Die wilde Jagd durchs Labyrinth

Das hier abgedruckte Spiel »Puck« ist ein spannendes Geschicklichkeitsspiel, das an den Klassiker »Pacman« erinnert.

Puck ist ein in Basic programmiertes Spiel, das zwar einfach aussieht, aber ganz schön schwer zu spielen ist. Ein mit dem Joystick gesteuerter kleiner Puck muß durch ein Labyrinth laufen und dort alle Punkte fressen. Dabei wird er von einer gefährlichen Spinne verfolgt. Die genaue Spielanleitung befindet sich im Programm und wird beim Starten mit »RUN« angezeigt. (Karl-Eugen Laubacher/bs)

```
1 DIM F%(40,25)
2 REM DATA IN FELD F% LADEN ***
3 SCNCLR : GOSUB 271: SCNCLR : PRINT "EIN WENIG
  GEDULD BITTE!"
4 GOSUB 150
5 REM SPIELFELD ZEICHNEN ***
6 PU=5: V=0: Y=0: LV=25
7 IF LV<10 THEN LV=10
8 GOSUB 157: FOR N=1 TO 10
9 GET NR$: NEXT N
10 AD=88: H=0: NR=0: ZV=0
11 PW=38: PS=23: POKE 4030,81
12 GW=1: GS=1: POKE 3113,42: RE=32: OL=3
13 IF PS=GS AND PW=GW THEN GOSUB 178: IF PU=0 TH
   EN Y=Y+V: GOSUB 194: GOTO 6
14 ZU=INT(RND(1)*LV): IF ZU=0 THEN 17
15 GOSUB 19: IF V=541 THEN PL=5: PU=PU+1: Y=Y+54
   1: V=0: LV=LV-5: VOL 7: SOUND 1,800,15: GOTO
   7
16 IF PS=GS AND PW=GW THEN GOSUB 178: IF PU=0 TH
   EN Y=Y+V: GOSUB 194: GOTO 6
17 GOSUB 49
18 GOTO 13
19 NR=JOY(1)
20 IF AD=NR THEN GOSUB 29: H=0: RETURN
21 IF NR=0 THEN 25
22 ZV=AD: AD=NR: GN=0: GOSUB 29
23 IF GN=0 THEN H=0: RETURN
24 H=NR: AD=ZV: GOSUB 29: RETURN
25 IF H=0 THEN GOSUB 29: RETURN
26 ZV=AD: AD=H: GN=0: GOSUB 29
27 IF GN=0 THEN H=0: RETURN
28 AD=ZV: GOSUB 29: RETURN
29 IF AD<>7 THEN 34
30 C=3071+PW+40*PS: B=PEEK(C)
31 IF B=160 THEN GN=1: RETURN
32 IF B=46 THEN V=V+1
33 POKE C,81: PW=PW-1: POKE C+1,32: RETURN
34 IF AD<>3 THEN 39
35 C=3073+PW+PS*40: B=PEEK(C)
36 IF B=160 THEN GN=1: RETURN
37 IF B=46 THEN V=V+1
38 POKE C,81: PW=PW+1: POKE C-1,32: RETURN
39 IF AD<>1 THEN 44
40 C=3072+PW+40*PS-40: B=PEEK(C)
41 IF B=160 THEN GN=1: RETURN
42 IF B=46 THEN V=V+1
43 POKE C,81: PS=PS-1: POKE C+40,32: RETURN
44 IF AD<>5 THEN RETURN
45 C=3072+PW+40*PS+40: B=PEEK(C)
46 IF B=160 THEN GN=1: RETURN
47 IF B=46 THEN V=V+1
48 POKE C,81: PS=PS+1: POKE C-40,32: RETURN
49 GP%=F%(GW,GS)
50 A=3072+GW+40*GS
51 RA=RE
52 ON GP% GOTO 54,58,62,66,70,74,78,90,102,114,1
   26
53 REM 1 ***
54 IF OL=3 THEN 56
55 RE=PEEK(A+1): GW=GW+1: POKE A+1,42: POKE A,RA
   : RETURN
56 RE=PEEK(A-1): GW=GW-1: POKE A-1,42: POKE A,RA
   : RETURN
57 REM 2 ***
58 IF OL=4 THEN 60
59 RE=PEEK(A+40): GS=GS+1: POKE A+40,42: POKE A,
   RA: RETURN
60 RE=PEEK(A-40): GS=GS-1: POKE A-40,42: POKE A,
   RA: RETURN
61 REM 3 ***
62 IF OL=2 THEN 64
63 RE=PEEK(A-40): GS=GS-1: POKE A-40,42: OL=4: P
   OKE A,RA: RETURN
64 RE=PEEK(A+1): GW=GW+1: POKE A+1,42: OL=1: POK
   E A,RA: RETURN
65 REM 4 ***
66 IF OL=4 THEN 68
67 RE=PEEK(A+40): GS=GS+1: POKE A+40,42: OL=2: P
   OKE A,RA: RETURN
68 RE=PEEK(A+1): GW=GW+1: POKE A+1,42: OL=1: POK
   E A,RA: RETURN
69 REM 5 ***
70 IF OL=4 THEN 72
71 RE=PEEK(A+40): GS=GS+1: POKE A+40,42: OL=2: P
   OKE A,RA: RETURN
72 RE=PEEK(A-1): GW=GW-1: POKE A-1,42: OL=3: POK
   E A,RA: RETURN
73 REM 6 ***
74 IF OL=2 THEN 76
75 RE=PEEK(A-40): GS=GS-1: POKE A-40,42: OL=4: P
   OKE A,RA: RETURN
76 RE=PEEK(A-1): GW=GW-1: POKE A-1,42: OL=3: POK
   E A,RA: RETURN
77 REM 7 ***
78 IF OL<>2 THEN 81
79 AR=ABS(GW+1-PW)+ABS(GS-PS): AU=ABS(GW-PW)+ABS
   (GS+1-PS)
80 IF AR<AU THEN 88: ELSE 87
81 IF OL<>3 THEN 84
82 A1=ABS(GW-PW): AU=ABS(GS+1-PS)+A1: AO=ABS(GS-
   1-PS)+A1
83 IF AO<AU THEN 86: ELSE 87
84 AR=ABS(GW+1-PW)+ABS(GS-PS): AO=ABS(GW-PW)+ABS
   (GS-1-PS)
85 IF AR<AO THEN 88: ELSE 86
86 RE=PEEK(A-40): GS=GS-1: POKE A-40,42: OL=4: P
   OKE A,RA: RETURN
87 RE=PEEK(A+40): GS=GS+1: POKE A+40,42: OL=2: P
   OKE A,RA: RETURN
88 RE=PEEK(A+1): GW=GW+1: POKE A+1,42: OL=1: POK
   E A,RA: RETURN
89 REM 8 ***
90 IF OL<>1 THEN 93
91 AR=ABS(GW+1-PW)+ABS(GS-PS): AU=ABS(GW-PW)+ABS
   (GS+1-PS)
92 IF AR<AU THEN 98: ELSE 99
93 IF OL<>3 THEN 96
94 AL=ABS(GW-1-PW)+ABS(GS-PS): AU=ABS(GW-PW)+ABS
   (GS+1-PS)
95 IF AL<AU THEN 100: ELSE 99
96 A1=ABS(GS-PS): AL=ABS(GW-1-PW)+A1: AR=ABS(GW+
   1-PW)+A1
97 IF AL<AR THEN 100: ELSE 98
98 RE=PEEK(A+1): GW=GW+1: POKE A+1,42: OL=1: POK
   E A,RA: RETURN
99 RE=PEEK(A+40): GS=GS+1: POKE A+40,42: OL=2: P
   OKE A,RA: RETURN
100 RE=PEEK(A-1): GW=GW-1: POKE A-1,42: OL=3: PO
    KE A,RA: RETURN
101 REM 9 ***
102 IF OL<>1 THEN 105
103 A1=ABS(GW-PW): AU=ABS(GS+1-PS)+A1: AO=ABS(GS
    -1-PS)+A1
104 IF AO<AU THEN 110: ELSE 111
```

Listing 1. Das Listing zu »Puck« für den C16

```
105 IF OL<>2 THEN 108
106 AL=ABS(GW-1-PW)+ABS(GS-PS): AU=ABS(GW-PW)+AB
    S(GS+1-PS)
107 IF AL<AU THEN 112: ELSE 111
108 AL=ABS(GW-1-PW)+ABS(GS-PS): AO=ABS(GW-PW)+AB
    S(GS-1-PS)
109 IF AL<AO THEN 112: ELSE 110
110 RE=PEEK(A-40): GS=GS-1: POKE A-40,42: OL=4:
    POKE A,RA: RETURN
111 RE=PEEK(A+40): GS=GS+1: POKE A+40,42: OL=2:
    POKE A,RA: RETURN
112 RE=PEEK(A-1): GW=GW-1: POKE A-1,42: OL=3: PO
    KE A,RA: RETURN
113 REM 10 ***
114 IF OL<>1 THEN 117
115 AR=ABS(GW+1-PW)+ABS(GS-PS): AO=ABS(GW-PW)+AB
    S(GS-1-PS)
116 IF AR<AO THEN 123: ELSE 122
117 IF OL<>2 THEN 120
118 A1=ABS(GS-PS): AL=ABS(GW-1-PW)+A1: AR=ABS(GW
    +1-PW)+A1
119 IF AL<AR THEN 124: ELSE 123
120 AL=ABS(GW-1-PW)+ABS(GS-PS): AO=ABS(GW-PW)+AB
    S(GS-1-PS)
121 IF AL<AO THEN 124
122 RE=PEEK(A-40): GS=GS-1: POKE A-40,42: OL=4:
    POKE A,RA: RETURN
123 RE=PEEK(A+1): GW=GW+1: POKE A+1,42: OL=1: PO
    KE A,RA: RETURN
124 RE=PEEK(A-1): GW=GW-1: POKE A-1,42: OL=3: PO
    KE A,RA: RETURN
125 REM 11 ***
126 IF OL<>1 THEN 131
127 A1=ABS(GW-PW): AO=ABS(GS-1-PS)+A1: AU=ABS(GS
    +1-PS)+A1
128 AR=ABS(GW+1-PW)+ABS(GS-PS)
129 IF AO<AU AND AO<AR THEN 148
130 IF AR<AU THEN 145: ELSE 146
131 IF OL<>2 THEN 136
132 A1=ABS(GS-PS): AL=ABS(GW-1-PW)+A1: AR=ABS(GW
    +1-PW)+A1
133 AU=ABS(GS+1-PS)+ABS(GW-PW)
134 IF AU<AR AND AU<AL THEN 146
135 IF AL<AR THEN 147: ELSE 145
136 IF OL<>3 THEN 141
137 A1=ABS(GW-PW): AO=ABS(GS-1-PS)+A1: AU=ABS(GS
    +1-PS)+A1
138 AL=ABS(GW-1-PW)+ABS(GS-PS)
139 IF AO<AU AND AO<AL THEN 148
140 IF AL<AU THEN 147: ELSE 146
141 A1=ABS(GS-PS): AL=ABS(GW-1-PW)+A1: AR=ABS(GW
    +1-PW)+A1
142 AO=ABS(GW-PW)+ABS(GS-1-PS)
143 IF AO<AR AND AO<AL THEN 148
144 IF AL<AR THEN 147: ELSE 145
145 RE=PEEK(A+1): GW=GW+1: POKE A+1,42: OL=1: PO
    KE A,RA: RETURN
146 RE=PEEK(A+40): GS=GS+1: POKE A+40,42: OL=2:
    POKE A,RA: RETURN
147 RE=PEEK(A-1): GW=GW-1: POKE A-1,42: OL=3: PO
    KE A,RA: RETURN
148 RE=PEEK(A-40): GS=GS-1: POKE A-40,42: OL=4:
    POKE A,RA: RETURN
149 REM ENDE SPINNE BEWEGEN ***
150 RESTORE 218
151 FOR S=1 TO 23
152 FOR W=0 TO 39
153 READ A%: F%(W,S)=A%
154 NEXT W
155 NEXT S
156 RETURN
157 SCNCLR
158 RESTORE 264
159 FOR N=0 TO 19
160 READ A: POKE 16360+N,A: NEXT N
161 SYS 16360
162 FOR N=1 TO 40
163 POKE 3071+N,160
164 POKE 4031+N,160
165 NEXT N
166 POKE 3112,160
167 FOR N=0 TO 880 STEP 40
168 POKE 3151+N,160: POKE 3151+N+1,160
169 NEXT N
170 RESTORE 196
171 FOR N=0 TO 331
172 READ A%
173 POKE 3072+A%,160
174 NEXT N
175 T=PU: IF PU>9 THEN T=-15
176 POKE 4071,T+48
177 RETURN
178 VOL 7: VL=7
179 FOR SO=0 TO 30
180 SOUND 3,800,1: VOL VL: VL=VL-.23
181 NEXT SO
182 PU=PU-1
183 IF PU=0 THEN RETURN
184 IF RE=81 THEN RE=32
185 POKE 3072+PW+PS*40,RE
186 T=PU: IF PU>9 THEN T=-15
187 POKE 4071,T+48
188 PW=38: PS=23: POKE 4030,81
189 GW=1: GS=1: POKE 3113,42: RE=32: OL=3
190 FOR N=1 TO 10
191 GET NR$: NEXT N
192 AD=88: H=0: NR=0
193 RETURN
194 SCNCLR : PRINT "PUNKTE:";Y: PRINT : Y=0
195 INPUT "DRUECKE RETURN";DUMMY$: RETURN
196 DATA 82,84,85,86,87,89,90,91,93,94,95,97,99,
    100,101,102,103,105,106
197 DATA 107,109,111,112,114,116,117,122,127,135
    ,145,149,152,154,162
198 DATA 163,164,165,167,168,170,171,173,175,177
    ,179,180,181,183,185
199 DATA 187,188,189,190,192,194,195,197,208,213
    ,225,242,244,246,248
200 DATA 250,251,253,254,255,257,259,261,263,265
    ,267,268,269,271,273
201 DATA 275,276,277,284,286,288,290,297,299,301
    ,303,307,311,313,317
202 DATA 322,326,330,332,334,336,337,339,341,343
    ,345,346,347,349,351
203 DATA 353,354,355,362,364,366,367,369,370,372
    ,374,376,379,389,393
204 DATA 397,418,419,421,422,423,425,427,428,429
    ,431,433,435,442,443
205 DATA 444,445,446,448,449,450,451,452,454,456
    ,463,465,471,475,477
206 DATA 494,496,498,499,500,501,503,505,507,509
    ,510,511,513,514,515
207 DATA 517,522,523,524,526,527,528,530,532,533
    ,534,536,541,545,570
208 DATA 576,577,579,581,582,583,585,586,588,589
    ,590,591,592,594,595,596,597
209 DATA 602,604,605,606,607,608,612,614,617,619
    ,623,630,637,642,646
210 DATA 650,652,654,655,657,659,660,661,663,665
    ,667,668,670,672,673,674
211 DATA 675,682,684,686,688,689,690,701,703,705
    ,708,717,724,732,733,735,736
212 DATA 738,739,745,746,748,749,750,751,752,754
    ,755,757,762,763,764,766
213 DATA 767,768,770,773,778,781,783,792,794,797
    ,804,810,811,813,814
214 DATA 815,817,818,820,821,823,825,826,827,828
    ,829,830,834,836,837,842
215 DATA 844,846,848,853,857,860,863,872,874,877
    ,882,884,886,888,889
216 DATA 890,892,893,895,896,897,899,900,902,903
    ,904,905,907,908,909,911
217 DATA 912,914,915,917
218 DATA ,4,1,8,1,1,1,1,8,1,1,1,8,1,8,1
219 DATA 1,1,1,1,8,1,1,1,8,1,8,1,8,1,1,5,
220 DATA ,2,,2,,,,,2,,,,2,,,,2,,2,
221 DATA ,,,,2,,,,2,,2,,,2,,2,,,2,
222 DATA ,2,,3,1,1,5,,3,8,1,1,11,1,5,,7,1,11,1
223 DATA 1,1,8,1,9,,4,1,6,,3,5,,2,,3,8,1,9,
224 DATA ,2,,,,,2,,,2,,,2,,2,,2,,
225 DATA ,,2,,2,,2,,,,,2,,2,,,2,,2,
226 DATA ,7,1,8,1,8,10,5,,7,1,1,9,,3,1,11,1,11,1
227 DATA 8,1,11,1,9,,7,1,1,1,8,10,8,10,8,1,10,1,
    9,
228 DATA ,2,,2,,2,,2,,2,,,,2,,2,
229 DATA 2,,2,,2,,2,,,,2,,2,,2,,,,2,
230 DATA ,7,1,9,,2,,2,,2,,4,10,8,1,8,6,,2,
231 DATA 2,,2,,7,1,6,,4,1,9,,2,,3,1,5,,2,
232 DATA ,2,,7,1,9,,3,8,6,,2,,2,,2,,,2,
233 DATA 2,,2,,2,,,,2,,2,,2,,,,7,1,9,
234 DATA ,2,,2,,2,,,2,,,2,,2,,2,,4,6,
235 DATA 7,1,10,1,11,1,8,1,6,,7,1,9,,4,1,9,,2,
```

```
236 DATA ,7,1,10,1,10,1,8,10,1,1,10,1,11,1,11,1,
    9,,
237 DATA 2,,,,2,,2,,,,2,,2,,2,,7,1,9,
238 DATA ,2,,,,,,2,,,,,,2,,2,,7,1,1
239 DATA 10,1,5,,2,,7,1,8,1,6,,7,1,6,,2,,2,
240 DATA ,7,1,1,1,8,1,10,1,8,1,8,1,6,,2,,2,,
241 DATA ,,2,,2,,2,,2,,,,2,,,,2,,2,
242 DATA ,2,,,,2,,,,2,,2,,,,2,,3,8,1
243 DATA 5,,3,1,9,,3,8,10,1,1,1,10,8,1,1,10,1,9,
244 DATA ,7,1,8,1,10,1,1,1,9,,7,1,8,1,9,,,2,
245 DATA 2,,,,2,,,2,,,,,,2,,,,,2,
246 DATA ,2,,2,,,,,,7,1,9,,2,,3,5,,2,
247 DATA 3,1,5,,7,1,8,10,1,5,,4,1,10,1,1,5,,2,
248 DATA ,2,,7,1,5,,4,1,6,,2,,2,,,2,,2,
249 DATA ,,2,,2,,2,,,2,,2,,,,,7,1,9,
250 DATA ,2,,2,,2,,2,,,,7,1,10,8,1,10,8,10,1
251 DATA 5,,2,,2,,3,5,,3,1,10,1,8,1,1,9,,2,
252 DATA ,7,1,6,,7,1,10,1,8,1,9,,,2,,,2,,
253 DATA 7,1,11,1,9,,,2,,,,,,2,,,2,,2,
254 DATA ,2,,,,2,,,,2,,3,5,,3,1,8,6,,4
255 DATA 6,,2,,7,1,1,10,1,1,1,5,,2,,4,6,,2,
256 DATA ,7,1,5,,7,1,8,1,9,,,2,,,,2,,,2
257 DATA ,,2,,2,,,,,,,7,1,9,,2,,,2,
258 DATA ,2,,2,,2,,2,,3,1,8,6,,4,1,6,,4,6
259 DATA ,4,6,,3,1,8,1,1,1,8,6,,2,,3,5,,2,
260 DATA ,2,,2,,2,,2,,,,2,,,,2,,,2,
261 DATA ,2,,,,,2,,,,2,,,,2,,,2,,2,
262 DATA ,3,1,10,1,10,1,10,1,1,1,10,1,1,10,1,1,1
    ,10,1
263 DATA 1,10,1,1,1,1,10,1,1,1,10,1,1,10,1,1,10,
    1,6,
264 DATA 169,46,162,250,157,255,11,157,249,12,15
    7,243,13,157,237,14,202,208
265 DATA 241,96
270 VOL 7: SOUND 1,800,15
271 PRINT "ZIEL IST ES, MOEGLICHST VIELE PUNKTE
    ZU"
272 PRINT "FRESSEN, OHNE SICH VON DER SPINNE FAN
    -"
273 PRINT "GEN ZU LASSEN. DIE ANZAHL DER PUCKS"
274 PRINT "WIRD RECHTS UNTEN ANGEZEIGT."
275 PRINT "DER PUCK KANN MIT DEM JOYSTICK IN ALL
    E"
276 PRINT "RICHTUNGEN BEWEGT WERDEN. EINE GEWAEH
    LTE"
277 PRINT "RICHTUNG WIRD BEIBEHALTEN, SOLANGE DI
    ES"
278 PRINT "MOEGLICH IST. DER JOYSTICK KANN DABEI
     IN"
279 PRINT "NEUTRALSTELLUNG STEHEN."
280 PRINT "DURCH DAUERBETAETIGUNG ODER KURZES AN
    -"
281 PRINT "TIPPEN KANN EINE NEUE RICHTUNG GEWAEH
    LT"
282 PRINT "WERDEN, DIE BEI DER NAECHSTEN GELEGEN
    -"
283 PRINT "HEIT EINGESCHLAGEN WIRD."
284 PRINT "WURDE KEINE NEUE RICHTUNG GEWAEHLT,"
285 PRINT "BLEIBT DER PUCK STEHEN, WENN EINE BEW
    E-"
286 PRINT "GUNG IN DER ALTEN RICHTUNG NICHT MEHR
    "
287 PRINT "MOEGLICH IST."
290 PRINT "GELINGT ES ABZURAEUMEN, GIBT ES EINEN
    "
291 PRINT "PUCK ALS BONUS."
292 PRINT "BIS ZUR VIERTEN RUNDE WIRD DIE SPINNE
    "
293 PRINT "IM VERHAELTNIS ZUM PUCK VON RUNDE ZU"
294 PRINT "RUNDE ETWAS SCHNELLER."
295 PRINT "SIND ALLE PUCKS GEFANGEN, WIRD DIE ZA
    HL"
296 PRINT "DER GEFRESSENEN PUNKTE ANGEZEIGT."
297 INPUT "DRUECKE RETURN! ";D$
298 RETURN
299 END
```

Listing 1. Das Listing zu »Puck« für den C16 (Schluß)

Cave – Sternenkampf im Labyrinth

Ihr Auftrag ist ein »Himmelfahrtskommando«. Doch vom Erfolg Ihres Unternehmens hängt das Überleben einer ganzen Flotte ab.

Auf dem Weg zurück zum Heimatplaneten Erde wird Ihr Geschwader durch feindliche Raumschiffe, Raketen und Minen bedroht (Bild). Die zivilen, unbewaffneten Raumschiffe folgen Ihrem Kampfjäger in sicherer Entfernung. Ihre Aufgabe ist es, ihnen mit drei Raumschiffen den Weg durch das feindliche Territorium zu bahnen. Besonders gefährlich sind die Raketen, die im Labyrinth auf Ihr Mutterschiff warten. Für die Zerstörung dieser stationierten Objekte gibt es deshalb auch die meisten Punkte. Feindliche Raumschiffe sind etwas leichter zu vernichten, deren Abschuß wird entsprechend geringer bewertet. Feindliche Minen, die sich häufig auch direkt in Ihrer Flugbahn befinden, dürfen auf keinen Fall abgeschossen werden, da die Detonation auch Ihr eigenes Raumschiff zerstören würde. Je tiefer Sie in das Höhlenlabyrinth eindringen, desto enger und schwerer wird es passierbar. Wenn es zum Manövrieren Ihres Schiffes zu eng wird, müssen Sie versuchen, sich einen Weg durch die Mauer zu schießen.

Zerstörte Raumschiffe und Raketen werden mit Pluspunkten belohnt. Wird Ihr Raumschiff jedoch durch Minen oder einen Zusammenprall zerstört, erhalten Sie einen Punkteabzug. Wieviel Punkte Ihnen abgezogen werden, hängt von der Anzahl bisher abgefeuerter Schüsse ab (Faktor 10). Wenn Sie beispielsweise bisher 30mal geschossen haben, verringert sich Ihr Punktekonto um 300. Wenn Sie bis zur Zerstörung Ihres letzten Raumschiffs genügend Punkte gesammelt haben, können Sie sich in die Liste der »Top Ten« eintragen. Finden Sie die optimale Strategie heraus.

(Alexander Jung/kn)

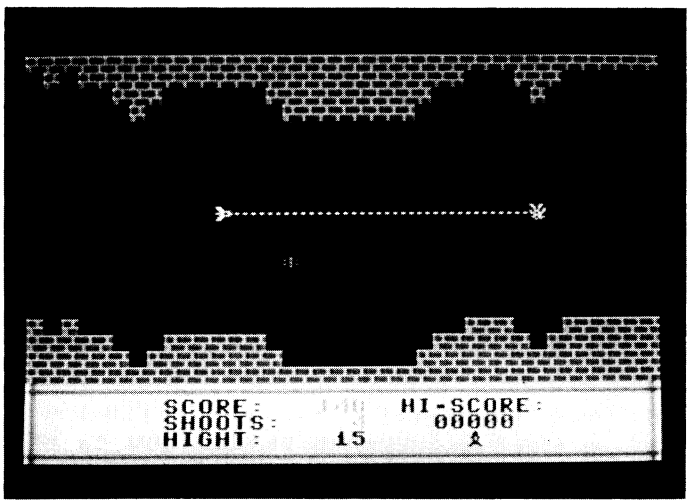

Bild. »Cave« besticht vor allem wegen seiner guten Grafik