXT I:IF IC=0 AND SQ=1 THEN PRINT"DIE <053>
  SOLL DAS GEHEN ?"
782 IF IC=0 AND SQ=0 THEN PRINT"SIE HABEN <175>
  DOCH GARNICHTS !"
800 : <119>
2000 REM RAUMSPEZIFISCHE ACTION ----- <056>
----- <227>
50000 REM ***** <009>
50001 REM * <169>
50002 REM * WORT-PARSER 4.0 * <011>
50003 REM * <086>
50004 REM * <C> 1986 BEI * <013>
50005 REM * * <082>
50006 REM * MICHAEL NICKLES * <015>
50007 REM * * <235>
50008 REM *****
50010 REM BEFEHLSSATZEINGABE ----- <069>
-----
50011 ZW=0:PRINT:GOSUB 53000:MO=MO+1:REM T <222>
  ITELLINE AUSGEBEN
50012 IF UD>0 THEN 50500:REM UND <171>
50015 SL=77:REM BEFEHLSATZLAENGE <048>
50020 BE$="":POKE 198,0:PRINT"<YELLOW>>"; <026>
50030 GET X$:IF PEEK(203)=1 THEN 50120 <226>
50040 IF X$="" THEN 50030 <171>
50050 IF LEN(BE$)=0 AND ASC(X$)=20 THEN 50 <148>
  030
50060 I=ASC(X$):IF I<32 OR I>133 AND I<159 <131>
  THEN IF I<>20 THEN 50030
50070 IF LEN(BE$)=SL AND I<>20 THEN 50030 <128>
50080 BE$=BE$+X$ <031>
50090 PRINT CHR$(20);X$;" "; <108>
50100 IF I=20 THEN BE$=LEFT$(BE$,LEN(BE$)- <101>
  2):GOTO 50030
50110 GOTO 50030 <001>
50120 PRINT"<LEFT,SPACE>" <227>
50400 REM SATZZEICHEN SUCHEN -----
----- <103>
50401 : <082>
50415 SN=0 <104>
50420 IF RIGHT$(BE$,1)=" " THEN BE$=LEFT$(B <233>
  E$,LEN(BE$)-1):GOTO 50420
50430 I$=RIGHT$(BE$,1) <180>
50440 IF I$=","OR I$="!" THEN SN=0:BE$=LEFT <010>
  $(BE$,LEN(BE$)-1):GOTO 50420
50450 IF I$="?" THEN SN=1:BE$=LEFT$(BE$,LEN <144>
  (BE$)-1):GOTO 50420
50500 REM BE$-ZEIGER SETZEN -----
----- <199>
50501 : <184>
50502 IF UD=0 THEN VE=0:RI=0:O1=0:O2=0:SP= <157>
  0:SQ=0
50504 IF UD>0 THEN RI=0:GOTO 50530 <199>
50505 IF LEFT$(BE$,1)=" " THEN BE$=RIGHT$(B <132>
  E$,LEN(BE$)-1):GOTO 50505
50510 BE$=BE$+" " <232>
50520 ZA=1:REM ZEIGER A SETZEN <058>
50530 ZB=ZA+1 <180>
50540 IF MID$(BE$,ZB,1)=" " THEN UD=0:CLOSE <170>
  1:OPEN 1,8,3,"TEXTE":RETURN
50550 IF MID$(BE$,ZB,1)<>" " THEN ZB=ZB+1:G <175>
  OTO 50550
50590 SU$=MID$(BE$,ZA,ZB-ZA) <222>

```

Listing 1. Ein vollständiges »Adventure-Programmier-Gerüst«

50600 REM SU\$ VORBEHANDLUNG -----

```

--<130>
50601 :<028>
50610 IF SU$="UND" THEN UD=1:GOTO 50630<093>
50620 IF SU$="," THEN UD=2:GOTO 50630<061>
50625 GOTO 50670<201>
50630 ZA=ZB+1<080>
50640 IF MID$(BE$,ZA,1)=" " THEN PRINT "UND W<153>
AS ?":GOTO 50530
50650 IF MID$(BE$,ZA,1)=" " THEN ZA=ZA+1:GO<242>
TO 50650
50660 CLOSE 1:OPEN 1,8,3,"TEXTE":RETURN<119>
50670 IF SU$="IHN"OR SU$="SIE"OR SU$="ES" T<069>
HEN O1=OM:GOTO 51300
50680 IF SU$="ALLES" THEN SP=1:GOTO 51300<029>
50690 IF SU$="AUSSER" THEN SQ=1:GOTO 51300<012>
51000 REM BINAERE SUCHROUTINE -----

```

```

--<113>
51001 :<176>
51002 CLOSE 1:OPEN 1,8,3,"WORTSCHATZ"<031>
51007 REM ANFANG UND ENDE DES SUCHBEREICHE<117>
S ERMITTELN
51008 I=ASC(LEFT$(SU$,1))<176>
51009 IF I-65<0 OR I-65>25 THEN GOSUB 5150<134>
0:GOTO 50000
51010 AN=IN(I-65):REM ANFANG DES SUCHBEREI<164>
CHES
51012 IZ=64<200>
51013 IF IN(I-IZ)=0 THEN IZ=IZ-1:GOTO 5101<086>
3
51015 EN=IN(I-IZ)-1:REM ENDE DES SUCHBEREI<070>
CHES
51016 IF AN=0 THEN GOSUB 51500:UD=0:GOTO 5<249>
0000
51020 SZ=197:N=INT( (LOG(EN-AN+1)/LOG(2))+1<121>
:REM MAXIMAL-FORMEL
51030 SA=AN-1+(2*N)/2:REM MITTE DER GES.DA<073>
TA
51040 GOSUB 52100:REM A$ LESEN<234>
51050 N=N-1 :REM 1.ABFRAGE<045>
51060 :<235>
51070 REM SU$ MIT A$ VERGLEICHEN -----<042>
51080 : IF LEN(SU$)<3 THEN 51110<140>
51090 : IF WA=1 AND SU$=LEFT$(A$,LEN(SU$))<157>
THEN VE=WC:GOTO 51300
51100 : IF WA<>3 OR SU$<>RIGHT$(A$,LEN(SU$<161>
)) THEN 51110
51102 : IF UD=1 OR UD=2 THEN UD=3:O1=0:O2<042>
=0
51104 : IF O1=0 THEN O1=WC:OM=O1:GOTO 51<133>
300
51105 : IF O2=0 THEN O2=WC:OM=O1:GOTO 51<170>
300
51110 : IF WA=2 AND SU$=A$<140>
THEN RI=WC:GOTO 51300
51115 : IF WA=5 AND SU$=A$<247>
THEN AD=WC:GOTO 51300
51120 : IF WA=1 AND SU$=A$<017>
THEN VE=WC:GOTO 51300
51125 : IF WA=4 AND SU$=A$<204>
THEN 51300
51130 : IF SU$<A$ THEN SA=SA-(2*(N-1)):GOSU<015>
B 52100:GOTO 51160
51140 : IF SA+(2*(N-1))>SA+EN-AN AND N>-1<204>
THEN N=N-1:GOTO 51140:REM "UEBERLAU
F"
51150 : SA=SA+(2*(N-1)):GOSUB 52100:REM " <141>
HALBIEREN"
51160 :<079>
51170 :N=N-1 :REM NAECHST. HALBIEREN<068>
51180 : IF N<0 THEN GOSUB 51500:UD=0:GOTO 5<201>
0000
51190 REM PRINT "SA=";SA;" N=";N<216>
:REM UNNOETIG !!
51200 :GOTO 51060:REM NAECHST. SUCH-VERS.<192>
51300 REM NAECHSTES WORT<027>
51310 ZA=ZB+1<254>
51320 IF MID$(BE$,ZA,1)=" " THEN 50530<128>
51330 IF MID$(BE$,ZA,1)=" " THEN ZA=ZA+1:GO<207>
TO 51330
51350 GOTO 50530<131>

```

51500 REM ICH KENNE SU\$ NICHT -----

```

--<232>
51501 :<168>
51510 I=INT(3*RND(1))+1<080>
51520 ON I GOTO 51530,51540,51550<173>
51530 X$=" "+ICH KENNE DAS WORT " "+SU$+" <011>
NICHT.":GOTO 52000
51540 X$=" "+DAS WORT " "+SU$+" IST MIR NI<122>
CHT BEKANNT.":GOTO 52000
51550 X$=" "+DEUTSCH IST LEIDER NUR MEINE<167>
ZWEITE SPRACHE. ICH KENNE DAS "
51555 X$=X$+" WORT " "+SU$+" NICHT.":GOTO 5<158>
2000
52000 REM FORMATIERTE TEXTAUSGABE -----
--<120>
52010 IF X$="" THEN 52090<213>
52020 IF LEN(X$)<40 THEN PRINT X$:GOTO 520<248>
90
52030 IF LEN(X$)=40 THEN PRINT X$;:GOTO 52<148>
090
52040 I$=LEFT$(X$,41):IC=0:FOR I=1 TO LEN(<242>
I$):IF MID$(I$,I,1)=" " THEN IC=I
52050 NEXT I:IF IC=0 THEN PRINT X$:GOTO 52<064>
090
52060 I$=LEFT$(I$,IC-1):IF LEN(I$)=40 THEN<046>
PRINT I$;
52070 IF LEN(I$)<40 THEN PRINT I$<116>
52080 X$=RIGHT$(X$,LEN(X$)-IC):GOTO 52010<206>
52090 RETURN<075>
52100 REM SATZ-TEILBEREICH LESEN -----
--<254>
52101 :<004>
52110 HB%=SA/256:LB%=SA-HB%*256<241>
52112 PRINT#15,"P";CHR$(3);CHR$(LB%);CHR$(<170>
HB%);CHR$(1)
52114 INPUT#15,A:IF A=50 THEN PRINT "SATZNU<245>
MMER ZU HOCH"
52120 IF A=50 THEN RETURN<165>
52130 INPUT#1,A$<199>
52140 WA=VAL(MID$(A$,22,1))<030>
52150 WC=VAL(MID$(A$,23,3))<089>
52160 A$=LEFT$(A$,21)<202>
52170 IF RIGHT$(A$,1)="(RIGHT)" THEN A$=LEF<154>
T$(A$,LEN(A$)-1):GOTO 52170
52180 RETURN<165>
53000 REM UP - TITELLINE -----
--<170>
53001 :<144>
53010 I1$="(37SPACE)";T$=""<253>
53020 T$=T$+CHR$(18)+R$:I$=STR$(SC)+" / "+RI<159>
GHT$(STR$(MO),LEN(STR$(MO))-1)
53030 T$=T$+RIGHT$(I1$,41-LEN(T$)-LEN(I$))<240>
+I$+CHR$(146):PRINT "HOME";T$;
53040 PRINT "{22DOWN}";:RETURN<073>
53200 REM UP - DISK EINLEGEN -----
--<023>
53201 :<088>
53205 IF IC=21 THEN GOSUB 53700:REM WEITER<136>
53210 PRINT "(RVSON) MITTE DISK-SEITE";SIDE;<119>
"(LEFT,SPACE)EINLEGEN ! (RVOFF)";:POK
E 198,0:WAIT 198,1
53220 PRINT "{29LEFT,29SPACE}";<121>
53221 PRINT "{29LEFT}";<122>
53222 OPEN 2,8,2,"#":PRINT#15,"U1 2 0 18 0<037>
":PRINT#15,"B-P";2;162:GET#2,I$
53224 CLOSE 2:IF SIDE<>VAL(I$) THEN 53200<046>
53225 IF SIDE=2 THEN OPEN 1,8,3,"TEXTE"<103>
53230 RETURN<199>
53400 REM ZEIGER POSITIONIEREN -----
--<108>
53401 :<034>
53420 HB%=SA/256:LB%=SA-HB%*256:PRINT#15,"<027>
P";CHR$(3);CHR$(LB%);CHR$(HB%);CHR$(
1)
53430 INPUT#15,IA:IF IA=50 THEN PRINT "SATZ<222>
NUMMER ZU HOCH"
53440 RETURN<155>

```

Listing 1. Ein »Adventure-Programmier-Gerüst« (Fortsetzung)

```

53500 REM SATZ EINLESEN -----
-----
53501 : <223>
53510 IF A=50 THEN 53520 <134>
53512 INPUT#1,I$ <148>
53515 ZW=ZW+1:IF ZW>23 THEN GOSUB 53700:ZW <058>
      =0 <010>
53520 RETURN <237>
53600 REM TEXT-EINLESEN -----
-----
53601 : <121>
53610 IF SIDE<>2 THEN SIDE=2:GOSUB 53200 <236>
53620 GOSUB 53400:REM POSITIONIEREN <080>
53630 GOSUB 53500:REM I$ EINLESEN <165>
53640 IF RIGHT$(I$,1)<>"*"THEN 53660 <068>
53650 :IF I$<>"*"THEN PRINT LEFT$(I$,LEN(I <056>
      $)-1) <118>
53652 :SA=0:RETURN <218>
53660 IF LEN(I$)<40 THEN PRINT I$ <182>
53670 IF LEN(I$)>=40 THEN PRINT I$; <144>
53680 SA=SA+1 <021>
53695 GOTO 53630 <014>
53700 REM WEITER -----
-----
53701 : <152>
53705 PRINT"[WEITER]";POKE 198,0:WAIT 198 <080>
      ,1:PRINT"(LEFT,8SPACE,4LEFT,UP)" <183>
53710 RETURN <171>
53800 REM KOMMENTAR LESEN -----
-----
53801 : <213>
53805 ZW=1 <182>
53810 PRINT:GOSUB 53600:RETURN <070>
53900 REM TEXTFORMER ----- <095>
53901 : <007>
53902 IF X$<>"*"THEN PRINT <026>
53905 IF X$="*"THEN 53945 <047>
53910 IF LEN(X$)<40 THEN PRINT X$:GOTO 539 <230>
      45 <004>
53915 IF LEN(X$)=40 THEN PRINT X$:GOTO 53 <055>
      945 <088>
53920 I$=LEFT$(X$,41):IC=0:FOR I=1 TO LEN( <239>
      I$):IF MID$(I$,I,1)="*"THEN IC=I <138>
53925 NEXT I:IF IC=0 THEN PRINT X$:GOTO 53 <203>
      945 <237>
53930 I$=LEFT$(I$,IC-1):IF LEN(I$)=40 THEN <152>
      PRINT I$; <116>
53935 IF LEN(I$)<40 THEN PRINT I$ <076>
53940 X$=RIGHT$(X$,LEN(X$)-IC):GOTO 53905 <206>
53945 RETURN <133>
53950 REM ARTIKEL-WANDLER -----
53951 : <095>
53952 I$=LEFT$(OB$,3) <035>
53955 IF I$<>"DER"AND I$<>"DAS"THEN 5396 <045>
      5 <070>
53960 X$="":X$="HIER LIEGT EIN"+RIGHT$(OB$ <133>
      ,LEN(OB$)-3)+". " <095>
53965 IF I$<>"DIE"THEN 53970 <035>
53966 X$="":X$="HIER LIEGT EINE"+RIGHT$(OB <045>
      $(I),LEN(OB$)-3)+". " <070>
53970 GOTO 53905
54000 REM RAUMTEXT-EINLESEN -----
-----
54001 : <055>
54020 PRINT:FOR I=1 TO OZ :REM OB.TAB. <126>
54030 :IF OB(I)<>ZN THEN 54080 <187>
54035 :IF OX(I)<1 THEN 54080 :REM NAMELOS <170>
54040 :SA=OX(I):GOSUB 53400 :REM POSIT. <186>
54050 :IF OA(I)=1 THEN GOSUB 53500:R$=I$ <071>
      :REM READ I$, TITELLINE <035>
54060 :IF WI(ZN)=1 AND VE<>11 AND OA(I)=1 <062>
      THEN PRINT R$:SA=0:GOTO 54080
54070 :GOSUB 53600 :REM LESEN <009>
54080 NEXT I <059>
54081 FOR I=1 TO OZ <179>

```

```

54083 IF OB(I)<>ZN OR OA(I)<>3 OR OX(I)<>0 <252>
      THEN 54088
54084 SA=OS(I):GOSUB 53400:GOSUB 53500:OB$ <250>
      ="D"+RIGHT$(I$,LEN(I$)-1)
54085 X$="HIER LIEGT "+OB$+"." :ZW=ZW+1:GOS <045>
      UB 53950
54088 NEXT I <067>
54095 WI(ZN)=1:PRINT:RETURN <205>
60000 REM TABELLEN EINLESEN -----
-----
60001 : <092>
63000 REM ***** <030>
63005 REM * MINI-TEXTEDITOR V.111.4 * <017>
63010 REM * <C> 1985 * <071>
63015 REM * VON MICHAEL NICKLES * <190>
63020 REM ***** <230>
63025 REM ANFANGSMENUE ----- <037>
63030 POKE 198,0:INPUT"(2DOWN)DATEINAME(2S <066>
      PACE)*3LEFT)";D$:POKE 198,0
63035 INPUT"SATZLAENGE(2SPACE)40(4LEFT)";S <094>
      L:IF SL<1 OR SL>254 THEN PRINT"GEHT
      NICHT !":GOTO 30 <211>
63040 OPEN 15,8,15 <203>
63045 PRINT#15,"I0":OPEN 1,8,3,D$:PRINT <200>
63050 REM ZWISCHENMENUE ----- <219>
63055 PRINT"(RVSON)S / L / ENDE(28SPACE,RV <014>
      OFF)";
63060 GET I$:IF I$="*"THEN 63060 <252>
63065 IF I$="E"THEN PRINT"(CLR)";GOTO 632 <043>
      65
63070 IF I$<>"S"AND I$<>"L"THEN 63060 <137>
63075 INPUT"SATZNR";SA:IF SA<0 OR SA>63000 <050>
      THEN 63070
63080 IF I$="L"THEN PRINT"(CLR)";GOTO 630 <222>
      95
63085 IF I$="S"THEN PRINT"(CLR)";MO=1:GOT <126>
      O 63150
63090 GOTO 63050 <212>
63095 REM LESEN ----- <046>
63100 :PRINT"(CLR,RVSON)";SA;"(LEFT)-";STR <030>
      $(SA+20);"(SPACE,RVOFF)" <130>
63105 IA=SA <040>
63110 FOR SA=IA TO IA+20 <042>
63115 :GOSUB 63195:GOSUB 63220
63120 :IF A=50 THEN SA=IA+21:GOTO 63135 <049>
63125 :IF LEN(A$)<40 THEN PRINT A$ <089>
63130 :IF LEN(A$)>=40 THEN PRINT A$; <252>
63135 NEXT:PRINT <000>
63140 IF A=50 THEN PRINT"SATZNUMMER ZU HOC <120>
      H!"
63145 GOTO 63050 <011>
63150 REM SCHREIBEN ----- <138>
63155 PRINT"EINGABESATZ(2SPACE)//=KEINE EI <185>
      NGABE"
63160 GOSUB 63195:REM POSITIONIEREN <146>
63165 IF A=50 THEN PRINT"SATZNUMMER ZU HOC <009>
      H!":GOTO 63050:REM FEHLERKANALABFR.
63170 GOSUB 63330:A$=BE$ <208>
      :REM INPUT-ROUT.
63175 IF A$="//"THEN MO=0:GOTO 63050:REM E <157>
      NDE
63180 GOSUB 63240:REM A$ SCHREIBEN <131>
63185 SA=SA+1 :REM NAECHSTER SATZ <149>
63190 GOTO 63160:REM NEXT (LOOP) <010>
63195 REM POSITIONIEREN ***** <220>
63200 HB%=SA/256:LB%=SA-HB%*256 <153>
63205 PRINT#15,"P";CHR$(3);CHR$(LB%);CHR$( <085>
      HB%);CHR$(1)
63210 INPUT#15,A <154>
63215 RETURN <022>
63220 REM SATZ EINLESEN ***** <057>
63225 IF A=50 THEN RETURN <092>
63230 INPUT#1,A$ <121>
63235 RETURN <044>
63240 REM SATZ SCHREIBEN ***** <043>
63245 IF A=50 THEN RETURN <114>
63250 IF A$="*"THEN A$="":REM ZUR SICHERHE <141>
      IT !!!
63255 PRINT#1,A$ <228>
63260 RETURN <069>
63265 REM BEENDEN ***** <107>
63270 CLOSE 1 <032>
63275 CLOSE 15 <120>

```

```

63280 PRINT "++ ENDE ++" <026>
63285 END <038>
63290 REM KORREKTUR-ROUTINE ----- <069>
63295 FOR I=0 TO LEN(BE$):PRINT CHR$(20);: <123>
NEXT I:BE$="" <123>
63300 SA=SA+SC : REM TEXTZEIGER <015>
63305 IF SA<0 THEN SA=0 : REM SPERRE <117>
63310 GOSUB 63195: REM POSITIONIEREN <042>
63315 IF A=50 THEN SA=SA-SC:GOTO 63310 <132>
63320 INPUT#1,A$ <213>
63325 RETURN <134>
63330 REM TEXTEINGABE ----- <248>
63335 IF AR=1 THEN AR=0:A$="(RVSON)SATZNUM <104>
MER ZU HOCH! (RVOFF)":GOTO 63345 <190>
63340 GOSUB 63195:GOSUB 63220 <042>
63345 PRINT" (HOME,40SPACE)":PRINT" (HOME,R <203>
VSON)SATZ";SA;" (RVOFF)" <086>
63350 PRINT" (40SPACE,UP)":A$ <059>
63355 PRINT"TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT <182>
TTTTTTTT";
63360 PRINT" (DOWN)"
63365 BE$="":POKE 198,0:POKE 211,0:POKE 21
4,22:SYS 58732:PRINT" (YELLOW)";
63370 GET X$:IF PEEK(203)=1 THEN 63430 <129>
63375 IF PEEK(203)=7 AND PEEK(653)=0 THEN <025>
SC=1 : GOSUB 63290: GOTO 63330 <162>
63380 IF PEEK(203)=7 AND PEEK(653)=1 THEN <154>
SC=-1 : GOSUB 63290: GOTO 63330 <125>
63385 IF X$=""THEN 63370 <155>
63390 IF LEN(BE$)=0 AND ASC(X$)=20 THEN 63 <147>
370 <134>
63395 I=ASC(X$):IF I<32 OR I>133 AND I<159 <153>
THEN IF I<20 THEN 63370 <225>
63400 IF LEN(BE$)=SL AND I<20 THEN 63370 <058>
63405 IF X$=" "THEN X$="(RIGHT)" <005>
63410 BE$=BE$+X$ <025>
63415 PRINT CHR$(20);X$;" "; <146>
63420 IF I=20 THEN BE$=LEFT$(BE$,LEN(BE$)- <122>
2):GOTO 63370
63425 GOTO 63370
63430 PRINT" (LEFT,SPACE)":GOSUB 63195
63435 IF BE$=""THEN BE$="":
63440 PRINT" (HOME,24DOWN)":RETURN

```

Listing 1. Ein »Adventure-Programmier-Gerüst« (Schluß)

Dokumentation

- 3-4 Laden und Aktivieren des deutschen Zeichensatzes (muß sich ebenfalls auf der ersten Diskseite befinden!).
- 5 Floppykanal öffnen.
- 6 Disk Seite 2 einlegen!
- 15 Ab 60000 stehen alle Tabellen wie der Wortschatz-Index.
- 20 Anhand des Feldes WI(1)-WI(100) merkt sich das Programm, in welchem Raum der Spieler bereits war. Genau wie bei INFOCOM erhält der Spieler von Räumen, in denen er bereits war, nur noch eine kurze Beschreibung.
- 30 Zähler für Punktzahl und Anzahl der Befehlseingaben werden auf Null gesetzt.
- 40 Der Titelvorspann des Spiels wird von der relativen Datei aufgerufen (SA ist dann meistens 1, da der Vorspann meistens beim ersten Satz der Texte-Datei beginnt).
- 50 Raumbeschreibung des ersten Zimmers ausgeben.
- 50000-52180 Hier steht Parser 4.0. Allerdings sind einige kleine Sachen hinzugefügt. Kommt im Befehlsatz das Wort »Alles« vor, so wird der Variablen SP der Wert 1 zugeordnet. Kommt »Ausser« vor, so erhält die Variable SQ den Wert 1. Dies ist nötig, damit wir auch Sätze wie zum Beispiel »NIMM ALLES AUSSER DEM SCHWERT« abhandeln können.
- 53000-53040 Unterprogramm zum Einblenden einer



Statuszeile in der obersten Bildschirmzeile. Wie bei Infocom stehen in dieser Zeile der Name des Raumes, in dem man sich gerade befindet, sowie Punktzahl und Anzahl der Versuche. Dieses Unterprogramm wird innerhalb der erweiterten INPUT-Routine von Parser 4.0 aufgerufen (Zeile 50011).

53200-53230 Dieses Unterprogramm haben wir bereits im Kapitel Dateiverwaltung kennengelernt. Es dient dazu, festzustellen, ob die richtige Diskseite (Wert von Variable SI) eingelegt ist. Ist Disk Seite 2 eingelegt, so öffnet das Unterprogramm außerdem automatisch die relative Datei »Texte«.

Soll Disk Seite 1 eingelegt werden, so sagen Sie einfache SIDE=1:GOSUB53200. Der Rücksprung aus dem Unterprogramm erfolgt erst dann, wenn die gewünschte Seite eingelegt ist.

53400-53440 Das altbekannte Unterprogramm zum Positionieren des Zeigers auf eine relative Datei.

53500-53520 Dito, zum Einlesen eines Satzes (!\$) aus einer relativen Datei.

53600-53695 Ein Text aus der relativen Texte-Datei wird eingelesen. Beginnend ab Satz SA bis zu dem nächsten Satz, der mit einem Sternchen endet. Wenn Sie also einen Text von der relativen Datei einlesen und auf dem Bildschirm ausgeben wollen, so ist ein

SA = (1. Satz des Textes): GOSUB 53600 notwendig.

53700-53710 Auf dem Bildschirm erscheint das Wort WEITER. Bei Tastendruck verschwindet es wieder, und das Unterprogramm wird mittels RETURN beendet. Dieses Unterprogramm wird zum Beispiel dann verwendet, wenn ein Text am Stück auf dem Bildschirm ausgegeben wird, der mehr als 25 Sätze enthält. Nach dem 24. Satz erscheint WEITER, damit dem Spieler genug Zeit zum Lesen bleibt und der Text nicht wegscrollt.

53800-53810 Dasselbe wie 53600-53695 nur, daß von der eigentlichen Textausgabe diesmal eine Leerzeile eingefügt wird.

53900-53945 Der altbekannte Textformer, der den String X\$ per word-wrapping auf dem Bildschirm ausgibt.

53950-53970 Die Gegenstände beziehungsweise transportablen Objekte sind in der Texte-Datei mit voranstehendem Geschlechtsartikel - der, die oder das - gekennzeichnet. Am Ende der Raumbeschreibungen soll jedoch für zum Beispiel Schwert der Kommentar »Hier liegt ein Schwert« kommen. (Natürlich wieder analog zu Infocom's THERE IS A SWORD LYING HERE.)

Die Artikelwandlung ist ganz einfach:

Aus der beziehungsweise das wird EIN. Aus die wird EINE.

54000-54095 Unterprogramm zum Einlesen eines Raumtextes. Das Einlesen eines Raumtextes ist nicht ganz so einfach wie das Einlesen eines gewöhnlichen Textes, da sich ein Raumtext aus mehreren Einzelteilen zusammensetzt:

- Zuerst wird der Haupt-Raumtext ausgegeben. Das ist der Text, der sich nie ändert.

- Dann werden Textteile ausgegeben, die sich ändern können. Ein Beispiel: Eine Tür, die bei der Beschreibung sowohl als offen als auch als geschlossen beschrieben werden kann.

- Im Raum befindliche Gegenstände. Dies sind in der Regel Objekte, die transportiert werden können. In der Raumbeschreibung erscheinen diese Objekte dann als »Hier liegt ein...« etc.

63000-63440 Hier steht eine Mini-Version des Relativen-Dateien-Editors. Er wird mittels GOSUB 63000 gestar-

tet. Beim Schreiben eines Textadventures fallen einem stets neue Kommentare ein, die man der Texte-Datei zufügen muß. Es wäre viel zu umständlich, wegen jedem neuen Kommentar die aktuelle Arbeit zu speichern, dann den Relative-Datei-Editor laden etc. Daher steht am Ende des Gerüsts eine Mini-Version, mit der nur gelesen und geschrieben werden kann. Am Ende der Programmierung kann dieser Editor entfernt werden.

Die Dokumentation ist an dieser Stelle noch nicht komplett. Sie müssen erst noch einige andere Dinge wissen.

Spielablauf der ersten Szene

So läuft die erste Spielszene ab: Der Spieler erwacht in der Glaskuppel. Er muß jetzt im Sand graben, um einen Zettel und ein Amulett zu finden. Anschließend gelangt er durch den Tunnel in eine Kristallgrotte, in der eine Statue steht. Im Marmorboden steckt ein Schwert, das sich nicht herausziehen läßt. Der Spieler muß jetzt die Statue berühren, um sich in die Figur zu verwandeln, die die Statue darstellt. Er nimmt so den Körper eines jungen Kriegers an. Jetzt kann er mühelos das Schwert aus dem Boden ziehen. Dieses Herausziehen des Schwertes wirkt gleichzeitig wie ein Schlüssel: In der westlichen Wand tut sich eine Öffnung auf, die zu einem Felsplateau hinausführt. Dort beginnt die nächste Szene.

Wir wollen jetzt gemeinsam das Gerüst um diese Szene ergänzen. Sie erfahren dabei gleichzeitig die Dokumentation zu den übrigen Gerüsten.

```

60000 REM TABELLEN EINLESEN -----
-----
--
60001 : <092>
60100 REM WORTSCHATZ-INDEX ----- <030>
60110 DIM IN(26) <213>
60120 FOR I=0 TO 26:READ IN(I):NEXT I <192>
60130 DATA 1,4,0,29,32,48,53,64,72,0,74,82 <181>
,0,88,96,100,107,109,124,160,167 <214>
60140 DATA 170,190,0,0,201 <188>
60150 DATA 211 : REM BUCHSTABE NACH Z ! <122>
60200 REM OBJEKT-TABELLE ----- <087>
60210 OZ=13:REM ANZAHL DER OBJEKTE <133>
60220 DIM OB(OZ),OX(OZ),OA(OZ),OU(OZ),OS(OZ) <103>
60230 FOR I=1 TO OZ <232>
60240 : READ OB(I),OX(I),OA(I),OU(I),OS(I) <003>
60250 NEXT I <133>
60260 DATA 1,24,1,24,164:REM GLASKUPPEL <183>
60262 DATA 1,0,2,95,165:REM SAND <198>
60264 DATA 2,34,1,34,166:REM TUNNEL <039>
60266 DATA 3,43,1,43,167:REM KR.GROTTE <066>
60268 DATA 3,102,2,0,168:REM QUADER <167>
60270 DATA 3,0,2,113,169:REM STATUE <177>
60272 DATA 3,106,0,0,170:REM SCHWERT <131>
60274 DATA 0,0,3,79,171:REM ZETTEL <127>
60276 DATA 0,0,3,87,172:REM AMULETT <092>
60278 DATA 0,76,2,91,173:REM GRUBE <098>
60279 DATA 0,161,2,0,174:REM OEFFNUNG <127>
60280 DATA 4,54,1,54,175:REM FELSPLAT. <123>
60282 DATA 3,0,2,97,176:REM KRISTALLE <037>
60500 REM LANDKARTE ----- <047>
60501 : <022>
60510 DIM MAP(4,10) <148>
60520 FOR I=1 TO 4 <037>
60530 : FOR I1=1 TO 10 <003>
60540 : READ MAP(I,I1) <126>
60550 : NEXT I1 <021>
60560 NEXT I <189>
60570 DATA 2,0,0,0,0,0,0,0,0,0,0,0 <042>
60572 DATA 3,1,0,0,0,0,0,0,0,0,0,0 <084>
60574 DATA 0,2,0,0,0,0,0,0,0,0,0,0 <094>
60576 DATA 0,0,0,3,0,0,0,0,0,0,0,0 <038>
60600 RETURN <203>

```

Listing 2. Ergänzung zu Listing 1

Zunächst müssen ab Zeile 60000 die Tabellen fertiggestellt werden, die im Rahmen von Gordon Saga II benötigt werden: die Wortschatz-Index-Tabelle, die Landkarte und die Objekt-Logik-Tabelle. Bitte ergänzen Sie das bisherige Gerüst-Programm um die Zeilen aus Listing 2.

Anlegen der Tabellen

Auf den Wortschatz-Index brauche ich nicht lange einzugehen. Es ist der Index des modifizierten Wortschatzes – erstellt mit dem Index-Maker-Programm und anschließend in Data-Zeilen abgelegt. Anders bei der Objekt-tabelle. Die Objekt-tabelle ist das wichtigste am ganzen Programm. Zu jedem der Objekte sind mehrere Daten gespeichert. Diese Daten sind in den Feldern OB(), OX(), OA(), OU() und OS() untergebracht. Was geben diese einzelnen Felder an? Betrachten wir sie am besten einmal im einzelnen:

In der Variablen OZ ist die Anzahl der Objekte gespeichert.
OB(1)–OB(Z): OB(X) gibt an, in welchem Raum sich Objekt X befindet. Der Wert OB(X) ist immer dann größer Null, wenn sich das Objekt X in einem Raum befindet.
Ist OB(X) = 0, so heißt dies, daß das Objekt X nicht bzw. nicht mehr vorhanden ist.

Ist OB(X) kleiner als Null, so befindet sich das Objekt irgendwo anders als in einem Zimmer. Zum Beispiel im Besitz des Spielers –OB(X)=–1. Soll das Objekt in einem Schrank liegen, so kann man ihm eine für den Schrank spezifische negative Zahl zuordnen und so weiter.

OX(1)–OX(OZ): OX(X) gibt an, ab welchem Satz der relativen Datei der Beschreibungs-Text zu Objekt X steht. Ist OX(X)=0, so heißt dies, daß das Objekt X bei der Raumbeschreibung nicht anhand eines Textes erläutert wird.

OA(1)–OA(OZ): OA(OZ) gibt an, um welche Art von Objekt es sich bei Objekt X handelt. Es gibt folgende Werte:
1. Das Objekt ist ein Raum (zum Beispiel Tunnel).
2. Das Objekt befindet sich im Raum und kann nicht transportiert werden (zum Beispiel Öffnung in der Wand).
3. Das Objekt kann transportiert werden vom Spieler (zum Beispiel Schwert).

OU(1)–OU(OZ): OU(OZ) zeigt auf den ersten Satz des Kommentars in der Texte-Datei, der ausgegeben wird, wenn das Objekt untersucht wird. Wird ein Objekt untersucht, dessen OU(X)-Wert gleich 0 ist, so erfolgt der Kommentar »DAS OBJEKT X HAT KEINE BESONDEREN MERKMALE.«

OS(1)–OS(OZ): OS(X) gibt an, in welchem Satz der Texte-Datei der Objektname von Objekt X gespeichert ist (Beispiel: die Glaskuppel = 164).

Die Objekt-tabelle steht nur bei der Entwicklung des Programmes in DATA-Zeilen. Sie wird später entweder als SEQ-File geladen oder als PRG-File geladen und irgendwo im RAM mit entsprechender Verwaltung untergebracht. Durch geschicktes Manipulieren der Objekt-tabelle ist es leicht, dem Adventure Intelligenz zu verleihen: Das Programm lacht den Spieler aus, wenn er auf die unsinnige Idee kommt, einen Raum wie den Tunnel nehmen zu wollen. Will der Spieler eine lebende Person nehmen (OA-Wert könnte zum Beispiel mit 4 definiert werden), so kann sich der Analyseteil des Adventures mit »ICH GLAUBE KAUM, DASS SICH DIE PERSON X SO OHNE WEITERES NEHMEN LÄSST« etc.

Nehmen wir einmal an, der Spieler will Objekt X nehmen – also BES=»NIMM OBJEKT X«.

Das Programm kann nun verschiedenes überprüfen und so feststellen, ob der Spieler Objekt X überhaupt nehmen kann.

Erweist sich OB(X) als –1, so meldet sich das Adventure mit »SIE HABEN OBJEKT X BEREITS«.

Kristallgrotte	0/5	I
Kristallgrotte		
Dies ist eine riesige Grotte, deren Wände mit Kristallen, die in allen Farben des Lichtspektrums glitzern, besetzt sind. Der Boden besteht aus glatten, strahlend weißen Marmorplatten. Trotz zahlreichen Unebenheiten ähnelt diese Grotte dem Inneren einer Halbkugel mit einem Radius von ca. 100 Metern. Der einzige Ausgang von hier ist ein in südliche Richtung verlaufender Tunnel.		II
Auf dem Marmorboden, exakt im Zentrum des Raumes, steht ein goldener Quader, von dem eine ca. drei Meter hohe Statue emporragt.		
Nur wenige Meter von der Statue entfernt steckt ein goldenes Schwert zu etwa einem Drittel seiner Gesamtlänge im Marmorboden.		III
Hier liegt ein Zettel.		
Hier liegt ein Amulett.		IV

Bild 2. Aufteilung der Raumbeschreibung

IST OB(X) ungleich ZN (der aktuelle Raum), so erfolgt der Kommentar »OBJEKT X IST HIER NICHT ZU SEHEN«. Außerdem läßt das Programm nur zu, Objekte zu nehmen, deren OA-Wert = 3 ist (also transportables Objekt beziehungsweise Gegenstand).

Mit den 5 Objekt-tabellen kann also jedes Objekt exakt analysiert werden. Ein Beispiel: Die Statue hat folgende Werte:

OB(6)=3: Das heißt, die Statue steht im Raum 3.

OX(6)=0: Das heißt, die Statue wird in der Raumbeschreibung nicht mehr besonders erwähnt. Sie kommt schließlich bereits in der Beschreibung des Quaders, auf dem sie steht, vor.

OA(6)=2: Das heißt, die Statue ist ein feststehendes Objekt und kann daher auch nicht transportiert werden.

OU(6)=113: Ab Satz 113 steht der Kommentar für die Untersuchung der Statue.

OS(6)=164: In diesem Satz der Texte-Datei steht der Objektname der Statue – die Statue.

Ganz klar wird Ihnen die Objekt-tabelle allerdings erst dann werden, wenn Sie sich mit der Abhandlung der Objekte im Programm selbst vertraut gemacht haben.

Zunächst möchte ich Ihnen jedoch noch einmal ganz genau zeigen, wie sich eine Raumbeschreibung zusammensetzt. Bild 2 zeigt die Raumbeschreibung der Kristallgrotte, nachdem man dort noch zwei transportable Gegenstände verloren hat.

I Dies ist die Statuszeile. Sie ist ständig zu sehen und gibt an, in welchem Raum man sich gerade befindet, die bisher erzielte Punktzahl und die Anzahl der Spielzüge, die man bisher gemacht hat.

II In diesem Bereich steht die Hauptraumbeschreibung – also die Beschreibung, die sich während des gesamten Verlaufs des Spieles nicht ändert.

III Hier sind alle Objekte aufgeführt, die sich zwar verändern können, jedoch (zumindest im Moment) nicht vom Spieler transportiert werden können. Der OA-Wert dieser Objekte beziehungsweise Beschreibungen ist 2. Der OA-Wert von Gegenständen (=transportable Objekte) ist in der Regel 0. Hier gibt es jedoch eine Ausnahme: Es ist möglich, ein solches Objekt bei sei-

nem ersten Auftreten ausführlicher zu beschreiben als mit HIER LIEGT EIN ... Wird das Objekt jedoch genommen, so wird innerhalb der NIMM-Routine sein OA-Wert für immer auf Null gesetzt. Wird es wieder verloren, so taucht es in der Raumbeschreibung anschließend mit der Beschreibung HIER LIEGT EIN ... auf.

- IV Ganz am Ende der Raumbeschreibung werden die Objekte aufgeführt, die dort vom Spieler verloren worden sind beziehungsweise bei denen es sich um Gegenstände handelt, deren OA-Wert von vorneherein 0 ist.

Das Unterprogramm zur Erzeugung der Raumbeschreibung steht im Programm ab Zeile 54000. Vergleichen Sie doch mal die obige Theorie mit diesem Programmteil – dann gewinnen Sie am schnellsten den Durchblick!

Die nächste Tabelle, die definiert wird, ist die Landkarte, die als Feld MAP(N,10) angelegt ist, wobei N für die Anzahl der Räume steht. In dieser Tabelle ist eingetragen, welche Verbindungen zwischen den einzelnen Räumen existieren. Dabei werden allerdings nur die stets bestehenden Verbindungen eingetragen, die nicht erst geschaffen werden müssen (zum Beispiel durch das Öffnen einer Tür mit einem Schlüssel). Die Richtungen stehen so: N,S,W,Q,NW,NO,SW,SO, rauf, runter – also MAP(N,1) für wohin es nach Norden geht und MAP(N,10) für Süden. Eine nicht existente Verbindung hat den Wert 0.

Beispiel: Ist MAP(3,3)=4, so heißt dies, daß man von Raum 3 nach Westen gehen kann, um zu Raum 4 zu gelangen.

Analyseteil

Wie schon beim ersten Adventure-Kurs unterscheiden wir auch diesmal zwei Arten von Action: allgemeine und raumspezifische Action.

Allgemeine Action sind Handlungen, die man in jedem Raum, jederzeit ausführen kann – zum Beispiel Objekte untersuchen, Sachen nehmen, verlieren, Inventur etc.

Raumspezifische Action hingegen sind Handlungen, die man nur in bestimmten Räumen ausführen kann. »Berühre Statue« hat nur in dem Raum einen Effekt, in dem die Statue steht.

Im Programm steht die allgemeine Action in den Zeilen 300–800, die raumspezifische Action ab Zeile 2000.

Beim Gerüst sind in den Zeilen 300 – 700 bisher nur die Routinen untergebracht, die bei jedem Adventure unentbehrlich sind: GEH-Routine, NIMM-Routine, INVENTUR-Routine, VERLIER-Routine, UNTERSUCHE-Routine.

Ich verzichte an dieser Stelle bewußt auf eine genaue Dokumentation dieser Programmteile: Die Routinen sind sauber programmiert, und es sollte Ihnen keine Schwierigkeiten bereiten, sie selbst zu analysieren, wenn Sie bisher aufmerksam mitgearbeitet haben!

Wenn Sie das bisherige Programm um die folgenden wenigen Zeilen ergänzen, haben Sie die erste Szene von Gordon Saga Teil II komplett im Computer, da diese Zeilen die noch fehlende raumspezifische Action darstellen.

800–900 Hier ist bei der allgemeinen Action noch eine kleine Routine eingefügt, mit der man den Brief lesen kann, der in Gordon Saga Teil 2 gefunden wird. Diese Routine habe ich erst jetzt beigefügt, da sie zwar zur allgemeinen Action zählt (man kann einen Brief überall lesen – nicht raumabhängige Action), aber nicht für jedes Adventure benötigt wird.

2100–2130 Hier steht die raumspezifische Action zur Glaskuppel.

2110 Hier wird festgestellt, ob der Spieler im Sand graben will: ist Verb = 15 (graben) und Objekt = 2 (Sand)?

```

800 REM LESEN ----- <182>
805 IF VE<>16 THEN 900 <198>
810 IF O1=8 AND OB(8)=-1 THEN SB=80 <056>
900 : <114>
2000 REM RAUMSPEZIFISCHE ACTION -----
- <056>
2010 ON ZN GOTO 2100,2200,2300,2400 <139>
2100 REM GLASKUPPEL <069>
2110 IF VE<>15 OR O1<>2 THEN 2120 <005>
2115 IF PA(0)<3 THEN SB=71:PA(0)=PA(0)+1 <114>
2117 IF PA(0)=2 THEN SB=73:OB(8)=ZN:OB(9)=
      ZN:OB(10)=1 <035>
2119 IF PA(0)=3 THEN SB=72 <112>
2120 IF VE=1 AND O1=3 THEN RI=1:GOTO 300 <245>
2130 GOTO 100 <026>
2200 REM TUNNEL <039>
2210 GOTO 100 <106>
2300 REM KRISTALLGROTTE <198>
2305 IF VE=17 AND O1=6 AND OU(6)=113 THEN
      SB=119:OU(6)=135:GOTO 2307 <081>
2306 IF VE=17 AND O1=6 AND OU(6)=135 THEN
      PRINT"ES PASSIERT NICHTS." <032>
2307 IF VE<>12 OR O1<>7 THEN 2310 <010>
2308 IF OU(6)=113 THEN SB=110 <202>
2309 IF OU(6)=135 THEN SB=139:OA(7)=3:OB(7)
      )=-1:OB(11)=3:MAP(3,3)=4:OX(7)=0 <152>
2310 IF VE=1 AND O1=3 THEN RI=3:GOTO 300 <183>
2320 GOTO 100 <218>
2400 REM FELSPATEAU <194>
2410 GOTO 100 <052>

```

Listing 3. Damit ist Gordon Saga II komplett

Gräbt der Spieler zum ersten Mal, so wird der Variablen SB der Wert 71 gesetzt (Zeiger auf Satz: Sie graben in dem Sand.).

In Zeile 110 wird der Wert der Variablen SB abgefragt. Ist SB ungleich Null, so wird in der Text-Datei der Kommentar ausgelesen, der ab Satz SB bis zum nächsten Sternchen steht. SB wird sodann auf 0 gesetzt. Mittels SB werden also Kommentare aus der Text-Datei abgerufen.

- 2117 Zweites Graben. Der Spieler findet ein Amulett und einen Zettel. Diesen beiden Objekten wird jetzt der OB-Wert der Zimmernummer ZN zugeordnet, da diese beiden Objekte gefunden sind und von nun an im Spiel enthalten sind.

Außerdem wird der OB-Wert der Grube auf ZN gesetzt. Dadurch wird die Grube das nächste Mal in der Raumbeschreibung erwähnt, wenn der Spieler Raum 1 untersucht: Es scheint so, als ob hier jemand vor kurzem gegraben hat....Durch solche Action werden die Raumbeschreibungen variabel – das Adventure gewinnt an »Leben«.

- 2119 Versucht der Spieler ein drittes Mal zu graben, so erfolgt der Kommentar, daß weiteres Graben sinnlos ist.

- 2120 Wenn Verb=geh und Objekt=Tunnel, dann ist dies gleichbedeutend mit der Befehlseingabe »Norden«, da man von der Glaskuppel nach Norden gehen muß, um in den Tunnel zu gelangen. Also einfach die RI-Variable auf Norden setzen – RI=1 – und Sprung zur GEH-Routine (GOTO 300).

- 2200 Im Tunnel kommt keine raumspezifische Action vor.
2300–2320 Raumspezifische Action der Kristallgrotte.

- 2305 Wenn Verb=berühre und Objekt=Statue then Kommentar ab Zeile 119: Die Statue hat jetzt die Form des Spielers angenommen. Beim Untersuchen derselben muß also ein anderer Kommentar als bisher erscheinen. Also OU(6)=135.

Sie sehen schon: Mit Hilfe der einzelnen Objekt- und

Wortschatz-Tabellen läßt sich die raumspezifische Action nahezu in Klartext übersetzen.

Damit wäre ich mit meinen Erklärungen auch schon am Ende. Die restlichen Action-Szenen können Sie leicht selbst deuten. Adventure-Programmierung ist ein sehr kompliziertes Gebiet. Ich bin mir sicher, daß einige von Ihnen bereits jetzt Ideen haben, wo sich hier und da noch so manches verbessern läßt.

Vielleicht haben Sie auch schon die Idee zu Parser 6.0 – dem NON-PLUS-ULTRA-Parser im Kopf.

Ich hoffe, Ihnen mit diesem Kapitel neue Anregungen im Bereich der Adventure-Programmierung gegeben zu haben. Wenn Sie noch irgendwelche Fragen oder Probleme mit Adventure-Programmierung haben, dann wenden Sie sich doch mal an die 64'er, vielleicht geben Ihre Nachfragen Anlaß zu einem kleinen Folgekurs.

Ich will allerdings noch nicht zum Schlußwort kommen, denn es steht noch ein wichtiges Kapitel aus:

Der Adventure-Cracker

Das Programm im Listing 4 macht's möglich: Adventures im Schlaf gelöst. Naja, ganz im Schlaf geht's nicht, aber Sie müssen sich lediglich im Sessel zurücklehnen und am Bildschirm mitverfolgen, wie der gesamte Text des Adventures ausgegeben wird.

Im Prinzip ist der Adventure-Cracker nichts weiter als ein ultra-primitiver Disk-Monitor, mit dem sich die einzelnen Bytes einer Diskette lesen lassen. Sie müssen einfach die Diskette, auf der sich das Adventure befindet, ins Laufwerk legen und dann mittels Angabe von Track und Sector nach Text suchen.

Der Adventure-Cracker filtert automatisch alle Zeichen heraus, die keinen Text darstellen (Grafikzeichen etc.). Es ist nur eine Frage der Zeit, bis man so den gesamten Text eines Adventures herausfindet.

Der Cracker knackt alle Basic-Adventures, Adventures, die mit relativen Dateien arbeiten (also unsere), und auch professionelle Spiele. Den INFOCOMs kann man mit dem Adventure-Cracker allerdings in keinsten Weise zu Leibe rücken: Ihr Text ist auf eine äußerst komplizierte Weise verschlüsselt und bringt auch harte Dechiffrierer zur Verzweiflung ...

Es ist also ratsam, Adventure-Texte nicht einfach unverschlüsselt in relativen Dateien abzulegen. Parallel zum Codieren bietet sich allerdings noch etwas anderes an:

Text-Raffung

Es gibt verschiedene Methoden, Text zu kürzen, um so letztendlich noch mehr Text auf einer Disk-Seite unterzubringen. Es bietet sich zum Beispiel an, einzelne Kürzel zu definieren. Schließlich haben wir 255 verschiedene Zeichen zur Verfügung, von denen zur reinen Textdarstellung nur ca. 70 benötigt werden (Groß-Klein-Buchstaben und diverse Satzzeichen). Man könnte nun doch eigentlich eine Tabelle anlegen, in der häufige Buchstabenkombinationen (er, die, ch, das, ng, en, ...) durch Grafikzeichen codiert werden.

Sie können insgesamt zirka 180 solche Kürzel definieren und dann ein Programm schreiben, das Ihre relativen Dateien automatisch kürzt und die Buchstaben anhand einer weiteren Tabelle – einem Code-Schlüssel – chiffriert.

Es gibt unglaublich viele Verschlüsselungs- und Text-Raffungsmethoden, die sich relativ einfach programmieren lassen. Denken Sie mal nach!

Bonbon zum Schluß

Zu Beginn dieses Kurses habe ich Ihnen versprochen, daß das freie RAM bis zum letzten Byte mit interessanten Routinen belegt wird. Das haben wir auch geschafft. Bis auf einen kleinen Bereich: von 1024 bis 2023 der alte Bildschirmspeicher.

Dort bringen wir jetzt noch eine Maschinenroutine unter, die in Listing 5 abgedruckt ist.

Diese Routine können Sie erst dann laden, wenn der deutsche Zeichensatz aktiviert und der Bildschirm an eine andere Stelle als 1024–2023 gelegt worden ist. Zum Laden können

```

0 REM ***** <138>
1 REM * * <050>
2 REM * ADVENTURE-CRACKER * <187>
3 REM * VERSION 1.0 * <247>
4 REM * VON MRIGHT$CHAE L NICKLES * <015>
5 REM * * <054>
6 REM ***** <144>
10 PRINT "CLR,RVSON)ADVENTURE-CRACKER VERS <038>
ION 1.0(11SPACE,RVOFF)";
20 PRINT "(DOWN)BITTE DISK MIT SPIEL EINLEG <022>
EN(SPACE,RVSON)RETURN(RVFF)"
25 POKE 198,0:WAIT 198,1 <253>
30 OPEN 1,8,15:OPEN 2,8,2,"#" <067>
50 REM HAUPTMENUE <188>
51 PRINT "(RVSON)EINZEL / ALLES" <099>
52 GET X$:IF X$=""THEN 52 <225>
55 IF X$="E"THEN 200 <166>
57 IF X$="A"THEN 100 <162>
58 GOTO 52 <044>
100 REM ALLE DATEN VON DISK LESEN <147>
105 INPUT"AB WELCHEM TRACK ";AN <041>
110 FOR T=AN TO 35:REM TRACKS <192>
112 REM AS=ANZAHL DER SEKTOREN <153>
113 IF T<18 THEN AS=21 <216>
114 IF T>17 AND T<25 THEN AS=19 <197>
115 IF T>24 AND T<31 THEN AS=18 <008>
116 IF T>30 THEN AS=17 <081>
118 FOR S=1 TO AS:REM SECTORS <063>
120 GOSUB 1000:REM DATEN IN BUFFER LESEN <093>
130 GOSUB 2000:REM DATEN AUSDRUCKEN <051>
140 NEXT S <048>
150 NEXT T <066>
200 REM EINZELNE DATEN LESEN <043>
201 PRINT <047>
205 INPUT"TRACK ";T <145>
206 INPUT"SEKTOR ";S <087>
210 IF T<1 OR T>35 THEN 200 <182>
211 IF S<1 THEN 200 <142>
212 IF T<18 AND S>21 THEN 200 <213>
213 IF T>17 AND T<25 AND S>19 THEN 200 <242>
214 IF T>24 AND T<31 AND S>18 THEN 200 <054>
215 IF T>30 AND S>17 THEN 200 <034>
220 GOSUB 1000 <176>
230 GOSUB 2000 <194>
250 GOTO 200 <186>
1000 : <214>
1001 REM DATEN IN BUFFER LESEN <212>
1002 : <216>
1005 PRINT#1,"U1 2 0 ";T;S <068>
1010 RETURN <050>
2000 : <198>
2001 REM BUFFER AUSLESEN <114>
2002 :M$="" <099>
2010 FOR I=0 TO 255 <074>
2020 :GET#2,A$ <046>
2021 IF A$=""THEN 2030 <195>
2022 M=ASC(A$):REM CHECKEN <008>
2023 IF M>64 AND M<91 THEN PRINT A$; <189>
2024 IF M>96 AND M<123 THEN PRINT A$; <004>
2030 IF M=46 OR M=58 OR M=59 OR M=63 OR M= <155>
32 OR M=160 OR VAL(A$)>0 THEN PRINT A <090>
$; <076>
2040 NEXT I
2050 RETURN

```

Listing 4. Der Adventure-Cracker

programm : ms 1400 16b0

```
1400 : a2 0f 20 c9 ff 20 fd ae c1
1408 : 20 8a ad 20 f7 b7 a9 50 61
1410 : 20 d2 ff a9 08 20 d2 ff 9b
1418 : a5 14 20 d2 ff a5 15 20 eb
1420 : d2 ff a9 01 20 d2 ff 20 55
1428 : cc ff a2 08 20 c6 ff a2 1b
1430 : 00 20 cf ff 9d fa 04 c9 89
1438 : 0d f0 03 e8 d0 f3 e0 28 1c
1440 : d0 05 a9 00 9d fa 04 e8 91
1448 : a9 00 9d fa 04 20 cc ff 2c
1450 : a2 ff e8 bd fa 04 d0 fa ed
1458 : ca bd fa 04 c9 0d f0 f8 fb
1460 : c9 2a f0 2f a2 00 bd fa 77
1468 : 04 f0 06 20 d2 ff e8 d0 dc
1470 : f5 a5 14 18 69 01 85 14 1d
1478 : a5 15 69 00 85 15 ee f9 b3
1480 : 04 ad f9 04 c9 16 d0 03 f0
1488 : 20 b9 04 a2 0f 20 c9 ff f3
1490 : 4c 0e 04 a9 0d 9d fa 04 cb
1498 : e8 a9 00 9d fa 04 a2 00 63
14a0 : bd fa 04 f0 06 20 d2 ff a6
14a8 : e8 d0 f5 ee f9 04 ad f9 be
14b0 : 04 c9 16 d0 03 20 b9 04 59
14b8 : 60 a9 00 8d f9 04 a2 00 e9
14c0 : bd e5 04 f0 06 20 d2 ff 3c
14c8 : e8 d0 f5 a9 00 85 c6 a5 5d
14d0 : c6 f0 fc a2 00 bd f0 04 5b
14d8 : f0 06 20 d2 ff e8 d0 f5 a4
```

```
14e0 : a9 00 85 c6 60 12 5b 57 76
14e8 : 45 49 54 45 52 5d 92 00 ea
14f0 : 14 14 14 14 14 14 14 f0
14f8 : 00 00 00 00 00 00 00 f9
1500 : 00 00 00 00 00 00 00 01
1508 : 00 00 00 00 00 00 00 09
1510 : 00 00 00 00 00 00 00 11
1518 : 00 00 00 00 00 00 00 19
1520 : 00 00 00 00 00 00 00 21
1528 : a9 40 8d 04 d4 a9 0f 8d c7
1530 : 18 d4 a2 1e a0 86 8e 00 97
1538 : d4 8c 01 d4 a2 00 a0 1b
1540 : 8e 05 d4 8c 06 d4 a9 00 c5
1548 : 8d f9 04 a9 3e 20 d2 ff 38
1550 : a9 a4 20 d2 ff a9 00 8d 16
1558 : af 06 a9 00 85 c6 a5 c6 27
1560 : f0 fc ad 77 02 a2 00 dd 1a
1568 : 5d 06 f0 08 e8 e0 52 d0 86
1570 : f6 4c 77 05 4c b5 05 c9 25
1578 : 0d d0 15 ad af 06 f0 2f 36
1580 : a9 9d 20 d2 ff a9 20 20 68
1588 : d2 ff 20 f8 05 4e 08 06 60
1590 : c9 14 d0 1b ad af 06 f0 4d
1598 : 16 a9 14 20 d2 ff 20 d2 df
15a0 : ff a9 a4 20 d2 ff 20 f8 41
15a8 : 05 ce af 06 4c 5a 05 20 ad
15b0 : 36 06 4c 5a 05 ac af 06 c8
15b8 : c0 4d f0 36 ae af 06 9d dd
15c0 : b0 06 a9 9d 20 d2 ff bd a5
```

```
15c8 : b0 06 20 d2 ff a9 a4 20 fe
15d0 : d2 ff ee af 06 bd b0 06 70
15d8 : c9 22 d0 10 20 d2 ff a9 d4
15e0 : 14 20 d2 ff 20 d2 ff a9 a5
15e8 : a4 20 d2 ff 20 f8 05 4c c7
15f0 : 5a 05 20 36 06 4c 5a 05 d2
15f8 : a9 41 8d 04 d4 a2 00 e8 5a
1600 : d0 fd a9 80 8d 04 d4 60 56
1608 : a9 00 85 c6 20 fd ae ad f3
1610 : af 06 20 7d b4 a0 00 b9 3e
1618 : b0 06 91 62 c8 cc af 06 3a
1620 : d0 f5 20 8b b0 85 fb 84 94
1628 : fc a2 02 a0 02 b5 61 91 80
1630 : fb ca 88 10 f8 60 a2 0a e6
1638 : 8e 00 d4 8e 01 d4 a9 21 6d
1640 : 8d 04 d4 a2 b0 a0 00 c8 fa
1648 : d0 fd e8 d0 f8 a9 80 8d 65
1650 : 04 d4 a6 1e a4 86 8e 00 e4
1658 : d4 8c 01 d4 60 41 42 43 ed
1660 : 44 45 46 47 48 49 4a 4b 50
1668 : 4c 4d 4e 4f 50 51 52 53 58
1670 : 54 55 56 57 58 59 5a 30 09
1678 : 31 32 33 34 35 36 37 38 68
1680 : 39 2c 2e 20 21 23 24 25 65
1688 : 26 27 28 29 2b 2d 3a 3b ed
1690 : 2f 3f 40 2a 22 c1 c2 c3 77
1698 : c4 c5 c6 c7 c8 c9 ca cb 87
16a0 : cc cd ce cf d0 d1 d2 d3 8f
16a8 : d4 d5 d6 d7 d8 d9 da 00 df
```

Listing 5. Einige nützliche Hilfsroutinen. Beachten Sie die Hinweise im Text.

Sie nicht den LOAD-Befehl, sondern nur Listing 6 verwenden.

Die Routine beherbergt zwei sehr wichtige Funktionen, die bei jeder Adventure-Programmierung benötigt werden:

1. Erweiterte INPUT-Routine

Mittels SYS 1160,BE\$ rufen Sie die erweiterte INPUT-Routine auf. Nach Rücksprung steht der eingegebene Satz in BE\$ – natürlich kann anstelle von BE\$ auch jeder andere beliebige String gesetzt werden.

2. Raumbeschreibung einlesen

Mittels SYS 1024,SA rufen Sie eine Maschinenroutine auf, die den Satzbereich einer relativen Datei von Satz SA ab bis zum nächsten Sternchen ausgibt – und das sehr, sehr schnell!

Die relative Datei muß von Basic aus per OPEN 8,8,8,»Filename« geöffnet werden, ebenso der Floppykanal mittels OPEN 15,8,15.

```
10 OPEN 1,8,0,"MS":GET#1,A$:GET#1,A$ <194>
20 FOR I=1024 TO 1718:GET#1,A$:A$=A$+CHR$(
   0):POKE I,ASC(A$):NEXT I:CLOSE 1 <102>
```

Listing 6. Lader für Listing 5

Nochmals sei darauf hingewiesen, daß die Routinen nur lauffähig sind, wenn der deutsche Zeichensatz verwendet wird, und daß sie nur mit den Zeilen aus Listing 6 geladen werden können.

Mit dieser Routine möchte ich mich von Ihnen verabschieden. Ich hoffe, daß Sie einige wertvolle Tips zur Programmierung von Abenteuerspielen erhalten haben.

Viel Spaß beim Programmieren!

(M. Nickles/bs)

