Dynamic Games for the ZX Spectrum

TIM HARTNELL



20 dynamische spellen voor de

zx spectrum

Tim Hartnell

20 dynamische spellen voor de zx spectrum

Colofon

Oorspronkelijke titel: Dynamic games for the ZX Spectrum
Oorspronkelijke uitgave: Sinclair Browne Ltd. Londen
© Tim Hartnell –MCMLXXXIII
© Ned. editie: Delfia Press B.V. – MCMXXXV
Vertaling: Rob de Jong
Vormgeving: Han Janssen
Zetwerk: Assist, Goes
ISBN 906449 0376

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Hartnell, Tim

20 dynamische spellen voor de ZX Spectrum / Tim Hartnell; [vert. uit het Engels door Rob de Jong]. –
Rijswijk: Delfia Press
Vert. van: Dynamic games for the ZX Spectrum. – Londen: Sinclair Browne, 1983.
Met cassette

Inhoud

Voorwoord 6

Videospelletjes 9
Level-5 9
Circus 16
Jogger 21
Mijnenveger 31
Copter 38
Mazurka 43
Alphabattle 50
Smashout 55
3-D Drive 59
Norway Panic 62
Death Race 2000 65
Clone Crazy 68

Bordspelen 79
Pirandello 79
Dammen 86
Boter, kaas en eieren 104
Beginners Schaken 108

Avontuur/Simulatie 143
De Wraak van Kasteel Angst 143
Joe Kapitalist Junior 178

Hersenbrekers 191
De put 191

Programmeertips 197
Verbeter uw programma's 197
Verhogen van de programmasnelheid 201

Appendix 205
Gotische generator 205

Voorwoord

In vele opzichten is de computer een ideale tegenstander in spelletjes.
De computer kan niet opscheppen over overwinningen, is galant bij verlies, onvermoeibaar, geduldig en altijd beschikbaar. Hoewel de computer niet het menselijke tegenspel kan bieden dat spelletjes vaak zo leuk maakt, hij is toch de perfecte partner om uren lang mee door te brengen.
Waarom u ook denkt uw Spectrum gekocht te hebben, ik weet zeker, dat u in ieder geval enige tijd op uw computer doorbrengt met het spelen van spelletjes. En dat is het punt waarop dit boek u van pas komt. In dit boek vindt u een grote verscheidenheid aan spelletjes, van bordspellen zoals SCHAKEN en DAMMEN, videospellen zoals JOGGER en LEVEL-5, tot het grote adventurespel, DE WRAAK VAN KASTEEL ANGST. We hebben gemikt op nuttige spelletjes die uw belangstelling lang vast kunnen houden, bovendien zijn er ook spelletjes die uw programma- en spel-ideeën bieden, die u bij uw eigen computerwerk weer kunt gebruiken.
Om u te helpen in het gebruik van de programma's hebben we twee dingen gedaan: ten eerste zijn alle programma's in dit boek voor u ingetypt en uitgetest. Deze kant-en-klare programma's vindt u op de bijgaande cassette. De programma's op de cassette zijn volledig getest en klaar om gebruikt te worden. Ten tweede vindt u in het boek de listings van alle programma's, en in de tekst worden de programma's regel voor regel verklaard. Hierbij worden de 'trucks' van de programmeurs uitgelegd, en ook ziet u hier, hoe u deze technieken in uw eigen programma's kunt gebruiken. Sommige pro-

Het merendeel van de programma's is geschreven door Tim Rogers, Paul Toland en Tim Hartnell. Tim Rogers en Paul zijn beiden studenten (Tim in Londen, Paul in Belfast), en zij verdienen een felicitatie voor de kwaliteit en originaliteit van hun werk, hetgeen weerslag vindt in de programma's die voor dit boek geselecteerd zijn. Een volledige lijst van programma-auteurs vindt u achter in dit boek.

De Nederlandse bewerking van de programma's is van Karin Kaptein. Zij verzorgde ook de invoer op cassette en de programma-listings en -voorbeelden.

En nu is het tijd voor u en uw Spectrum!

Veel plezier met deze spelletjes, die nadrukkelijk de benaming 'Dynamisch' verdienen!

Videospelletjes

LEVEL-5

Ons eerste programma is gelijk al een erg spannend spel. LEVEL-5 is een snelbewegende versie van een bekend videospel.
Het spelgebied wordt gevormd door vijf niveaus, die onderling verbonden zijn door een aantal ladders. Vier monsters beginnen op het onderste niveau, en klimmen langzaam naar boven. Uw taak is de monsters gevangen te nemen in de kerker onder het onderste niveau. Het voornaamste probleem is, dat de monsters altijd naar boven klimmen. De enige manier waarop u ze dus naar beneden kunt krijgen, is door het graven van een gat in één van de niveaus, zodat de monsters naar beneden vallen. U kunt een gat graven, door op de toets '0' te drukken. Dit werkt echter alleen, als er geen ander gat op het scherm aanwezig is. Ieder gat bestaat slechts voor een korte tijd, daarna wordt het automatisch weer opgevuld.
U kunt de bewegingen besturen met de volgende toetsen: '5' (links), '8' (rechts), '6' (naar boven) en '7' (naar beneden). Het spel eindigt, als u alle monsters gevangen heeft. Als u dit voor elkaar heeft, geeft het programma aan, hoelang u er over gedaan heeft. Het programma wordt echter voortijdig beëindigd, als één van de monsters u te pakken krijgt. Zoals vaak het geval is bij Spectrum programma's, wordt de kwaliteit van de grafische weergave niet erg goed benaderd door de print-uitvoer, die hieronder afgedrukt is. Als u het programma probeert, zult u versteld staan van de snelheid en de versla-

Regel 1 laat zien, welke grafische tekens er op welke toets gebruikt worden, de stenen staan hierbij op de 'A', een gedeelte van de ladder op 'B', uzelf (het rennende figuurtje) op toets 'C' en de 'aantrekkelijke' monsters op toets 'E'. Regel 25 geeft de besturing over een regel 9100, nadat de INK op wit gezet is, en de PAPER en BORDER op blauw. In deze subroutine worden de instructies op het scherm gezet. Regel 9130 wacht, totdat u iets ingetoetst heeft, en maakt daarna het scherm schoon. Daarna ziet u het laatste stuk van de instructies, gevolgd door de boodschap: 'Veel geluk!!'. De RETURN-instructies op regel 9160 geven de besturing weer terug aan regel 30, waar de subroutine op regel 9000 aangeroepen wordt. Regel 9000 begint met een RESTORE van de DATA-pointer naar zichzelf (eigenlijk naar de eerste DATA-opdracht na het getal, dat volgt op RESTORE). Dit mechanisme zorgt er voor, dat als de I- en J-loops uitgevoerd worden (regel 9001 tot 9040), deze verwijzen naar de getallen, die in de grafische tekens gePOKEd moeten worden vanaf regel 9050 tot regel 9090 en dus niet vanaf de eerste DATA-opdracht op regel 1070.
Nadat het programma de grafische karakters heeft gedefinieerd, wordt de uitvoering vervolgd (door de tweede opdracht op regel 30) vanaf regel 700. Hier worden de vijf niveaus afgedrukt, voor ieder niveaus één rij 'bakstenen'. Regel 910 is een RESTORE voor zichzelf, hier wordt net als bij regel 9000 de DATA-pointer naar de eerste DATA-opdracht verplaatst, die op de RESTORE volgt. In dit geval is dit regel 1030. Die bevat informatie, die gebruikt wordt in de I- en J-loops (1000 tot 1050) om de ladders op het scherm te plaatsen. Regel 1055 maakt de onderste regel van het scherm schoon, en het programma keert terug naar regel 40, waar de echte

in het tweede gedeelte van deze regel, die de monsters bevat, voordat deze op hun beginpositie worden geplaatst met de loop op regel 60. De H-array (waar het 'gat' opgeslagen wordt), wordt gedimensioneerd op regel 70, en zowel de INK als de PAPER wordt in regel 80 op 8 gezet. Daarna wordt uiteindelijk het mannetje op het scherm afgedrukt in regel 90.
De variabele TIME (denk aan Paul's beste score, 60!) wordt in regel 100 op nul gezet, en vervolgens in regel 395 verhoogd, daar wordt ook de nieuwe waarde van TIME op het scherm afgedrukt. De titel van het programma wordt in regel 110 aan de bovenzijde van het scherm geplaatst, waarbij ruimte open wordt gelaten om later TIME af te drukken, en de flag FREE wordt in regel 190 op nul gezet.
De I-loop bestuurt de positie en de bewegingen van de monsters. Deze loop is van regel 200 tot 390 en bevat eigenlijk alle belangrijke acties in het programma. Hieronder vallen (o.a.) het bewaken van de vertikale beweging van de monsters (regels 205), het opnieuw afdrukken (regel 220), het controleren van hun direkte omgeving (regels 225 en 230), en een sprongopdracht naar regel 430 als ze u te pakken hebben, of naar regel 280 als ze een ladder opklimmen. De horizontale coördinaat van de monsters wordt verhoogd in regel 240, en de daarop volgende regel verandert de richting, waarin een monster zich beweegt, als hij aan het einde van een niveau is gekomen. 'Weg Monster' wordt uitgevoerd op regel 270, als een monster in een gat loopt, en het monster wordt in regel 280 op zijn nieuw positie afgedrukt.
Een gat bestaat gedurende 40 'cycli' van het programma, en regel 295 vult een gat na afloop van deze periode. Regel 300 leest het toetsenbord, en springt over het grootste gedeelte van de volgende routine indien er

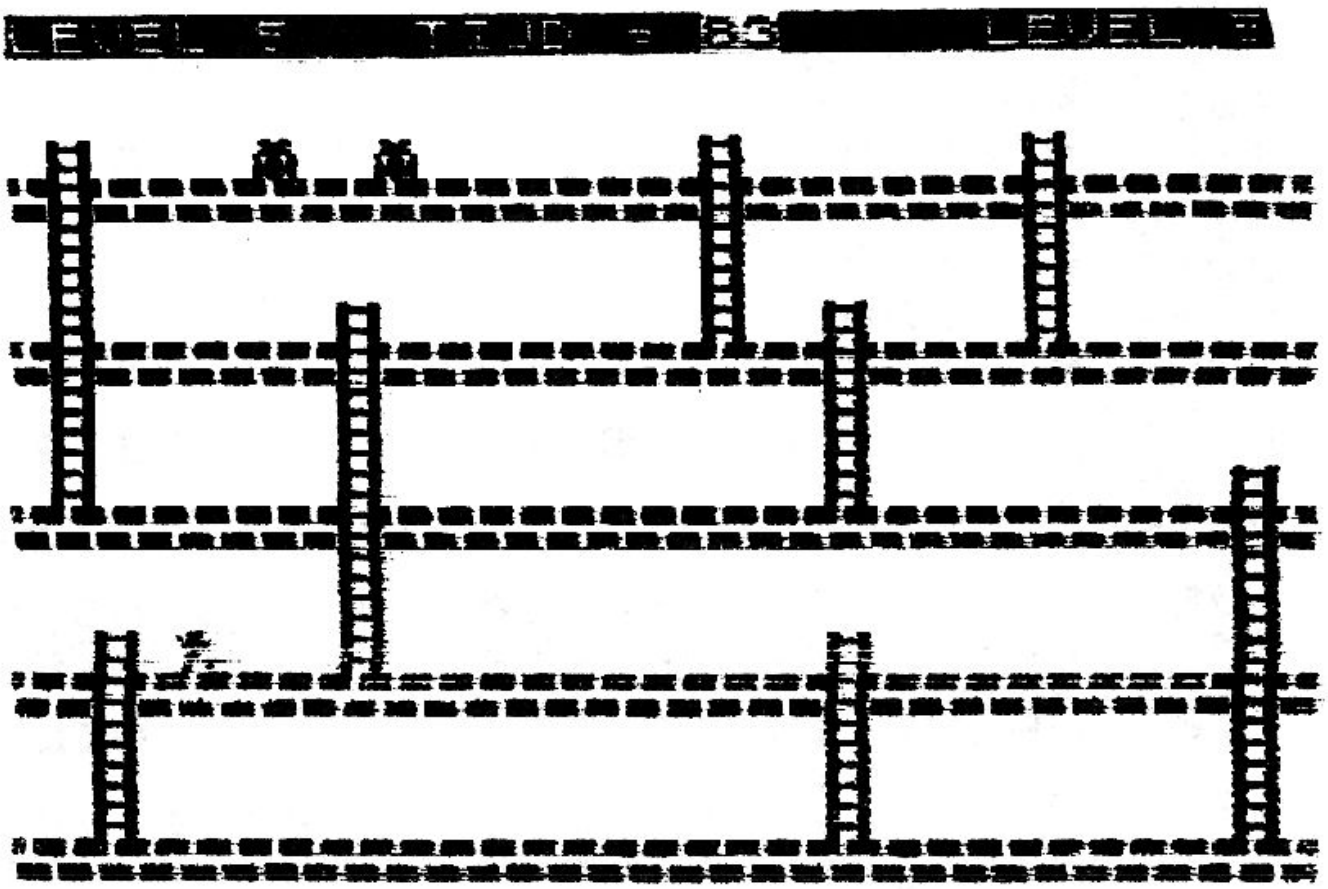
geen andere gaten op het scherm heeft gemaakt, en de coördinaten van een nieuw gat worden ingevuld, volgens de zelfde coördinaten waar het mannetje op dat moment staat, en één regel lager, voordat erop die positie een gat afgedrukt wordt, daarna volgt een sprong naar regel 390.
De computer weet, dat als u een nul ingetoetst heeft op regel 315, u geen 5, 6, 7, of 8 in de volgende regels ingetoetst kunt hebben. Het programma wordt daarom maximaal versneld door het overslaan van de regels, waarvan bekend is, dat deze regels dit keer niet uitgevoerd behoeven te worden. Regel 317 bekijkt, of u geen andere toetsen intikt, dan tussen '4' en '9', en als dit niet het geval is, wordt er naar regel 390 gesprongen. de volgende regel plaatst het mannetje op de juiste positie, en regel 325 POKEd het ATTRibute-adres van Uw positie met 15, om u zo van uw huidige posite te verwijderen, voordat u opnieuw afgedrukt wordt in regel 390 (ATTRibuten zijn opgeslagen na de DISPLAY FILE, vanaf adres 22528).
Regel 327 controleert, of u op een looppad bent, als deze routine ontdekt, dat dit niet het geval is, wordt er naar regel 350 gesprongen om te controleren, of u op een ladder naar boven of beneden wilt bewegen. N, de waarde die gebruikt wordt om de ATTRibute van uw positie te POKEn, wordt op uw huidige ATTRibute gezet, voordat u op regel 380 afgedrukt wordt. Regel 330 wordt bereikt, als u op een looppad bent, waar het toetsenbord u naar wens verplaatst, al naar gelang u een '5' of een '8' ingedrukt hebt. Regel 340 zorgt er voor, dat u niet van het scherm af kunt lopen. Regel 390 tenslotte start de gehele cyclus opnieuw. Nadat u de I-loop vier maal heeft doorgemaakt, wordt de variabele FREE bekeken, om te zien of alle monsters gevangen zijn .

en op regel 430 wordt opgemerkt, dat ze u te pakken hebben. Er wordt dan nog de mogelijkheid geboden, om een nieuw spel te starten, op regel 500 wordt ieder antwoord buiten 'n' en 'N' gebruikt, om een nieuw spel te starten.

```
    1 REM UDGA=A    B=B   C=C   E=E
    2 REM
   10 REM LEVEL 5    P.TOLAND
   20 INK 7: PAPER 1: BORDER 1: C
LS
   25 GO SUB 9100
   30 GO SUB 9000: GO SUB 700
   40 LET X=14: LET Y=5: LET C=1
   50 LET M=4: DIM M(M,3)
   60 FOR I=1 TO M: LET M(I,1)=19
: LET M(I,2)=I*3: LET M(I,3)=1-I
NT (RND+.5)*2: PRINT AT 19,I*3;"
E": NEXT I
   70 DIM H(3): LET H(1)=0: LET H
(2)=0: LET h(3)=0
   80 INK 8: PAPER 8
   90 PRINT AT Y,X; INK 5;"C"
  100 LET N=15: LET TIME=0
  110 PRINT AT 0,0; INVERSE 1;"LE
VEL 5    TIJD =              LEVEL 5"
  190 LET FREE=0
  200 FOR I 1 TO 4
  205 IF M(I,1)=21 THEN GO TO 295
  207 LET FREE=1
  210 LET D=M(I,1): LET A=M(I,2)
  220 PRINT AT D,A; OVER 1;"E"
  225 IF ATTR (D-1,A)=13 THEN GO
TO 430
  230 IF ATTR (D-1,A)=14 THEN LET
  D=D-1: GO TO 230
  240 LET A=A+M(I,3)
  250 IF A=0 OR A=31 THEN LET M(I
  3)=-M(I,3)
  260 IF ATTR (D,A)=13 THEN GO TO
  430
  270 IF ATTR (D+1,A)=15 THEN LET
  D=D+4: BEEP .1,30: IF D>21 THEN
  LET D=21
  280 PRINT AT D,A; OVER 1;"E"
  290 LET M(I,1)=D: LET M(I,2)=A
  295 IF H(3)>0 THEN LET H(3)=H(3
)+1: IF H(3)=40 THEN PRINT INK 2
; PAPER 7;AT H(1),H(2);"A": LET
```

```
TR (Y+1,X)=58 THEN LET H(1)=Y+1:
 LET H(2)=X: LET H(3)=1: PRINT P
APER 1; INK 7;AT H(1),H(2);" ":
GO TO 390
 317 IF I$<"5" OR I$>"8" THEN GO
 TO 390
 320 PRINT AT Y,X; OVER 1;"C"
 325 POKE 22528+Y*32+X,N
 327 IF (Y+1)/4<>INT ((Y+1)/4) T
HEN GO TO 350
 330 LET X=X+(I$="8")-(I$="5")
 340 LET X=X+(X<0)-(X>31)
 350 IF I$="7" AND ATTR (Y-1,X)=
14 THEN LET Y=Y-1
 360 IF I$="6" AND ATTR (Y+1,X)=
14 THEN LET Y=Y+1
 370 LET N=ATTR (Y,X)
 380 PRINT AT Y,X; OVER 1; INK 5
;"C"
 390 NEXT I
 395 LET TIME=TIME+1: PRINT AT 0
,17;TIME
 400 IF FREE THEN GO TO 190
 410 PRINT OVER 1;AT 9,5;"ALLE M
ONSTERS GEVANGEN": BEEP 3,30
 420 GO TO 440
 430 PRINT AT Y,X;"E"; PAPER 1;
BRIGHT 1; OVER 1;AT 9,0;"DE MONS
TERS HEBBEN U TE PAKKEN !": BEEP
 3,-20
 440 PRINT AT 13,10;"NA ";TIME;"
 SEC."
 500 INPUT "NOG EEN SPELLETJE ?"
;A$
 510 IF A$<>"N" AND A$<>"n" THEN
 RUN
 699 STOP
 700 FOR I=1 TO 5
 800 PRINT INK 2; PAPER 7;AT I*4
,0;"AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA"
 900 NEXT I
 910 RESTORE 910
1000 FOR J=1 TO 8
1010 READ X,Y,L
1020 FOR I=0 TO L
1030 PRINT INK 6;AT Y+I,X;"B"
1040 NEXT I
1050 NEXT J
1055 PRINT AT 21,0; PAPER 0,,
```

```
9000 RESTORE 9000
9001 FOR I=0 TO 4
9010 FOR J=0 TO 7
9020 READ N: POKE USR "A"+I*8+J,
N
9030 NEXT J
9040 NEXT I
9045 RETURN
9050 DATA 207,207,207,0,0,126,12
6,126
9060 DATA 195,255,195,195,195,25
5,195,195
9070 DATA 56,144,124,20,48,47,34
,96
9080 DATA 28,9,62,40,12,116,68,6
9090 DATA 102,60,126,153,187,255
,255,153
9100 PRINT INVERSE 1;"     LEVEL
5          LEVEL 5          "
9110 PRINT ,,"Er zijn 4 monsters
 die over uw  scherm bewegen, zi
j proberen    naar de top te kom
en. Er zijn 5 niveaus verbonden
door ladders "
9120 PRINT "en uw taak is, de mo
nsters        gevangen te nemen in
 de kerker   onder het onderste n
iveau."
9121 PRINT "De enige manier waar
op u de       monsters naar benede
n kunt        krijgen, is door een
 gat te       graven in een van de
 niveaus,"
9122 PRINT "waar de monsters dan
 door         vallen. Echter, iede
r gat sluit  zich na enige tijd e
n er kan      maar een gat tegelij
k bestaan."
9123 PRINT "U sterft als een mon
ster u te     pakken krijgt en uw
opdracht is   de 4 de monsters zo
snel          mogelijk gevangen te
 nemen."
9125 PRINT FLASH 1;"Druk een toe
ts om door te gaan."
9130 PAUSE 0: CLS
9140 PRINT "Gebruik 5,6,7 en 8 o
m te bewegenen '0' (NUL) om een
gat te        graven. VEEL GELUK.,
.             U zult het nodig heb
ben!!"
9150 PAUSE 200: CLS
```



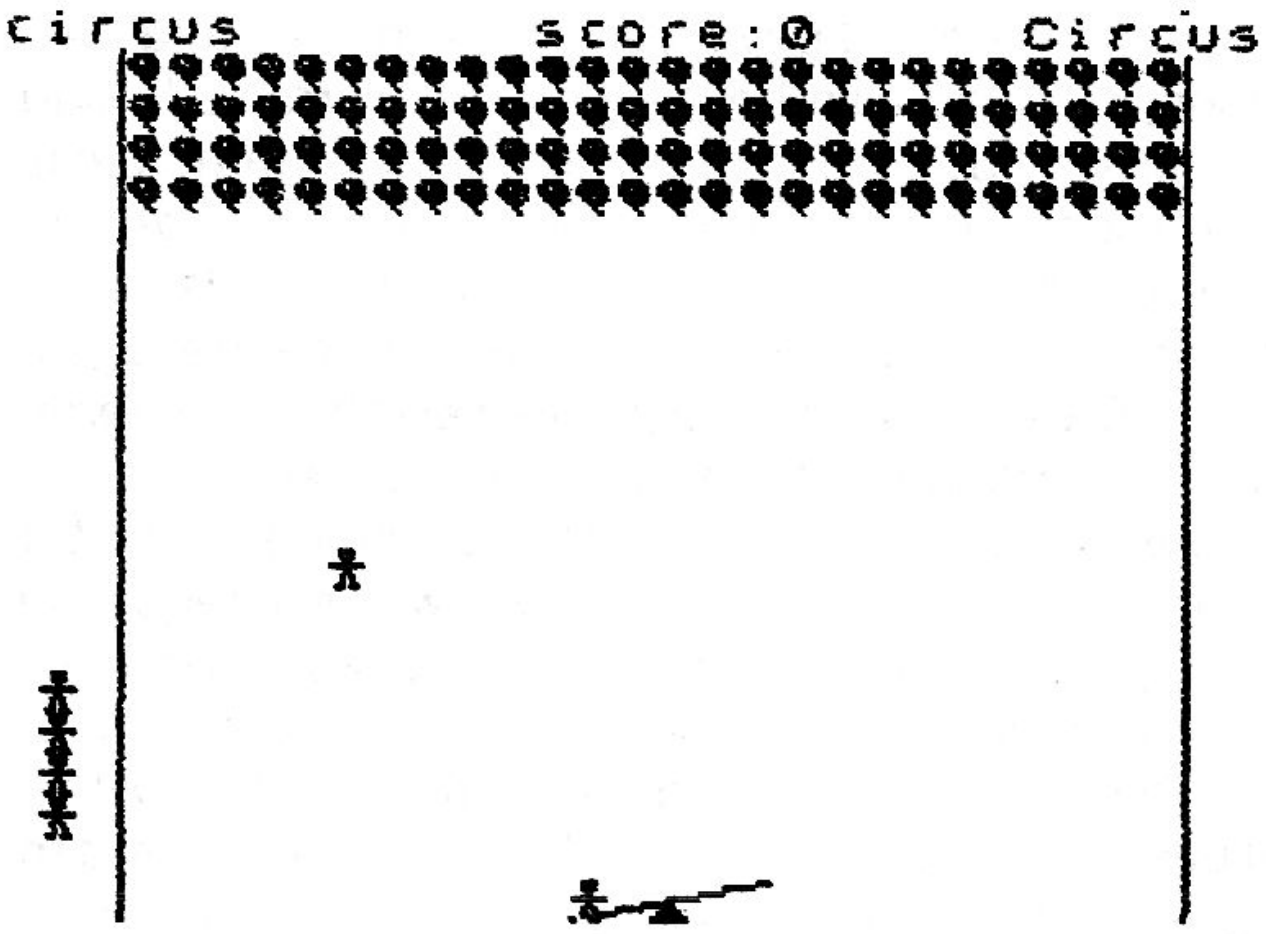
Circus

Circus is een videospel, dat wel iets lijkt op 'Breakout', maar waarbij een veel interessantere omgeving gebruikt wordt, en ook aan de spelers een grotere uitdaging geboden wordt. U moet trachten de rijen ballonnen aan de bovenzijde van het scherm uiteen te laten spatten door omhoog te springen met twee clowns, die aan de onderzijde van het scherm op een wip staan. U kunt de wip besturen met de toetsen '5' en '8'. Als u de toets '1' indrukt, wordt de clown op de wip aan de andere zijde van de wip geplaatst. De vallende clown moet op het vrije eind van de wip vallen, liefst zo ver mogelijk op het uiteinde, om de andere clown in de lucht te laten schieten. Als de clown niet ver genoeg op het uiteinde van de plank landt, dan komt de andere clown niet hoog genoeg om de ballonnen te raken.

Het spelletje is zó moeilijk om te leren, dat we de originele versie aangepast hebben. U krijgt nu een aantal 'reserve'-clowns, zodat u een kans krijgt om een redelijk

HET CIRCUS IS NAAR DE STAD GEKOMEN MET ALS HOOFDATTRAKTIE 'DE CLOWNS OP DE WIP'. TWEE CLOWNS LANCEREN ELKAAR OP DE WIP, OM ZO DE BALLONNEN, DIE IN DE LUCHT HANGEN, DOOR TE PRIKKEN. DE DALENDE CLOWN MAG NIET OP DE GROND OF OP DE VERKEERDE KANT VAN DE WIP LANDEN. ALS HIJ TE DICHT BIJ HET MIDDEN LANDT ZAL DE ANDERE CLOWN NIET HOOG GENOEG KOMEN. HET CIRCUS HEEFT MAAR 6 CLOWNS. U BEDIENT DE WIP MET DE TOETSEN 5 EN 8. U KUNT DE CLOWN OP DE WIP VAN ZIJDE LATEN VERANDEREN DOOR 1 IN TE TOETSEN. GEBRUIK N OM TE STOPPEN EN ELKE ANDERE TOETS OM TE BEGINNEN.

En zo ziet het programma er uit, als het uitgevoerd wordt:



De listing laat zien, welke grafische tekens er gebruikt worden (regel 5), die 'A' is het midden van de wip, de 'B' voor de clown en de 'C' voor de ballon. Nadat de random-getallengenerator in regel 20 ingesteld is, gaat het programma naar regel 80, waar OVER op 1 gezet

maakt, hiermee wordt de nieuwe PAPER kleur in gebruik genomen.
Nu wordt het programma voortgezet op regel 900, waar twee andere subroutines aangeroepen worden (regels 1100 en 1020). De eerste hiervan (vanaf regel 1100) print de titel van het programma en de instructies, de tweede (vanaf regel 1020) definieert de grafische tekens. Nadat uit deze subroutines is teruggekeerd, wordt het frame rond het scherm getekend (regels 902 tot 905), en de ballonnen worden op hun plaats gezet door de loop van regel 910 tot regel 930). U ziet dat deze loop ook de 'reserve'-clowns aan de zijkant van het scherm neerzet.
In de volgende twee regels worden een aantal variabelen gedefinieerd. SC bevat de score van het aantal ballonnen, dat u geraakt hebt. Dit aantal wordt verhoogd in regel 180. SS is de horizontale positie van de basis van de wip, die in regel 50 gebruikt wordt om de wip en de clown te plaatsen, in de volgende regels wordt de plank van de wip getekend. Merk op, dat de regels 40 en 60 een subroutine zijn, die verschillende keren vanuit het programma worden aangeroepen.
M is een teller, die het aantal clowns bijhoudt, dat u tot nu toe gebruikt hebt. Deze variabele wordt in regel 300 verhoogd en aangeroepen vanuit regel 230. Het array B wordt gebruikt om 26 spaties af te drukken, nadat een clown gevallen is. Dit gebeurt in regel 330. OD en AN besturen de positie van de clown. Als deze waarden negatief gemaakt worden (regel 107), dan wordt hierdoor de clown op de plank naar de andere zijde gebracht. OD en AN worden gebruikt op regels 50, 260 en 280.
De variabele CA wordt gebruikt, om te zien, of AN gewijzigd moet worden (zie regels 95 en 270). HL wordt in regel 250 ingesteld, en in regel 160 gebruikt om te zien,

sitie van de clown, die wordt gebruikt in regel 130 en verhoogd in regel 140 met de waarde A. De variabele A wordt in regel 280 veranderd, om de horizontale beweging van de clown om te keren. Y (wordt ingesteld in regel 950) is de start-coördinaat van de clown. Deze waarde ligt tussen de 5 en 8. Regel 100 leest het toetsenbord voor '5' en '8'. Hiermee wordt de verticale coördinaat van de basis van de wip gewijzigd. Deze waarde wordt in regel 105 gecontroleerd, om er zeker van te zijn, dat u de wip niet te ver naar links of naar rechts kunt bewegen.

```
  5 REM U.D.G.  ▲=A ⁂=B ♥=C
  7 REM
 10 REM CIRCUS.  P.TOLAND.
 15 REM ------------------
 20 RANDOMIZE
 40 GO TO 80
 50 PRINT AT 21,ss;"▲";AT 21,ss
+od;"⁂"
 60 PLOT ss*8-16,7*(an=1): DRAW
 38,7*-an: RETURN
 80 OVER 1: INK 0: PAPER 7: BOR
DER 2: CLS
 85 GO SUB 900: GO SUB 50
 90 GO SUB 50
 95 IF ca THEN LET an=-an: LET
ca=0
100 LET ss=ss+(INKEY$="8")-(INK
EY$="5")
105 LET ss=ss+(ss=4)-(ss=27)
107 IF INKEY$="1" THEN LET an=-
an
110 GO SUB 50
130 PRINT AT y,x;"⁂"
140 LET x=x+a: LET y=y+d
150 IF x=3 OR x=28 THEN LET a=-
a: BEEP .01,1
160 IF y<h( THEN LET d=-d
165 LET no=ATTR (y,x)
167 PRINT AT y,x;"⁂"
170 IF no=56 THEN GO TO 210
180 LET sc=sc+61-no: BEEP .01,1
0
185 PRINT AT y,x; OVER 0;"⁂";AT
```

```
 210  IF y<21 THEN GO TO 90
 220  LET dis=x-ss: BEEP .01,5
 230  IF dis<-2 OR dis>2 OR SGN d
is<>-an THEN GO TO 310
 250  LET ht=11-ABS dis*5
 260  LET x=ss+od
 270  LET od=dis: LET ca=1
 280  LET a=SGN od: LET d=-1
 290  GO TO 90
 310  LET m=m+1
 320  BEEP .2,-10: BEEP .7,-20
 330  IF m<5 THEN PRINT AT 21,3;
OVER 0;b$: GO SUB 950: GO SUB 50
: GO TO 90
 335  BEEP 1,-40: RUN
 340  PRINT AT 9,4; FLASH 1;"U HE
EFT HET VOOR ELKAAR!!MAXIMALE SC
ORE"
 345  BEEP 1,20: BEEP 2,30
 350  PAUSE 1: PAUSE 100
 360  RUN
 900  GO SUB 1100: GO SUB 1020
 902  PLOT 23,0: DRAW 0,167
 905  PLOT 232,0: DRAW 0,167
 910  FOR i=1 TO 4
 920  PRINT AT i,3; INK i;"▀▀▀▀▀▀
▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀";
 925  PRINT AT 15+i,1;"▚"
 930  NEXT i
 940  PRINT AT 0,0;"circus
score:0       Circus"
 945  LET sc=0: LET ss=16: LET m=
0: DIM b$(26)
 950  LET od=-2: LET ca=0: LET ht
=0: LET x=3: LET y=5+INT (RND*4)
: LET a=1: LET d=1: LET an=-1
 955  PRINT AT 15+m,1; OVER 0;" "
 957  FOR i=15+m TO y STEP -1
 960  PRINT AT i,1;"▚"
 970  BEEP .05,21-i
 980  PRINT AT i,1;"▞"
 990  NEXT i
1000  PRINT AT y,x;"▛"
1010  RETURN
1020  RESTORE 1070
1030  FOR i=0 TO 23
1040  READ n: POKE USR "a"+i,n
1050  NEXT i
1060  RETURN
1070  DATA 0,0,0,0,16,56,124,254
1080  DATA 56,56,16,254,16,40,40,
```

```
1110 PRINT "Het circus is in de
stad en de  hoofd-attraktie zijn
 de clowns  met de wip-wap act."
1111 PRINT "Twee clowns springen
 van de     wip-wap, om hoog in
de lucht    ballonnen kapot te p
rikken."
1115 PRINT "De vallende clown ma
g niet op   de grond of op de ve
rkeerde     zijde van de wip val
len."
1116 PRINT "Als hij te dicht bij
 het midden van de wip valt, dan
 springt de andere clown niet ho
og"
1117 PRINT "genoeg om de ballonn
en te raken.Het circus heeft sle
chts zes    clowns, dus wees voo
rzichtig."
1120 PRINT "U bestuurt de wip me
t de toetsen5 en 8. U kunt de cl
own op de   wip van zijde doen v
erwisselen  met de toets '1'."
1121 PRINT "Toets 'N' om te stop
pen of een  andere toets om te s
tarten."
1130 IF INKEY$="" THEN GO TO 113
0
1140 IF INKEY$="n" OR INKEY$="N"
 THEN STOP
1150 CLS
1160 RETURN
```

Jogger

U moet vijf vermoeide joggers helpen oversteken over een drukke zesbaans snelweg. Eén voor één geven de joggers hun leven in uw handen, terwijl u ze door het drukke verkeer leidt. U gebruikt hiervoor de toetsen '5' (links), '8' (rechts), '6' (omhoog) en '7' (naar beneden). Eén fout, en uw jogger is er geweest! En dan heeft u niet alleen het langsrazende verkeer, om u zorgen over te maken, maar bovendien de klok. Want iedere keer,

staan er weer vijf nieuwe joggers klaar, om door u naar de overzijde geleid te worden.
Zoals u op de print-out kunt zien, zijn er altijd vijf joggers op het scherm. Deze joggers wachten geduldig aan de onderzijde van het scherm, of ze zijn aan het oversteken en lopen dus op de weg, of staan aan de rechter-bovenzijde van het scherm, veilig aan de overkant. Aan de bovenzijde van het scherm wordt opgegeven hoeveel joggers u nog over moet helpen, en hoe vaak u de reis reeds gemaakt hebt. Op de tweede regel ziet u het aantal joggers, dat u veilig aan de overkant gebracht heeft. Deze waarde wordt verhoogd, zodra u weer een volgende jogger door het drukke snelverkeer heeft geloodst.
Als u het spel een beetje onder de knie heeft, kunt u ook de tweede versie proberen. In deze variatie stoppen de joggers nooit met lopen, hierdoor wordt een grote concentratie gevraagd om de jogger door het verkeer te leiden. U kunt dus beter de eerste versie goed onder de knie hebben, voordat u aan de tweede versie begint.
Het programma begint op regel 20 met een aanroep naar de subroutine op regel 410. Hier wordt de INK op blauw gezet, de PAPER op wit, de BORDER op rood en het scherm wordt vervolgens schoongemaakt. De A-loop vanaf regel 440 tot regel 500 definieert de zes grafische tekens. De eerste (grafische A, of ASCII teken nummer 144) is de jogger, de anderen zijn de auto's. U ziet in deze loop, dat E$ ingesteld wordt op ASCII teken nummer 143 plus de waarde van A. De binnenste loop, B, leest de DATA-opdracht van regel 560 tot 610 met een READ-opdracht, en POKEd deze waarden op de juiste positie met behulp van regel 480.
Regel 510 zet de variabele TIME (dit staat voor de verstreken tijd per ronde van vijf joggers, gemeten in 'hartslagen') en stelt de variabele BESTTIME op een zeer

variabele instelt op het moment, dat u het eerste stel van vijf joggers veilig aan de overzijde van de snelweg heeft gebracht.
De strings A$, B$, C$ en D$ – waarin de auto's zich bevinden – worden opgebouwd in de regels 520 tot 550. U kunt zo veel – of zo weinig – auto's in deze strings plaatsen als u wilt, zolang u er maar voor zorgt, dat de totale lengte van de strings 32 tekens is. Het is dus mogelijk, om het aantal auto's te vergroten, om het spelletje moeilijker te maken, voordat u met de tweede versie begint. In deze tweede versie kunt u dan weer met een klein aantal auto's beginnen. De auto's in de strings kunnen echter niet volledig willekeurig geplaatst worden. Zij zijn namelijk zó ontworpen, dat zij met de voorkant naar links of naar rechts staan. Bovendien zijn er ook auto's die er even goed uitzien, welke kant ze ook op rijden. De bovenste drie rijen verkeer rijden van links naar rechts, de onderste drie van rechts naar links. U kunt de grafische 'B' in iedere string gebruiken, maar de 'C' moet gebruikt worden in A$ en B$, terwijl de tekens 'D', 'E' en 'F' gereserveerd zijn voor de strings C$ en D$. Als de variabelen zijn geïnitialiseerd, en de grafische tekens gedefinieerd, stuurt de RETURN-opdracht in regel 555 het programma terug naar regel 30, waar de J-loop begint. Deze loop bestuurt de belangrijkste elementen van het programma, en zoals u kunt zien, wordt deze loop 5 maal doorlopen, voor iedere jogger één maal. Regel 40 print uw 'aantal joggers veilig', dit is dus altijd één minder dan de waarde van de J-variabele uit de loop, en de verstreken tijd. In regel 50 wordt het aantal joggers afgedrukt, dat de snelweg nog niet veilig heeft overgestoken. Dit aantal is dus altijd 5 min de huidige waarde van J (dus als J de waarde 1 heeft, is het aantal nog niet veilige joggers 4, omdat er dan

Regel 100 maakt de onderste regel schoon, zodat eventuele overblijvende joggers van het scherm verwijderd worden. De loop van regel 110 tot regel 130 print daarna een jogger voor iedere nog wachtende jogger. De variabele JA is de positie van de huidige jogger, die op weg is over het scherm, en JD is de vertikale positie van deze jogger. De jogger begint te lopen vanaf positie (16,19), dit is ongeveer het midden van de onderzijde van het scherm onder de eerste rij auto's. Regel 150 print een spatie op de plaats, waar de jogger gestaan heeft na een beweging, zodat deze posities schoongemaakt worden als de jogger in beweging is. Regels 160 en 170 lezen het toetsenbord, zodat u de jogger volgens uw wensen kunt bewegen. Op regel 180 wordt tenslotte de jogger geprint.
De lange regel 190 print alle rijstroken met auto's. De opdracht PRINT wordt hier gebruikt voor maximale snelheid. Regels 200 tot 230 vergelijken de huidge positie van de jogger met de inhoud van de string op die positie. Als hier blijkt, dat de jogger en een auto op de zelfde positie staan, wordt de besturing overgegeven aan regel 350, waar de routine wordt uitgevoerd, die een jogger laat verongelukken. De regels 250 tot 280 besturen de beweging van de auto's. Hierbij wordt gebruik gemaakt van Sinclair BASIC string verwerking om de elementen van de strings te verschuiven.
De variabele TIME wordt verhoogd in regel 290. In regel 300 wordt de waarde van de variabele JD bekeken. Als deze waarde 2 is, weet de computer dat er weer een jogger veilig aan de overkant is gekomen. Er wordt dan een aardig melodietje gespeeld, voordat de NEXT J-opdracht op regel 300 de besturing weer terugstuurt naar regel 40. Hier vandaan begint dan de volgende jogger aan zijn gevaarlijke oversteek.

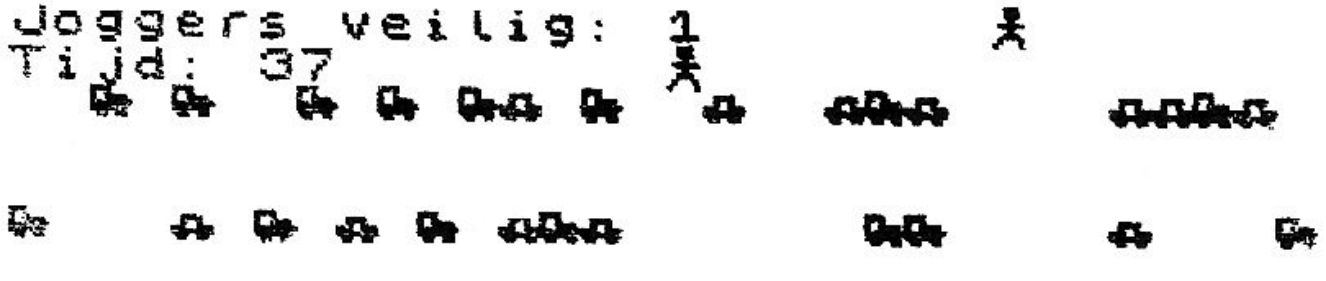
is dan 6, wordt de besturing overgegeven aan regel 150, waar de jogger van het scherm verwijderd wordt, voordat hij door regel 180 opnieuw geprint wordt.
Als alle joggers veilig zijn, wordt door regel 320 de tekst 'Deze ronde:' geprint, samen met de tijd, die u nodig had om de jogger over te krijgen. Daarna wordt er een '5' geprint achter de tekst 'Joggers veilig:', en op het scherm komt de melding 'U HEEFT HET VOOR ELKAAR'. Regel 330 vertelt u vervolgens, hoelang u er over gedaan heeft, en regel 340 stuurt het prorgramma verder naar regel 620.
Op regel 620 wordt de tijd van deze ronde vergeleken met de waarde BESTTIME, en als de waarde van TIME kleiner is, dan wordt de waarde BESTTIME gelijk gemaakt aan de waarde van TIME. Als de waarde van BESTTIME ongelijk is aan 10000 (wat voor kan komen als u dit spelletje voor het eerst probeert), dan wordt de waarde van BESTTIME op het scherm afgedrukt, zodat u kunt zien hoe goed (of slecht!) u het ervan af gebracht heeft. Daarna klinken nog een aantal contrasterende tonen (regel 620), de variabele TIME wordt op 0 gezet, en het programma gaat verder op regel 30 om zo de volgende ronde te starten.
Als u er niet in geslaagd bent, de vijf joggers aan de overkant te krijgen, dan wordt de routine vanaf regel 350 opgestart. (Zoals u zich zult herinneren, wordt dit gedaan door de IF/THEN-regels van regel 200 tot 230). De loop (van regel 350 tot regel 370) print daarna een knipperende jogger op de plaats, waar deze geraakt is door een auto, er klinken tonen, en er wordt u meegedeeld (hoewel u dit waarschijnlijk al begrepen had), dat u geraakt bent. Vervolgens krijgt u te zien, hoeveel joggers u aan de overkant heeft gekregen (regel 390). De variabele TIME wordt ingesteld op de waarde 10000,

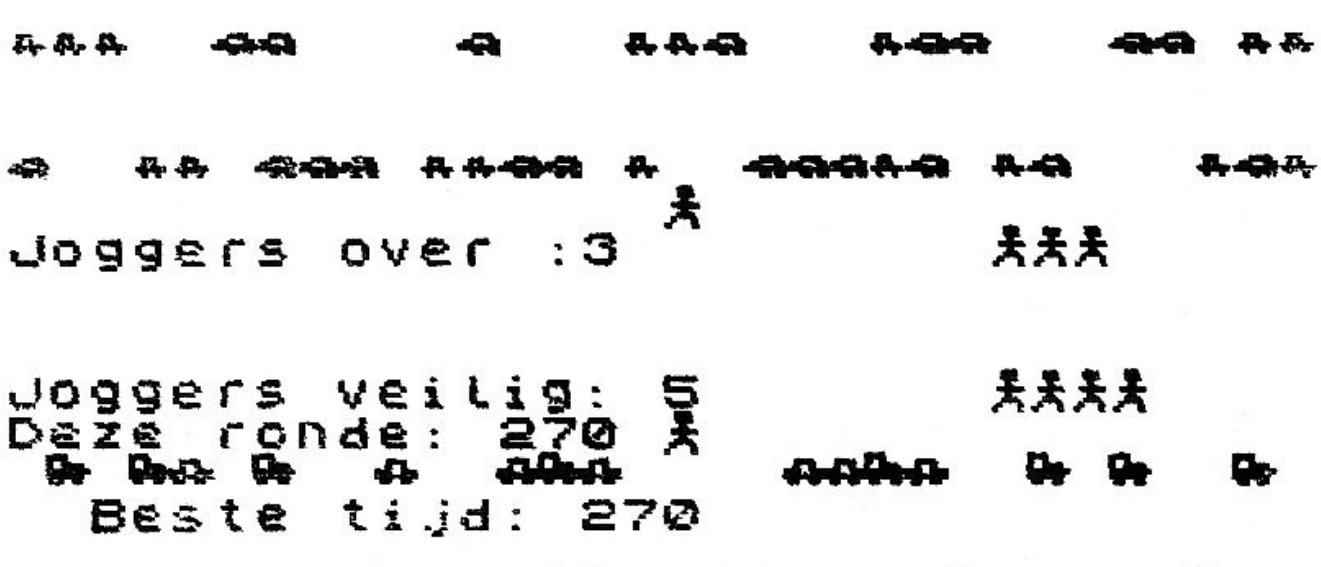
worden, krijgt u namelijk op een heel eenvoudige wijze een heel goede BESTTIME, eenvoudig door onder de eerste auto te lopen, die u op het scherm ziet.
Als u dit spelletje een beetje onder de knie hebt, kunt u het naar wens moeilijker maken, door meer auto's toe te voegen, of door uw eigen voertuigen te ontwerpen en toe te voegen. Een aantal paarden, een bromfiets of een trekkar kunnen het spel nog verder verlevendigen. Als zelfs deze eigengemaakte versie van JOGGER u niet langer kan boeien, probeer dan deze wijzigingen ook aan te brengen in de tweede versie van JOGGER. In deze versie begint de jogger namelijk direct te lopen, en blijft gedurende de gehele oversteek in beweging. Dit is dan werkelijk een hele uitdaging, en wellicht vindt u het prettiger, om het spelletje een beetje te vereenvoudigen als u begint. Dit kunt u doen, door wat minder auto's op de weg te laten rijden, of door het spel op een wat langzamere snelheid te laten verlopen. Dit laatste doet u (bijvoorbeeld) door de regel:

```
195 BEEP .05,RND*50
```

in het programma toe te voegen.
Veel geluk, Jogger, en dat u de weg veilig moge oversteken!

Dit is een voorbeeld van de uitvoering van het programma:





En dit is de listing:

```
   10 REM JOGGER
   20 GO SUB 410
   30 FOR J=1 TO 5
   40 PRINT AT 1,0;"Joggers veili
g:  ";J-1;AT 2,0;"Tijd: ";time;
   50 PRINT AT 20,0;"Joggers over
 :";5-J
   60 PRINT AT 1,24;
   70 FOR Z=1 TO J-1
   80 PRINT INK 2;"A";: REM GRAPH
IC A
   90 NEXT Z
  100 PRINT AT 20,24;"      ";AT 2
0,24;
  110 FOR z=1 TO 5-J
  120 PRINT INK 2;"A";
```

```
 170 LET JD=JD+(INKEY$="6")-(INK
EY$="7")
 180 PRINT AT JD,JA; INK 2;"X"
 190 PRINT AT 3,0; INK 2;A$;AT 9
,0; INK 0;A$;AT 6,0; INK 4;B$;AT
 12,0; INK 3;C$;AT 18,0; INK 1;C
$;AT 15,0; INK 2;D$
 200 IF (JD=3 OR JD=9) AND A$(JA
+1)<>" " THEN GO TO 350
 210 IF Jd=6 AND B$(JA+1)<>" " T
HEN GO TO 350
 220 IF (jd=12 OR jd=18) AND c$(
ja+1)<>" " THEN GO TO 350
 230 IF jd=15 AND d$(ja+1)<>" "
THEN GO TO 350
 250 LET a$=a$(31)+a$( TO 31)
 260 LET b$=b$(31)+b$( TO 31)
 270 LET c$=c$(2 TO )+c$(1)
 280 LET d$=d$(2 TO )+d$(1)
 290 LET time=time+1
 300 IF jd=2 THEN FOR z=1 TO 20:
 BEEP .01,z: NEXT z: NEXT j
 310 IF j<6 THEN GO TO 150
 320 PRINT INK 2;AT 2,0;"Deze ro
nde: ";time;" ";AT 1,16;5;AT 10,
8; FLASH 1; PAPER 6; BRIGHT 1;"U
 heeft het voor elkaar!"
 330 PRINT '"U had ";time;" har
tslagen nodig"
 340 GO TO 620
 350 FOR z=1 TO 20
 360 PRINT AT jd,ja; FLASH 1; BR
IGHT 1; INK 2;"X": BEEP .008,z
 370 NEXT z
 380 PRINT AT 10,8;"U bent geraa
kt"
 390 PRINT AT 12,2;j-1;" Joggers
 hebben het gehaald"
 400 LET time=10000: GO TO 620
 410 INK 1: PAPER 7: BORDER 2: C
LS
 440 FOR a=1 TO 6
 450 LET e$=CHR$ (143+a)
 460 FOR b=0 TO 7
 470 READ c
 480 POKE USR e$+b,c
 490 NEXT b
 500 NEXT a
 510 LET time=0: LET besttime=9e
7
```

```
 550 LET d$="
     "
 555 RETURN
 560 DATA 28,28,8,62,8,28,34,66
 570 DATA 0,60,36,231,255,102,0,
0
 580 DATA 240,144,158,255,254,10
2,0,0
 590 DATA 0,0,30,99,255,27,0,0
 600 DATA 0,0,56,40,126,36,0,0
 610 DATA 0,0,30,51,255,34,0,0
 620 IF time<besttime AND time<>
10000 THEN LET besttime=time
 630 IF besttime<>9e7 THEN PRINT
 AT 4,2; FLASH 1;"Beste tijd:  ";
besttime
 640 FOR z=50 TO 1 STEP -1: BEEP
 .05,z: BEEP .05,-z: NEXT z
 650 CLS
 660 LET time=0
 670 GO TO 30


  10 REM JOGGER versie 2
  20 GO SUB 410
  30 FOR J=1 TO 5
  40 PRINT AT 1,0;"Joggers veili
g:  ";J-1;AT 2,0;"Tijd:  ";time;
  50 PRINT AT 20,0;"Joggers over
 :";5-J
  60 PRINT AT 1,24;
  70 FOR Z=1 TO J-1
  80 PRINT INK 2;"A";: REM GRAPH
IC A
  90 NEXT Z
 100 PRINT AT 20,24;"      ";AT 2
0,24;
 110 FOR z=1 TO 5-J
 120 PRINT INK 2;"A";
 130 NEXT Z
 140 LET JA=16: LET JD=19
 150 PRINT AT JD,JA;" "
 155 LET z$=INKEY$: IF z$<"5" OR
 z$>"8" THEN LET z$=x$
 160 LET ja=ja+(z$="8" AND ja<28
)-(z$="5" AND ja>3)
 170 LET jd=jd+(z$="6" AND jd>1)
-(z$="7" AND jd<20)
 172 IF jd=20 THEN LET jd=19
 175 LET x$=z$
 180 PRINT AT JD,JA; INK 2;"A"
 190 PRINT AT 3,0; INK 2;A$;AT 9
```

```
 200 IF (JD=3 OR JD=9) AND A$(JA
+1)<>" " THEN GO TO 350
 210 IF Jd=6 AND B$(JA+1)<>" " T
HEN GO TO 350
 220 IF (jd=12 OR jd=18) AND c$(
ja+1)<>" " THEN GO TO 350
 230 IF jd=15 AND d$(ja+1)<>" "
THEN GO TO 350
 250 LET a$=a$(31)+a$( TO 31)
 260 LET b$=b$(31)+b$( TO 31)
 270 LET c$=c$(2 TO )+c$(1)
 280 LET d$=d$(2 TO )+d$(1)
 290 LET time=time+1
 300 IF jd=2 THEN FOR z=1 TO 20:
 BEEP .01,z: NEXT z: NEXT j
 310 IF j<6 THEN GO TO 150
 320 PRINT INK 2;AT 2,0;"Deze ro
nde: ";time;" ";AT 1,16;5;AT 10,
8; FLASH 1; PAPER 6; BRIGHT 1;"U
 heeft het voor elkaar!"
 330 PRINT '"U had ";time;" har
tslagen nodig"
 340 GO TO 620
 350 FOR z=1 TO 20
 360 PRINT AT jd,ja; FLASH 1; BR
IGHT 1; INK 2;"*": BEEP .008,z
 370 NEXT z
 380 PRINT AT 10,8;"U bent geraa
kt"
 390 PRINT AT 12,2;j-1;" Joggers
 hebben het gehaald"
 400 LET time=10000: GO TO 620
 410 INK 1: PAPER 7: BORDER 2: C
LS
 420 LET x$="8"
 440 FOR a=1 TO 6
 450 LET e$=CHR$ (143+a)
 460 FOR b=0 TO 7
 470 READ c
 490 NEXT b
 500 NEXT a
 510 LET time=0: LET besttime=9e
7
 520 LET a$="
 530 LET b$="
 540 LET c$="
 550 LET d$="
 555 RETURN
```

```
2,0,0
  590 DATA 0,0,30,99,255,27,0,0
  600 DATA 0,0,56,40,126,36,0,0
  610 DATA 0,0,30,51,255,34,0,0
```

```
 610>DATA 0,0,30,51,255,34,0,0
 620 IF time<besttime AND time<>
10000 THEN LET besttime=time
 630 IF besttime<>9e7 THEN PRINT
 AT 4,2; FLASH 1;"Beste tijd: ";
besttime
 640 FOR z=50 TO 1 STEP -1: BEEP
 .05,z: BEEP .05,-z: NEXT z
 650 CLS
 660 LET time=0: LET z$="8"
 670 GO TO 30
```

Mijnenveger

Dit spelletje is erg moeilijk, omdat er TWEE objecten TEGELIJK bestuurd moeten worden. Uw taak is, de mijnen van het scherm te vegen. Een taak die extra moeilijk is, omdat het verwijderen van een mijn in twee stappen gedaan moet worden. Eerst moet u de mijnen onschadelijk maken. Hiertoe heeft u de beschikking over een INVERTER. Als u de inverter over een mijn beweegt, die op scherp staat, dan wordt deze daardoor onschadelijk gemaakt. Maar let op: als u met de inverter over een onschadelijk gemaakt mijn beweegt, dan wordt deze daardoor opnieuw geactiveerd. U bestuurt de inverter met de toetsen '5' tot '8'.

Behalve de inverter heeft u ook de besturing over de VEGER. Met de veger kunt u een onschadelijk gemaakte mijn van het scherm verwijderen. Maar pas op: u moet er echt zeker van zijn, dat de veger een onschadelijke mijn benadert. Als de mijn namelijk nog (of weer) actief

Er zijn 40 mijnen en een keuze uit vijf verschillende moeilijkheidsgraden, zodat u dit spelletje u niet snel zal vervelen. Zoals u op het programmavoorbeeld kunt zien, wordt het u niet gemakkelijk gemaakt, en het is niet eenvoudig om alles onder controle te houden. Als u het programma voor de eerste keer opstart, dan wordt vanaf regel 30 de subroutine, beginnende op regel 600 aangeroepen. Hier worden de grafische tekens 'A', 'B' en 'C' gedefinieerd. De 'Z' in de laatste DATA-opdracht geeft het einde van de subroutine aan, en de besturing wordt dan teruggegeven aan het hoofdprogramma door de RETURN-opdracht op regel 600. Als uit deze subroutine teruggekeerd is, dan wordt de besturing doorgegeven naar regel 520 (door de GOSUB in regel 30), waar de instructies voor het programma uitgeprint worden op het scherm:

UW OPDRACHT IS DE MIJNEN OP HET SCHERM TE VERWIJDEREN. DEZE OPERATIE IS IN TWEE HANDELINGEN ONDERVERDEELD. EERST MOET DE MIJN ONSCHADELIJK GEMAAKT WORDEN, DOOR DE INVERTER OVER DE MIJN TE BEWEGEN (M.B.V. TOETSEN 5 TOT 8), DAARNA MOET DE MIJN VERWIJDERD WORDEN DOOR DE VEGER. DEZE BEDIENT U MET DE TOETSEN 5 EN 8 MET DE CAPS LOCK EN SHIFT TOETS INGEDRUKT. DE VEGER MAG GEEN ONSCHADELIJKE MIJN RAKEN.

Als u dit gelezen heeft, dan kunt u uw gewenste moeilijkheidsgraad opgeven na de vraag:

GEEF MOEILIJKHEIDSGRAAD 1-5.
5=EENVOUDIG
TOETS 6 OM TE STOPPEN.

De INKEY$-routine op regels 540, 550 en 560 accepteert een geldige invoer, en verwerpt alle invoer, die

een waarde, die overeenkomt met de gekozen moeilijkheidsgraad van 1 tot 5. SP bestuurt in het verdere verloop van het spel de moelijkheidsgraad. Regel 580 stuurt het programma naar regel 40, de regel na de GOTO die het voorgaande deel opgestart heeft. In regel 40 wordt het scherm opgemaakt. De INK en PAPER worden beide ingesteld op 5 (cyaan), het scherm wordt daarna schoon gemaakt om de kleur in te stellen, en ook de BORDER wordt op cyaan ingesteld. De INK-kleur wordt dan gewijzigd in zwart, en in de variabele V$ wordt één spatie opgenomen. Regels 50 en 60 initialiseren de verschillende variabelen, die verschillende zaken besturen, zoals de positie van de inverter en de veger, en het aantal mijnen, dat door u onschadelijk is gemaakt. De routine in regel 70, 80 en 90 tekent de grenzen, waarbinnen de acties van het spelletje zich afspeelt. De routine op regels 100 tot 150 plaatst de mijnen op het scherm. Hierbij wordt er door regel 130 voor gezorgd, dat er geen mijnen op een positie geplaatst worden, waar reeds een mijn staat. Als dit namelijk zou gebeuren, is het aantal mijnen op het scherm kleiner dan 40, en wordt het einde van het spel, als 40 mijnen verwijderd zijn (zie regel 370) nooit bereikt. Zodra de mijnen op hun positie staan (afgedrukt in rood, zie de INK-verandering aan het einde van regel 90), wordt de INK weer op cyaan ingesteld in regel 155. I$ wordt gebruikt om uw commando's te accepteren. Het eerste gedeelte (regels 180 tot 210) wordt gebruikt om de 'normale' toetsen '5' tot '8' te interpreteren. Hiermee bestuurt u de inverter. In het tweede deel (regels 220 tot 260) wordt er gekeken naar de zelfde toetsen, maar dan in combinatie met de SHIFT-toets. Merk op dat in regel 220 de CODE van I$ wordt bepaald, en dat deze code gebruikt wordt om de ingetoetste waarde vast te stellen.

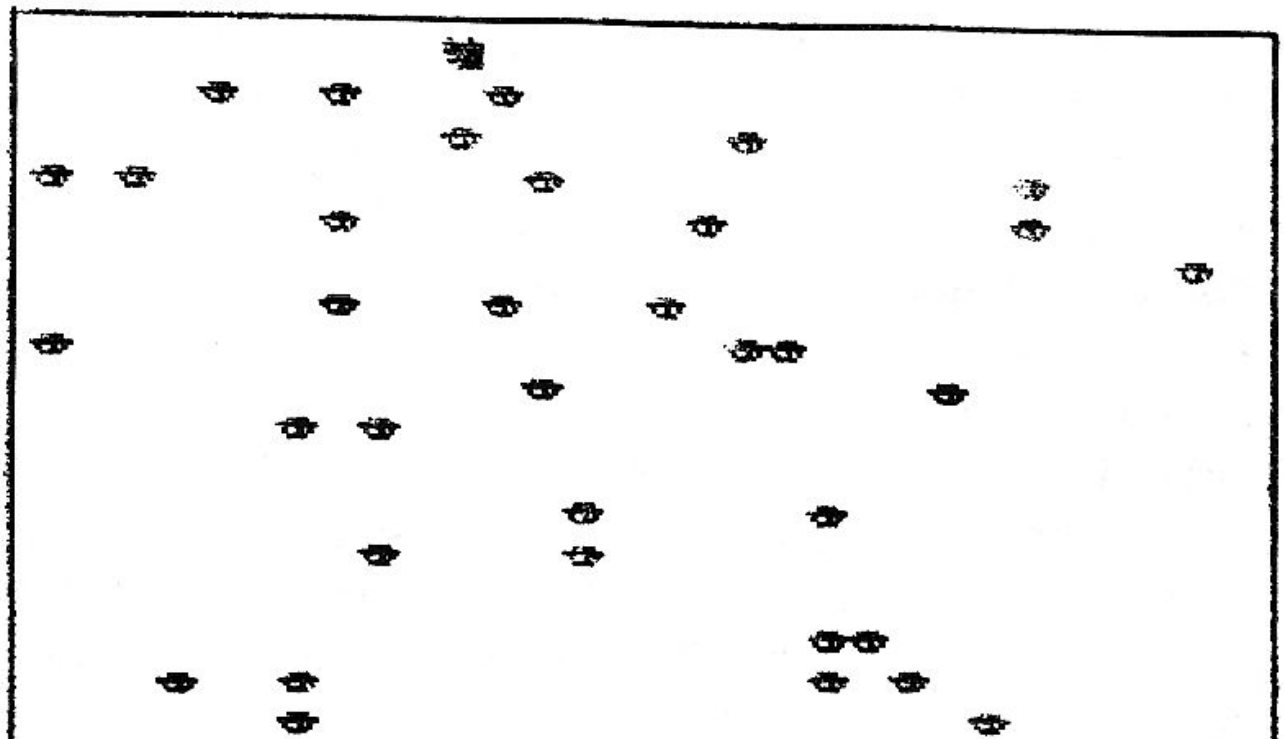
kunt u een eenvoudige routine schrijven, die de CODE uitprint van de toets, die u intikt.
Het resultaat van deze berekening wordt toegekend aan de variabelen A, D, CA en CD (A en D voor de inverter, CA en CD voor de veger). Deze variabelen worden daarna opgeteld bij de waarde van de positie van de inverter en de controler, nadat de 'oude' posities ervan in regel 270 met spaties overschreven zijn.
De variabele N krijgt nu de waarde van de ATTRibute van de nieuwe positie van de inverter, waarbij CN de waarde krijgt, van de ATTRibute van de positie van de veger. Regel 320 print vervolgens de beide objecten en regel 325 voegt een korte BEEP toe. Zoals u kunt zien, wordt de waarde van de BEEP bepaald, door de moeilijkheidsgraad, die u aan het begin van het spel hebt opgegeven. Hoe hoger dat getal is, hoe langer de BEEP is, die door regel 325 geproduceerd wordt. Zoals u weet, wordt de uitvoering van een programma op de Spectrum gestopt, zolang de BEEP-opdracht actief is, dus op deze manier krijgen we zowel een handige manier om de moeilijkheidsgraad van het programma vast te stellen, terwijl tegelijkertijd een BEEP gegeven wordt om het programma te verlevendigen. Als u het programma zelfs op niveau 5 te moeilijk vindt, kunt u de '9' op regel 570 veranderen in een kleinere waarde.
Als N gelijk is aan 40 of CB gelijk is aan 40, dan weet het programma, dat u tegen de muur aan de grens van het scherm gebotst bent, en het programma print dan: 'U BENT TEGEN DE ELEKTRISCHE BARRIÈRE GEBOTST'.
Regels 340 en 350 brengen de nodige veranderingen aan in de mijnen, als u over een mijn beweegt met de inverter. Hier worden dus actieve mijnen onschadelijk gemaakt en vice versa. Als de waarde van CN 42 is,

EEN ACTIEVE MIJN GERAAKT' wordt afgedrukt in rode, knipperende letters.
Als u een actieve mijn onschadelijk heeft gemaakt (ATTRibute is CN is 42, regel 370), dan wordt uw score (S) van verwijderde mijnen met 1 verhoogd. Als S daarna gelijk is aan 40 (einde van regel 370), dan weet de computer, dat u alle mijnen heeft verwijderd, en regel 470 wordt dan geactiveerd. De boodschap 'U HEEFT HET VOOR ELKAAR' wordt nu afgedrukt. Als dit geen van beide het geval is (u heeft geen actieve mijn geraakt, en u heeft niet alle mijnen verwijderd), dan stuurt regel 380 het programma terug naar regel 160 om het gehele proces voort te zetten.
De rest van het programma, van 400 tot 510, bevat de boodschappen voor het einde van het spelletje, en een aantal korte uitbarstingen van Spectrum's muzikaal geweld. Als u een bijzonder vertrouwde speler bent geworden, verwijder dan regel 325 en kijk eens hoe u het ervan af brengt. Dat is de laatste uitdaging die we u met dit programma kunnen bieden.

```
  30 GO SUB 500: GO TO 520
  40 INK 5: PAPER 5: CLS : BORDE
R 5: INK 0: LET t$="  "
  50 LET s=0: LET x=16: LET y=1
  60 LET cx=16: LET cy=20: LET c
d=0
  65 LET a=1: LET d=0: LET ca=1
  70 PLOT 4,4: DRAW 247,0
  80 DRAW 0,167: DRAW -247,0
  90 DRAW 0,-167: INK 2
 100 FOR i=1 TO 40
 110 LET tx=INT (RND*30)+1
 120 LET ty=INT (RND*18)+2
 130 IF SCREEN$ (ty,tx)<>"  " THE
N GO TO 110
 140 PRINT AT ty,tx;"■"
 150 NEXT i
 155 INK 5
 160 LET i$=INKEY$
```

```
 d=0: GO TO 270
 200 IF i$="6" THEN LET a=0: LET
 d=1: GO TO 270
 210 IF i$="7" THEN LET a=0: LET
 d=-1: GO TO 270
 220 LET i=CODE i$
 230 IF i=8 THEN LET ca=-1: LET
cd=0: GO TO 270
 240 IF i=9 THEN LET ca=1: LET c
d=0: GO TO 270
 250 IF i=10 THEN LET ca=0: LET
cd=1: GO TO 270
 260 IF i=11 THEN LET ca=0: LET
cd=-1: GO TO 270
 270 PRINT AT cy,cx;" ";AT y,x;t
$: LET t$=" "
 280 LET x=x+a: LET y=y+d
 290 LET cx=cx+ca: LET cy=cy+cd
 300 LET n=ATTR (y,x)
 310 LET cn=ATTR (cy,cx)
 320 PRINT INK 7;AT y,x;"*";AT c
y,cx;"■"
 325 BEEP sp,0
 330 IF n=40 OR cn=40 THEN GO TO
 400
 340 IF n=42 THEN LET t$=CHR$ 16
+CHR$ 1+"■"
 350 IF n=41 THEN LET t$=CHR$ 16
+CHR$ 2+"■"
 360 IF cn=42 THEN GO TO 440
 370 IF cn=41 THEN LET s=s+1: IF
 s=40 THEN GO TO 470
 380 GO TO 160
 400 PRINT FLASH 1; INK 0;AT 0,0
;"U bent tegen de barriere gelop
en"
 410 FOR i=1 TO 10: BEEP .02,-0
 420 BEEP .02,5: NEXT i
 430 GO TO 500
 440 PRINT FLASH 1; INK 2;AT 0,2
;"U bent op een mijn gelopen"
 450 BEEP .2,-20: BEEP .5,-5
 470 PRINT AT 1,3; INK 0;"U heef
t het gehaald!!"
 480 BEEP 1,9
 500 INK 0
 510 PRINT "U heeft ";s;" mijnen
 opgeruimd"
 520 PRINT "■■■■■■■■■■■■■■■CLEARWAY
■■■■■■■■■■■■■■■Uw taak is de mijnen
 op het      scherm onschadelijk
```

```
nverter          bestuurt U met de to
etsen 5-8.       Daarna kunt U de mij
nen met de       VEGER (■) opruimen.
De veger         bestuurt U met de to
etsen 5-8        en de toets CAPS SHI
FT               ingedrukt. De veger
mag geen         scherpe mijn raken."
  530 PRINT      "Geef moeilijkheids
graad 1-5          5=makkelijk
                   druk 6 om te stopp
en"
  540 LET i$=INKEY$
  550 IF i$<"1" OR i$>"6" THEN GO
  TO 540
  560 IF i$="6" THEN STOP
  570 LET sp=VAL i$/9
  580 GO TO 40
  600 READ A$: IF A$="Z" THEN RET
URN
  610 FOR i=0 TO 7
  620 READ n: POKE USR A$+i,n
  630 NEXT i
  640 GO TO 600
  650 DATA "A",0,0,0,60,82,255,60
,0
  660 DATA "B",36,36,255,60,60,25
5,36,36
  670 DATA "C",0,0,248,254,254,25
5,255,34
  680 DATA "Z"
```



Copter

Dit is een programma om uw vliegkunst te testen. U heeft vier helikopters, en uw taak is, een aantal aanvallende rode lijnen te verslaan. U kunt deze lijnen verslaan, door er op te landen. U moet daartoe eerst op de kortste lijn landen, die u ziet, daarna op de op een na kortste, en zo verder totdat u de langste lijn als laatste verslaat. Eén verkeerde beweging en u verliest een helikopter. U heeft vier helikopters waarmee u de opdracht dient te voltooien, en u krijgt een bonus als het u lukt zonder alle vier de helikopters te gebruiken.

U bestuurt de kleine machines met de toets '0' om te stijgen, en de toetsen '5' en '8' om naar links en rechts te bewegen. U moet constant naar boven sturen, want de zwaartekracht trekt u automatisch naar beneden. Uw uiteindelijke score wordt vastgesteld aan de hand van de hoeveelheid brandstof, die u verbruikt heeft. Als alle vier de helikopters neergestort zijn, voordat u de rode lijnen heeft vernietigd, dan krijgt u de grimmige boodschap 'COPTER BEËINDIGD'. Bovendien wordt de hoogste score door het programma vastgehouden.

Regel 10 stelt de BORDER in op de kleur zwart, de variabele H wordt in regel 15 op 0 gezet (voor de hoogste score, zie regels 532 en 535), daarna wordt INK op geel gezet (regel 20) en PAPER op zwart (regel 30). Hierna wordt het programma naar de subroutine op regel 610 gestuurd, waar de DATA vanaf regel 900 in de grafische 'A' gePOKEd wordt om de helikopters te maken, die u op regel 45 ziet.

Nadat de grafische 'A' gedefinieerd is, wordt het scherm schoongemaakt, zodat de zwarte PAPER geactiveerd wordt. De loop van regel 50 tot regel 70 print een solide geel blok, waarbij gebruik gemaakt wordt van een geel PAPER en een reeks spaties tussen aanhalingstekens.

bruik gemaakt van een RANDOM getal, dat in regel 110 gegenereerd wordt, en de PLOT (regel 150) en DRAW (regel 160) produceren een zeer effectieve landingsbasis. Dit wordt duidelijk, als u het programma uitvoert. De routine op regel 120 en 130 zorgen er voor, dat er tien lijnen van verschillende lengte in de D-array opgeslagen worden. De variabele R krijgt in regel 185 de waarde 1. Hiermee wordt de lijn waarop de helikopter moet landen, bestuurd (als r=1, dan mikt u op de lijn, die aangegeven wordt met d(r)). Uw brandstof-variabele f, wordt op regel 190 ingesteld op de waarde 25000. Naast het besturen van de brandstof wordt f ook gebruikt om het geluid op regel 240 te genereren (hier wordt f dan wel gedeeld door 1000). Zoals u zult horen, produceert de helikopter een geluid, dat zeer goed bij dit spelletje past. De brandstof wordt op regel 250 en (als u stijgt) op regel 280 met 21 verminderd. De variabele P, uw horizontale coördinaat, wordt op 16 gezet in regel 210 voordat uw kleine helikopter in regel 220 gelanceerd wordt. Regel 245 print de 'Helikopters over' (door gebruikt te maklen van I$, die in regel 45 als drie helikopters gedefinieerd is) in de rechter-benedenhoek van het scherm, en – nadat de brandstof in regel 250 verminderd is, en de overgebleven hoeveelheid in regel 255 in de linker-benedenhoek van het scherm is afgedrukt – wordt de helikopter op de vorige positie verwijderd, voordat de nieuwe helikopter wordt afgedrukt. De variabele U, die een initiële waarde van 1 in regel 200 heeft gekregen, bestuurt uw vertikale verplaatsing: hoe hoger de waarde van variabele U is, hoe lager uw helikopter vliegt. De regels 275 tot 290 worden gebruikt om uw commando's van het toetsenbord in te lezen ('0' om te stijgen, '5' om naar links te bewegen, '8' om naar rechts te bewegen), voordat het programma opnieuw

regel 230, hierdoor wordt het programma naar regel 400 gestuurd, waar de landing gecontroleerd wordt. Als de plaats waar geland wordt, niet de kortste lijn op het scherm is (dus p< > d (r)), dan wordt een akelig geluid hoorbaar, en er worden 1000 eenheden brandstof door het programma van uw brandstofvoorraad afgenomen. Daarna wordt de helikopter in een inverse kleur afgedrukt op de plaats van de onjuiste botsing (regel 412). Er klinkt opnieuw dat akelige geluid, en de helikopter verdwijnt (regel 413 en 414). De string I$ wordt gecontroleerd. Als deze string gelijk is aan de nul-string (''), dan weet de computer, dat u door uw helikopters heen bent. U zult zich herinneren, dat I$ in het begin van het spel op drie helikopters werd ingesteld (regel 45). Regel 416 verwijdert aan het einde van de string één helikopter, zodat als I$ gelijk is aan '', het programma weet, dat alle helikopters gebruikt zijn (let daarbij op, dat dit in totaal 4 helikopters zijn, één in het spel en 3 als reserve!). Regel 418 stuurt het programma terug naar regel 200, waar de volgende helikopter op zijn missie gestuurd wordt.
Regel 400 controleert of er een succesvolle landing uitgevoerd is. Als dit inderdaad het geval is, dan wordt al het slechte nieuws op regels 405 tot 418 overgeslagen en gaat het programma direct verder bij regel 420, waar de helikopter verwijderd wordt van het scherm. Bovendien wordt de bovenzijde van de rode lijnen door regel 425 veranderd. Een kleine BEEP-loop op regel 532 geeft u de kans even op adem te komen, daarna wordt de variabele r met één verhoogd, zodat u aan de volgende lijn kunt beginnen. Als het aantal lijnen, dat u reeds vernietigd hebt, gelijk is aan elf, dan zijn er geen lijnen meer om op te landen, en is het spelletje afgelopen. Het programma verwijdert dan de nog aanwezige reserve-he-

likopters. De nieuwe waarde voor de hoeveelheid brandstof wordt in regel 520 langzaam bijgesteld.
De hoogste score wordt bijgewerkt, indien u deze behaald hebt, en deze wordt door de volgende regel (532-534) afgedrukt. Daarna gaat het programma naar regel 310, waar u wordt verteld, dat het spelletje afgelopen is. Als u er niet in geslaagd bent, alle lijnen te vernietigen, dan wordt de boodschap 'COPTER BEËINDIGD' op het scherm afgedrukt, om aan te geven, dat u gefaald hebt (dit ziet het programma doordat r ongelijk is aan 11). Op regels 330 en 340 kunt u een toets indrukken en weer loslaten, daarna gaat het programma verder op regel 40. Hier wordt dus een nieuwe ronde opgestart, waarin u opnieuw een kans krijgt om de beste score te verbeteren. Vanaf regel 40 wordt het scherm schoongemaakt en daarna opnieuw opgebouwd, maar de grafische tekens worden niet opnieuw gedefinieerd, en de hoogste score h, wordt ook niet opnieuw geïnitialiseerd. Daarna kunt u met de nieuwe helikopters aan de gang. Goede vlucht !!

```
  5 REM Copter
 10 BORDER 0
 15 LET h=0
 20 INK 6
 30 PAPER 0
 35 GO SUB 610
 40 CLS
 45 LET l$="AAA"
 50 FOR a=11 TO 21
 60 PRINT AT a,0; PAPER 6;"
 70 NEXT a
 80 DIM d(10)
 90 DIM e(32)
100 FOR a=1 TO 10
110 LET b=INT (RND*32)
120 IF e(b+1)=1 THEN GO TO 110
130 LET e(b+1)=1
140 LET d(a)=b
150 PLOT b*8+4,67
```

```
 200 LET u=0
 210 LET p=16
 220 PRINT INK 4;AT u,p;"A"
 230 IF u=9+r THEN GO TO 400
 240 BEEP .01,f/1000
 245 PRINT INVERSE 1;AT 21,27;l$
;" "
 250 LET f=f-21
 255 PRINT AT 21,0; INVERSE 1;"B
randstof:";f;" "
 260 PRINT AT u,p;" "
 270 LET u=u+1
 275 LET i=IN 61438
 280 IF INT (i/2)=i/2 THEN LET u
=u-2*(u>1): LET f=f-21
 288 LET p=p+(i=187 OR i=186)*(p
<31)-(IN 63486=175)*(p>0)
 289 GO TO 300
 290 LET p=p+(i=187)*(p<31)-(IN
63486=175)*(p>0)
 300 GO TO 220
 310 PRINT AT 7,10; INK 5;"EINDE
 COPTER"
 320 IF r<11 THEN PRINT AT 1,8;
PAPER 1; FLASH 1;"copter beeindi
gd"
 330 IF INKEY$<>"" THEN GO TO 33
0
 340 IF INKEY$="" THEN GO TO 340
 350 GO TO 40
 400 IF p=d(r) THEN GO TO 420
 405 BEEP .01,-10
 410 LET f=f-1000
 412 PRINT INVERSE 1;AT u,p;"A"
 413 BEEP .1,-20
 414 PRINT AT u,p;" "
 415 IF l$="" THEN GO TO 310
 416 LET l$=l$(2 TO )
 418 GO TO 200
 420 PRINT AT u,p;" "
 425 PRINT AT 10+r,0;"
                      "
 430 FOR a=0 TO 20
 440 BEEP .01,a
 450 NEXT a
 460 LET r=r+1
 470 IF r<11 THEN GO TO 200
 480 FOR a=1 TO LEN l$
 490 PRINT AT 21,26+a;"■"
 500 BEEP .5,a*10
 510 LET f=f+a*1000
```

```
ogste score:"; FLASH 1; BRIGHT 1
; INVERSE 1; PAPER 7; INK 2;h
 540 GO TO 310
 610 FOR b=0 TO 7
 620 READ c
 630 POKE USR "a"+b,c
 640 NEXT b
 650 RETURN
 900 DATA 0,85,8,28,62,62,20,34
```

Mazurka

In MAZURKA krijgt u de schietervaring van uw leven. Dit spelletje lijkt wel wat op de schiettenten, die men wel op kermissen aantreft, waar men eenden en andere voorwerpen moet zien te raken.

Deze computerversie kent echter nog een extra moeilijkheid. In MAZURKA bewegen er rijen objecten boven u, en deze rijen worden beschermd door vier barrières, die in de tegenovergestelde richting bewegen. Uw doel is, zo veel mogelijk objecten te raken zonder de barrières te raken. U kunt uw geweer bewegen (de grafische 'H'), door gebruik te maken van de toetsen '5' en '8', u beweegt dan in de richting, die door de pijlen op die toetsen wordt aangegeven. Met de 'F'-toets schiet u uw geweer af. Er kan slechts één kogel tegelijk op het scherm zijn. U heeft slechts een beperkte hoeveelheid munitie, en het aantal schoten, dat u nog kunt lossen, wordt aangegeven door de lengte van de lijn onder u. Zoals u kunt zien in het voorbeeld van dit spelletje, is het scherm een wilde verzameling rondvliegende objecten, die tezamen een opwindend beeld vormen. Dit is echter niet alles. Van tijd tot tijd komt er een eend overgevlogen, ongeveer zoals het moederschip in spelletjes zoals Space Invaders. U krijgt en bonus van 6 punten, als u deze eend raakt, en u verliest 15 punten als u mist. De eend is betrekkelijk gemakkelijk te raken, omdat hij,

hiervoor worden gebruikt 'A', 'B', 'C', 'D', 'E', 'F', 'H', 'I' en 'J'. Als regel 600 (READ C$) de 'Z'-regel in 750 tegenkomt, dan wordt de return geactiveerd. Terug bij het begin van het programma (regel 30) worden de PAPER en de BORDER ingesteld op geel (kleur 6), en het scherm wordt daarna schoongemaakt om deze kleuren te activeren. Regel 35 stelt de INK in op zwart (kleur 0), en het programma gaat verder naar regel 500, waar de instructies geprint worden:

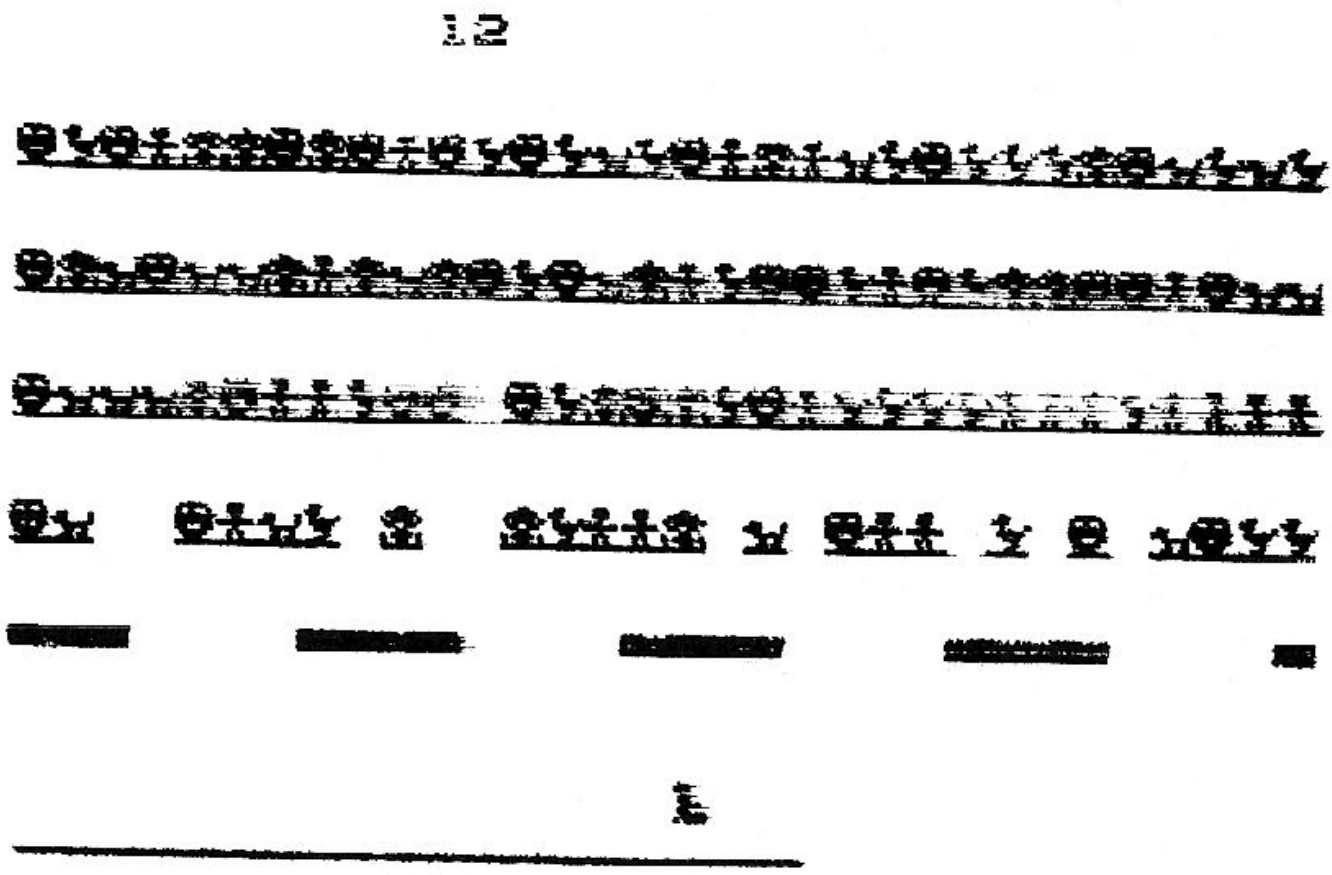
U MOET DE BEWEGENDE VOORWERPEN RAKEN, ZONDER DE ZWARTE LIJNEN TE RAKEN. HOE HOGER DE RIJ HOE HOGER DE SCORE. U BEWEEGT UW GEWEER MET DE TOETSEN 5 EN 8 EN SCHIET MET F. ER KAN SLECHTS EEN KOGEL TEGELIJK OP HET SCHERM ZIJN. HET OVERIGE AANTAL KOGELS WORDT AANGEGEVEN DOOR DE LIJN ONDERIN HET SCHERM. ELKE KEER DAT EEN EEND HET EIND VAN HET SCHERM BEREIKT WORDEN ER 15 PATRONEN VAN UW MUNITIE AFGETROKKEN.

Hierna wordt u verzocht, om 'J' in te tikken om het spel te beginnen. Deze instructie en de 'TOETS J'-routine komen voor aan het einde van ieder spelletje, zodat u altijd op dit punt het spel kunt verlaten door 'N' of 'n' in te toetsen. Iedere andere toets, behalve 'N' en 'n', geven u een nieuw spelletje. Volgende spelletjes gebruiken natuurlijk niet de subroutine vanaf regel 600 om de grafische tekens te initialiseren.
Terug op regel 40 worden de variabelen geïnitialiseerd, en het string-array, dat de bewegende doelen bevat, wordt gedimensioneerd (A$). De loops van 60 tot 100 wijzen willekeurige doelen toe aan de strings. Deze worden geprint door regel 95, en een stijgende toon wordt door regel 97 gemaakt, terwijl het programma zichzelf opzet. De B op het einde van regel 100 bestuurt de

verwijderd door de PLOT OVER, die hierop volgt. In A$(5) worden de bewegende barrières opgeslagen, we hebben nu dus A$(1) tot A$(4) als objecten om op te schieten, en A$(5) als de barrière, die zijn best zal doen, om uw schoten af te weren. Regel 107 geeft een BEEP om te laten weten, dat het spel nu direct zal beginnen.
De I-loop van regel 110 tot 350 is het voornaamste gedeelte van het programma. Hierin worden de bewegingen van de objecten in de loop bestuurd. Dit wordt gedaan, door I$ in te stellen op het eerste element van de string in regel 120. Daarna wordt in regel 130 dit eerste element toegevoegd aan het einde van de zelfde string, waarvan het eerste element is verwijderd. In regel 140 wordt de nieuwe string geprint, waarbij voor de verschillende strings verschillende INK-kleuren gekozen worden (de bovenste regel is blauw, de tweede rood, de derde magenta en de vierde regel wordt in groen afgedrukt). Hierna maakt regel 170 de positie schoon, waar de schutter afgedrukt was, en regel 180 verandert de horizontale positie van de schutter, afhankelijk van de toets, die in deze regel gelezen wordt. In regel 190 wordt de positie aangepast indien deze te ver naar links of naar rechts is.
Op regel 200 wordt de schutter opnieuw afgedrukt. Indien B gelijk is aan nul, dan zijn alle schoten gebruikt, en de actie gaat dan verder op regel 205 tot regel 430, waar u verteld wordt, dat u zonder munitie zit. Regel 210 checkt, of u kunt schieten (F-toets ingedrukt, en geen kogel in de lucht, dus OK om te schieten!). Regel 210 zet ook de horizontale coördinaat van de kogel (FX) gelijk aan die van de schutter (G). F is de vertikale coördinaat van de kogel, dus (F,FX) is de lokatie, waar de kogel afgedrukt en vervolgens weer uitgeveegd moet worden. Regel 240 verlaagt de F-coördinaat met één,

Uw scherpschutterskunde wordt in regel 250 aan de kaak gesteld. Als SCREEN$(F,FX) leeg is, dan ligt er geen object direct boven de kogel. Als er een object wordt gevonden, dan wordt er een test gedaan om te zien, of dit de barrière is (regels 260 en 265 testen dit; als P gelijk is aan vijf, dan heeft u de barrière geraakt.) Als u de barrière geraakt heeft, dan is het spel afgelopen, en de computer gaat verder naar regel 400 om u het goede nieuws mede te delen. Als u de barrière niet geraakt heeft, maar u heeft wel iets geraakt, dan moet dat iets dus één van de vliegende objecten zijn, waar u op mikt. Regel 270 geeft u een score, die afhankelijk is van de hoogte van het geraakte object (hoe lager de waarde van P, hoe hoger u in het scherm bent). Als u de eend geraakt hebt, moet P gelijk zijn aan zes, en u krijgt dan een bonus van zes punten. De variabele D wordt dan op nul gezet. Het programma springt dan hier vandaan naar regel 300 als u een eend heeft geraakt. Als dit niet het geval is, dan wordt de score (knipperend) door regel 275 afgedrukt, en het element van de string, die het geraakte object bevat, wordt op een spatie gezet, zodat het object verdwijnt. Als u niets geraakt heeft (dus de GOTO 300 aan het einde van regel 280 wordt niet uitgevoerd), dan print regel 290 de kogel. Ongeveer om de 50 doorgangen van de routine wordt door regel 300 een nieuwe eend gegenereerd, als er op dat moment geen eend op het scherm is aangekomen. Als dit het geval is, wordt de eend van het scherm verwijderd door deze regel, de variabele D wordt op nul gezet (de Geen-Eend conditie), en er wordt 15 afgetrokken van uw score, voordat deze opnieuw afgedrukt wordt. Het laatste gedeelte van regel 330 (GOTO 350) springt over het afdrukken van de eend op regel 340. Regel 350 is het einde van de hoofd-loop in het programma.

zijde van A$(5) toegevoegd op regel 370. Dit is het tegenovergestelde van wat er gedaan wordt met de andere elementen van het A$-array, zodat de barrières ook in tegenovergestelde richting van de objecten bewegen. Regel 380 print de barrière en 390 stuurt de actie terug naar regel 110 waar de hoofd-loop opnieuw begint. Regels 400 en 420 worden gebruikt als u de barrière geraakt hebt (P is gelijk aan 5, zie voorgaande gedeelte programma), en nadat dit u is meegedeeld gaat het programma naar 450 om uw score af te drukken. Als u zonder kogels zit (dus variabele B is gelijk aan nul) worden regels 430 en 440 uitgevoerd: 'U HEEFT GEEN KOGELS MEER OVER'. Aan het einde van de spelboodschappen wordt de titelpagina opnieuw afgedrukt, en u toetst alles behalve 'n' of 'N' in voor een nieuw spel. Hieronder ziet u het programma in actie:



En dit is de programma-listing:

```
  60 FOR i=1 TO 4
  70 FOR j=1 TO 32
  80 LET a$(i,j)=CHR$ (INT (RND*
5)+14+4)
  90 NEXT j
  95 PRINT AT i+3,0;a$(i)
  97 BEEP .4,i*2
 100 NEXT i: LET b=256
 102 LET d=0: DRAW 255,0
 105 LET a$(5)="
                "
 107 BEEP 1,9
 108 PRINT AT 20,g;"
 110 FOR i=1 TO 4
 120 LET t$=a$(i,1)
 130 LET a$(i)=a$(i,2 TO )+t$
 140 PRINT INK i;AT i*3,0;a$(i)
 150 LET x$=INKEY$: LET g=g+(x$=
"8")-(x$="5")
 190 LET g=g+(g=-1)-(g=32)
 200 IF x$<>"" THEN PRINT AT 20,
g-1;"  "
 205 IF b=0 THEN GO TO 430
 210 IF f=1 AND x$="f" THEN PRIN
T AT 1,fx;" ": LET f=19: LET fx=
g: BEEP .02,-15: LET b=b-1: PLOT
 OVER 1;b,0: PRINT AT 20,g;"
 220 PRINT AT f,fx;" "
 230 IF f=1 THEN GO TO 300
 240 LET f=f-1
 250 IF SCREEN$ (f,fx)=" " THEN
GO TO 290
 260 LET p=f/3: BEEP .02,6
 265 IF p=5 THEN GO TO 400
 270 LET s=s+5-p: LET f=1
 272 IF p=6 THEN LET s=s+6: PRIN
T AT 18,fx;" ": LET d=0: GO TO 3
00
 275 PRINT FLASH 1;AT 0,10;s
 280 LET a$(p,fx+1)=" ": GO TO 3
00
 290 PRINT AT f,fx;"
 300 IF d=0 AND RND>.98 THEN LET
 d=1
 310 IF d=0 THEN GO TO 350
 320 LET d=d+1
 330 IF d=31 THEN PRINT AT 18,31
;" ": LET d=0: LET b=b-15: LET b
=b*(b>0): DRAW OVER 1;-PEEK 2367
7+b,0: BEEP .02,9: GO TO 350
 340 PRINT AT 18,d;"
```

```
 390  GO TO 110
 400  BEEP 2,-9
 410  PRINT FLASH 1;AT 0,0;"U hee
ft de barriere geraakt !"
 420  GO TO 450
 430  BEEP 1,1: BEEP 1,9
 440  PRINT FLASH 1;AT 0,0;"U hee
ft geen kogels meer !"
 450  PRINT FLASH 1; INVERSE 1;"U
w score is ";s
 460  IF INKEY$="" THEN GO TO 460
 470  CLS
 500  PRINT "LLLLLLLLLL MAZURKA LL
LLLLLLLLLLL";"" U moet de voorw
erpen die boven   U bewegen neers
chieten, maar U   mag de zwarte l
ijnen niet        raken. Hoe hoge
r het voorwerp,   hoe hoger de sc
ore. U kunt het   geweer bewegen
met de toetsen    5 en 8, en schi
eten met de       F-toets. U kunt
 slechts 2 schot  tegelijk afvure
n, en het aantal  schoten, dat U
over heeft, kunt  U aflezen aan d
e lengte van de   regel onder U.
Als een EEND het  einde van het s
cherm bereikt,    raakt U 15 scho
ten kwijt."
 510  PRINT '" Toets J voor star
t, N voor stop"
 520  IF INKEY$="" THEN GO TO 520
 530  IF INKEY$="N" OR INKEY$="n"
 THEN STOP
 540  GO TO 40
 595
 600  READ c$: IF c$="Z" THEN RET
URN
 610  FOR i=0 TO 7
 620  READ n: POKE USR c$+i,n
 630  NEXT i
 640  GO TO 600
 650  DATA "a",BIN 00110000,BIN 0
1110000,BIN 00100001,BIN 0011111
0,BIN 00011100,BIN 000100,BIN 00
011000,255
 660  DATA "B",0,0,BIN 01100001,B
IN 11100001,BIN 00111110,BIN 000
10010,BIN 00110110,255
 670  DATA "C",BIN 00011000,BIN 0
1111110,BIN 10111101,BIN 011001
```

```
  685 DATA "E",BIN 00111000,BIN 0
0111000,16,254,16,40,40,255
  690 DATA "H",BIN 00111000,BIN 0
0111000,BIN 00011000,BIN 0001100
0,BIN 00011110,BIN 00011000,BIN
00011111,BIN 00101111
  700 DATA "I",0,0,8,28,28,28,26,
0
  710 DATA "J",0,0,32,18,255,BIN
10011000,16,32
  720 DATA "Z"
```

Alphabattle

In dit ongewone spel bestuurt u een groene helikopter, terwijl de Spectrum de besturing heeft over een blauwe gevechtsstraaljager. U vecht tegen de Spectrum om het bezit van het alfabet. De letters, die u bezit, worden afgedrukt aan de linkerzijde van het scherm en de letters van de Spectrum aan de rechterzijde.

Door de '5' en de '8'-toets te gebruiken, beweegt u horizontaal, met de 'w' en 'z'-toetsen vertikaal. U moet uw helikopter vóór een niet-knipperende letter van de vijand plaatsen. Als u dan de '0'-toets indrukt, begint deze letter onmiddellijk aan uw zijde te knipperen. De vijandelijke letter verdwijnt dan, uw score wordt verhoogd, en u wordt teruggezet naar de rechterzijde van het scherm voor een volgende aanval in de strijd. Merk op, dat u meer dan één toets tegelijk in kunt drukken, omdat IN gebruikt wordt om het toetsenbord te lezen, in plaats van het meer gebruikelijk INKEY$ (zie regels 410 en 420). U kunt dus ook diagonaal bewegen. Terwijl u dit alles doet, is de Spectrum met deze zelfde handelingen bezig om uw letters te stelen.

Voor de letters gaan staan en op de '0' drukken is één manier om een letter te pakken te krijgen. Er is echter ook nog een andere methode. Als de blauwe straaljager van de Spectrum op de zelfde regel staat als een van de niet-knipperende letters, kunt u zelf ook op die regel

Het spel is afgelopen, als alle letters door één van beide partijen gevangen zijn. De computer drukt dan de winnende score af. (U wordt hier trouwens aangeduid als de 'humanoïde' in de score-regel.)
Regels 100 tot 150 definiëren de grafische tekens (de gebruikte letters zijn 'A', 'B', 'C', en 'D'), deze worden ingelezen vanaf de DATA vanaf regel 900. Er worden twee arrays gedimensioneerd, P en T, en 22 letters van het alfabet worden over deze arrays willekeurig verdeeld en vervolgens afgedrukt.
De positie van uw helikopter wordt gedefinieerd door de waarden van U en P. De vertikale coördinaat één positie boven die, waar de laatste letter is afgedrukt. De positie van de computer wordt overeenkomstig bepaald aan de hand van zijn laatste letter, waarbij de variabelen A en T gebruikt worden. Deze beginnen op 1, zie regel 310. Regels 410 en 420 lezen het toetsenbord, waarbij, zoals hierboven reeds aangegeven werd, gebruik gemaakt wordt van de functie IN. Uw helikopter wordt afgedrukt op regel 430. Als u hebt besloten een letter te vangen, of op het vijandelijke vliegtuig te schieten, stuurt regel 440 de actie naar regel 580, waar de besturing verder doorgegeven wordt aan een subroutine, beginnende vanaf regel 1050. Hier wordt, indien nodig, uw laser-wapen afgevuurd. Als A in regel 440 gelijk is aan U, dan gaat het programma verder naar regel 700 om de letter te vangen, en wordt uw score verhoogd. Als u aan de uiterst linkerzijde van het scherm staat, is de variabele P gelijk aan 1, dus de computer gaat direct naar regel 700 om deze twee taken te vervullen.

Regel 460 verwijdert het straalvliegtuig van de Spectrum van het scherm, en gebruikt de volgende routine

de regel veranderd om de jet dichter bij uw helikopter te brengen. Regels 610 en 620 brengen een willekeurig effect in de bewegingen van de helikopter.
Regel 650 print de straaljager opnieuw op zijn nieuwe positie en het begin van regel 660 bepaalt, of de jet al dan niet op u zal schieten. Als het programma besluit een schot op u te lossen, wordt de subroutine op regel 1000 aangeroepen. Als T gelijk is aan 29, weet de computer, dat hij aan uw zijde van het scherm is, en stuurt hij de besturing door naar regel 1100, waar een letter van u gevangen wordt. Daarna wordt de zelfde letter aan de linkerzijde van het scherm verwijderd. U hoort dan een overwinningsmelodie, die gemaakt wordt door de subroutine op regel 800, voordat de letter op zijn juiste plaats gezet wordt.
Als de score, die u en de Spectrum tezamen behaald hebben, gelijk is aan 22, is het spel tot een einde gekomen, en regel 735 stuurt het programma dan naar regel 1200, waar de resultaten van de ALPHABATTLE bekend worden gemaakt. Regel 745 verwijdert de jet van zijn oude positie, en reset de waarden van P en U. Daarna gaat het programma naar regel 400, zodat het spel opnieuw kan beginnen.
Hieronder de listing van ALPHABATTLE:

```
 10 REM Alphabattle
100 FOR a=1 TO 4
110 FOR b=0 TO 7
120 READ c
130 POKE USR CHR$ (143+a)+b,c
140 NEXT b
150 NEXT a
200 DIM p(2,22)
205 DIM t(2,22)
210 FOR p=1 TO 22
215 FOR f=1 TO 2
220 LET e=INT (RND*22)
230 IF p(f,e+1)>0 THEN GO TO 22
```

```
 255 LET t(f,p)=e+1
 260 NEXT f
 270 NEXT p
 280 LET u=t(2,22)-1
 290 LET p=29
 300 LET a=t(1,22)-1
 310 LET t=1
 320 DIM h(2,22)
 330 LET s=0
 340 LET x=0
 400 PRINT AT u,p;"  "
 410 LET p=p+(IN 61438<>191)*(p<
29)-(IN 63486<>191)
 420 LET u=u+(IN 65278<>191)*(u<
21)-(IN 64510<>191)*(u>0)
 430 PRINT AT u,p; INK 4;"■■"
 440 IF IN 57342<>191 AND p>t TH
EN GO SUB 580: IF a=u THEN GO TO
 700
 450 IF p=1 THEN GO TO 700
 460 PRINT AT a,t;"  "
 470 IF p(1,u+1)<11 THEN GO TO 5
50
 480 LET a=a+(a<u)-(a>u)
 490 LET t=t+(ABS (a-u)<ABS (t-p
))-(ABS (a-u)>ABS (t-p))*(t>1)
 500 GO TO 650
 550 IF h(1,p(2,a+1))=1 OR h(2,p
(2,a+1))=1 THEN GO TO 600
 560 LET t=t+1
 570 GO TO 650
 580 GO SUB 1050
 590 BEEP .005,20: GO SUB 1050
 595 RETURN
 600 LET r=RND
 610 IF RND<.5 OR u=21 THEN LET
a=a-(a>0)
 620 IF RND>.5 OR a=0 THEN LET a
=a+(a<20)
 650 PRINT AT a,t; INK 5;"▒▒"
 660 IF ABS (a-u)<3 AND p>t AND
h(1,p(1,u+1))=0 AND h(2,p(1,u+1)
)=0 THEN GO SUB 1000: BEEP .005,
0: GO SUB 1000: IF a=u THEN GO T
O 850
 670 IF t=29 THEN GO TO 1100
 680 GO TO 400
 700 IF h(1,p(1,u+1))=1 OR h(2,p
(1,u+1))=1 THEN GO TO 745
 705 PRINT AT u,0;" "
 710 GO SUB 800
 720 PRINT FLASH 1;AT t(2,p(1,u+
```

```
 745 PRINT AT u,p;"   "
 750 LET p=29
 760 IF u=a THEN LET u=(a=0)
 765 PRINT AT 0,4;"Score ";(x),"
Score  ";(s);
 770 GO TO 400
 800 BEEP .005,10
 810 BEEP .01,20
 820 BEEP .005,10
 830 RETURN
 850 PRINT AT a,t;"   "
 860 LET a=u
 870 GO TO 1100
 900 DATA 0,64,48,62,31,31,31,30
 905 DATA 0,0,0,0,192,252,0,0
 910 DATA 0,3,7,15,31,8,127,0
 920 DATA 0,230,230,246,254,16,2
54,0
1000 PLOT OVER 1;(t+2)*8,(21-a)*
8+3
1010 DRAW OVER 1; INK 3;(p-t)*8,
0
1030 RETURN
1050 PLOT OVER 1;p*8,(21-u)*8
1060 DRAW OVER 1; INK 5;(t-p)*8,
0
1070 RETURN
1100 IF h(2,p(2,a+1))=1 OR h(1,p
(2,a+1))=1 THEN GO TO 1160
1105 PRINT AT a,31;" "
1110 GO SUB 800
1120 PRINT FLASH 1;AT t(1,p(2,a+
1))-1,0;CHR$ (64+p(2,a+1))
1130 LET x=x+((h(1,p(2,a+1)))=0)
1140 IF x+5=22 THEN GO TO 1200
1150 LET h(1,p(2,a+1))=1
1160 PRINT AT a,t;"   "
1170 LET t=1
1180 IF u=a THEN LET a=(u=0)
1190 GO TO 765
1200 CLS
1205 PRINT AT 11,12;"SPEL AFGELO
PEN"
1210 PRINT AT 19,0;
1220 PRINT "spectrum ";
1230 FOR a=1 TO x
1240 PRINT "/";
1250 NEXT a
1260 PRINT TAB 25;x;"humanoide "
;
1270 FOR a=1 TO s
```

```
10
1320 IF INKEY$="" THEN GO TO 132
0
1330 RUN
```

Smashout

In deze kleurige versie van 'Breakout' moet u trachten door een stenen muur heen te breken met behulp van een scherpe, diamant-vormige bal. Zoals u ziet, als u het spelletje speelt, is de grafische 'A' gedefinieerd om op een steen te lijken. De muur die u moet doorbreken ziet er dan ook erg effectief uit. U krijgt 12 ballen om op de muur te kaatsen. De bewegingen van uw paddle (aan de onderzijde van het scherm) bestuurt u met de '5'-toets (links) en de '8'-toets (rechts). De score is doorlopend zichtbaar, en wordt indien nodig veranderd. Regel 15 zet de BORDER en PAPER op rood, en de INK op wit. De grenzen, waarbinnen het spel zich afspeelt, worden afgedrukt in regel 20 (de bovenzijde) en in de loop op regel 30. Dit speelveld krijgt de kleur groen. (Er is natuurlijk geen enkele reden, waarom u zich aan dit kleurenschema zou moeten houden. Kleurentelevisies reageren verschillend op sommige kleuren, en combinaties die goed werken op de ene TV, kunnen op een andere wellicht ongeschikt zijn. Experimenteer hier eens mee, en zoek de best geschikte combinatie voor uw TV). De variabele S (voor de Score) wordt in regel 35 op nul gezet, en A en B die de vlucht van de bal bepalen, worden beide op 1 geïnitialiseerd. Regels 40 en 42 zijn loops, die de stenen (regel 0) en de bal (regel 42) definiëren, hierbij gebruik maken van de DATA-opdrachten op de regels 210 en 215.

De volgende loop, van regel 45 tot 55, print de stenen op het scherm. Merk op, dat regel 55 bestaat uit 28 grafische 'A'-s. Regel 50 kiest een willekeurig getal om de kleur van iedere rij stenen te bepalen. Het tweede

de zelfde kleur zou geven als de achtergrond. Als de regel ziet, dat de kleur toch rood is , dan gaat het programma terug naar het begin van de regel om een nieuwe kleur te vinden.
Regel 60 bepaalt de startpositie van de bal. De bal wordt afgedrukt op positie X,Y en de variabelen, die gebruikt worden om de bal te verwijderen (DX en DY), worden gelijk gemaakt aan de waarden van X en Y in deze regel. Als Y groter is dan 28 of kleiner dan 3, dan heeft de bal één van de zijkanten geraakt, en regel 65 stuurt dan het programma door naar regel 155, waar de richting van de bal wordt omgekeerd (op deze wijze 'stuitert' de bal dus op de zijkant).
Als X kleiner is dan twee, dan heeft de bal de bovenzijde van het speelveld bereikt, en regel 70 laat de bal dan naar beneden stuiteren. Regel 75 kijkt, of de positie, die de bal gaat innemen, een steen bevat, en als dit het geval is, dan wordt een subroutine aangeroepen, die begint op regel 125. Hier wordt een steen-verpulverend geluid ten gehore gebracht, en de waarde van B, die gebruikt wordt om de Y-waarde te verhogen wordt 'omgeklapt'. Als X kleiner is dan 20, is de bal aan de onderzijde van het speelveld. Regel 80, die de X-waarde van de balpositie in de gaten houdt, kijkt of de paddle in de juiste positie staat om de bal terug te kaatsen. Als dit zo is, wordt het kaatsgeluid gemaakt (BEEP .008,10), en de variabele A, die verandering van de positie van de bal bestuurt, wordt met –1 vermenigvuldigd.
Het eerste gedeelte van regel 85 gebruikt de variabelen DX en DY om de oude positie van de bal schoon te maken, de bal wordt daarna opnieuw op de nieuwe positie geprint met gebruikmaking van de variabelen X en Y. Regel 90 print de paddle. Merk hierbij op, dat de paddle een spatie aan iedere zijde heeft, zodat de oude

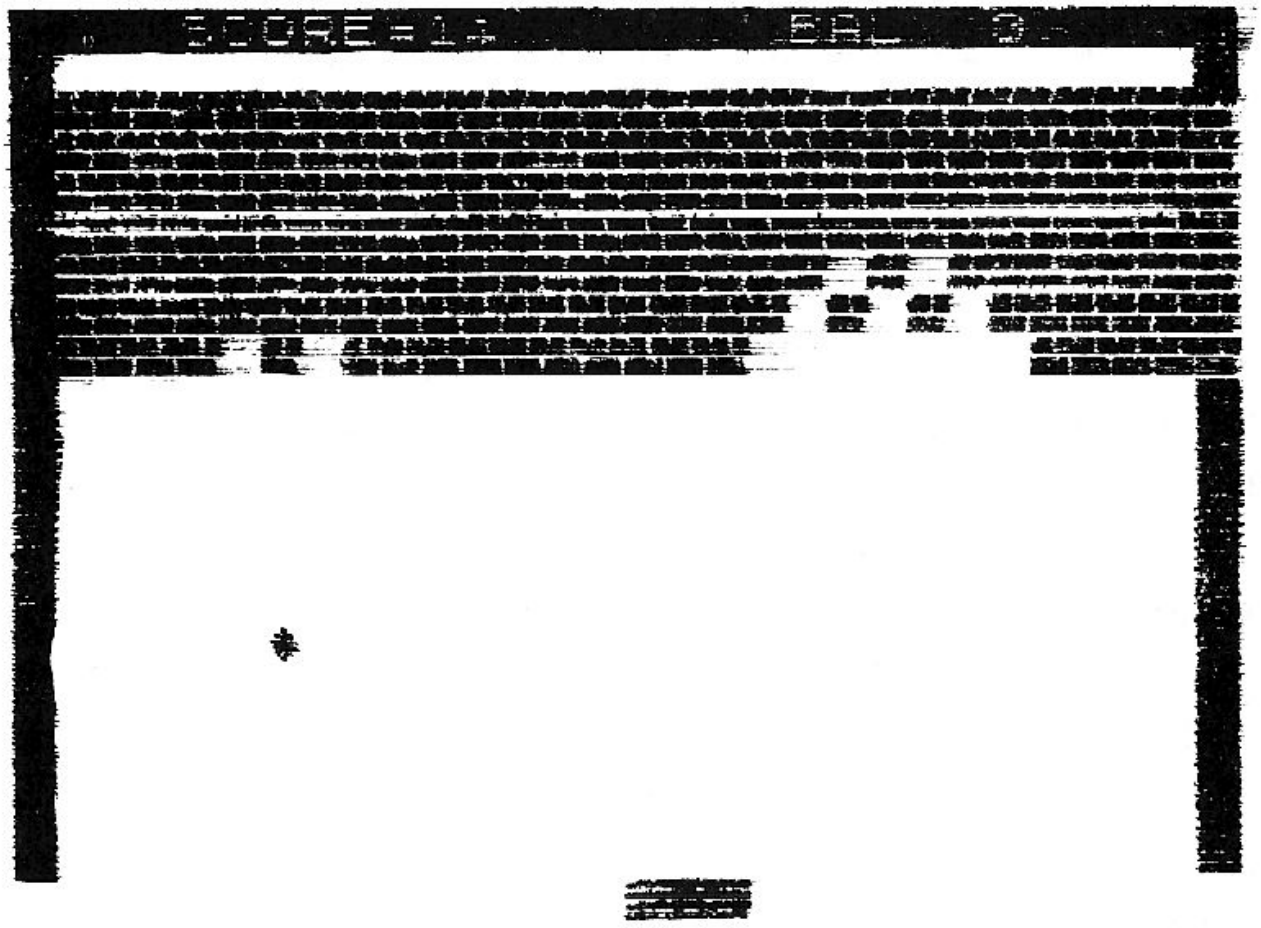
waarde van N om de waarde van N te wijzigen. Deze N bepaalt de positie van de paddle.
De balpositie wordt in regel 105 aangepast. Als A groter is dan 21 (regel 110), dan weet de computer dat de bal de paddle gemist heeft, en daarna uit het veld gevallen is. De subroutine vanaf regel 165 print het nummer van de bal, de balvariabele (Q) wordt verhoogd met 1 en een 'NIEUWE BAL'-geluid wordt daarna door regel 175 gecreëerd. Het aantal gebruikte ballen wordt in regel 180 gecontroleerd, en als dit kleiner is dan 12, gaat het programma verder op de volgende regel. Als alle 12 ballen gebruikt zijn, wordt de boodschap 'SPEL VOORBIJ' op het scherm afgedrukt in een aantal verschillende kleuren, waarbij een BEEP-loop verschillende geluiden produceert bij de flitsende kleuren. Na dit 'ademloze' display wordt een nieuw spel begonnen.
Als de maximale waarde van Q nog niet overschreden is, wordt er een nieuwe startpositie voor de bal bepaald, de onderste regel van het speelveld wordt geheel 'overgeprint', zodat de oude positie van de paddle verwijderd wordt, en de richting van de bal (omhoog en naar rechts, of omlaag en naar beneden) wordt bepaald aan de hand van de waarde van de variabele M, die in regel 185 vastgesteld is, en naar variabele B overgezet is in regel 200. Daarna wordt de besturing teruggegeven naar regel 60, waar het gehele spelletje opnieuw begint. De laatste twee regels van het programma – 219 en 215 – bevatten DATA-opdrachten voor respectievelijk de stenen van de muur en de bal.
Dit is de programma-listing van SMASHOUT:

```
   10 REM Smashout
   15 BORDER 2: PAPER 2: INK 7: C
LS
   20 PRINT INK 4;"
```

```
  35 LET s=0: LET a=1: LET b=1
  40 FOR n=USR "A" TO USR "A"+7:
 READ k: POKE n,k: NEXT n
  42 FOR n=USR "B" TO USR "B"+7:
 READ k: POKE n,k: NEXT n
  45 FOR n=2 TO 8
  50 LET z=INT (RND*8): IF z=2 T
HEN GO TO 50
  55 PRINT AT n,2; INK z; BRIGHT
 1;"[graphics]
[graphics]": NEXT n
  60 LET n=15: LET x=20: LET y=I
NT (RND*10)+5: LET dx=x: LET dy=
y
  65 IF y>28 OR y<3 THEN GO SUB
150
  70 IF x<2 THEN LET a=-a
  75 IF SCREEN$ (x,y)<>" " THEN
GO SUB 125
  76 IF x<>21 THEN GO TO 85
  80 IF y=n+3 THEN LET b=-1: BEE
P .008,10
  81 IF y=n+1 THEN LET b=1: BEEP
 .008,10
  82 IF y=n+2 THEN LET b=-b: BEE
P .008,10
  83 LET y=y+1: IF y>=n+1 AND y<
=n+3 THEN LET a=-a
  85 PRINT AT dx,dy;" ";AT x,y;"
[graphics]": LET dx=x: LET dy=y
  90 PRINT AT 21,n;"     "
  95 LET n=n+(INKEY$="8" AND n<=
26)-(INKEY$="5" AND n>0)
 105 LET x=x-a: LET y=y+b
 110 IF x>21 THEN GO TO 165
 115 PRINT AT 0,5; INVERSE 1;"SC
ORE=";s: INVERSE 0
 120 GO TO 65
 125 BEEP .008,20
 130 PRINT AT dx,dy;" ";AT x,y;"
[graphics]"
 135 LET a=-1
 140 LET s=s+1
 145 RETURN
 150 BEEP .008,30
 155 LET b=1-2*(y>28 OR y<2)
 160 RETURN
 165 PRINT AT 0,20; INVERSE 1;"B
AL: ";q: INVERSE 0
 170 LET q=q+1
 175 BEEP 1,0: BEEP 0.02,20
```

```
EP .008 -(d): NEXT d: NEXT z: RU
N
  185 LET m=RND
  190 PRINT AT 21,0;"
                    "
  195 LET a=1
  200 LET b=-b*(m<0.5)+b*(m>=0.5)
  205 GO TO 60
  210 DATA BIN 1110111,BIN 111101
11,BIN 11110111,0,BIN 11111110,B
IN 11111110,BIN 11111110,0
  215 DATA BIN 1000,BIN 11100,BIN
  111110,BIN 111110,BIN 11100,BIN
  1000,0,0
```

En zo ziet het spel er in actie uit:



3-D Drive

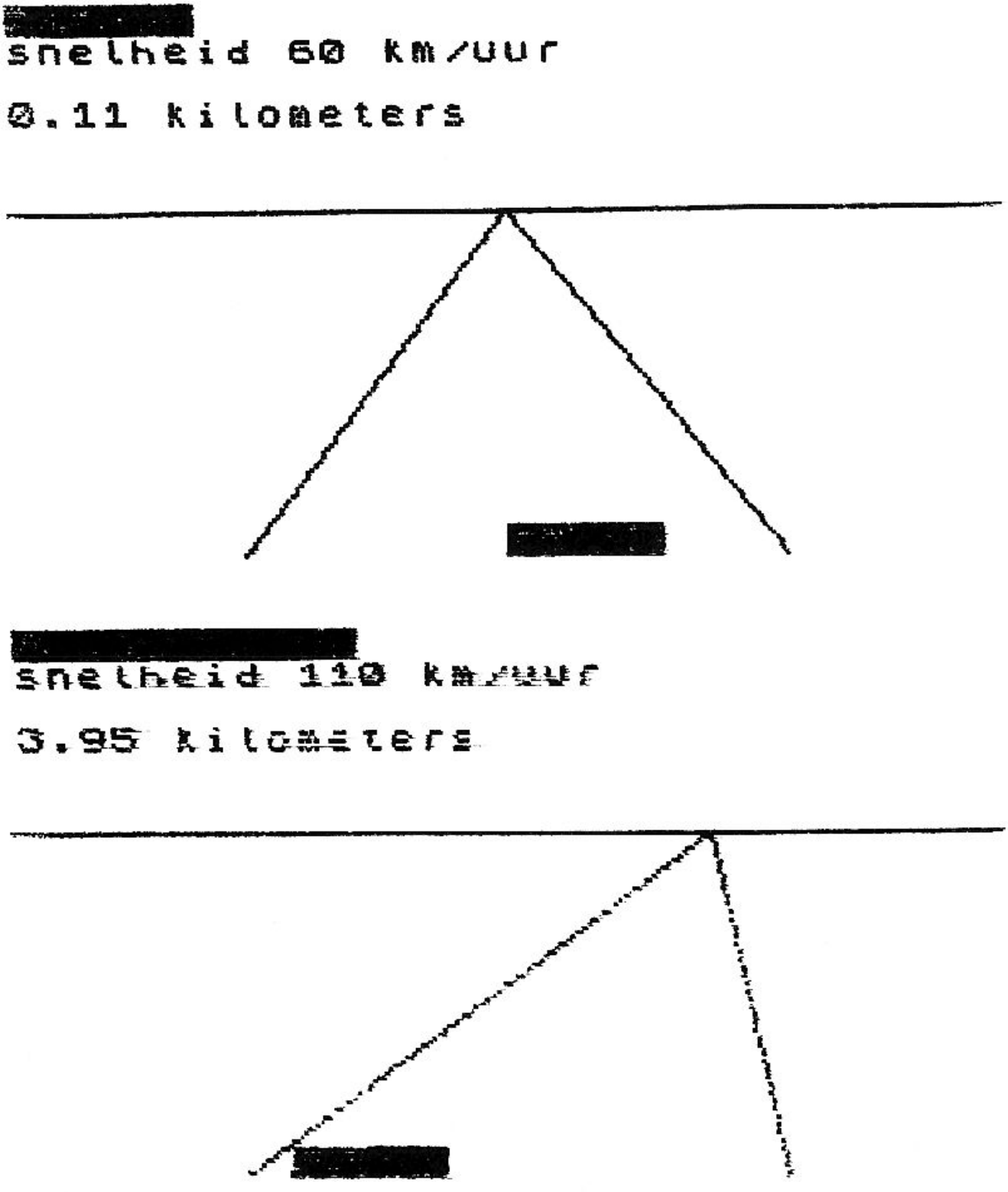
Dit korte programma is erg effectief en als u de smaak ervan te pakken heeft, zal het een echte uitdaging voor u kunnen gaan worden. U rijdt op een weg, die bestaat uit twee geDRAWde lijnen, die naar de horizon toe naar elkaar lopen. U gebruikt de '5'-toets (om naar links te

Het programma begint met het definiëren van een aantal variabelen (O, P, I$, Q, D en R), en stelt de PAPER en BORDER in op zwart en de INK op wit. Regel 35 leest het toetsenbord voor een '5' of een '8', en vermenigvuldigt deze met de snelheid I. In regel 40 wordt P opgeteld bij D (die initieel op 0 gesteld is in regel 25) en die gebruikt wordt om de coördinaten voor de lijnen te bepalen, die met een DRAW-opdracht in de subroutine vanaf regel 100 getekend worden. Deze lijnen stellen de weg voor. De variabele O is uw horizontale positie op het scherm (zie regel 70) en deze wordt berekend met behulp van de variabele P op regel 45. Uw snelheid wordt willekeurig gevarieerd op regel 50, en als P tussen 120 en 140 ligt, dan wordt het scherm op regel 55 schoongemaakt, voordat de weg in de subroutine op regel 100 opnieuw geprint wordt.
Hoewel het scherm iedere keer opnieuw schoongemaakt wordt en vervolgens de weg opnieuw geprint is, werkt DRAW zó snel, dat de 'nieuwe weg' vrijwel ogenblikkelijk op het scherm verschijnt. Er treedt slechts een minimum aan flikkering op.
Regel 65 controleert, of u één van beide zijden van de weg geraakt heeft, als dit het geval is, dan wordt het programma doorgestuurd naar de subroutine vanaf regel 150. Uw voertuig wordt op regel 70 opnieuw afgedrukt en de afgelegde afstand (variabele S) wordt verhoogd met een waarde, die afgeleid wordt van uw snelheid, voordat de afgelegde afstand op het scherm wordt uitgedrukt. Regel 77 geeft u de mogelijkheid uw snelheid te wijzigen (als D gelijk is aan 0), en regel 80 print een regel over de breedte van de bovenzijde van het scherm, die als een soort snelheidsmeter dienst doet. Regel 90 stuurt het programma terug naar regel 35 voor de volgende seconden van uw rit. De subroutine van

finable grafische tekens gebruiken om een voorruit en een stuur voor te stellen.
Dit is de programma-listing van 3-D Drive:

```
   1 REM 3-D Drive
   3 INK 7: BORDER 0: PAPER 0: C
LS
   5 LET o=16: LET s=0
  10 LET p=125
  12 LET i$="[redacted]
  15 LET i=1
  20 LET q=50
  25 LET d=0
  30 LET r=200
  35 LET p=p+(i*2)*((INKEY$="8")
-(INKEY$="5"))
  40 LET r=r+d
  45 LET o=(255-p)/6
  50 IF p>120 AND p<140 THEN LET
 i=i+1: LET d=i*(INT  (RND*2)-IN
T (RND*2))
  55 CLS
  60 PLOT 0,87: DRAW 255,0: PLOT
 q,0: DRAW INK 6;p-q,87: PLOT r,
0: DRAW INK 6;p-r,87
  65 IF ATTR (21,o)=6 OR ATTR (2
1,o+4)=6 THEN GO TO 150
  70 PRINT AT 21,o;"[redacted]"
  75 LET s=s+i/100: PRINT AT 3,0
;s;" kilometers "
  77 IF INKEY$="1" AND d=0 THEN
LET i=i-2*(i>2): BEEP .01,0
  80 PRINT AT 0,0; INK 2; BRIGHT
 1;i$( TO i); INK 6;AT 1,0;"snel
heid ";i*10;" km/uur"
  90 GO TO 35
 100 PLOT 0,87: DRAW 255,0: PLOT
 q,0: DRAW INK 6;p-q,87: PLOT r,
0: DRAW INK 6;p-r,87: RETURN
 150 PRINT AT 11,1; FLASH 1; BRI
GHT 1; INK 2; PAPER 6;"u heeft "
;s;" kilometer gereden"
 160 BORDER RND*7
 165 BEEP .007,30+RND*20
 180 IF RND<.96 THEN GO TO 160
 190 RUN
 200 PRINT AT 0,0;p;q;r
 205 INK 0: PLOT p,0
```

En zo ziet het programmascherm er uit:



Norway Panic

Als u 3-D Drive een beetje onder de knie heeft, dan kunt u uw geluk beproeven met een ander autosport-programma, Norway Panic genaamd. In dit programma rijdt u met uw auto in Noorwegen door een reeks tunnels en muren en tussen bergen door. U ziet de muren naar

u leert om de grafische weergave te 'zien' als een weg en wegzijden, maar als u dit eenmaal voor elkaar heeft, dan zult u het een uitstekende test vinden, om met de '5'- en '8'-toets uw stuurkunsten te oefenen.
Het array A$, dat in regel 120 gedimensioneerd is, bevat de negen verschillende posities van de muren, die, terwijl ze naderbij komen, verschillend van onderlinge afstand en grootte worden (door de veranderingen in het perspectief). De variabele T (regel 70) bepaalt uw startpositie voordat de muren naar beneden toe op het scherm afgedrukt worden in de B-loop (van regel 90 tot regel 130). Uw score, S, wordt in regel 107 verhoogd, en regel 120 leest het toetsenbord (voor een '5' of een '8'). De gelezen waarde wordt omgezet in een element van de muur, die afgedrukt wordt en de waarde van T. Hierbij zorgt het programma er wel voor, dat de waarde van T nooit groter wordt dan 21, om er voor te zorgen, dat de muur altijd op het scherm blijft.
In regel 140 wordt onderzocht of de variabele T tussen 13 en 9 ligt. Als dit het geval is, bent u verongelukt, en wordt het programma voortgezet op regel 200, waar uw score afgedrukt wordt, en u een nieuw spelletje aangeboden krijgt. Als u echter nog niet verongelukt bent na een doorgang van de B-loop, dan wordt de variabele I (die in regel 5 op de waarde 1 gezet was) gelijk gemaakt aan de negatieve waarde van I (dus –1 verandert in 1 en 1 verandert in –1). Deze waarde bepaalt, welke PAPER/BORDER/INK-combinatie gebruikt zal worden voor de volgende weg, die u moet berijden. De combinaties met een zwarte omgeving en gele muren stellen wegen voor, die tussen de bergen door lopen, de andere combinaties zijn wegen in de Noorse buitenlucht. Als u dit programma overmeesterd heeft, dan kunt u bijvoorbeeld de hoogste score vast leggen, of een manier om

Dit is de programma-listing:

```
    1 REM Norway Panic
    5 LET i=1
    7 BORDER 4: INK 0: PAPER 7: C
LS
   10 DIM a$(9,10)
   20 FOR a=1 TO 9
   30 READ a$(a)
   40 NEXT a
   42 DATA "          "
   43 DATA "          "
   44 DATA "          "
   45 DATA "          "
   47 DATA "          "
   48 DATA "          "
   49 DATA "          "
   50 DATA "          "
   51 DATA "          "
   55 LET s=1
   60 LET a=9
   70 LET t=INT (RND*22)
   80 PRINT AT 21,14; INK 2;"####
#"
   90 FOR b=1 TO 12
  100 PRINT AT a+b,t;a$(INT ((b/1
.3)+.5))
  105 BEEP 1/s,b
  107 LET s=s+1: PRINT AT 0,0;s
  110 PRINT AT a+b,0;"
                      "
  120 LET t=t-((INKEY$="8")*(t>0)
-(INKEY$="5")*(t<21)*(INT (b/4)
)
  121 PRINT AT 0,20;t
  130 NEXT b
  140 IF t>14 OR t<9 THEN GO TO 2
00
  160 LET i=-i
  170 IF i=-1 THEN PAPER 0: BORDE
R 6: INK 6
  180 IF i=1 THEN PAPER INT (RND*
7)+1: BORDER 4: INK 0
  190 CLS
  195 GO TO 60
  200 FOR a=1 TO 20
  205 LET n=INT (RND*8): IF n=2 T
HEN GO TO 205
  210 PAPER n: CLS
  220 NEXT a
  225 PAPER 6: INK 0: CLS
  230 PRINT AT 11,10;"Score ";s
```

Death Race 2000

Kunt u de gevaarlijke groene gluiper-kruiper ontlopen, en door Death Race 2000 ontsnappen, voordat u gegrepen wordt? Dit programma zal u de waarheid leren! Het programma tekent een rechthoekig doolhof op het scherm, met u aan de linkerzijde en de voornoemde gluiper-kruiper ergens in het doolhof. U kunt de '5'- en '8'-toetsen gebruiken om naar links en rechts te bewegen, en de 'w'- en 'z'-toetsen voor onder en boven. Hierdoor kunt u het kleine mannetje (het figuurtje aan het einde van regel 240) manipuleren, zodat deze van de linkerzijde naar de uiterste rechterzijde van het doolhof loopt.

Op regel 5 wordt de random-getallen -generator ingesteld en regels 10 tot 30 zetten de BORDER en PAPER op zwart en de INK op geel. Regel 40 zendt de besturing naar de initialisatie-subroutine op regel 500, waar de DATA, die voor de twee gebruiker-gedefinieerde grafische trekken gebruikt worden, opgeslagen is. Zodra de initialisatie gedaan is, keert het programma terug naar regel 50. Hier wordt een functie gedefinieerd om willekeurige (random) getallen te genereren. Het scherm wordt schoongemaakt in regel 55, zodat de PAPER-kleur uit regel 30 in werking treedt, en de twee variabelen X (het getal, dat gebruikt moet worden in functie A om de willekeurige gaten in het doolhof te produceren) en S (U begin-score, die in regel 245 met 16 verlaagd wordt, als het programma aan het einde van een doorgang van de hoofd-loop) worden geïnitialiseerd.

Regel 100 print een solide gele balk aan de bovenzijde van het scherm, de loop van regel 110 tot 130 print balken naar beneden over het scherm in cyaan (regel 120), daarbij is de rechterzijde van het scherm ook weer geel (regel 125). Regel 140 completeert de ruwe achter-

op het scherm gaten te printen in de solide balken. Dit zijn de plaatsen, die u moet gebruiken (en die de gluiper-kruiper gebruikt) in uw adembenemde race naar de andere zijde van het scherm. Regels 190 tot 220 geven een waarde aan u (de horizontale start-coördinaat) en P (deze variabele krijgt de waarde 1, omdat het mannetje op de meeste linker positie in het scherm moet starten).
Regel 235 bekijkt of de gluiper-kruiper en het mannetje op dezelfde plaats staan. Als dit zo is, dan heeft de gluiper-kruiper toegeslagen, en gaat het programma naar regel 400. Na een korte melodie uit de 'gluiper-kruiper Overwinnings Symfonie' wordt u meegedeeld, dat u verloren heeft. De rest van de hoofd-loop van regel 230 tot 315 kijkt naar wat u ingetikt heeft, en verandert aan de hand hiervan de positie van het mannetje.
'U heeft de Death Race gewonnen' is de boodschap, die u op regel 360 te zien krijgt. Op regel 365 wordt gecontroleerd, of u de hoogste score verbeterd heeft, als dit zo is, dan wordt de variabele H, waarin de hoogste score is opgeslagen, veranderd in uw laatste score. De regels 370 en 375 printen de score en de hoogste score. Merk hierbij op, dat de lengte van de stringversie van de score (LEN STR$ S) gebruikt wordt om de print-output te positioneren. Regels 380 en 390 wachten totdat u uw handen van het toetsenbord heeft afgehaald (380), en daarna totdat u weer een toets heeft ingedrukt om een nieuwe race te starten (vanaf regel 55). Als u erg veel zelfvertrouwen heeft in uw mogelijkheden, om dit spelletje te winnen, dan kunt u het veranderen, zodat het meer dan één gluiper-kruiper bevat.
Dit is de programma-listing van Death Race 2000:

```
5 RANDOMIZE
7 LET h=0
```

```
  50 DEF FN a(x)=INT  (RND*x)+1
  55 CLS
  60 LET x=20
  70 LET s=10000
  80 REM [illegible]
  90 REM [illegible]
 100 PRINT INVERSE 1;"
                      "
 110 FOR a=0 TO 20
 120 PRINT INK 5;"■ ■ ■ ■ ■ ■ ■
■ ■ ■ ■ ■ ■ ■ ■ ■ ■"
 125 PRINT AT a,30, INVERSE 1;"
■"
 130 NEXT a
 140 PRINT AT 21,0, INVERSE 1,"
                              "
 150 FOR a=2 TO 28 STEP 2
 160 PRINT AT FN a(x),a;" "
 170 PRINT AT FN a(x),a;" "
 180 NEXT a
 190 LET u=FN a(x)
 200 LET a=FN a(x)
 210 LET p=1
 220 LET t=FN a(15)*2-1
 230 PRINT AT INT a,t; INK 4;"◆"
 235 IF ATTR (u,p)=4 THEN GO TO
400
 240 PRINT AT u,p;"■"
 245 LET s=s-16
 250 BEEP .03,s/200
 255 PRINT AT 0,1; INVERSE 1;s;"
"
 260 PRINT AT u,p;" "
 265 IF INKEY$="6" AND ATTR (u,p
+1)=6 THEN LET p=p+2: BEEP .1,p:
 BEEP .1,p-12
 270 IF INKEY$="5" AND ATTR (u,p
-1)=6 THEN LET p=p-2: LET s=s-10
0: BEEP .1,0
 275 IF p=29 THEN GO TO 320
 280 LET u=u+(INKEY$="z")*(u<20)
-(INKEY$="w")*(u>1)
 285 PRINT AT INT a,t;" "
 290 LET a=a+RND*(INT a<u)-RND*(
INT a>u)
 295 IF ATTR (INT a,t+1)=6 AND t
<p THEN LET t=t+2
 300 IF ATTR (INT a,t-1)=6 AND t
>p THEN LET t=t-2
 315 GO TO 230
 320 FOR a=30 TO 60
```

```
ft de Death Race gewonnen"
 365 IF h<s THEN LET h=s
 370 PRINT AT u+1-2*(u=20),30-LE
N STR$ s;s
 375 PRINT AT 0,1; FLASH (h=s);
INVERSE 1;"HOOGSTE SCORE ";h
 380 IF INKEY$<>"" THEN GO TO 38
0
 385 IF INKEY$="" THEN GO TO 385
 390 GO TO 55
 400 FOR f=60 TO 30 STEP -1
 410 BEEP .01,f
 420 BEEP .01,f-12
 430 NEXT f
 445 PRINT AT u+1-2*(u=20),0; OV
ER 1;"U heeft de Death Race verl
oren"
 450 PRINT AT INT a,t; FLASH 1;
INK 4;"♦"
 460 GO TO 375
 500 DATA 24,24,48,94,16,104,76,
0
 510 DATA 0,34,84,12,12,64,3,0
 520 FOR a=0 TO 1
 530 FOR b=0 TO 7
 540 READ c
 550 POKE USR CHR$ (144+a)+b,c
 560 NEXT b
 570 NEXT a
 580 RETURN
```

Clone Crazy

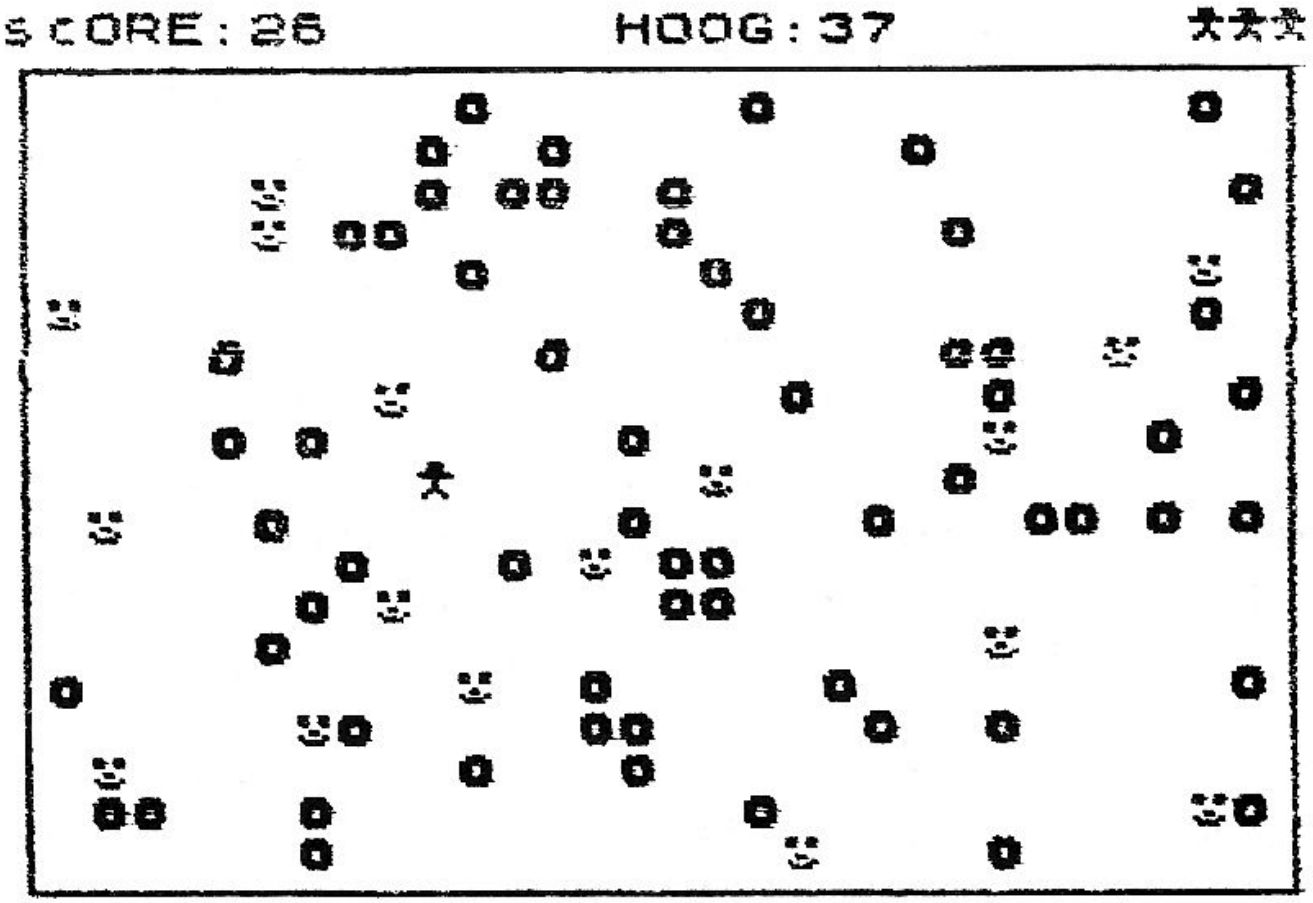
Een zombie heeft zichzelf 19 maal gecloond, en er zijn nu 20 kwaadaardige zombies op pad om u te grazen te nemen. U zit gevangen in een rechthoekig gebied in Zombieland, en de topografie van het gebied laat u zien, dat er hier behalve uzelf en de zombies, een aantal putten in de grond zitten.

De zombies hebben alle twee dingen gemeen: ze hebben een zeer slimme, ingebouwde vaardigheid om mensen op te sporen, en ze hebben een aan het maniakale grenzende haat tegen mensen. Deze 20 waanzinnige zombies zijn er dus allemaal op uit, uw leven snel en doeltreffend te beëindigen. Er is echter een lichtpuntje:

Zoals u zich herinneren zult, zijn er een aantal putten in de grond in dit gebied van Zombieland. De ingebouwde vaardigheid van de zombies om mensen op te sporen, laat verstek gaan op het gebied van putten. Lok dus een zombie in put en hij stort hulpeloos naar beneden. De beste methode om te ontsnappen is dus uw voordeel te doen met het slechte gezichtsvermogen van de zombies, en ze in een put te lokken. Dit kunt u doen door zó te lopen , dat er een put tussen u en de zombie in zit. Denk er echter wel aan, dat u 20 belagers heeft, dus terwijl u in de buurt van een put staat om een zombie in de val te laten lopen, is het mogelijk, dat een andere zombie u te pakken krijgt.
U kunt naar links, naar rechts, onder of boven bewegen, en – als u zeer moedig bent – kunt u ook stil blijven staan. De zombies bewegen horizontaal, vertikaal en diagonaal. U begint iedere ronde van Clone Crazy met vier levens en een complete groep van 20 zombies. U krijgt 1 punt voor iedere zombie, die u in een put weet te lokken, en 6 bonus-punten, als u het voor elkaar krijgt, om alle zombies te vernietigen.
Iedere keer dat u overleeft, wordt het land opnieuw getekend, met een nieuwe kolonie zombies, en met minder putten. Als u in een put valt, door een zombie opgegeten wordt, of van de rand van het land valt, bent u een leven kwijt. Vier van zulke blunders en u bent 'overleden'.
Het programma houdt de hoogste score bij, dus u kunt de eerst komende vijf jaar of zo trachten de beste score te verkrijgen. U kunt over zombieland lopen met de toetsen 'a' (omhoog), 's' (omlaag), 'k' (links) en 'l' (rechts). Met iedere andere toets stopt u met lopen, en staat u op die plaats stil.
Na het instellen van de random-getallen-generator op regel 1 wordt een array gedimensioneerd dat alle zom-

bies en uzelf maakt. In regel 9100 wordt de BORDER op blauw gezet, de PAPER op geel en de INK-kleur op zwart. Daarna krijgt u de gebruiksaanwijzing te zien, waarop u kunt aflezen welke toetsen u kunt gebruiken. Op regel 9200 krijgt u de boodschap 'Druk een toets in om te starten' te zien, waarna een PAUSE 0-opdracht het programma laat wachten totdat u een toets indrukt.
De RETURN in regel 990 stuurt het programma terug naar regel 20, waar het aantal levens, dat u kunt krijgen, op 4 gezet wordt. De gebruikte variabele is MEN, en de inhoud hiervan is 1 minder dan het aantal levens dat u nog heeft, dus als MEN=0, dan heeft u nog 1 leven over. De variabele SCORE wordt op nul gezet en andere variabelen, die het programma gaat gebruiken, worden geïnitialiseerd (regel 30).
Het adres 23560 wordt met een 0 gePOKEd. Dit adres bevat de waarde van de laatst-ingedrukte toets, en het POKEn met een 0 betekent, dat deze toets-indruk weggegooid wordt. Regel 100 stuurt het programma naar de subroutine vanaf regel 1000 die het zombieland tekent. De rand wordt getekend in regel 1010, en de dubbele loop (I en J) vanaf regel 1020 plaatst de putten op het scherm. Het aantal putten wordt bepaald door de waarde van de variabele RAND in regel 1040 en is ook afhankelijk van de waarde, die gegenereerd wordt door de RAND op die regel.
Nadat uit deze subroutine is teruggekeerd, wordt door regel 110 de besturing overgegeven aan de subroutine op regel 3000, die de zombies op het scherm plaatst, daarna alle zombies telt (begin van regel 3050) en de zombies bestuurt vanuit het array Z (laatste twee delen van 3050). Vanuit deze subroutine wordt teruggekeerd naar regel 120, die daarna de subroutine vanaf regel 2000 aanroept. Deze subroutine plaatst het mannetje

naast een zombie of een put is. Als dit niet gedaan zou worden, dan zou u wel eens door een zombie gegrepen kunnen worden, of in een put lopen, voordat het programma werkelijk begonnnen was.



Als u naar dit programmavoorbeeld van zombie kijkt, dan ziet u, dat het aantal levens, dat u nog over heeft, aangegeven wordt door het aantal mannetjes, dat in de rechter bovenhoek staat, direct boven het land.
De regels 130 tot 150 plaatsen deze mannetjes op het scherm, regel 160 print de score bovenaan het scherm, en de hoogste score ziet u midden boven het land. Dit wordt verzorgd door regel 170.
U zult zich herinneren, dat in regel 40 het adres 23560 met de waarde 0 gePOKEd was, om de input-buffer van het toetsenbord schoon te maken. Nu we hier in het programma zijn aangekomen, verwachten we, dat u een toets indrukt, die een bepaalde actie voor het

heeft. Voordat de computer op deze informatie reageert (regels 210 tot 240), wordt eerst uw huidige positie opgeslagen in de variabelen TX en TY. Deze variabelen worden gebruikt om uw positie schoon te maken, in het verloop van het programma (regel 450). Als u geen van de vier toetsen heeft ingedrukt, dan springt regel 250 over de rest van de routine heen, die uw positie controleert en u opnieuw print. Regel 260 bekijkt even vlug, of u niet zo dom bent geweest om over de rand heen te stappen, en regel 270 bekijkt of de plaats waar u naar toe wilde nog vrij is. Als de regel iets vindt op uw nieuwe positie, dan bent u er geweest, en de routine vanaf regel 4000 laat u een dramatisch 'Son et lumière'-display zien om uw stommiteit nog eens te benadrukken. Als u de test op regel 270 wel doorkomt, dan wordt uw mannetje op regel 280 op de nieuwe positie afgedrukt.

De routine van regel 300 tot regel 490 wordt uitgevoerd totdat de variabele NUM (verhoogd in regel 400) groter wordt dan de waarde van de variabele GO (die oorspronkelijk ingesteld was op de waarde 4 in regel 30, en die veranderd wordt door regel 485). Deze gehele routine bevat de kwade geest achter de bewegingen van de zombies. Een bijbehorende routine, van regel 500 tot regel 530, wordt aangeroepen door regel 410 als de SCREEN$-test uitwijst, dat er een lege positie gevonden is. Als het aantal zombies, dat nog in leven is (de informatie hierover straat in de variabele LEFT), gelijk wordt aan nul, dan heeft u alle zombies vernietigd en krijgt u een bonus-score (regel 8000 tot 8050). Dit wordt nog eens bevestigd door regel 470. De waarde van de variabele RAND wordt verlaagd, dit betekent dat u in de volgende ronde minder putten op het scherm te zien krijgt. Het aantal putten op het scherm wordt

ces weer opnieuw begint. De loop vanaf regel 7000 print de zombies, met daarbij een kleine muzikale ondersteuning (regels 7020 en 7040). Als het aantal levens, dat u nog over heeft (variabele MEN, zie regel 7060) gelijk is aan nul, dan gebruikt de computer de routine vanaf regel 7070 om de hoogste score, indien nodig, aan te passen, en om u de mogelijkheid van een nieuw spelletje aan te bieden. Als u 'n' intikt, om aan te geven, dat u niet meer wilt spelen, dan wordt door regel 7090 de INK, BORDER en PAPER op startwaarde gezet, zodat het daarna makkelijker is om het programma op uw scherm uit te listen, of om een ander programma in te toetsen. Als u een nieuw spel wilt spelen (het antwoord 'j' wordt in regel 7080 gelezen), dan gaat de besturing terug naar regel 25, die er voor zorgt, dat uw hoogste score HI, niet op nul wordt gezet.
Eventuele veranderingen die u aan het programma aan kunt brengen, kunnen bijvoorbeeld de te gebruiken toetsen veranderen. Dit kunt u doen, door regels 210 tot 240 aan te passen (waar de numerieke waarde van de laatst-ingedrukte toets opgehaald wordt). Als u dit doet, moet u ook de instructies op de regels 92150 tot 9180 veranderen. Om meer of minder putten te genereren, kunt u de initiële waarde van variabele RAND op regel 15 en regel 7063 veranderen. Om het veranderen van het aantal putten aan het einde van iedere ronde van het spel te veranderen, moet u regel 8006 aanpassen.
De gebruikte USER-DEFINED grafische tekens zijn de 'A' (het mannetje), die 'B' (voor de zombies) en de 'C' (voor de putten).
Dit is de programmalisting van Clone Crazy.

```
1 RANDOMIZE
```

```
  25 LET men=3: LET score=0
  30 LET left=0: LET max=20: LET
 zombie=0: LET go=0
  40 POKE 23560,0
 100 GO SUB 1000
 110 GO SUB 3000
 120 GO SUB 2000
 130 IF men>2 THEN PRINT AT 0,29
;"☥"
 140 IF men>1 THEN PRINT AT 0,30
;"☥"
 150 IF men>0 THEN PRINT AT 0,31
;"☥"
 160 PRINT AT 0,0;"SCORE:";score
 170 PRINT AT 0,15;"HOOG:";hi
 200 LET a=PEEK 23560
 205 LET tx=px: LET ty=py
 210 IF a=97 THEN PRINT AT py,px
;" ": LET py=py-1: GO TO 260
 220 IF a=115 THEN PRINT AT py,p
x;" ": LET py=py+1: GO TO 260
 230 IF a=107 THEN PRINT AT py,p
x;" ": LET px=px-1: GO TO 260
 240 IF a=108 THEN PRINT AT py,p
x;" ": LET px=px+1: GO TO 260
 250 GO TO 300
 260 IF px<1 OR px>30 OR py<2 OR
 py>20 THEN GO TO 4000
 270 IF SCREEN$ (py,px)<>" " THE
N GO TO 5000
 280 PRINT AT py,px;"☥"
 300 LET num=0
 305 IF num>go THEN GO TO 200
 310 LET zombie=zombie+1: IF zom
bie>max THEN LET zombie=1
 320 IF z(zombie,1)=0 THEN GO TO
 310
 330 LET zx=z(zombie,1): LET zy=
z(zombie,2): LET tx=zx: LET ty=z
y
 340 LET dx=0: LET dy=0
 350 IF zx<px THEN LET dx=1
 360 IF zx>px THEN LET dx=-1
 370 IF zy<py THEN LET dy=1
 380 IF zy>py THEN LET dy=-1
 390 LET zx=zx+dx: LET zy=zy+dy
 400 LET num=num+1
 410 IF SCREEN$ (zy,zx)=" " THEN
 GO TO 500
 420 IF SCREEN$ (zy,zx)="S" THEN
 GO TO 305
```

```
  460 LET score=score+1: PRINT AT
  0,0;"scORE:";score
  470 LET left=left-1: IF left=0
THEN GO TO 8000
  480 FOR n=1 TO 10: BEEP 0.01,n:
 NEXT n
  485 LET go=1+INT (left/5)
  490 GO TO 305
  500 PRINT AT ty,tx;" "
  510 PRINT AT zy,zx; FLASH 1; IN
K 2;"☺"
  520 LET z(zombie,1)=zx: LET z(z
ombie,2)=zy
  525 BEEP 0.02,30-left
  530 GO TO 305
1000 PAPER 6: CLS
1010 PLOT 4,4: DRAW 248,0: DRAW
0,159: DRAW -248,0: DRAW 0,-159
1020 FOR i=2 TO 20
1030 FOR j=1 TO 30
1040 IF RND>rand THEN GO TO 1060
1050 PRINT AT i,j; INK 1;"■"
1060 NEXT j: NEXT i
1100 RETURN
2000 LET px=INT (RND*13+5)
2010 LET py=INT (RND*17+5)
2020 IF SCREEN$ (py,px)<>" " THE
N GO TO 2000
2021 IF SCREEN$ (py-1,px-1)<>" "
 THEN GO TO 2000
2022 IF SCREEN$ (py,px-1)<>" " T
HEN GO TO 2000
2023 IF SCREEN$ (py+1,px-1)<>" "
 THEN GO TO 2000
2024 IF SCREEN$ (py-1,px)<>" " T
HEN GO TO 2000
2025 IF SCREEN$ (py+1,px)<>" " T
HEN GO TO 2000
2026 IF SCREEN$ (py-1,px+1)<>" "
 THEN GO TO 2000
2027 IF SCREEN$ (py,px+1)<>" " T
HEN GO TO 2000
2028 IF SCREEN$ (py+1,px+1)<>" "
 THEN GO TO 2000
2030 PRINT AT py,px;"☠"
2100 RETURN
3000 FOR j=2 TO 20
3020 LET i=INT (RND*30+1)
3030 IF SCREEN$ (j,i)<>" " THEN
GO TO 3020
3040 PRINT AT j,i; FLASH 1; INK
```

```
3100 RETURN
4000 FOR i=1 TO 20
4010 PRINT AT py,px;" ": PRINT A
T ty,tx;"*"
4020 BEEP .02,i
4030 PRINT AT py,px;"*": PRINT A
T ty,tx;" "
4040 BEEP .02,20-i
4050 NEXT i
4060 GO TO 7060
5000 FOR i=1 TO 20
5020 PRINT AT py,px;" ": PRINT A
T ty,tx;"*"
5030 BEEP .02,20+i
5040 PRINT AT py,px;"*": PRINT A
T ty,tx;" "
5050 BEEP .02,20-i
5060 NEXT i
5070 GO TO 7060
7000 FOR i=1 TO 20
7010 PRINT AT py,px; INK 3;"S":
PRINT AT ty,tx;" "
7020 BEEP .02,20+i
7030 PRINT AT py,px;"*": PRINT A
T ty,tx;"S"
7040 BEEP .02,20-i
7050 NEXT i
7060 LET men=men-1: IF men<0 THE
N GO TO 7070
7062 FOR i=1 TO 100: NEXT i
7063 LET rand=0.1
7065 GO TO 30
7070 IF score>hi THEN LET hi=sco
re: PRINT AT 0,15;"HOOG:";hi
7075 PRINT AT 1,6;"NOG EEN KEER
(J/N) ";
7080 IF INKEY$="j" THEN GO TO 25
7090 IF INKEY$="n" THEN BORDER 7
: PAPER 7: INK 0: STOP
7100 GO TO 7080
8000 PRINT AT 0,10; FLASH 1;"***
 BONUS ***"
8005 LET score=score+5
8006 LET rnd=rand*0.6
8010 FOR i=1 TO 3
8020 FOR j=1 TO 50
8030 BEEP 0.01,j
8040 NEXT j
8050 NEXT i
8060 GO TO 30
9000 POKE USR "A",BIN 00011100
```

```
9005 POKE USR "A"+5,BIN 00010100
9006 POKE USR "A"+6,BIN 00100010
9007 POKE USR "A"+7,BIN 00000000
9010 POKE USR "B",BIN 00000000
9011 POKE USR "B"+1,BIN 01100110
9012 POKE USR "B"+2,BIN 01100110
9013 POKE USR "B"+3,BIN 00000000
9014 POKE USR "B"+4,BIN 00011000
9015 POKE USR "B"+5,BIN 01000010
9016 POKE USR "B"+6,BIN 00111100
9017 POKE USR "B"+7,BIN 00000000
9020 POKE USR "C",0
9021 POKE USR "C"+1,BIN 00111100
9022 POKE USR "C"+2,BIN 01111110
9023 POKE USR "C"+3,BIN 01100110
9024 POKE USR "C"+4,BIN 01100110
9025 POKE USR "C"+5,BIN 01111110
9026 POKE USR "C"+6,BIN 00111100
9027 POKE USR "C"+7,0
9100 BORDER 1: PAPER 6: INK 0: C
LS
9110 PRINT AT 1,13;"Clone Crazy"
9120 PRINT AT 1,13; OVER 1;"_____
_____"
9130 PRINT AT 4,4;"  Mike O'Neil
l Juli 1982"
9140 PRINT AT 8,4;"Toetsen:"
9150 PRINT AT 10,4;"A=Omhoog"
9160 PRINT AT 11,4;"S=Omlaag"
9170 PRINT AT 12,4;"K=Links"
9180 PRINT AT 13,4;"L=Rechts"
9190 PRINT AT 15,4;"Iedere ander
e toets=STOP"
9200 PRINT AT 19,2;"Druk een toe
ts om te starten"
9210 PAUSE 0
9900 RETURN
```

```
```

BORDSPELEN

Pirandello

Dit spel is gebaseerd op het in 1880 uitgevonden spel Reversi, en wordt ook wel 'Othello' genoemd. Reversi wordt gespeeld op een schaakbord. In Reversi kunnen de spelers hun eerste twee stukken op een willekeurig veld van de vier centrale velden plaatsen. In PIRANDELLO is de plaats van de eerste vier stukken vast, en het spel start dan ook met deze stukken reeds op het bord geplaatst, zoals u bij de uitvoering van het programma ook kunt zien.
Hierna worden de zetten bepaald door één simpele regel: ieder nieuw te plaatsen stuk moet direct aangrenzend aan een stuk van de tegenstander geplaatst worden en aan de andere zijde van de reeks van stukken van de tegenstander moet een eigen stuk staan. Alle stukken van de tegenstander op die lijn veranderen dan in 'eigen' stukken. De reeks stukken kan in iedere richting lopen, zowel horizontaal, vertikaal als diagonaal. Als een stuk van een speler een lijn afmaakt in meer dan één richting, dan veranderen de stukken in alle richtingen, waarin een lijn afgemaakt wordt. Het deel van het spel is, zo veel mogelijk stukken te veranderen in 'eigen' stukken, zodat als het bord vol is, of als er geen zetten meer mogelijk zijn, u het meeste aantal stukken op het bord heeft. De winnaar wordt bepaald door eenvoudig het aantal stukken van beide partijen te tellen.
Om te laten zien hoe de stukken in PIRANDELLO veranderen, moet u zich voorstellen, dat uw stukken X zijn, en die van de computer O. Stel, er staat ergens op het speelbord een reeks tekens zoals XOOO. Als u nu uw

van de tegenstander zijn dan geheel omsloten door uw stukken, dus de O-en van de computer veranderen in X-en van uzelf.
Het spel is min of meer gebaseerd op de kennis van sterke posities, meer dan op het 'pakken' van zo veel mogelijk vijandige stukken in één zet. De sterkste posities in het spel zijn de vier hoeken van het speelbord, om dat een stuk, dat in een van deze hoeken geplaatst wordt, een lijn aanvalt die zowel horizontaal, vertikaal als diagonaal over het gehele bord loopt, terwijl dit stuk zelf nooit aangevallen kan worden (de tegenstander kan er nooit zijn stukken 'omheen' plaatsen). De posities, die hieraan grenzen zijn daarentegen waarschijnlijk de zwakste posities. Andere sterke posities zijn de velden, die direct grenzen aan de centrale vier, die in het begin van het spel automatisch opgezet worden.

De Spectrum heeft een behoorlijk goed inzicht in de relatieve sterkte van iedere positie op het bord. Nu is het niet de beste strategie om uitsluitend op de sterkte van positie te vertrouwen. Het is echter wel zo, dat kennis van de belangrijkste velden veel belangrijker is dan welke andere overweging ook, en de computer kan dan ook snel beslissen welke van de mogelijke zetten hij als volgende zal kiezen. Daarom vertrouwt dit programma volledig op de kennis van sterke posities, en geeft daardoor een snel en (meestal) intelligent antwoord op iedere zet die u doet. Op deze wijze wordt een snel en uitdagend spel gecreëerd.
Regel 30 stuurt de besturing van het programma naar regel 4000, waar de naam van het programma wordt afgedrukt, waarna het spel zelf gaat beginnen. De BORDER wordt op 5 ingesteld (regel 4000), de PAPER op 2 en de INK op 9 (een INK-kleur van 9 wordt automatisch

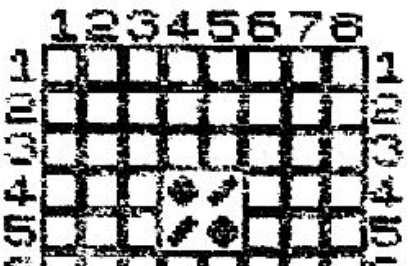
muzikale begeleiding. Daarna ziet u de boodschap 'Even geduld..', knipperend aan de bovenzijde van het scherm, terwijl de arrays gedimensioneerd worden en de variabelen geïnitialiseerd.
De initialisatie-routine begint op regel 5000. Er worden drie arrays geDIMensioneerd. Het array A (regel 5010) bevat de positie op het bord, gesorteerd in volgorde van relatieve sterkte. Het array D wordt gebruikt tijdens het spel, om de lokaties van de stukken op te slaan, die door de ene speler in de soort van de andere speler omgezet worden.
De loops van de regels 5030 tot 5070 vullen het A-array met het teken met code 146 (dit is de CODE van een user-defined vierkantje). Deze CODE wordt gebruikt om het bord af te drukken, waarbij met de vierkantjes de lege posities aangegeven worden. De volgende loop (regel 5080 tot 5110) leest de informatie van de DATA-opdrachten goed bekijkt, dan zult u zien, dat hierin de getallen van 12 tot 89 in een schijnbaar willekeurige volgorde zijn opgeslagen. Deze volgorde is echter, in tegenstelling tot wat het lijkt, niet willekeurig. In de volgorde wordt de relatieve sterkte van ieder veld volgens de programmeur weergegeven. Bijvoorbeeld veld 19 is volgens de programmeur een van de sterkste velden, dus de computer zal altijd kijken, of hij dit veld met een geldige zet kan innemen, voordat hij een andere zet overweegt. De minst waardevolle velden zijn de velden in de buurt van het einde van de DATA-opdracht op regel 5330, dit zijn de velden 28, 78 en 23.
De volgende loop (van regel 5120 tot 5150) vult het array D met de getallen in de DATA-opdracht op regel 5340 (1, 9, 10, 11, – 1, –9, –10, –11). Deze getallen geven de 'afstand' aan van een bepaalde positie op het bord, en worden gebruikt als de computer stukken bekijkt om te

wordt om aan te geven, welke speler er aan de beurt is op 145 gezet in regel 5180, hiermee wordt ingesteld, dat de computer als eerste begint.
Als dit is gebeurd, gaat de Spectrum aan de slag bij het definiëren van de user-defined grafische tekens. Er worden drie verschillende tekens gebruikt: een vierkant voor een lege positie, een ruit (afgedrukt in rood) voor een stuk van de computer, en een diagonale lijn (afgedrukt in wit) voor een van uw eigen stukken. De DATA voor de grafische tekens staat in de regels 5350 tot 5370, met de 'Z' aan het einde, die aangeeft, dat alle data gelezen zijn. Daarna wordt de BORDER kleur ingesteld, het scherm schoongemaakt, en het programma springt terug naar regel 400 vanaf het einde van regel 5270.
Nadat het bord afgedrukt is (regel 400 tot 490) wordt de zet-wijzer gecontroleerd. Als deze de waarde 145 heeft, dan is de computer aan zet. Als dit niet het geval is, dan verandert de computer de waarde van variabele Y in de waarde 144. Op regel 530 wordt een zet van de speler gevraagd, deze zet dient ingetikt te worden als één enkel getal van twee cijfers. Zetten boven de 88 of onder de 11 worden niet geaccepteerd.
Als u ziet, dat u geen geldige zet kunt doen, dan kunt u dit aangeven door een 0 in te tikken. De computer gaat dan naar regel 100 om zijn zet te doen. (Als de computer daar ziet, dat hij ook niet kan zetten, dan bekijkt hij in regel 380, of u hiervoor ook heeft aangegeven, dat u niet kunt zetten, en als dit het geval is, gaat het programma verder op regel 1000, waar het spel beëindigd wordt. Vanaf regel 1000 worden de stukken geteld, het programma maakt de winnaar bekend, en het spel is afgelopen.) Regel 560 zorgt er voor, dat de zet binnen de numerieke grenzen is (11 tot 88) en dat het vakje,

afstanden in array E gebruikt worden, om te bekijken, welke stukken van de tegenstander er veranderd moeten worden. Aan het einde van deze routine, die van regel 160 tot 320 loopt, wordt de zet-wijzer opnieuw bekeken. Als deze gelijk is aan 144, of als de 'print-bord-flag' (J, ingesteld in regel 160 en op regel 290 gewijzigd) gelijk is aan 1, dan gaat het programma verder op regel 400, waar het bord opnieuw geprint wordt. Als nog niet alle velden bekeken zijn (K in regel 340 is ongelijk aan 60) dan gaat de computer naar regel 130, waar het zoeken verder gaat. Als speler net een zet heeft gedaan, dan wordt deze regel natuurlijk nooit uitgevoerd. Als regel 350 bereikt wordt, dan geeft de computer de melding 'ik kan niet'.

Regel 500 bekijkt de waarde van Y. Als deze 144 is, dan weet de computer, dat hij aan zet is. Daarna gaat het programma dan terug naar regel 100, waar hij naar een zet zoekt (regel 100 tot 150, plus 340). Merk op, dat de computer hier gebruik maakt van de informatie, die afkomstig is uit de DATA-opdrachten, waarin de volgorde staat, waarin de velden bekeken moeten worden.

U zult zien, dat PIRANDELLO een uitdagend spel is en de Spectrum een waardige tegenstander. Als u het spel onder de knie heeft, dan kunt u zelf de user-defined grafische tekens aanpassen, of u kunt kijken, of het veranderen van de volgorde van de getallen in de DATA-opdrachten tot een verbeterde strategie kunnen leiden.



```
  10 REM Piradello
  30 GO TO 4000
 100 LET Y=145
 110 LET Z=144
 120 LET K=0
 130 LET K=K+1
 140 LET B=E(K)
 150 IF A(B)<>146 THEN GO TO 340
 160 LET J=0
 170 FOR X=1 TO 8
 180 LET N=D(X)
 190 LET E=0
 200 LET F=B
 210 IF A(F+N)<>Y THEN GO TO 250
 220 LET E=1
 230 LET F=F+N
 240 GO TO 210
 250 IF A(F+N)<>Z THEN GO TO 310
 260 IF E=0 THEN GO TO 310
 270 FOR A=B TO F STEP N
 280 LET A(A)=Z
 290 LET J=1
 300 NEXT A
 310 NEXT X
 320 IF Y=144 THEN GO TO 400
 330 IF J=1 THEN GO TO 400
 340 IF K<>60 THEN GO TO 130
 350 IF G<>0 THEN PRINT AT 0,0;
FLASH 1; INK 2; PAPER 7;"Ik kan
niet..": BEEP 1,1: BEEP 1,2: BEE
P 1,1: PRINT AT 0,0;"
     "
 380 IF G=0 THEN GO TO 1000
 400 PRINT AT 6,11; BRIGHT 1;"12
345678"
 410 BEEP 0.05,32: BEEP 0.05,12:
 BEEP 0.05,32
 420 FOR A=1 TO 8
 430 PRINT TAB 10; BRIGHT 1; INK
 6;A;
 440 FOR B=2 TO 9
 445 LET W=A(A*10+B): BEEP 0.002
,(W=144)+50*(W=145)-60*(W=146)
 450 LET R=2*(W=144)+7*(W=145)+4
*(W=146)
 455 PRINT BRIGHT 1; INK R;CHR$
(W);
 460 NEXT B
 470 PRINT BRIGHT 1; INK 6;A
 480 NEXT A
 490 PRINT TAB 11; INK 6; BRIGHT
```

```
  530 INPUT INK 2; PAPER 6; FLASH
  1; BRIGHT 1;"  Toets uw zet in
  ";G
  540 IF G=0 THEN GO TO 100
  550 LET B=G+1
  560 IF B<12 OR B>89 OR A(B)<>14
6 THEN GO TO 530
  590 GO TO 160
1000 LET C=0
1010 LET H=0
1020 FOR A=1 TO 100
1030 IF A(A)=144 THEN LET C=C+1
1040 IF A(A)=145 THEN LET H=H+1
1050 NEXT A
1055 PRINT AT 18,0; FLASH 1; BRI
GHT 1; INK 2; PAPER 7;"  Uiteind
elijke score:","Spectrum> ";C,,"
Mens> ";H,,
1060 IF C>H THEN PRINT AT 0,0; B
RIGHT 1; FLASH 1; INK 4;"IK win,
..bedankt voor het spelen": GO T
O 1060
1070 IF C<H THEN PRINT AT 0,0; B
RIGHT 1; FLASH 1; INK 4;"U wint,
..bedankt voor het spelen": GO T
O 1070
1080 IF C=H THEN PRINT AT 0,0; B
RIGHT 1; FLASH 1; INK 4;"Het is
gelijk spel...bedankt voor het s
pelen": GO TO 1080
1090 STOP
4000 BORDER 5: PAPER 2: INK 9: C
LS
4010 FOR A=1 TO 22: PRINT TAB A;
  INK A/5+3;"Pirandello"
4020 BEEP 0.008,A: BEEP 0.008,A*
2: NEXT A
4030 PRINT AT 5,5; FLASH 1; INK
7; PAPER 0;" Even geduld...  "
5000 PAPER 0: INK 9
5010 DIM A(100): DIM E(60)
5020 DIM D(8)
5030 FOR A=1 TO 8
5040 FOR B=2 TO 9
5050 LET A(A*10+B)=146
5060 NEXT B
5070 NEXT A
5080 FOR A=1 TO 60
5090 READ B
5100 LET E(A)=B
5110 NEXT A
```

```
5160 LET A(45)=144: LET A(46)=14
5
5170 LET A(56)=144: LET A(55)=14
5
5180 LET Y=144
5270 READ A$: IF A$="Z" THEN BOR
DER 0: CLS : GO TO 400
5280 FOR C=0 TO 7
5290 READ B: POKE USR A$+C,B
5300 NEXT C
5310 GO TO 5270
5320 DATA 19,81,82,12,62,17,32,6
7,69,14,39,84,64,67,37,34,49,15,
59,85,16,52,42,86,65,54,66,44,35
,57,36,47
5330 DATA 63,48,58,76,24,27,38,2
9,74,77,43,26,66,33,53,75,72,13,
29,18,86,22,83,79,73,28,78,23
5340 DATA 1,9,10,11,-1,-9,-10,-1
1
5350 DATA "A",0,24,60,126,126,60
,24,0
5360 DATA "B",0,4,14,28,56,112,3
2,0
5370 DATA "C",255,129,129,129,12
9,129,129,255,"Z"
```

Dammen

Het volgende, uitdagende spel is dammen. Dit is één van 's werelds meest geliefde bordspelen, en alleen schaken is nog populairder. Sommige historici denken, dat dammen zelfs een voorloper is van het schaakspel. Het is zeker, dat dam-achtige spelen gespeeld werden in het oude Egypte, in Griekenland en Rome. Het spel is door de eeuwen heen aangepast en geëvolueerd, en heeft haar huidige vorm in de tweede helft van de zeventiende eeuw gekregen. Er bestaan dan ook vele regionale variaties maar het door onze computer gespeelde spel ligt erg dicht bij het standaardspel. In het standaard damspel wordt er tussen twee tegenstanders gespeeld op een bord van 8 bij 8 vakken, meestal op de zwarte velden. Om het eenvoudiger op het scherm leesbaar te

Iedere speler begint met 12 stukken. In deze versie heeft u de kleine x-en als stukken, de computer speelt met de kleine o-en. Er is echter geen enkele reden, waarom u het programma naderhand niet zou veranderen, om eigen gedefinieerde tekens te gebruiken. De computer begint bovenaan het scherm, en speelt naar beneden, terwijl u onderaan het scherm begint.
U kunt uitsluitend diagonaal bewegen en dus op de witte velden spelen. Als er een vijandelijk stuk op een diagonaal veld direct voor u staat, en er een leeg veld op deze diagonaal is, dan kunt u het vijandelijke stuk slaan, door over het vijandelijke stuk te springen (het stuk wordt dan van het veld verwijderd) en te landen op het lege veld direct daar achter. U kunt meer dan één stuk in één zet slaan, als er vanaf het veld, waar u geland bent, nog een stuk te slaan is.
In de meeste spelregels over dammen, is het slaan van een steen van de tegenstander verplicht. In deze versie van dammen wordt dit niet gecontroleerd. De computer slaat echter wel altijd, dus wees eerlijk, en sla wanneer dit mogelijk is.
Het doel van het spel is, alle stenen van de tegenstander van het bord te verwijderen, of het voor de tegenstander onmogelijk te maken om nog een geldige zet te doen. In dit programma zal de Spectrum trachten tot het einde toe door te spelen, maar er is een mogelijkheid ingebouwd, waardoor de Spectrum kan zien, dat hij verslagen is. Als hij dit opmerkt, zal hij het spel beëindigen. U kunt op ieder punt in het spel opgeven, door 99 in te tikken. Als u dit doet, dan zal de Spectrum met veel medeleven accepteren, dat hij gewonnen heeft.
Als u een steen geheel aan de overkant weet te krijgen, dan wordt het stuk gepromoveerd tot een DAM. Dit wordt in het programma aangegeven, door deze stuk-

een veel grotere mobiliteit dan een normaal stuk, omdat hij zowel vooruit als achteruit kan bewegen. Met een dam kunt u dus het spel sterk in uw voordeel laten lopen. Merk op dat, wanneer u op de laatste rij van het spel terecht komt uw steen niet direct als dam gebruikt mag worden om zo verder te slaan. De dam geldt pas in de volgende zet. De computer weet dit, en laat dit ook zien, door een steen pas te veranderen na de volgende zet.
U zult zien, dat de computer dit spel behoorlijk snel speelt, en met een zekere mate van intelligentie en inzicht. Dit betekent, dat het spel u een redelijke tegenstander biedt. Om het spel nog aantrekkelijker te maken, print de computer iedere zet, die hij in overweging neemt, op het scherm, en laat een BEEP horen, als hij een zet gevonden heeft, die hij als redelijk aanmerkt. De computer zoekt het gehele bord af, veld voor veld, beginnend in de linker bovenhoek, dan van links naar rechts over iedere rij, en zo tot aan de rechter hoek. Als hij een eigen steen vindt, dan kijkt hij, of hij daarmee kan slaan. Als hij een zet vindt, waarmee een steen van een tegenstander geslagen kan worden, dan doet hij deze zet, zonder verder te kijken. Dan print hij het nieuwe bord en geeft u de mogelijkheid om uw antwoord in te toetsen. Terwijl de Spectrum op zoek is naar een zet, bekijkt hij ook, of hij een 'veilige' zet kan vinden. Hiervoor gebruikt hij een eenvoudige techniek, die hem in staat stelt, een potentiële zet uit te kiezen, die het verplaatste stuk niet in gevaar brengt. Daarnaast is er nog een faciliteit, die kijkt, of een stuk, dat aangevallen wordt, beschermd kan worden.
Nadat het hele bord op deze wijze onder de loep genomen is, en elke, mogelijke zet in een array is opgeslagen, wordt er gekeken, of een van de stukken in een enkele zet tot dam gepromoveerd kan worden. Als zo'n zet

te beschermen. Als een steen niet tot dam gepromoveerd kan worden, dan kiest de computer een willekeurige zet uit het array van veilige zetten. Door een willekeurige zet te kiezen uit een array, waarin zetten zijn opgeslagen, die de computer allen gelijkwaardig acht, kan de computer een reeks spelen afwerken, zonder in herhaling te vervallen, zelfs als u dit tracht te forceren door een zelfde spel twee maal te spelen.
Als het spelbord geheel afgetast is, en er geen veilige zetten zijn gevonden, dan kiest de Spectrum een willekeurig stuk, en verplaatst dit. Deze zetten zullen slechts zeer zelden voorkomen, in het begin of het midden van het spel. Als er nog maar een klein aantal stukken op het bord over zijn, dan is het mogeijk, dat de Spectrum zijn positie hopeloos vindt. Hij geeft dan op, en laat de overwinning aan u.
Zoals reeds vermeld, speelt dit programma redelijk snel, omdat voor de meeste zetten slechts een klein gedeelte van het spelbord afgezocht hoeft te worden, of – in het slechtste geval – het hele bord slechts één maal. De snelheid van de respons wordt vertraagd door de regels, die de zetten afdrukken, die de computer in overweging heeft. Deze regels voegen echter zoveel spanning toe, dat ze de moeite waard zijn. Als u een spel verkiest, dat op maximale snelheid speelt, dan kunt u bepaalde regels verwijderen, die later aangegeven zullen worden. De snelheid wordt dan werkelijk bijzonder hoog. Er zijn ook een aantal commentaarregels, zoals 'Goede zet' of 'Ik heb er al een!', die van tijd tot tijd opduiken. Deze zinnen vertragen het programma ook enigszins, en kunnen daarom ook verwijderd worden, indien uw voorkeur uitgaat naar een zeer snelle speler. Zoals u zult zien ,vergaat het u waarschijnlijk net zoals vele anderen, die reeds met dit programma gespeeld

zetten, en één supersnelle versie. Daarna kunnen ze zelf kiezen, welke versie ze willen spelen.
Na de random-getallen-generator op regel 50 te hebben ingesteld, gaat het programma verder naar regel 1070, waar een subroutine begint, die de variabelen initialiseert. Als u deze sectie van het programma nader bestudeert, dan ziet u dat de PAPER en BORDER op blauw gezet worden, en de INK op geel, voordat het scherm schoongemaakt wordt. Ook hier geldt, dat u deze kleuren het beste zelf kunt kiezen. De woorden 'Even geduld...' verschijnen aan de bovenzijde van het scherm gedurende een betrekkelijk kort initialisatie-proces. (U kunt in overweging nemen, om in ieder programma, dat een wat langere initialisatie heeft, zo'n regel in te voeren.)
Regel 1080 geeft de waarden van de stukken aan. de namen van de variabelen zijn als volgt: C is een stuk van de computer, H een stuk van de Humanoide (u dus), CK is een dam van de computer, HK een dam van uzelf, E is een leeg veld en B een zwart veld. Deze waarden worden in deze regel aan de variabelen toegewezen, en in het hele programma wordt daarna aan deze namen gerefereerd. Dit is zo gedaan, opdat u zelf de grafische tekens van de verschillende stukken kunt wijzigen, door alleen deze regel aan te passen. Als u bijvoorbeeld, gebruiker-gedefinieerde tekens zou willen gebruiken, dan hoeft u alleen 144 te gebruiken voor de H (144 is de CODE van user-defined grafisch teken 'A'), daarna 145 voor C, etc.
Regel 1090 zet de variabele OF om de situatie 'van het bord af' weer te geven. De computer heeft in zijn gedachten namelijk een onzichtbare rij van dummy-velden rondom het bord, zodat hij weet waar de rand van het bord is, en dus niet probeert om een slag te doen, waar

het array wordt gevuld met de dummy-waarde. De volgende loop (regel 1110 tot 1130) gebruikt de DATA in regels 1140 tot 1230 om de velden op het bord te vullen met de juiste waarden. Alle elementen van het Q-array, die hier geen waarde krijgen, behouden hun oude waarde, –99, zodat de computer weet, dat dit velden zijn, die buiten het speelbord liggen, en dus niet gebruikt mogen worden.
Regel 1240 is een loop, die het N-array (gedimensioneerd in regel 1090) vult met getallen, die de mogelijke zetten aangeven van ieder veld naar een ander veld zonder te slaan. De computer weet, dat een slag-zet verkregen kan worden, door de inhoud van deze velden te verdubbelen. In regel 1260 worden de variabelen, die het aantal geslagen stukken voor beide partijen aangeven, op nul gezet, CO voor de computer, op HU voor de speler.
Nadat we terugkeren op regel 90 (vanaf regel 1260), zien we, in de REM-opdracht de tekst: 'Verwijder de volgende regel om de Spectrum niet de eerste zet te geven'. Als u zelf een openingszet wilt doen, dan verwijdert u eenvoudig regel 100. U hoeft geen enkel ander gedeelte van het programma te wijzigen. De Spectrum is niet de sterkste dammer in de wereld, dus het minste wat u kunt doen, is hem de eerste zet laten houden.
Het programma roept in haar loop van regels 110 tot 1140 een aantal subroutines aan, regel 150 laat het geheel weer opnieuw beginnen. De verschillende subroutines voeren de volgende vitale gedeelten van het programma uit:
680 print het bord uit
850 accepteert een zet van de speler
680 print het bord opnieuw uit na een zet van de speler
160 start het zoeken van een zet door de computer

belangrijke en interessante gedeelte van het programma is.
Regel 160 dimensioneert het array S, dat gebruikt wordt om veilige zetten voor de computer op te slaan, terwijl hij op zoek is naar een zet om een steen van de tegenstander te slaan. SC, de 'veilige-zetten-teller', wordt in regel 170 op nul gezet, waarbij A (de variabele, die gebruikt wordt om aan te geven, welke zet er in overweging genomen wordt) op 89 ingesteld wordt, één meer dan het element 88, dat de linker bovenhoek van het speelveld aangeeft. Regel 180 verlaagt A met één, om zo het linker bovenveld aan te geven, en regel 190 kijkt, of dit veld een normale steen van de computer bevat (dus het programma kijkt, of de waarde van het stuk op het veld gelijk is aan C, de waarde van een steen van de computer), of een dam van de computer (waarde CK). Regel 200 zet B (die aangeeft, welke potentiële zet er vanaf dit veld overwogen wordt) op nul. Als A kleiner of gelijk is aan 29, dan weet hij, dat hij op de onderste regel aan de kant van zijn tegenstander is aangekomen, en dus alleen nog maar omhoog kan spelen (waarden 3 en 4 van B geven een zet naar beneden aan), dus het programma bekommert zich daarna niet meer om zetten als B is 1 of B is 2. Regel 210 telt 1 op bij B (die dan 1 wordt, of 3 als een stuk op de onderste regel wordt overwogen), en regel 220 zet M (die door deze hele procedure heen het veld aangeeft, waar de computer naar toe zou willen zetten, terwijl A het veld aangeeft, waar de computer van vandaan zou willen zetten) op A plus het element van het array N, dat aangewezen wordt door de waarde B. Als M groter is dan 88 of kleiner is dan 11, dan weet de computer, dat het stuk van het bord af staat, en hoeft hij dus niet verder te kijken. Regel 240 is één van de regels, die weggelaten kunnen wor-

steen van de tegenstander te slaan. Als het veld, een steen van de mens bevat (H) of een dam van zijn tegenstander (HK), en het veld daarachter is leeg (E), dan weet de computer, dat hij een steen kan slaan, en dan wordt de besturing direct doorgegeven aan regel 320, waar de tekst 'Hebbes!' in knipperende letters op het scherm wordt afgedrukt. Als u een programma wilt, dat het spel op maximale snelheid afwikkelt, dan kunt u daartoe de regels 330, 340 en 350 verwijderen, deze regels zijn alleen in het spel gezet, zodat de computer kan opscheppen over zijn overwinningen.
Teruggekomen in het voornaamste gedeelte van het programma, voert het volgende gedeelte van de computer-zet-routine een zoekopdracht uit naar een veilige zet. Hiertoe bekijkt hij de velden, die liggen direct ná het veld, waar hij een stuk naar toe wil verschuiven. Als hij door het doen van deze zet zijn eigen stuk of een van zijn andere stukken niet in gevaar brengt, dan wordt dit bevestigd door het programma. Het programma voert namelijk een uitgebreid aantal IF-opdrachten uit, die er allen op gericht zijn, een veilige zet te vinden. Als een zet al deze IF-opdrachten op deze regel doorkomt, dan gaat de Spectrum naar een routine op regel 470. De computer kan maximaal 10 veilige zetten opslaan (omdat het array, dat deze zetten opslaat, maximaal 10 elementen kan bevatten, en in de praktijk is gebleken, dat dit vrijwel altijd voldoende is). De teller van de veilige zetten SC, die op regel 170 de waarde nul heeft gekregen, wordt nu met één verhoogd, zolang deze teller ten minste kleiner is dan 10. Regel 480, een andere regel, die u voor maximale snelheid kunt verwijderen, print de tekst 'Veilige zet ... naar ...', en laat een BEEP horen, om de veilige zet er uit te doen springen. Regel 490 is nogal interessant. Deze regel converteert zowel het

decodeerd kan worden om de zet uit te voeren. Regel 490 keert terug naar de hoofdlijn van het programma, maar terwijl we op dit punt zijn, kunnen we even kijken, hoe de veilige zet gedaan wordt. Als er geen vijandelijke steen geslagen is, dan gaat de computer naar regel 500. Als hij ziet, dat SC gelijk is aan nul, dan weet hij, dat er geen veilige zet gedaan kan worden en gaat naar regel 550 om een willekeurige zet te kiezen. Als SC groter is dan nul, dan 'valt' het programma door de volgende regels en wordt er een van de gecodeerde getallen uit het SC-array gekozen, deze zet wordt dan opgeslagen in de variabele XC in regel 510. Het 'van'-veld wordt uit het gecodeerde getal gehaald in regel 520, en het 'naar'-veld in regel 530. Regel 540 stuurt het programma verder naar regel 650, waar de zet gedaan wordt.
Als we terugkeren naar het eerste segment van het programma, dan komen we op regel 270. Hier wordt B verhoogd. De elementen 1 en 2 van het array N bevatten zetten die gemaakt kunnen worden door iedere steen op het bord (behalve door de stenen op de onderste rijen) en de elementen 3 en 4 bevatten uitsluitend damzetten. Als B, de teller die de elementen uit het array N selecteert die voor de zet gebruikt worden, kleiner is dan zijn maximum (2 voor een gewone steen, 4 voor een dam), dan keert het programma terug naar regel 210, waar B verhoogd wordt. Het programma gaat nu opnieuw zoeken naar een zet. Regel 280 kijkt of ieder veld van het bord gecontroleerd is. Zo niet, dan wordt er teruggekeerd naar regel 180, waar de veldteller A met één verlaagd wordt, waarna het zoeken verder gaat. Als echter alle velden getest zijn (dus A is niet groter dan 11, zoals in regel 280 getest wordt), dan gaat het voorprogramma verder naar de volgende twee regels, waar een potentiële dam gezocht wordt. Een flag FL,

maken. Als de computer hier een steen van zichzelf vindt, dan stuurt hij het programma naar een subroutine vanaf regel 1270.

Als deze subroutine een zet vindt, die een dam kan maken, dan wordt in de subroutine de flag FL op één gezet, hetgeen de besturing van het programma verder stuurt naar regel 650, waar de werkelijke zet gedaan wordt.
Als zo'n zet niet mogelijk is, dan springt regel 310 over het gedeelte heen, dat een steen van de tegenstander slaat (waar we straks nog even naar kijken), en gaat naar regel 500. Houd nu in gedachte, dat de computer de volgende strikte hiërarchie van mogelijkheden aanhoudt:

- Kan ik slaan, en indien ja, kan ik verschillende malen slaan?
- Kan ik een dam halen?
- Kan ik een zet doen, die het verplaatste stuk of elk ander stuk niet in gevaar brengt?
- Kan ik een legale zet doen?

Als het antwoord op één van deze vragen 'ja' is, dan wordt de zet gedaan, en wordt het nieuwe speelbord afgedrukt, klaar voor de zet van de tegenstander. Als het antwoord op de eerste vraag 'nee' is, dan wordt de tweede vraag gesteld, enzovoorts. Als het antwoord op de laatste vraag 'nee' is, dan beëindigt de computer het spel. (Er is nog een andere vraag, die iedere keer gesteld wordt, als het bord opnieuw afgedrukt wordt. Deze vraag luidt: 'Heeft een van de spelers 12 stenen meer dan de andere speler?'. Als het antwoord op deze vraag 'ja' luidt, dan gaat het programma naar een rou-

de volledigheid vermeld, zodat u eenvoudiger kunt begrijpen, hoe de computer tot een zet komt). Deze wijze van ordenen van zetten is de meest voorkomende in intelligente spelprogramma's, vanaf de meest eenvoudige videospelletjes zoals Space Invaders, tot de meest gecompliceerde, zoals Schaken.
Als de computer regel 500 heeft bereikt, dan was het antwoord op de eerste twee vragen 'nee'. Om de derde vraag te kunnen stellen ('Kan ik een veilige zet doen?') hoeft de computer alleen maar naar de teller SC te kijken, die het aantal veilige zetten bevat. Als deze teller op nul staat, zijn er geen zetten gevonden, die het etiket 'veilig' verdienen, dus het antwoord op deze vraag luidt dan ook 'nee'. We hebben al gezien, hoe het 'veiligezet'-systeem werkt, dus nu kijken we naar het gedeelte van het programma, waar de computer zijn toevlucht zoekt tot zijn laatste reddingsmogelijkheid: het doen van een willekeurige zet, zoals dit gebeurt vanaf regel 550.
Hier wordt de variabele SC opnieuw gebruikt, uitsluitend om te voorkomen, dat we een andere variabele moeten initialiseren. Deze variabele moet namelijk altijd nul zijn, anders zou deze routine niet uitgevoerd worden (zie ook regel 170). Regel 550 telt 1 op bij de waarde van SC en kiest een willekeurig veld tussen 1 en 88. Als dit veld geen steen van de computer of een dam van de computer bevat, dan gaat het programma naar regel 630, waar zolang er minder dan 300 veldnummers geprobeerd zijn, de besturing teruggezonden wordt naar regel 550 om het opnieuw te proberen.
Als echter het gevonden veld een C of een CK bevat, dan kijkt de routine van regel 570 tot 620 of dit stuk bewogen kan worden. Als een zet gevonden is, dan stuurt regel 610 het programma naar regel 650, waar

gevonden is, dan beëindigt de Spectrum het programma via regel 640. Voordat we kijken naar het gedeelte van het programma, dat de menselijke tegenstander een zet laat doen, gaan we nog een klein stukje terug in het programma, naar het gedeelte dat begint op regel 370. Deze regel wordt uitgevoerd nadat een stuk van de tegenstander is geslagen. De voorgaande regel heeft de score van de computer (CO) met 1 verhoogd voordat de regel 370 de bord-print-routine aanroept, die het verplaatste stuk laat zien, de nieuwe score van de spectrum, en – natuurlijk – een leeg veld waar het Spectrumstuk eerst stond. U zult zich herinneren, dat A het veld was waar de computer vandaan kwam, en M het veld waar hij naartoe ging. A wordt nu gelijk gemaakt aan het veld, waar de computer naartoe gesprongen is, en de regels van 390 tot 460 kijken, of er nog een sprong gemaakt kan worden, zodat er verschillende stenen in één zet geslagen kunnen worden, dan voert het programma ze uit, en het bord wordt opnieuw geprint. Hierna wordt opnieuw bekeken of er nogmaals geslagen kan worden. Als de hele reeks van slagen gedaan is, dan geeft de Spectrum de beurt over aan de menselijke tegenstander, die dan zijn of haar zet kan doen.
Het gedeelte van het programma, dat de menselijke zetten behandelt, is eenvoudig. U doet een zet, door het veld aan te geven, waar vandaan u een steen wilt verplaatsen, door het nummer van het veld op de zijde aan te geven, gevolgd door het nummer van het veld in de breedte. Tik beide getallen aan elkaar in (zoals '31'). Daarna drukt u de ENTER-toets in, gevolgd door de twee getallen, die het veld aangeven, waar u naartoe wilt bewegen. De zet wordt dan gedaan, het bord wordt opnieuw afgedrukt, en als u een steen geslagen heeft, wordt u gevraagd, of u nogmaals kunt slaan. Als u 'nee'

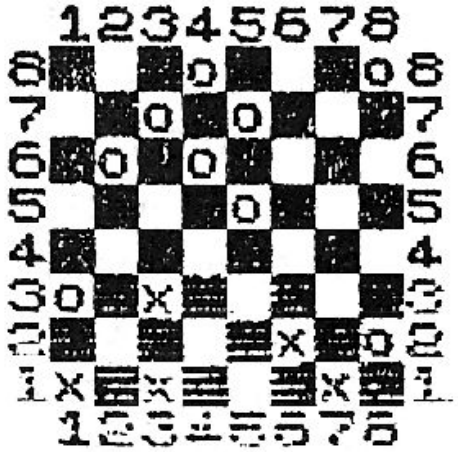
gen (omdat de computer het veld, waar u vandaan komt natuurlijk al weet, daar bent u immers zojuist naar toe gesprongen).
Het gedeelte van het programma, dat de zet van de menselijke tegenstander accepteert, begint op regel 810, waar een REM-opdracht aangeeft, dat u het spel op dit punt kunt beëindigen, door 99 in te toetsen. Als u een '1' intoetst, dan geeft u daarmee te kennen, dat u het scherm naar de printer wil laten kopiëren. Na deze print-out van het scherm te hebben gemaakt, wordt u opnieuw om uw zet gevraagd. Nadat de zet uitgevoerd is in het array, dat de velden van het bord bevat (in regel 910) worden er twee loops gebruikt om uit te vinden of één van de stenen van de speler tot dam gepromoveerd moet worden (regels 920 tot 950). Als dit het geval is, dan wordt het symbool van dit stuk overeenkomstig aangepast. U zult zich herinneren, dat een steen niet als een dam mag bewegen in het spel, totdat de tegenstander een zet heeft gedaan. Op dit moment worden dan ook alle stukken van de computer, die tot dam gepromoveerd zijn in de laatste zet van de Spectrum in een dam veranderd. We doen dit hier, in het midden van het gedeelte van het programma, dat de menselijke zet behandeld, zodat we er dan zeker van kunnen zijn, dat de computer op geen enkele manier vals kan spelen met zijn nieuw-gecreëerde koning. Wij nemen trouwens aan, dat hoewel het stuk van de menselijke tegenstander onmiddellijk in een dam veranderd wordt, de tegenstander niet vals zal spelen door het stuk direct daarna te verplaatsen. Er is in het programma geen enkele voorziening getroffen om te voorkomen, dat u vals speelt, maar het hele spelletje stelt natuurlijk niets voor als u dit wel doet, omdat de Spectrum er nooit van zou dromen om vals te spelen.

verschil niet kleiner is dan 12, dan heeft u een steen geslagen, en de computer zal dit (twee van de drie keer) bevestigen met een commentaar, dat gegenereerd wordt in regel 970 tot 990. Na de eventuele opmerking wordt het slaan van een steen altijd begeleid door een stijgende toon, die gegenereerd wordt in regel 995. Daarna wordt de score van de tegenstander (u dus) met 1 verhoogd (variabele HU, in regel 1000), het geslagen stuk wordt van het veld verwijderd (door de tweede opdracht in regel 1000) en het bord wordt opnieuw geprint, voordat (in regel 1030) de tegenstander de kans krijgt om nog een steen te slaan. Als hij dit niet doet, of geen steen geslagen heeft, dan gaat het programma terug naar de hoofdloop. Als hij echter wel verder wil slaan, dan wordt in regel 1060 het oude 'naar'-veld gevuld met het nieuwe 'van'-veld, en stuurt het programma naar regel 860, waar een nieuwe 'naar' opgevraagd wordt.
We hebben al naar de rest van het programma gekeken, dat de initialisatie bevat, en de routine 'Kan ik een dam halen?'. Het enige gedeelte van regel 680 tot 780, dat het bord uitprint. Dit gedeelte van het programma legt zichzelf uit. Merk regel 770 op, die de return naar het hoofdprogramma overslaat, als de mens (HU) of de computer (CO) teller gelijk is aan twaalf.
Dit is het bord aan het begin van het spel, en een korte tijd later in het spel, waarbij de Spectrum van boven naar beneden speelt.

```
Spectrum:  1     Mens:  0

   1 2 3 4 5 6 7 8
 8 ■ o ■ o ■ o ■ o 8
 7 o ■ o ■ o ■ o ■ 7
 6 ■   ■ o ■ o ■ o 6
 5   ■   ■   ■   ■ 5
```

Spectrum: 6 Mens: 2



```
  10 REM Spectrum Checkers
  50 RANDOMIZE
  80 GO SUB 1070
  90 REM Verwijder de volgende r
egel om zelf de eerste zet te do
en
 100 GO TO 130
 110 GO SUB 680
 120 GO SUB 850
 130 GO SUB 680
 140 GO SUB 160
 150 GO TO 110
 160 DIM S(10)
 170 LET SC=0: LET A=89
 180 LET A=A-1
 190 IF Q(A)<>C AND Q(A)<>CK THE
N GO TO 280
 200 LET B=0: IF A<29 THEN LET B
=2
 210 LET B=B+1
 220 LET M=A+N(B)
 230 IF M>68 OR M<11 THEN GO TO
280
 240 IF 10*INT (M/10)<>M THEN PR
INT AT 0,0;A;" naar ";M;"?"
 250 IF (Q(M)=H OR Q(M)=HK) AND
Q(M+N(B))=E THEN GO TO 320
 260 IF Q(M)=E THEN IF (Q(M-11)<
>H AND Q(M-11)<>HK) THEN IF (Q(M
-9)<>H AND Q(M-9)<>HK) AND Q(M+9
)<>HK THEN IF ((Q(M+22)<>HK OR Q
(M+10)<>HK) AND (Q(M+9)<>C OR Q(M
+9)<>CK OR Q(M+11)=C OR Q(M+11)=
CK)) AND Q(M+11)<>HK) THEN GO SU
B 470
```

```
4)=C OR Q(26)=C OR Q(28)=C THEN
GO SUB 1270
  300 IF FL=1 THEN GO TO 650
  310 GO TO 500
  320 LET Q(M+N(B))=Q(A): LET Q(M
)=E: LET Q(A)=E
  330 PRINT AT 0,0; FLASH 1; INK
0;" **********************
*"
  340 FOR T=-1 TO 55 STEP 3: BEEP
 .008,T: NEXT T
  350 PRINT AT 0,0; INK 2; PAPER
7; FLASH 1; BRIGHT 1;"          Heb
ben je ...."                    ": PAUSE 12
0: PRINT AT 0,0;"
                "
  360 LET CO=CO+1
  370 GO SUB 680
  380 LET A=M+N(B)
  390 LET B=0
  400 LET B=B+1
  410 IF (A+2*N(B)<11 OR A+2*N(B)
>88) AND B<4 THEN GO TO 400
  420 LET M=A+N(B)
  430 IF Q(M)=C AND B>0 THEN RETU
RN
  440 IF (Q(M)=M OR Q(M)=MK) AND
Q(M+N(B))=L THEN GO TO 320
  450 IF B=4 OR (Q(M)=C OR Q(M)=L
) THEN GO TO 400
  460 RETURN
  470 IF SC<10 THEN LET SC=SC+1
  480 PRINT AT 0,0; INVERSE 1;"Ve
rloren  ";O;" naar ";M: BEEP
.5,50: PRINT AT 0,0;"
                "
  490 LET S(SC)=100*A+B+20: RETUR
N
  500 IF SC=0 THEN GO TO 550
  510 LET XC=INT (RND*SC)+1
  520 LET A=INT (S(XC)/100)
  530 LET M=A+N(S(XC)-100*A-20)
  540 GO TO 650
  550 LET SC=SC+1: LET A=INT (RND
*88)+1
  560 IF Q(A)<>C AND Q(A)<>CK THE
N GO TO 630
  570 LET B=1
  580 LET B=B+1
  590 LET M=A+N(B)
  600 IF M>88 OR M<11 THEN GO TO
620
```

```
  640 PRINT AT 0,0;"Ik geef het o
p": STOP
  650 LET Q(M)=Q(A): LET Q(A)=E
  660 PRINT AT 0,0;"Van ";A;" naa
r ";M: FOR T=30 TO -30 STEP -2:
BEEP .008,T: NEXT T: PRINT AT 0,
0;"
  670 RETURN
  680 PRINT AT 4,2; BRIGHT 1; INK
 6;" Spectrum: ";CO;"    Mens: ";
HU;" "
  710 PRINT TAB 6; BRIGHT 1; INK
6;"12345678"
  720 FOR F=60 TO 10 STEP -10
  730 PRINT TAB 5; BRIGHT 1;F/10;
  740 FOR G=1 TO 8: PRINT CHR$ Q(
F+G);: NEXT G
  750 PRINT F/10: NEXT F
  760 PRINT TAB 6; BRIGHT 1; INK
6;"12345678"
  770 IF CO=12 OR HU=12 THEN GO T
O 790
  780 RETURN
  790 IF HU=12 THEN PRINT AT 0,0;
"U heeft gewonnen...bedankt voor
 het spelen": STOP
  800 IF CO=12 THEN PRINT AT 0,0;
"Ik heb gewonnen...bedankt voor
het spelen": STOP
 8010 REM 99 OM OP TE GEVEN
 8020 REM 1 OM HET BORD TE PRINTE
N
 8050 INPUT FLASH 1;" Van? ";A
 8060 IF A=99 THEN PRINT AT 0,0;"
Bedankt voor het spelen": STOP
 8070 IF A=1 THEN COPY : GO TO 85
0
 8080 INPUT FLASH 1;(A);" Naar? "
;
 8090 LET Q(B)=Q(A): LET Q(A)=E
 8100 FOR T=11 TO 7: IF Q(T)=C TH
EN LET Q(T)=CK
 8130 NEXT T
 8140 FOR T=82 TO 88: IF Q(T)=H T
HEN LET Q(T)=HK
 8160 NEXT T
 8170 IF ABS (A-B)<12 THEN RETURN

 8180 LET TY=RND
 8190 IF TY<0.3 THEN PRINT AT 0,0
;"Goed Zet"
 8200 IF TY>0.7 THEN PRINT AT 0,0
```

```
1060 LET HU=HU+1: LET Q(A+B)/2)
=E: GO SUB 680
1010 FOR T=82 TO 88: IF Q(T)=H T
HEN LET Q(T)=HK
1020 NEXT T
1030 INPUT FLASH 1;"Kunt U nogm
als slaan?  (J/N) ";A$
1040 IF A$<>"J" AND A$<>"j" THEN
 RETURN
1060 LET A=B: GO TO 860
1070 PAPER 1: BORDER 1: INK 7: C
LS
1075 PRINT AT 0,0;"Even geduld..
."
1080 LET H=120: LET HK=75: LET C
=111: LET CK=36: LET E=82: LET B
=143
1090 LET OF=-99: DIM Q(99): DIM
N(4)
1100 FOR M=1 TO 99: LET Q(M)=OF:
 NEXT M
1110 FOR M=1 TO 64
1120 READ D: READ G
1130 LET Q(D)=G: NEXT M
1140 DATA 81,B,82,C,83,B,84,C,85
,B,86,C,87,B
1150 DATA 88,C,71,C,72,B,73,C,74
,B,75,C,76,B
1160 DATA 77,C,78,B,61,B,62,C,63
,B,64,C
1170 DATA 65,B,66,C,67,B,68,C,51
,E,52,B
1180 DATA 53,E,54,B,55,E,56,B,57
,E,58,B
1190 DATA 41,B,42,E,43,B,44,E,45
,B,46,E
1200 DATA 47,B,48,E,31,H,32,B,33
,H,34,B,35,H
1210 DATA 36,B,37,H,38,B,21,B,22
,H,23,B,24,H
1220 DATA 25,B,26,H,27,B,28,H,11
,H,12,B,13,H
1230 DATA 14,B,15,H,16,B,17,H,18
,B
1240 FOR M=1 TO 4: READ N(M): NE
XT M
1250 DATA -11,-9,11,9
1260 LET CO=0: LET HU=0: RETURN
1270 IF Q(22)=C AND Q(11)=E THEN
 LET A=22: LET M=11: LET FL=1: R
ETURN
1280 IF Q(22)=C AND Q(13)=E THEN
```

```
 LET A=24: LET M=13: LET FL=1: R
ETURN
1300 IF Q(24)=C AND Q(15)=E THEN
 LET A=24: LET M=15: LET FL=1: R
ETURN
1310 IF Q(26)=C AND Q(15)=E THEN
 LET A=26: LET M=15: LET FL=1: R
ETURN
1320 IF Q(26)=C AND Q(17)=E THEN
 LET A=26: LET M=17: LET FL=1: R
ETURN
1340 RETURN
```

Boter, kaas en eieren

Boter, kaas en eieren is een zeer bekend spelletje, waarbij u en uw tegenstander om beurten een symbool (kruis of cirkel) op de lege posities van een drie-bij-drie rooster plaatsen, waarbij getracht wordt een rij van drie gelijke symbolen te maken. Dit mag zowel horizontaal, verticaal als diagonaal gebeuren. De speler, die de rij maakt, is de winnaar. Het spelletje is al zo vaak geanalyseerd en bestudeerd, dat er zeer zelden verrassingen zijn bij het spelen ervan. In plaats van een voorspelbaar programma, waarbij de speler op z'n best kan gelijkspelen, maar vaak ook verliest, is er door de auteur een programma geschreven dat een onvoorspelbare, maar intelligente partij speelt. Het programma geeft u de mogelijkheid om zelf te beginnen, of de computer te laten beginnen (de beslissing, wie het eerste mag zetten, wordt genomen in regel 50). Behalve het bezetten van het middenveld zijn de zetten van de Spectrum moeilijk te voorspellen (behalve natuurlijk als hij een rij van drie af kan maken, of als hij u moet blokkeren om een rij te maken). Als RND in regel 50 groter is dan 0,5, dan beslist de computer, dat hij als eerste zet, en gaat hij na een pauze naar regel 90, waar het programma doorgestuurd wordt naar regel 700. Hier begint een subroutine, die het bord uitprint. Bij het terugkeren uit deze subroutine gaat de

Deze routine, die alle mogelijke winnende posities nagaat, gaat verder tot aan regel 550.
Nadat uit de 'winnende zet'-routine is teruggekeerd, kijkt de computer naar het centrale vlakje, veld nummer 5. Als hij ziet, dat deze positie leeg is, dan neemt hij deze positie in en gaat hij daarna direct naar regel 60, waar eerst de subroutine aangeroepen wordt, die het bord print. Daarna kan de tegenstander zijn zet intoetsen.
Als deze eenvoudige zet niet gemaakt kan worden, dan geeft de computer de boodschap 'Even geduld...'. Hij zoekt dan naar een zet, die hem direct de overwinning geeft bij de eerste maal, dat de regels 140 tot 280 uitgevoerd worden, en als er geen winnende zet gevonden wordt, dan kijkt hij, of een potentiële winnende zet van de tegenstander geblokkeerd kan worden. Als deze zetten niet gevonden worden, dan kiest hij maximaal 20 willekeurige zetten, omdat hij ergens moet zetten. Als hij geen zet kan vinden, dan gaat hij (van regel 350 tot 380) ieder veld afzonderlijk controleren, om te zien, of alle velden bezet zijn. Als hij inderdaad ziet, dat hij dan nog geen zet kan doen, dan weet hij, dat alle velden gevuld zijn. Hij weet ook, door de uitvoering van de 'winnende-zet'-subroutine, dat u noch hijzelf gewonnen heeft, dat het spel gelijk geëindigd is. Na een aantal korte tonen en een boodschapo (regel 410 en 590 tot 600) wordt het programma dan automatisch herhaald.
De subroutine, die de zet van de tegenstander uitvoert, loopt vanaf regel 620. Hierbij worden uw zetten ingelezen met behulp van INKEY$, dus u hoeft geen ENTER in te toetsen, nadat u uw keuze heeft gemaakt. Merk op hoe het gehele spel binnen een eindeloze loop van regel 60 tot regel 100 wordt uitgevoerd. Hierbij wordt eerst het bord afgedrukt en daarna een zet van de speler gevraagd, het bord weer afgedrukt, de computer aan

het programma als een eindeloze loop en subroutine aanroepen naar de verschillende onderdelen van het programma is, dat nu verschillende delen van het programma eenvoudig en overzichtelijk aangepast kunnen worden, zonder dat daardoor de samenhang binnen het gehele programma verstoord wordt.
Als u de speelsterkte van het programma wilt verbeteren (hoewel dit ten koste gaat van de 'onvoorspelbaarheid' van het programma), dan kunt u het 'aftasten' van de velden in regel 290 e.v. in een vooropgestelde volgorde laten doen.

```
   10 REM BOTER KAAS EN EIEREN
   20 DIM A(9)
   30 BORDER 1: INK 7: PAPER 1: C
LS
   50 IF RND>0.5 THEN PRINT AT 5,
5; FLASH 1;"Ik begin...": FOR E=
0 TO 50: BEEP 0.08,E: NEXT E: CL
S : GO TO 90
   60 GO SUB 700
   70 GO SUB 430
   80 GO SUB 620
   90 GO SUB 700
  100 GO SUB 430
  110 IF A(5)=0 THEN LET A(5)=1:
GO TO 60
  120 REM RIJ/BLOK AFMAKEN
  130 PRINT AT 1,1; FLASH 1;"Ew
edu(d..."
  140 LET D=1
  150 LET E=1
  160 IF B=1 THEN LET X=1: LET Y=
4: LET Z=7
  170 IF B=2 THEN LET X=1: LET Y=
5: LET Z=9
  180 IF B=3 THEN LET X=3: LET Y=
6: LET Z=9
  190 IF B=4 THEN LET X=D: LET Z=
D
  200 LET D=1
  210 IF A(X)=D AND A(Y)=D AND A(
Z)=0 THEN LET A(Z)=1: GO TO 60
  220 IF A(X)=D AND A(Y)=0 AND A(
Z)=D THEN LET A(Y)=1: GO TO 60
```

```
 250 IF B=2 THEN LET X=X+1: LET Y=Y+1: LET Z=Z+1
 260 IF C<3 THEN LET C=C+1: GO TO 210
 270 IF B<4 THEN LET B=B+1: GO TO 170
 280 IF D<2 THEN LET D=D+1: GO TO 150
 290 REM DOE WILLEKEURIGE ZET
 300 LET B=1
 310 LET C=INT (RND*9)+1
 320 IF A(C)=0 THEN LET A(C)=1: GO TO 60
 330 LET B=B+1
 340 IF B<21 THEN GO TO 310
 350 LET B=0
 360 LET B=B+1
 370 IF A(B)=0 THEN LET A(B)=1: GO TO 60
 380 IF B<9 THEN GO TO 360
 390 GO TO 700
 400 PRINT '''' FLASH 1;TAB 5;"Gelijk spel!"
 410 GO TO 590
 430 REM WINST CONTROLE
 440 FOR B=1 TO 4
 450 IF B=1 THEN LET X=1: LET Y=2: LET Z=3
 460 IF B=2 THEN LET X=1: LET Y=4: LET Z=7
 470 IF B=3 THEN LET X=1: LET Y=5: LET Z=9
 480 IF B=4 THEN LET X=3: LET Z=7
 490 FOR C=1 TO 3
 500 IF A(X)=A(Y) THEN IF A(Y)=A(Z) THEN IF A(X)<>0 THEN GO TO 560
 510 IF B=1 THEN LET X=X+3: LET Y=Y+3: LET Z=Z+3
 520 IF B=2 THEN LET X=X+1: LET Y=Y+1: LET Z=Z+1
 530 NEXT C
 540 NEXT B
 550 RETURN
 560 BEEP 1,1: BEEP 2,2
 570 IF A(X)=1 THEN PRINT '''' FLASH 1;TAB 5;"Ik heb gewonnen!"
 580 IF a(x)=2 THEN PRINT ''' FLASH 1;TAB 5;"U heeft gewonnen!"
```

```
EEP .05,X: BORDER RND*7: NEXT X
 610 RUN
 620 REM ZET VAN SPELER
 630 PRINT AT 1,1; FLASH 1;"Uw z
et..."
 640 LET A$=INKEY$
 650 IF A$<"1" OR A$>"9" THEN GO
 TO 640
 660 LET B=VAL (A$)
 670 IF A(B)<>0 THEN GO TO 640
 680 LET A(B)=2
 690 RETURN
 700 REM PRINT OUT
 710 PRINT AT 1,1;"
                  "
 720 PRINT AT 6,5;"1  2  3     ";
 725 LET FLAG=0
 730 FOR B=1 TO 9
 735 IF A(B)=0 THEN LET FLAG=1
 740 IF A(B)=0 THEN PRINT "  _  ";
: BEEP .05,10
 750 IF A(B)=1 THEN PRINT "  O  ";
: BEEP .09,50
 760 IF A(B)=2 THEN PRINT "  X  ";
: BEEP .05,-3
 770 IF B=3 THEN PRINT AT 8,5;"4
  5  6     ";
 780 IF B=6 THEN PRINT AT 10,5;"
7  8  9     ";
 790 NEXT B
 800 IF FLAG=0 THEN GO TO 400
 810 RETURN
```

Beginners Schaken

In 1978, toen schakende computers nog een zeldzaamheid waren, speelden lezers van het Roemeense weekblad Magazinul een schaakpartij tegen een computer, de Felix C-256. Het programma was gedurende twee volle jaren ontwikkeld door Vioreal Darie van het Instituut voor Computer Techniek in Boekarest, Vioreal noemde zijn programma Astro-64. Felix deed een zet, en de lezers werden uitgenodigd, om een goede tegenzet in te sturen. De meest populaire zet werd in Felix ingevoerd, en zijn antwoord werd in de volgende uitgave

erg sterk, en hij werd dan uiteindelijk ook verslagen door de lezers van Magazinul. Een schaak-expert zei, dat het programma soms bijzonder sterke zetten deed, maar op andere momenten 'als een beginner' speelde, en zelfs elementaire fouten maakte.
Zoals de titel van het programma en het verhaaltje over Felix u al hebben laten zien, gaat die zelfde kritiek ook op voor ons programma, 'Beginners Schaken'. Het speelt geen goede partij schaak en moet eenvoudig te verslaan zijn, maar het is om verschillende redenen toch in dit boek opgenomen.
Schaken is altijd een uitdaging geweest voor programmeurs. In de onschuldige dagen voor de micro-computers heerste veelal de mening, dat het schaken door een computer als bewijs door kon gaan voor de stelling, dat computers een zekere mate van intelligentie bezitten. Om met het enorme aantal mogelijkheden rekening te kunnen houden dat bij een schaakspel op kan treden, is een schaakspel dan ook nauwelijks te vergelijken met een programma, dat Dammen of Boter, kaas en eieren speelt. Er zijn ongeveer 10^{120} mogelijke zetten in een partij van 40 zetten, een getal dat ongeveer gelijk is aan het aantal atomen in het heelal! Het programmeren van een computer om met dit aantal mogelijkheden rekening te houden, is dus een geweldige uitdaging.
Er zijn zeer weinig schaakprogramma's in BASIC gepubliceerd. Het feit dat een schaakprogramma kan worden geschreven in BASIC, dat binnen en redelijke tijd zijn zetten doet, is al een belangwekkend feit. Er is ook geen twijfel, dat het bekijken van dit programma fascinerend is, zelfs als het u nooit laat zien, hoe u beter kunt leren schaken. Het programma speelt zeer snel. Het zoekt, sorteert en slaat zijn stukken in minder dan 4 seconden en doet meestal na 6 tot 15 seconden een

waarbij het programma tegen zich zelf speelt. Het is fascinerend om te zien, hoe de computer om beurten met zwart en wit speelt, en het geeft u een goede mogelijkheid om inzicht te krijgen in de zwakheden van het programma dat u hopelijk tot verbetering ervan zult inspireren. Het programma is opzettelijk in z'n geheel in modulaire vorm geschreven, om het mogelijk te maken een stuk programma te vinden, dat zich met een bepaald stuk op het bord bezighoudt. Het is dan ook mogelijk om het programma stap voor stap naar uw eigen inzicht te verbeteren. Het feit, dat Felix zelf maar slecht schaken kon, werd in 1978 al als groot nieuws gezien. Het is daarom misschien verrassend dat 10 jaar na de publikatie in 1949 van het eerste boek over het programmeren van een computer om te schaken (Programming a Computer for playing chess, door Claude Shannon) de eerste, werkelijke schakende computer pas verscheen. In 1960 kregen Alex Bernstein en drie van zijn collega's bij IBM het voor elkaar om het spel te laten lopen op een IBM 704. Ondanks deze relatief late start, zijn er thans een behoorlijk aantal schaakprogramma's beschikbaar, waaronder verschillende zeer goede versies voor de ZX Spectrum.
Maar schaken is, zoals gezegd, een bijzondere uitdaging voor een programmeur. Als u dit programma een beetje bestudeerd heeft, dan heeft u gelijk al een veel beter begrip van hoe een spel-programma ontwikkeld kan worden. Dan bent u beter in staat om te waarderen hoe echte schaakmachines werken, en kunt u ze naar uw hand zetten, door een volledig eigen schaakprogramma te schrijven. Ik heb een aantal ideeën gebruikt, die fundamenteel zijn bij het schrijven van bordspelen in dit programma, en het moet voor u niet moeilijk zijn om andere programma's te bedenken, waar deze tips

een realistisch doel. Voordat ik dit programma schreef, zat ik eens te worstelen met een vroege versie van het damprogramma. Ik probeerde te ontdekken, hoe ik de computer een meervoudige slag kon laten vinden. Trevor Sharples (die dit boek samen met mij geschreven heeft) stelde voor, dat ik een programma zou schrijven, dat meervoudig slaan eenvoudig zou 'vergeten'. Hij gaf aan dat, omdat het programma niet bijzonder sterk speelde, de kans, dat de menselijke tegenstander de mogelijkheid voor meervoudig slaan aan zou bieden, zeer klein was, zodat het programma hier niet naar hoefde te kijken. Later ontwikkelde ik een subroutine, die het probleem oploste. Claude Shannon, de man die het boek over computerschaak in 1949 schreef zei, dat er ongeveer 19^{120} zetten mogelijk waren in een spel van 40 zetten. Shannon voegde hieraan toe, dat een snelle computer ongeveer 10^{90} jaar nodig zou hebben, om alle mogelijke zetten te beoordelen, voordat hij de eerste pion zou verplaatsen.

Ik besloot, dat 10^{90} jaar een beetje lang was en dat 40 maal 10^{90} jaar mijn geduld wel uit zou putten. Een uitgebreide zoekactie in het programma was dus duidelijk niet mogelijk, zelfs als ik de programmeer-kennis had om deze zetten op één of andere manier te beoordelen en de juiste uit te kiezen.

Hoewel ik geen plannen had om een programma te maken, dat iets meer zou kunnen dan de meest essentiële mogelijkheden om naar de mogelijke consequenties van zijn zetten te kijken, wist ik dat het enorme aantal mogelijkheden en combinaties op het eenvoudigste niveau het programma zeer langzaam zou maken. Ik was dan ook aangenaam verrast dat het spelen erg snel ging, toen ik het programma voor de eerste keer uitvoerde. Ik had natuurlijk wel getracht, het programma op

voerd te worden, dan een programma, dat deze juist vooraan heeft staan. Dit komt, omdat de computer bij de aanroep van een subroutine begint te zoeken bij het eerste regelnummer in het programma, en zo verder naar beneden, totdat hij de juiste regel gevonden heeft. Als de routine in de buurt van het eerste regelnummer in het programma is, dan hoeven er minder regels afgezocht te worden. Ik had het gehele programma zorgvuldig gepland, en had het eerste gedeelte geheel op papier uitgeschreven, voordat ik de eerste toets op de Spectrum aanraakte. Dit gaf de mogelijkheid om het gehele programma met de hand 'uit te voeren'. Ik heb toen de verschillende programma-onderdelen zodanig verschoven, dat de programmastructuur zo logisch mogelijk was. Het bleek, dat dit de responstijd erg laag hield. De eerste twee ideeën van het programma waren dus, te mikken op een bescheiden horizon, en de programma-structuur zó te maken, dat de responstijd zo weinig mogelijk verhoogd wordt.

Het volgende belangrijke concept betrof de opbouw van het bord. Mijn eerste pogingen voor een computerbordspel maakte gebruik van een nummeringssysteem, dat ik ontwikkeld had naar een idee van A.L. Samuels in een artikel in Scientific American (Zie Strachey, C. 'Systems Analyses and Programming', in Readings from Scientific American, W.H. Freeman and Co., San Francisco, 1971).

Hoewel dit bevredigend werkte, was er een lange routine nodig om de zetten, die de speler intikte, te converteren in getallen, die de computer kon gebruiken. Een veel betere oplossing werd me geboden door Graham Charlton in 1981 (mede-auteur van The Turing Criterion – Machine Intelligent Programs for the 16K ZX81, Interface, Londen 1982), en deze oplossing ligt in het hart

van het array om te wisselen, waarbij de elementen, die door een stuk worden verlaten, gevuld worden met de CODE van een leeg veld, en de elementen, die het veld aangeven, waar het stuk naar toe beweegt, gevuld worden met de code van het stuk, dat bewogen is. Als het bord uitgeprint wordt, hoeft de computer alleen maar de CHR$ van ieder element van de array uit te printen. Het Charlton bord-nummeringssysteem maakt het ook eenvoudig om getallen, die door de gebruiker ingetikt worden om een zet op het bord aan te geven, te converteren naar elementnummers in het array, die het betreffende veld weergeven.
Ik wist voordat ik begon, dat de speler de mogelijkheid moest hebben, om de velden op het bord aan te geven in de algebraïsche notatie, die redelijk standaard is in schaakkringen, dus heb ik mijn schaakbord zó ingericht, dat dit zo makkelijk mogelijk te regelen was. Dit diagram laat de velden op het bord zien, rondom het bord staan de letters en cijfers, die in de algebraïsche schaaknotatie gebruikt worden, op het bord zelf staat in ieder veld het elementnummer van het array, dat dit veld weergeeft:

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

Het werkelijke voordeel van het Charlton systeem boven andere systemen is, dat iedere zet beschreven kan worden als een eenvoudige optelling bij, of aftrekking van het beginveld. Een uitleg kan dit wellicht verduidelijken. Stelt u zich voor, dat u een paard heeft op veld 45. Zoals u weet, beweegt een paard zich in het schaakspel met een L-vormige sprong, eerst twee velden en dan nog een, of eerst een veld en dan nog twee. Eén van de mogelijke zetten voor het paard is van het huidige veld (45) naar het veld met nummer 66, en toename van 21. Begin op veld nummer 11, tel hierbij 21 op en we krijgen 32, een volgende geldige sprong voor een paard. Begin op 52, tel 21 op, en we krijgen een (geldig) veld 73. De acht zetten van een paard vanaf een willekeurige positie (neem even aan, dat het paard niet van het bord af springt) kunnen gevonden worden door de volgende getallen bij het beginveld op te tellen:

19, –19, 21, –21, 8, –8, 12, –12

Op een overeenkomstige manier kunnen de zetten van een pion (met uitzondering van de eerste zet, die één of twee velden vooruit kan zijn) gevonden worden, door eenvoudige 1 op te tellen bij het veldnummer, waar de pion op staat. Een pion kan dus geldig bewegen van veld 45 naar 46, of van 23 naar 24. Een pion, die een ander stuk slaat, beweegt diagonaal over één veld, dus een pion kan slaan naar velden met als nummer het eigen nummer van de pion + 11 of – 9. Nu we dit weten, kunnen we een computer betrekkelijk eenvoudig programmeren, om een geldige zet te berekenen vanaf een beginveld. Dit is precies wat ik in het programma heb gedaan. Als u naar de regels na 2640 in de initialisatiecode kijkt, dan ziet u hoe dit gedaan wordt. Eerst wor-

zetten van een stuk vasthoudt. Het array N wordt gevuld met data van regel 2870. Als u naar deze regel kijkt, dan ziet u dezelfde acht getallen, die we een aantal paragrafen terug hebben besproken voor de zetten van het paard. Eén manier om een paard te laten springen (en deze methode wordt, met een aantal beperkingen, op een bepaald punt in het schaakprogramma gebruikt) is een random getal te genereren tussen 0 en 8, dat element van N te testen, om te zien, of dat veld vrij is, of een stuk bevat van de tegenstander. Als het veld een stuk van de tegenstander bevat, en een aantal tests wijzen uit, dat dit veld niet door andere stukken aangevallen wordt, dan kan de zet gedaan worden. Dit is de techniek, die de basis vormt van het Beginners Schaakprogramma. Neem even een moment om dit opnieuw te lezen en te overdenken. Als u bereid bent om de nodige hoeveelheid methodisch werk op u te nemen, bent u nu waarschijnlijk wel in staat een eigen schaakprogramma te schrijven, gewapend met alle informatie, die u tot nu toe heeft.
Een programma, dat zijn zetten doet, door eenvoudig met een dobbelsteen te gooien, wordt het slachtoffer van zelfs de meest eenvoudige beginner. Het genereren van zetten moet dus wel enigszins veranderd worden, zoals in dit programma gedaan is, maar de zetten werken precies zoals hier beschreven is. De array voor alle stukken in het programma zijn als volgt:

Paard – 2860 tot 2870
Toren – 2880 tot 2920
Loper – 2930 tot 2970

Koningin – 2990 laadt het array Q met de elementen van de array's van de toren en de loper, omdat de zetten van de koningin eenvoudig een combinatie zijn van alle

Het volgende array, dat gevuld wordt, S (regel 3020 tot 3100), bevat het volgende 'geheim' van ons programma. Vele 'intelligente' spel-programma's (zoals Dammen in dit boek) zoeken het bord veld voor veld af, beginnend bij de linkerbovenhoek tot aan de rechter benedenhoek, waarbij gezocht wordt naar stukken, slagen en zetten. Mijn beslissing – en deze beslissing is waarschijnlijk verantwoordelijk voor de snelheid van het programma – was het testen van de velden in een volgorde, die ik het meest geschikt vond voor het schaakspel. De volgorde, waarin de velden bekeken worden, staan in de DATA-opdrachten van regel 3030 tot 3100. Hier zijn de getallen die u kunt vergelijken met de lokaties op het schaakbord, dat een eindje terug al afgedrukt staat:

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

Het programma kijkt altijd eerst naar veld 46, dan 56, enzovoorts, totdat als laatste van de lijst veld 58 bekeken is. Misschien is het de moeite waard om dit voor een ogenblik te bestuderen, omdat deze volgorde zeer goed voor verandering in aanmerking komt, als u denkt daarmee het spel te verbeteren. Laten we de gedeelten van het programma, die we behandeld hebben, nog even op een rijtje zetten. We hebben al gezien, dat dit programma in het boek opgenomen is, omdat schaken een constante uitdaging voor computerprogrammeurs is, en hoewel Beginners Schaken niet erg sterk speelt, kan het u nogal wat inzicht geven in het programmeren

De programma-listing wordt gevolgd door een aantal variaties van de eerste tien regels, zodat de computer ook tegen zichzelf kan spelen. Dit geeft een fascinerend beeld, en het is een uitstekende manier om fouten uit het programma te halen ('debuggen'). U kunt het programma de gehele dag tegen zichzelf laten spelen, om te zien, of het op en bepaald punt fout loopt, hetgeen aangegeven wordt door het uittypen van een foutmelding. Het bord-nummeringssysteem, dat we gebruikt hebben, is aan de orde geweest, en ik heb verteld, dat door dit nummeringssysteem iedere zet van een schaakstuk uitgedrukt kan worden als een optelling bij, of een aftrekking van het veldnummer, waar het stuk nu staat. De mogelijke zetten van een stuk worden in een array opgeslagen. Het S-array bevat de volgorde, waarin de velden op het bord gecontroleerd worden, als het programma op zoek is naar een zet.
Hier volgt de lay-out van het programma:

10- 30	Startinformatie, zorgt er voor, dat CAPS LOCK is ingedrukt
40- 60	Deze drie regels printen het bord opnieuw nadat een zet door de computer gedaan is, vragen de speler een zet in te voeren, en printen daarna het bord weer.
70-2350	Dit is het belangrijkste gedeelte van het programma, waar al het denkwerk uitgevoerd wordt. We komen hier later uitgebreid op terug.
2360-2460	Hier wordt het bord opnieuw geprint, en (in regels 4210 en 2420) wordt een pion, die tot de achterste rij is doorgelopen, in een koningin veranderd.
2470-2560	Dit is de 'laatste uitweg' als de koning in de

als er geen zet meer gedaan kan worden.

2570-2630 Dit gedeelte accepteert de zet van de speler. De commentaar-subroutine wordt aangeroepen, indien een stuk van de computer geslagen is, en kan de speler verschillende dingen opgeven aan het programma, zodra hij zijn zet heeft ingetikt. Het programma verwacht de invoer van de speler in de vorm van een letter en een cijfer (zoals A5), aan elkaar ingetoetst, om het veld aan te geven, waar u vandaan wilt bewegen, en daarna een volgende letter en een cijfer om het veld aan te geven, waar u naar toe wilt springen. Als dit gedaan is (het programma controleert wel de lengte van de invoer, maar niet de geldigheid van de zet!) dan krijgt u de volgende keuze-mogelijkheden:

C – U heeft de computer schaak gezet
P – U wilt het bord uit laten printen
X – U wilt van zijde wisselen
S – U wilt het spel beëindigen
ENTER – om het spel voort te zetten

Zodra u uw keuze ingetikt hebt, voeren de regels 2625 en 2630 uw zet werkelijk uit.

Dit is het werkende gedeelte van het programma, op de subroutine na, die commentaar levert op het slaan van een stuk van de computer door de speler, en de routine, die aangeroepen wordt, als u van speelzijde wilt verwisselen. De rest van het programma is initialisatie, en bestaat uit het volgende:

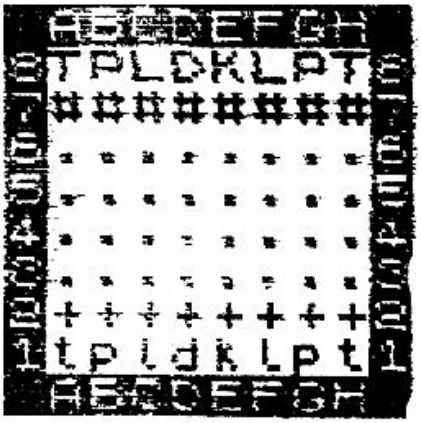
2640 Hier wordt de random-getallen generator ingesteld, de flag MM wordt op nul gezet, de string A$ (die het eerste gedeelte van

gramma. Als de computer een zet vindt, dan wordt MM op één gezet. De flag MM wordt daarna door de gehele uitvoering van het programma gecontroleerd, en alleen als de waarde van MM één is, dan stopt de computer met het zoeken van een zet. Het proces van het zoeken naar een zet wordt in totaal maximaal 8 maal uitgevoerd, voordat de computer besluit, dat hij het spel moet beëindigen, omdat hij geen zet heeft, die MM van nul naar één omzet.

2650 Hier worden de arrays geDIMensioneerd. Dit is de functie van de arrays:
A – bevat het bord, het voornaamste array
R – de zetten van de toren
B – de zetten van de loper
N – de zetten van het paard
Q – de zetten van de koningin
K – de zetten van de koning
Z – wordt gebruikt om de zijden te wisselen
S – volgorde, waarin de velden bekeken worden
T – bevat de plaats van ieder stuk, en wordt bij iedere zet veranderd

2660-2670 Deze twee regels bevatten de variabelen, die voor de stukken gebruikt worden. De stukken worden trouwens weergegeven op het bord door de eerste letter van hun naam (zoals L voor een loper en T voor een toren). De computer speelt met zwart van boven naar beneden, en de stukken van de computer worden in hoofdletters afgedrukt. Uw stukken worden in kleine letters afgedrukt. Zo ziet het bord er aan het begin van het

Even geduld...



Merk op, dat de pionnen en het paard dezelfde beginletter P hebben, het paard wordt daarom met de letter R aangegeven. Ook de koning en de koningin hebben dezelfde beginletter, de koningin heeft daarom als letter de Q (Queen) gekregen.
Zoals u kunt zien, bestaat het bord uit punten in plaats van zwarte en witte vlakken, en ziet u geen user-defined grafische tekens. De uiterlijke verschijning is bij dit programma namelijk op de tweede plaats gekomen. Als u echter echte schaakstukken wilt gebruiken, doe dit dan (en stuur mij, via de uitgever, een kopie van het door u gebruikte programma, omdat ik erg geïnteresseerd ben!). Ieder veld wordt aangegeven door een letter over de breedte en een cijfer over de lengte. Uw koningin staat dus op D1 en de konings-loper van de computer op G8. Dit zijn de variabelen van de stukken:
Mens – wit: P, R, N, B, Q, K
Computer – zwart: PB, RB, NB, BB, QB, KB

3110 Zet de PAPER en BORDER op blauw en de INK op wit. Verander dit in de kleur die u wilt.

3120-3210 Deze routine geeft u de mogelijkheid om van zijde te wisselen op ieder punt van het spel dat u wilt, direct nadat u een zet gedaan heeft. Het wisselen werkt als volgt: de computer doet alsof er een spiegel op het midden van het bord geplaatst is, en de stukken in deze spiegel gereflecteerd worden. Een stuk op A2 komt dus terecht op A7. Hoewel de stukken van de spelers omgewisseld worden, blijft de computer met de hoofdletters en u met de kleine letters spelen. U heeft wel elkaar posities geërfd. De 'auto-play'-optie doet eenvoudig een omwisseling na iedere zet, en laat ieder tweede bord zien (als de stukken weer 'goed' staan), zodat u het spel kunt volgen. U zult zien, dat dit omwissel-systeem eenvoudig te begrijpen is, als u er eenmaal aan gewend bent.

3310-3380 Als u een zet doet naar een veld, dat door een computerstuk bezet wordt (dit wordt gezien door regel 2625), dan gebruikt de Spectrum deze subroutine, om bij het slaan van een van zijn stukken commentaar te geven. Drie van de vier maal laat het de BORDER knipperen, en zegt iets als 'Hebbes!', en speelt een korte melodie, daarna drie treurige noten. De vierde keer laat het programma alleen de drie noten horen, plus een zeer korte flits van de BORDER kleur om zo het slaan aan te geven.

Voordat we uiteindelijk tot de listing van het programma

programma geheel te begrijpen, zodat u het eventueel kunt wijzigen.
In regel 70 bevat MM (zoals u al weet) de zet van de machine. MM moet na iedere zet op nul gezet worden. Als u trouwens zelf als eerste wilt zetten, verwijdert u dan de GO TO 60 aan het einde van regel 30. UK telt het aantal malen, dat het programma naar een zet gezocht heeft. Dit is zo gedaan, om het programma er toe te bewegen, de koning niet te bewegen zonder een goede reden, en om het programma aan te sporen, zichzelf niet zo snel gewonnen te geven. De volgende regels (80 tot 100) volgen de instructies van de speler, die hij geeft, nadat hij een zet gedaan heeft.
Regel 110 dimensioneert het T-array. Door dit iedere keer te doen, wordt het array automatisch met nullen gevuld, zodat het klaar is om de lokatie van ieder van de stukken in op te slaan. U is de variabele die het aantal stukken van de computer telt. De boodschap 'even geduld...' komt aan de bovenzijde van het scherm te staan, terwijl de computer op zoek gaat naar een zet. De routine in regels 120 en 130 gaat langs alle velden op het bord in de volgorde, die aangegeven wordt door het array S, waarbij de stukken op deze velden opgeslagen worden in de volgorde, waarin ze gevonden worden in het array T. De stukken worden dan ook direct geteld. De variabele KM is de positie van de koning, deze variabele wordt door de computer gebruikt om zijn meest waardevolle stuk te allen tijd bij de hand te hebben. KM wordt ingesteld op de lokatie van de koning door regel 120. Regel 130 sluit de loop af. De variabele U telt het aantal stukken op het bord. Als deze kleiner dan 3 is, dan geeft de computer het spel op.
Het programma gaat nu verder op regel 580, waar Z gelijk gesteld wordt aan de positie van de koning in KM.

u uw zet heeft gedaan. Zodra het gevaar gevonden is, gaat de computer naar de routine vanaf regel 1710, waar hij kijkt, of hij zijn koning in veiligheid kan brengen. Terug naar de start van het programma, gaat de computer op regel 150 al zijn stukken na, en plaats de koning aan het einde van deze serie, zodat de koning alleen wordt verplaatst, als er geen andere zetten meer mogelijk zijn. Hij weet uit de tests van de routine vanaf regel 580, dat de koning niet aangevallen is, dus is er geen reden, waarom hij de koning in gevaar zou brengen, door hem van zijn veilige plaats te verplaatsen.

Regel 170 kiest een stuk uit de reeks van stukken, die hij wil verplaatsen, beginnende bij de eerste drie. Dit aantal wordt gereduceerd tot 1 als de speler 'Schaak!' heeft aangegeven aan het einde van regel 170. Als het gekozen getal kleiner is dan het totale aantal stukken op het bord, dan wordt 1 opgeteld bij de waarde van variabele Q, om zo te bepalen, welk stuk van het array T als eerste voor een zet gekozen zal worden.
Regel 190 zet de variabele Z (die door het gehele programma heen gebruikt wordt als de 'van'-lokatie) gelijk aan het element van T, dat gekozen is, en het programma springt drie regels vooruit naar een subroutine, die begint op regel 230. Een BEEP, die door regel 230 geproduceerd wordt, geeft aan, dat de computer aan het werk is. De BEEP klinkt vanaf deze plaats, zodra de computer op zoek is naar een zet, zodat u weet, dat het programma nog steeds aan de gang is, zelfs als het lijkt, dat de Spectrum problemen heeft bij het vinden van een zet.
De volgende vier regels sturen het programma naar een specifieke subroutine, afhankelijk van het stuk, dat het programma wil gaan verzetten. Van deze subroutine-adressen kunt u zien, welk deel van het programma de

Koningin – 820
Toren – 1300
Paard – 1540
Pion – 2070

Regel 280 keert terug naar regel 200, waar de variabele MM bekeken wordt. Als MM gelijk is aan 1, dan gaat het programma naar regel 2310, waar de zet gedaan wordt, en terug naar regel 40, die de subroutine aanroept om het bord opnieuw te printen. Daarna kan de speler zijn zet intikken. Als MM gelijk is aan 0, dan gaat het porgramma verder naar de volgende regel en – als Q kleiner is dan U – wordt het programma voortgezet op regel 180. Hier wordt U verhoogd met 1, en het zoeken wordt voortgezet. Als Q niet kleiner is dan U, dan betekent dit, dat alle willekeurige gekozen stukken nagekeken zijn, en niet konden slaan.
Vanaf dit punt springt het programma naar regel 2180 (mijn weloverwogen plannen om een duidelijk stroomschema voor het programma aan te houden werden hier doorkruist door de eisen van de computer) waar de variabele Q de waarde krijgt van een willekeurige getal tussen 0 en 4 (met het dubbele gebruik van de RND-functie, zodat de meeste getallen aan de onderkant uitkomen). Dit getal wordt nu gecontroleerd, om er zeker van te zijn, dat het niet groter is dan het aantal beschikbare stukken. U kunt het cijfer 5 in deze rekenkundige uitdrukking vervangen door een hogere waarde, als u wilt, dat de computer niet op steeds dezelfde, voorspelbare wijze bepaalde stukken ontwikkelt. Ieder getal onder de tien is redelijk bruikbaar. Hier wordt dan weer 1 bij opgeteld, en Z wordt gelijk gemaakt aan het stuk, dat nu

Pion – 2140
Paard – 1630
Loper – 1420
Toren – 1180
Koningin – 940
Koning – 1710 (alleen als de koning niet schaak staat en het willekeurig gekozen getal is kleiner dan 0,07. Hiermee wordt voorkomen, dat de koning zinloos gaat 'wandelen').

Zodra de computer de relevante subroutine heeft uitgevoerd, wordt bekeken, of de flag MM nog steeds nul is. Als dit het geval is, en nog niet alle stukken uit het willekeurig gekozen beginstuk zijn gecontroleerd, dan gaat het programma naar 2180, waar een nieuw beginstuk gekozen wordt. Als MM gelijk is aan 1, dan gaat het programma naar regel 2310, waar een subroutine de zet doet, en keert daarna terug naar regel 40, om het bord te laten printen, voordat de speler weer aan de beurt is om een zet te doen. De volgende regel kijkt of de waarde van de variabele UK groter is dan 8, als dit zo is, dan is de gehele routine om een zet te zoeken 8 maal doorlopen. De computer heeft dan in 8 maal zoeken nog een goede zet kunnen vinden, dus hij geeft het spel op. Dit gebeurt in regel 2060.
De volgende regels maken de werkelijke zet van de computer. Regel 2319 verwerpt een zet van een koning negen van de tien keer als de koning niet schaak staat, door MM weer op nul terug te zetten en terug te gaan naar regel 2180, waar een ander stuk gezocht wordt om mee te spelen. Regel 2312 wordt aangeroepen als het te verplaatsen stuk een pion is. Deze routine kijkt, of het programma niet probeert een pion naar achteren te verplaatsen, en zorgt er voor, dat die zet niet gemaakt kan worden. Als de witte koning op het veld staat, waar

'SCHAAK!' afgedrukt op het scherm. De computer gaat dan terug naar regel 190, waar een andere slag gezocht wordt. Als de gevonden zet al deze 'horden' heeft doorstaan (dit is een eenvoudig stuk van het programma, dat u zelf kunt wijzigen, als u bepaalde zetten wilt voorkomen, of de computer wilt ontmoedigen, een bepaalde zet te doen), dan doet regel 2315 de zet, en regels 2320 tot 2330 vertellen u (met gebruikmaking van een op het eerste gezicht bijzonder complexe regel, die gebruikt wordt om het veldnummer om te zetten in een veldnaam dat de zet gedaán is. Regel 2340 keert terug naar het begin van het programma, de sectie waar het bord uitgeprint wordt.
Ondanks de routine om gevaar voor de koning op te merken, is het programma niet altijd in staat om zichzelf uit een 'Schaak!'-situatie te redden. U kunt zo'n fout opvatten als een wens om het spel op te geven (of als u tolerant bent, het programma toestaan gedurende één zet schaak te staan). Als de computer zichzelf in schaak begeeft (een zeer uitzonderlijk geval), dan kunt u ook aannemen, dat hij opgeeft.

```
   10 REM BEGINNERS SCHAAK
   20 PRINT "SCHAKEL "; FLASH 1;"
CAPS LOCK"; FLASH 0;" IN.": PRIN
T "DAARNA ENTER INDRUKKEN": INPU
T A$: CLS : PRINT '"Dank U, eve
n geduld..."
   30 GO SUB 2640: GO TO 60
   40 GO SUB 2350
   50 GO SUB 2570
   60 GO SUB 2350
   70 LET MM=0: LET UK=0
   80 IF A$="E" THEN STOP
   90 IF A$="Z" THEN PRINT AT 0,0
; FLASH 1;"Zijden wisselen": GO
SUB 3120: PRINT AT 0,0;"
          ": LET A$=""
  100 IF A$="P" THEN COPY
  110 DIM T(16): LET U=0: PRINT A
```

```
PB THEN LET U=U+1: LET T(U)=S(Q)
: IF A(S(Q))=KB THEN LET KM=S(Q)
 130 NEXT Q: IF U<3 THEN GO TO 2
060
 140 GO TO 560
 150 FOR Q=1 TO U: IF A(T(Q))=KB
 THEN LET T(Q)=T(U): LET T(U)=KM
 160 NEXT Q
 170 LET Q=INT (RND*3)*((A$="S")
+1)
 180 IF Q<U THEN LET Q=Q+1
 190 LET Z=T(Q): GO SUB 230
 200 IF MM=1 THEN GO SUB 2310: G
O TO 40
 210 IF Q<U THEN GO TO 180
 220 GO TO 2160
 230 BEEP .008,RND*30+20: IF A(Z
)=QB THEN GO SUB 620
 240 IF A(Z)=RB THEN GO SUB 1060
 250 IF A(Z)=BB THEN GO SUB 1300
 260 IF A(Z)=NB THEN GO SUB 1540
 270 IF A(Z)=PB THEN GO SUB 2070
 290 IF A(X)=K THEN PRINT AT 0,0
; FLASH 1;"    Schaak!": LET Q=Q
+1: LET MM=0: GO TO 190
 300 IF X+9>88 THEN GO TO 320
 310 IF A(X+9)<83 AND A(X+9)>65
AND RND<.96 THEN RETURN
 320 IF X-11<11 THEN GO TO 340
 330 IF A(X-11)<83 AND A(X-11)>6
5 AND RND<.96 THEN RETURN
 340 LET AD=0
 350 LET AY=1
 360 LET AX=X+O(AY+AD)
 370 IF AX<11 OR AX>88 THEN GO T
O 400
 380 LET AP=A(AX)
 390 IF AP=0 OR AP=R AND RND>.8
OR AP=B AND RND>.5 THEN RETURN
 400 LET AY=AY+1
 410 IF AY<8 THEN GO TO 360
 420 LET AD=AD+7
 430 IF AD<56 THEN GO TO 350
 440 LET AY=1
 450 LET AX=X+N(AY)
 460 IF AX<11 OR AX>88 THEN GO T
O 480
 470 IF A(AX)=N THEN RETURN
 480 LET AY=AY+1
 490 IF AY<9 THEN GO TO 450
 500 LET AY=1
 510 LET AX=X+K(AY)
```

```
RND>.1 THEN RETURN
 540 LET AY=AY+1
 550 IF AY<9 THEN GO TO 510
 560 LET MM=1
 570 RETURN
 580 LET Z=KM
 590 LET Y=0
 600 LET Y=Y+1
 610 LET X=Z+N(Y)
 620 IF X<11 OR X>88 THEN GO TO
640
 630 IF A(X)=N THEN GO TO 1710
 640 IF Y<6 THEN GO TO 600
 650 LET D=0
 660 LET Y=1
 670 LET X=Z+Q(Y+D)
 680 IF X<11 OR X>88 THEN GO TO
730
 690 IF A(X)=B OR A(X)=Q OR A(X)
=R THEN GO TO 1710
 700 IF A(X)<>E THEN GO TO 730
 710 LET Y=Y+1
 720 IF Y<8 THEN GO TO 670
 730 LET D=D+7
 740 IF D<49 THEN GO TO 670
 750 LET X=Z+11
 760 IF X>88 THEN GO TO 780
 770 IF A(X)=P THEN GO TO 1710
 780 LET X=Z-11
 790 IF X<1 THEN GO TO 150
 800 IF A(X)=P THEN GO TO 1710
 810 GO TO 150
 820 LET D=0
 830 LET Y=1
 840 LET X=Z+Q(Y+D)
 850 IF X<11 OR X>88 THEN GO TO
910
 860 IF A(X)=42 OR A(X)>=BB AND
A(X)<=RB THEN GO TO 910
 870 IF A(X)>=S AND A(X)<=R THEN
 GO SUB 290:  IF MM<>1 THEN GO TO
 910
 880 IF MM=1 THEN RETURN
 890 LET Y=Y+1
 900 IF Y<8 THEN GO TO 840
 910 LET D=D+7
 920 IF D<49 THEN GO TO 830
 930 RETURN
 940 LET D=0
 950 LET Y=1
 960 LET X=Z+Q(Y+D)
```

```
IF MM=0 THEN GO TO 1030
900 IF MM=1 THEN RETURN
910 LET Y=Y+1
920 IF Y<8 THEN GO TO 960
930 LET D=D+7
940 IF D<49 THEN GO TO 950
950 RETURN
960 LET D=0
970 LET Y=1
980 LET X=Z+R(Y+D)
990 IF X<11 OR X>86 THEN GO TO
1000
1010 IF A(X)=42 OR A(X)>=0B AND
A(X)<=R6 OR A(X)=P6 THEN GO TO 1
090
1020 IF A(X)>=0 AND A(X)<=R THEN
GO SUB 290: IF MM=0 THEN GO TO
1050
1030 IF MM=1 THEN RETURN
1040 LET Y=Y+1
1050 IF Y<8 THEN GO TO 1080
1060 LET D=D+7
1070 IF D<28 THEN GO TO 1070
1080 RETURN
1090 LET D=0
1100 LET Y=1
1110 LET X=Z+R(Y+D)
1120 IF X<11 OR X>86 THEN GO TO
1270
1220 IF A(X)<>E THEN GO TO 1270
1230 IF RND<.1 THEN GO SUB 290
1240 IF MM=1 THEN RETURN
1250 LET Y=Y+1
1260 IF Y<8 THEN GO TO 1200
1270 LET D=D+7
1280 IF D<28 THEN GO TO 1190
1290 RETURN
1300 LET D=0
1310 LET Y=1
1320 LET X=Z+B(Y+D)
1330 IF X<11 OR X>86 THEN GO TO
1390
1340 IF A(X)=42 OR (A(X)>=0B AND
A(X)<=RB) OR A(X)=P6 THEN GO TO
1390
1350 IF (A(X)>=0 AND A(X)<=R) OR
A(X)=P THEN GO SUB 290: IF MM<>
1 THEN GO TO 1390
1360 IF MM=1 THEN RETURN
1370 LET Y=Y+1
1380 IF Y<8 THEN GO TO 1320
1390 LET D=D+7
```

```
1430 LET Y=1
1440 LET X=Z+B(Y+D)
1450 IF X<11 OR X>88 THEN GO TO
1510
1460 IF A(X)<>E THEN GO TO 1510
1470 IF RND>.05 THEN GO SUB 290:
 IF MM<>1 THEN GO TO 1510
1480 IF MM=1 THEN RETURN
1490 LET Y=Y+1
1500 IF Y<8 THEN GO TO 1440
1510 LET D=D+7
1520 IF D<28 THEN GO TO 1430
1530 RETURN
1540 LET Y=1
1550 LET X=Z+N(Y)
1560 IF X<11 OR X>88 THEN GO TO
1500
1570 IF A(X)=42 THEN GO TO 1600
1580 IF (A(X)>=0 AND A(X)<=R) OR
 A(X)=P THEN GO SUB 290
1590 IF MM=1 THEN RETURN
1600 LET Y=Y+1
1610 IF Y<9 THEN GO TO 1550
1620 RETURN
1630 LET Y=0
1640 LET X=Z+N(INT (RND*8+1))
1650 IF X<11 OR X>88 THEN GO TO
1640
1660 IF A(X)=42 THEN GO TO 1640
1670 LET Y=Y+1
1680 IF A(X)=E THEN GO SUB 290
1690 IF MM=1 OR Y>20 THEN RETURN
1700 GO TO 1640
1710 LET YK=1
1720 LET Z=KM
1730 LET X=Z+K(YK): LET X1=X
1740 IF X<11 OR X>88 THEN GO TO
2030
1750 IF A(X)=42 OR A(X)>65 AND A
(X)<83 THEN GO TO 2030
1760 IF A(X)>97 AND A(X)<115 THE
N GO TO 2030
1770 LET Z=X
1780 LET Y=0
1790 LET Y=Y+1
1800 LET X=Z+N(Y)
1810 IF X<11 OR X>88 THEN GO TO
1830
1820 IF A(X)=N THEN GO TO 2030
1830 IF Y<8 THEN GO TO 1790
1840 LET D=0
```

```
1860 IF A(X)=B OR A(X)=0 OR A(X)
=R THEN GO TO 2030
1890 IF A(X)<>E THEN GO TO 1920
1900 LET Y=Y+1
1910 IF Y<8 THEN GO TO 1860
1920 LET D=D+7
1930 IF D<49 THEN GO TO 1860
1940 LET X=Z+11
1950 IF X>86 THEN GO TO 1970
1960 IF A(X)=P THEN GO TO 2030
1970 LET X=Z-11
1980 IF X<11 THEN GO TO 2000
1990 IF A(X)=P THEN GO TO 2030
2000 LET X=X1: LET Z=KM
2010 LET MM=1
2020 GO SUB 2310: GO TO 40
2030 LET YK=YK+1
2040 LET Z=KM
2050 IF YK<9 THEN GO TO 1720
2060 PRINT AT 0,0; FLASH 1;"Ik g
ef oP, kampioen!": STOP
2070 LET X=Z-9
2080 IF (A(X)>=0 AND A(X)<=R) OR
 A(X)=P THEN LET MM=1: IF A(X)=P
 AND RND<.2 THEN LET MM=0
2090 IF MM=1 THEN RETURN
2100 IF Z=12 THEN RETURN
2110 LET X=Z-11
2120 IF (A(X)>=0 AND A(X)<=R) OR
 A(X)=P THEN LET MM=1: IF A(X)=P
 AND RND<.2 THEN LET MM=0
2130 RETURN
2140 IF Z<>12 THEN IF Z-10*(INT
(Z/10))=7 AND A(Z-1)=E AND A(Z-2
)=E AND (A(Z-13)=E OR A(Z-13)=42
) AND (A(Z+7)=E OR A(Z+7)=42) TH
EN LET X=Z-2: LET MM=1: RETURN
2145 IF Z=12 THEN RETURN
2150 IF A(Z-1)=E AND A(Z-12)<98
AND A(Z+8)<98 THEN LET X=Z-1: LE
T MM=1: RETURN
2160 IF RND<0.5 AND A(Z-1)=E THE
N LET X=Z-1: LET MM=1
2170 RETURN
2180 LET Q=INT (RND*RND*5): IF Q
>U THEN GO TO 2180
2190 IF Q<U THEN LET Q=Q+1
2200 LET Z=T(Q)
2210 IF A(Z)=PB THEN GO SUB 2140
2220 IF A(Z)=NB THEN GO SUB 1630
2230 IF A(Z)=BB THEN GO SUB 1420
```

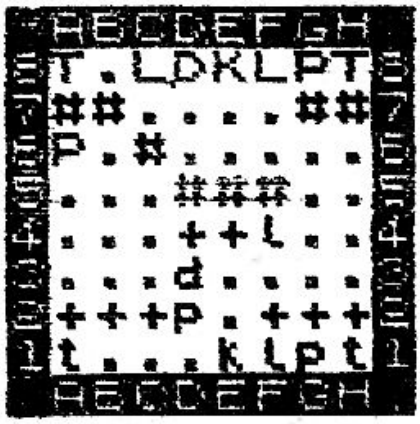
```
2270 IF MM=0 AND 0<U THEN GO TO
2280
2280 IF MM=1 THEN GO SUB 2310: G
O TO 40
2300 LET UK=UK+1: IF UK>8 THEN G
O TO 2060
2305 GO TO 2180
2310 IF A(Z)=KB AND A$<>"S" AND
ND>.1 THEN BEEP .05,-3: LET MM=
0: GO TO 2160
2312 IF A(Z)=PB AND ((X-10*INT (
X/10))>Z-10*INT (Z/10) OR ABS (X-
Z)>11)) THEN LET MM=0: GO TO 216
0
2314 IF A(X)=K THEN PRINT AT 0,0
;"          Schaak!          ": LET MM=
1: LET U=U+1: GO TO 190
2315 LET A(X)=A(Z): LET A(Z)=E
2320 PRINT AT 0,0;"Ik heb gezet
    ";
2330 LET FZ=INT (Z/10): PRINT CH
R$ (FZ+64);Z-10*FZ;" naar ";: LE
T FX=INT (X/10): PRINT CHR$ (FX+
64);X-10*FX
2340 RETURN
2350 IF MM=0 THEN GO TO 2370
2360 BEEP 1,RND*10: BEEP 1,10+RN
D*10: BEEP 1,-RND*10
2370 PRINT AT 0,0;"
                    ";AT 4,0;: GO SU
B 2450
2380 FOR X=8 TO 1 STEP -1
2390 PRINT TAB 10; INVERSE 1;X;
INVERSE 0;
2400 FOR Y=10 TO 80 STEP 10
2410 IF A(Y+1)=PB THEN LET A(Y+1
)=0B
2420 IF A(Y+8)=B THEN LET A(Y+8)
=0
2430 PRINT CHR$ A(X+Y);
2440 NEXT Y: PRINT INVERSE 1;X:
NEXT X: LET MM=0
2450 PRINT TAB 10; INVERSE 1;" A
BCDEFGH "
2460 RETURN
2470 LET Z=KM
2480 LET QK=0
2490 LET M=Z+K(QK)
2500 IF A(M)=42 OR A(M)>65 AND A
(M)<83 OR MM=0 THEN GO TO 2540
```

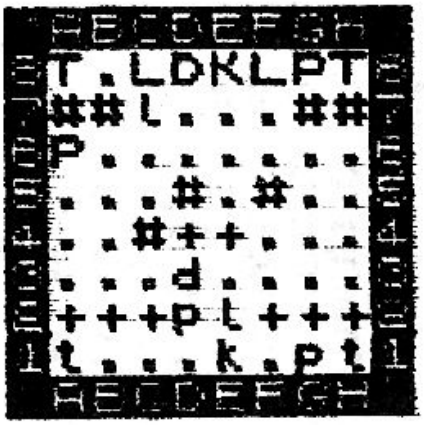
```
2560 GO TO 2060
2570 INPUT "(VAN (LETTER, NUMMER
) ";A$: BEEP .008,-3
2580 IF LEN A$<>2 THEN GO TO 257
0
2590 INPUT "VAN ";(A$);" NAAR? "
;B$: BEEP .008,-10
2595 IF LEN B$<>2 THEN GO TO 259
0
2600 LET X=10*(CODE A$-64)+VAL A
$(2)
2610 LET Y=10*(CODE B$-64)+VAL B
$(2)
2620 INPUT "Toets S - Schaak
                P - Print bord
                Z - Zijden wis
                E - Einde spel
selen           ENTER - Spel voor
zetten ";A$
2625 IF (A(Y)>=0B AND A(Y)<=RB)
OR A(Y)=PB THEN GO SUB 3310
2630 BEEP .008,-3: LET A(Y)=A(X)
: LET A(X)=46: RETURN
2640 RANDOMIZE : LET MM=0: LET A
$="  "
2650 DIM A(99): DIM R(28): DIM B
(28): DIM N(8): DIM Q(56): DIM K
(8): DIM Z(88): DIM S(64): DIM T
(16)
2660 LET P=CODE "+": LET R=CODE
"t": LET N=CODE "p": LET B=CODE
"(": LET WQ=CODE "d": LET K=CODE
 "k": LET E=46
2670 LET PB=CODE "#": LET RB=COD
E "T": LET NB=CODE "P": LET BB=C
ODE "L": LET QB=CODE "D": LET KB
=CODE "K"
2680 FOR Z=1 TO 99: LET A(Z)=42:
NEXT Z
2690 FOR Z=1 TO 64: READ X: READ
Y: LET A(X)=Y: NEXT Z
2700 DATA 16,84,26,80,36,76,46,6
2710 DATA 56,75,66,76,76,80,66,8
2720 DATA 17,35,27,35,37,35,47,3
2730 DATA 57,35,67,35,77,35,67,3
2740 DATA 16,46,26,46,36,46,46,4
2750 DATA 56,46,66,45,76,46,86,4
```

```
2780 DATA 14,46,24,46,34,46,44,4
2790 DATA 54,46,64,46,74,46,84,4
2800 DATA 13,46,23,46,33,46,43,4
2810 DATA 53,46,63,46,73,46,83,4
2820 DATA 12,43,22,43,32,43,42,4
2830 DATA 52,43,62,43,72,43,82,4
2840 DATA 11,116,21,112,31,108,4
1,100
2850 DATA 51,107,61,108,71,112,8
1,116
2860 FOR Z=1 TO 8: READ N(Z): NE
XT Z
2870 DATA 19,-19,21,-21,8,-8,12,
-12
2880 FOR Z=1 TO 28: READ R(Z): N
EXT Z
2890 DATA 10,20,30,40,50,60,70
2900 DATA -1,-2,-3,-4,-5,-6,-7
2910 DATA -10,-20,-30,-40,-50,-5
0,-70
2920 DATA 1,2,3,4,5,6,7
2930 FOR Z=1 TO 28: READ B(Z): N
EXT Z
2940 DATA -11,-22,-33,-44,-55,-6
6,-77
2950 DATA 11,22,33,44,55,66,77
2960 DATA 9,18,27,36,45,54,63
2970 DATA -9,-18,-27,-36,-45,-54
2980 RESTORE 2690
2990 FOR Z=1 TO 56: READ Q(Z): N
EXT Z
3000 FOR Z=1 TO 8: READ K(Z): NE
XT Z
3010 DATA 1,11,9,10,-10,-9,-11,-
3020 FOR Z=1 TO 64: READ S(Z): N
EXT Z
3030 DATA 46,56,36,66,47,57,45,5
3040 DATA 37,67,35,65,28,78,27,7
3050 DATA 44,54,26,76,38,68,17,8
```

```
3080 DATA 53,33,63,23,73,52,42,6
3090 DATA 32,83,13,72,22,12,62,4
3100 DATA 51,31,61,21,71,11,81,5
3110 PAPER 1: BORDER 1: INK 7: C
S : RETURN
3120 FOR Z=11 TO 88: LET Z(Z)=A(
Z): NEXT Z
3130 FOR Z=11 TO 88: LET X=Z-10*
INT (Z/10)
3140 IF X=0 OR X=9 THEN GO TO 31
0
3150 LET A(Z)=Z(Z+9-X*2)
3160 NEXT Z
3170 FOR Z=11 TO 88: LET M=A(Z)
3180 IF M=P THEN LET A(Z)=PB
3181 IF M=PB THEN LET A(Z)=P
3182 IF M=R THEN LET A(Z)=RB
3183 IF M=RB THEN LET A(Z)=R
3184 IF M=NB THEN LET A(Z)=N
3185 IF M=N THEN LET A(Z)=NB
3186 IF M=WQ THEN LET A(Z)=QB
3187 IF M=QB THEN LET A(Z)=WQ
3188 IF M=KB THEN LET A(Z)=K
3189 IF M=K THEN LET A(Z)=KB
3190 IF M=B THEN LET A(Z)=BB
3191 IF M=BB THEN LET A(Z)=B
3200 NEXT Z
3210 RETURN
3310 LET COMMENT=INT (RND*4): BO
RDER RND*5+2: GO SUB 3330+10*COM
MENT
3320 BEEP 1,1: BEEP 1,2: BEEP 1,
3
3330 BORDER 1: RETURN
3340 PRINT AT 0,0;"Goed gedaan":
GO SUB 3370: RETURN
3350 PRINT AT 0,0;"Goede zet": G
O SUB 3370: RETURN
3360 PRINT AT 0,0;"Hebbes!"
3370 FOR I=1 TO 10: BEEP .05,I:
NEXT I
3380 PRINT AT 0,0;"                "
RETURN
```

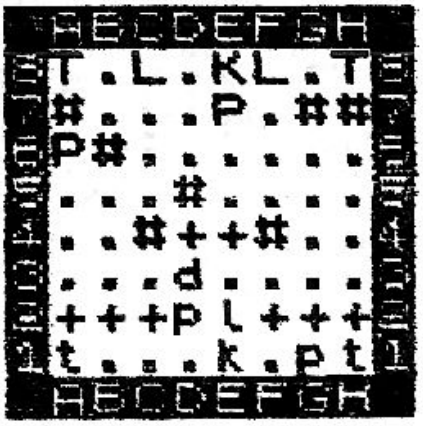
Even geduld...





Dit is de positie, die in de vroege stadia van het middenspel in en andere partij bereikt werd:

Even geduld...



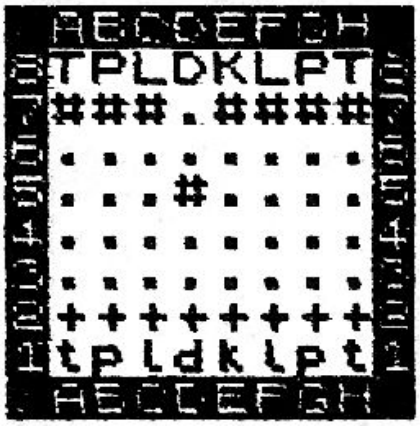
Het is mogelijk om het programma te veranderen, zodat het tegen zichzelf speelt. Dit gebeurt door aan het programma op te geven, dat na iedere zet de zijden verwisseld moeten worden. Er wordt dan ook een print-out van het scherm gemaakt, dit gebeurt na iedere tweede omwisseling. Om het programma dus 'auto-schaak' te laten spelen, dient u de eerste regels uit het programma als volgt te veranderen:

```
    10 REM BEGINNERS SCHAAK
    20 LET PR=1
    30 PRINT "SCHAKEL "; FLASH 1;"
    40 LOCK"; FLASH 0;" IN,": PRIN
T "DRUK ENTER INDRUKKEN": INPU
```

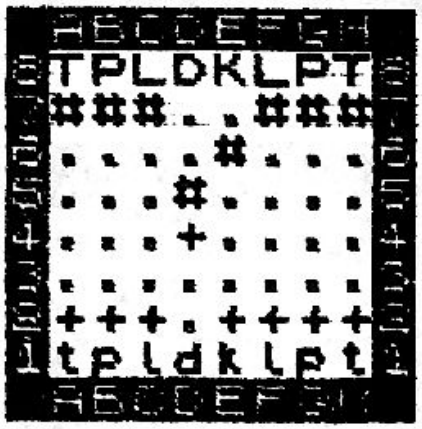
```
   70  LET MM=0:  LET UK=0
   80  IF A$="E" THEN STOP
   90  LET A$="Z"
  100  IF A$="Z" THEN PRINT AT 0,0
;"Zijden wisselen":  GO
  120  PRINT AT 0,0;"
       ":  LET A$=""
  100  IF PR=0 THEN COPY : LET PR=
1:  GO TO 110
  105  LET PR=0
  110  DIM T(16):  LET U=0:  PRINT A
T 0,0;  INK RND*5+2;  PAPER 9;" Ev
en geduld..."
  120  FOR Q=1 TO 64:  IF (A(S(Q))>
```

Als u geen print-out wilt, dan kunt u regels 12, 100 en 105 verwijderen.
Hier is het begin van een partij, die door het veranderde programma gespeeld werd:

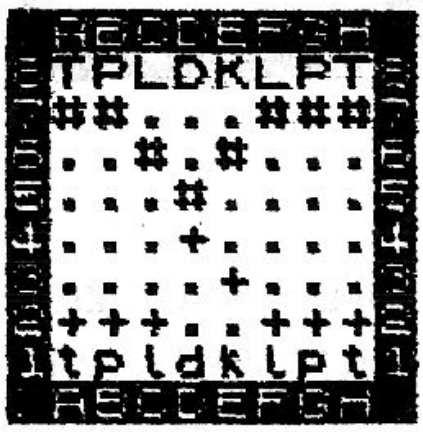
1



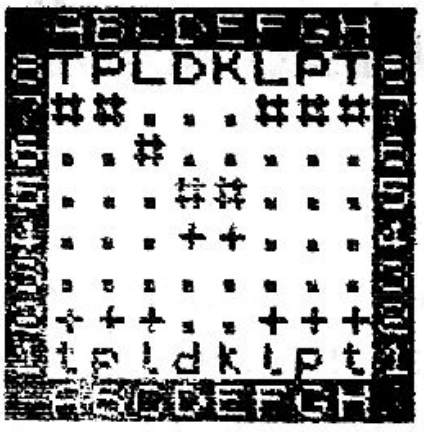
2



3



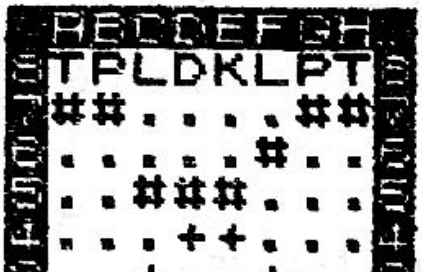
4



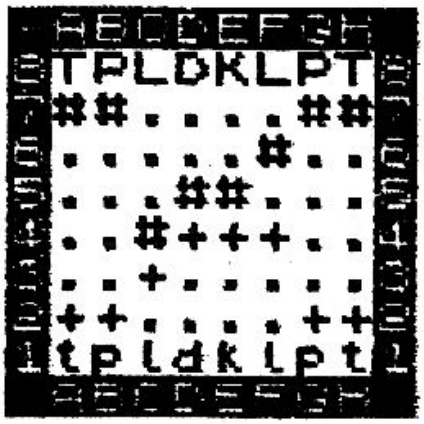
5



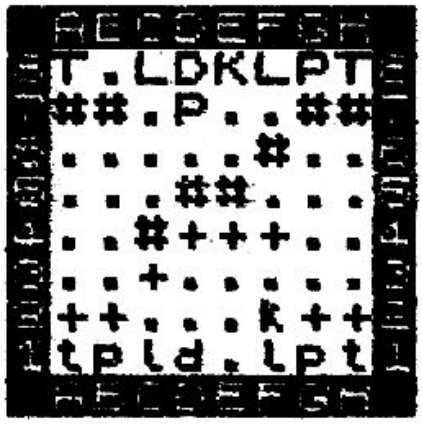
6



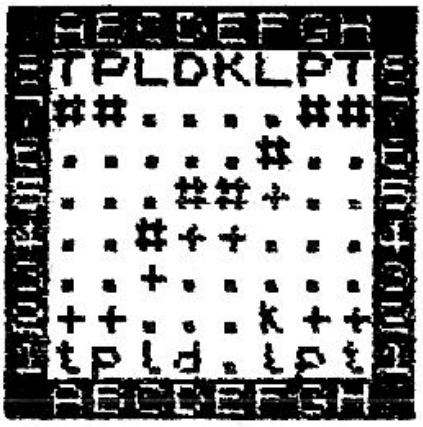
7



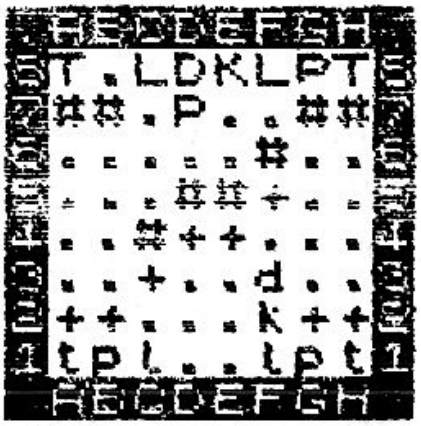
8



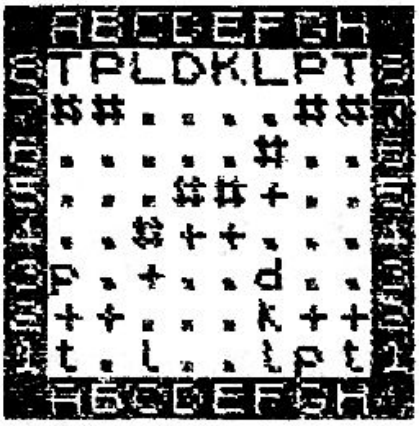
9



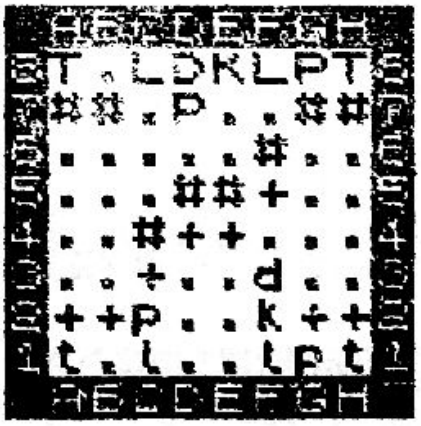
10



11



12



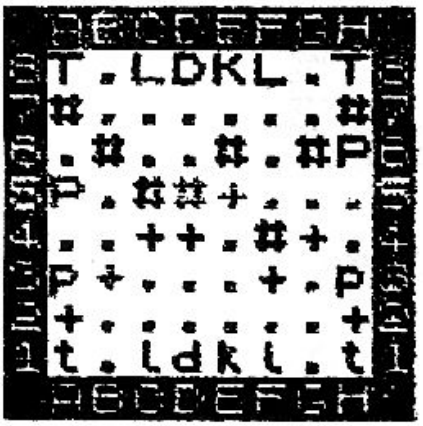
Door het programma op deze wijze tegen zichzelf te laten spelen, kunt u op eenvoudige wijze fouten opsporen. Het is namelijk zo, dat uiteindelijk de meeste van de aanwezige GOTO- en GOSUB-opdrachten wel een keer uitgevoerd worden. Verwijder, indien nodig, de regels, die voor een print-out zorgen, en laat de computer gewoon de gehele dag lopen, spel na spel spelend. Als

Het programma kan nog iets gewijzigd worden, om de mogelijkheid te geven, om niet alle zetten uit te printen, maar van tijd tot tijd eens een scherm naar de printer te kopiëren. Om deze 'random' print-outs te krijgen, moet het begin van het programma er als volgt uitzien:

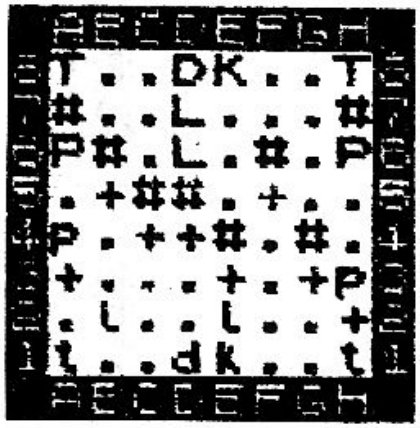
```
100 IF PR=0 THEN LET PR=1
101 IF PR=1 AND RND<.1 THEN COPY
 : GO TO 110
105 LET PR=0
```

Hier is een aantal van deze 'snapshots' van een partij, die op deze wijze geproduceerd zijn:

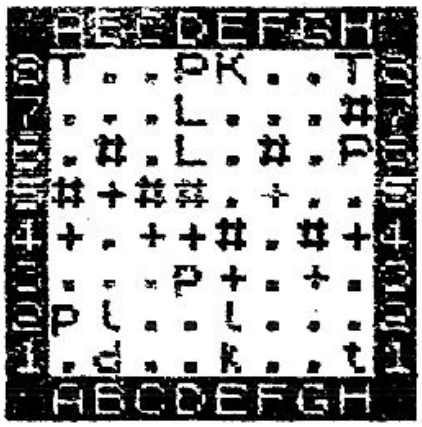
1



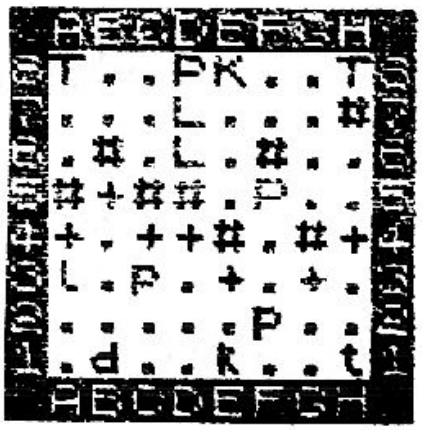
2



3



4

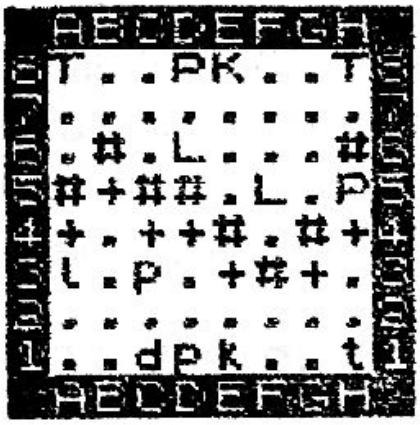


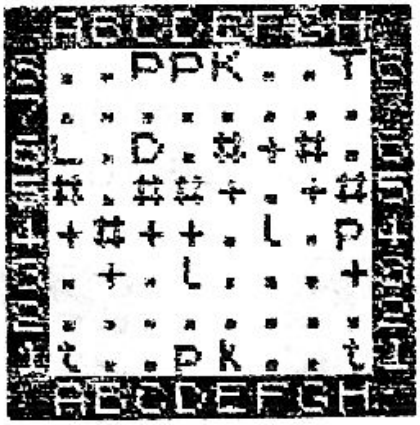
5

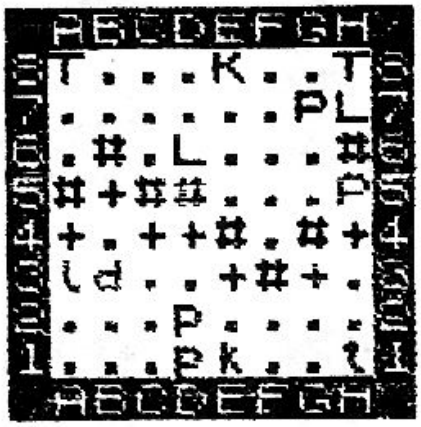


6









Als u meer over schaken en computer-schaak wilt lezen, zijn er een groot aantal publicaties over dit onderwerp verschenen. We noemen er enkele:

Intelligent Computer Games Levy, D., *Creative Computing*, november 1980 (p. 158-163) en december 1980 (p. 208-213)

The Sargon Chronicle Ehara, T. H., *Creative Computing,* mei 1980 (p. 42-44)

World Chess Championship Computer Ehara, T. H., *Creative Computing,* januari 1979 (p. 134-136)

Attention, Chess Phreaks! Palenik, L., *Creative Computing,* januari 1979 (p. 78)

Tumult and the Toronto Tournament Dowhal, D., *Personal Computing,* mei 1978 (p. 98-113)

A Romanian Rhapsody (Bron voor het verhaal over Felix), Friedberg, U., *Personal Computing,* oktober 1978 (p. 76-88)

Sargon vs. Microchess Martellaro, J., *on Computing,* winter 1979 (p. 26-30)

Programming a Computer for Playing Chess Shannon, C. E., *Philosophical Magazine,* vol. 41 (7e jaargang) (p. 256-275)

Chess Skill in Man and Machine Frey, Peter W. (Springer Verlag, 1977)

Technique in Chess Abrahams, G., (Dover Publications Inc., USA, 1973)

The Complete Book of Chess Horowitz, I. A. en Rothenberge, P. L., (Collier Macmillan, 1969)

AVONTUUR/SIMULATIE

De Wraak van Kasteel Angst

Geen enkele verzameling spelprogramma's zou compleet zijn zonder een avontuur. Ik heb besloten, dat het avontuur in dit boek een van de belangrijkste programma's moet zijn, dus heb ik een nogal uitgebreid programma geschreven, dat de onheilspellende titel 'De Wraak van Kasteel Angst' mee heeft gekregen. De eersten, die een avonturenprogramma geschreven hebben voor een computer zijn Crowther en Woods. Zij hadden hun programma, met de eenvoudige naam 'Adventure', op het computersysteem van de Stanford University gemaakt. Het computersysteem was uitgerust met een aantal timesharing-terminals, en de uitvoer werd uitgeprint op lange rollen papier, met terminals, die wel iets van een Telexmachine weg hadden.
Adventure werd onmiddellijk razend populair. Er werden kopieën over de gehele Verenigde Staten gebruikt, en al snel veroverde het spel de hele wereld. Toen meer en meer computerfanaten het spel in handen kregen, werd het al snel veranderd en verbeterd. Dit leidde uiteindelijk tot versies, die niets meer weg hadden van de originele versie. Het is echter zo, dat er twintig jaar na de eerste versie van Adventure, nog steeds Amerikaanse softwarebedrijven zijn, die een programma verkopen, dat zichzelf de 'originele Adventure, noemt. Het schrijven van een avonturenprogramma is een fascinerende oefening. In essentie bedenkt de ontwerper een volledig, logisch consistent universum, en brengt dat in kaart. Met 'logisch consistent' bedoel ik, dat in een correct gecontstrueerd avonturen-programma, een speler

geven wordt, waarom de brug er niet meer is. Deze verklaring hoeft natuurlijk niet alledaags te zijn, en zou bijvoorbeeld kunnen luiden: 'De buitenaardse wezens van Epsilon IV hebben de brug meegenomen naar hun museum-planeet met een materie-transmitter'. Willekeurige avonturen-programma's, waarin geen verklaringen voor veranderingen worden gegeven, zijn frustrerend voor de speler, en kunnen hem nooit lang boeien.
In 'De Wraak van Kasteel Angst' speelt het spel zich af in een kasteel, waar een aantal monsters huizen, zoals 'de boze tovenaar', 'de vuurspugende Angstbrenger' en de onheilspellende 'Bewaker van de Zwarte Lagune', die gedurende het spel door het kasteel lopen, en die bevochten moeten worden (u kunt ook trachten te vluchten, hoewel dat slechts een kans van slagen van één op de twee keer heeft). Daarbij zijn er tot maximaal 4 kisten aanwezig, die goede en slechte verrassingen bevatten, een fles, die een magische drank bevat, die zowel een goede als een boosaardige uitwerking kan hebben, enzovoorts. Het is onwaarschijnlijk, dat u alle monsters en alle verrassingen in ieder spel tegenkomt. U krijgt geen kaart van het kasteel. Een gedeelte van het speelplezier bestaat uit het ontdekken van de ruimtelijke samenhang tussen de verschillende 'kamers', die u op uw reis tegenkomt. Het kasteel in dit spel is tot op zekere hoogte in kaart gebracht, en ik heb er voor gezorgd, dat de aanwijzingen, die u in de PRINT-opdrachten te zien krijgt, zorgvuldig overeenkomen met de relaties tussen de verschillende kamers. Dit is gedaan, zodat u zelf een kaart kunt maken van het kasteel, terwijl u op ontdekkingsreis bent en om voorbeelden aan te geven, hoe u een geschikte 'Avontuur-omgeving' kunt maken. Als u een kaart heeft, die redelijk goed met het programma overeenkomt, dan kunt u die uittesten, door

de kaart een beetje moeilijker te maken. Alle aanwijzingen in het programma worden als richtingen gegeven (noord, zuid, oost en west), en alle deuren in de vertrekken zijn in een van deze richtingen. U kunt het maken van uw kaart het beste beginnen door de vier richtingen aan de bovenzijde van een stuk papier te schrijven. U begint het avontuur aan de bovenzijde van het papier met uw gezicht naar beneden (naar het zuiden dus). De eerste beschrijving van een 'ruimte' in het programma, die u te zien krijgt, ziet er als volgt uit:



```
U staat voor de ingang van een
eeuwen oud, naargeestig kasteel.
U bent aan de noord-zijde van
het kasteel en als U zuidwaards
kijkt, ziet U, dat de poort in
de bouwvallige muur open is en
niet bewaakt wordt.

Wat wilt U nu doen?
```

De laatste regel , 'wat wilt u nu doen?', volgt iedere beschrijving van de omgeving (op een aantal vragen na, zoals 'wilt u de drank drinken?', waar als antwoord 'J' of 'N' op verwacht wordt), en kan op één van volgende zeven wijzen beantwoord worden:

Toets in:	En de computer zal het vertalen als:
N	Ga naar het noorden
Z	Ga naar het zuiden
E	Ga naar het oosten
W	Ga naar het westen
VE	Vecht

de kamer en de vraag opnieuw worden geprint. (U kunt ook altijd 'Q' gebruiken om het spel te verlaten.) Avonturenspelen, die door mensen gespeeld worden (vaak met een scheidsrechter, die Meester van het Spel) volgen vaak bepaalde regels, die voorschrijven, dat de kracht en de persoonlijkheid van een speler aan het begin van het spel worden vastgesteld door het gooien van een dobbelsteen, en het uitlezen van het karakter van de speler uit een tabel (zoals 'Schranderheid of Kracht). Als twee persoonlijkheden elkaar in het spel tegenkomen (avontuurspelers noemen dit Interactie, het hoogtepunt van het spel), dan wordt er nog een aantal malen met de dobbelsteen gegooid, en een aantal tabellen wordt geraadpleegd, om te zien, hoe de afloop van het gevecht zal zijn.

Om het systeem eenvoudig te houden, heb ik mij vastgehouden aan de geaccepteerde conventies, maar er wel voor gezorgd, dat de Spectrum al het werk doet, zowel het genereren van uw persoonlijkheid als dat van de monsters en het oplossen van problemen, die u onderweg tegenkomt.
Het verhaal achter het programma en het doel van het spel zijn eenvoudig. U als een welbekende bent de ligging te weten gekomen van een eeuwenoude ruïne van een kasteel. Omdat u verschrikkelijk nieuwsgierig bent, gaat u de hoofdpoort binnen in de noordelijke muur, waarvan u ontdekt heeft, dat hij openstaat. Als u het kasteel echter door de poort naar binnen bent gegaan, dan blijkt opeens, dat de poort gesloten wordt, en u niet meer langs deze weg het kasteel kunt verlaten. U moet het kasteel doorzoeken om een andere uitgang te vinden. Terwijl u in de grote hal van het kasteel staat, die rijkelijk met tapijten behangen is, herinnert u zich

verschrikkingen gemaakt om zijn huis tegen indringers te beschermen. Behalve de noordelijke ingang, die nu door magische krachten voor altijd gesloten is, is er maar één weg naar buiten. Dit is via de, zich onder het kasteel bevindende, Zwarte Lagune die door een doodgewone deur vanuit het kasteel bereikt kan worden. Er wordt echter gefluisterd, dat de tovenaar zijn kasteel nog een extra bescherming heeft meegegeven, door een onheilspellend monster te scheppen, waarvan weinig bekend is, behalve zijn naam, De Bewaker van de Zwarte Lagune.
Uw persoonlijkheid in het spel wordt bepaald door drie factoren,: magische kracht, fysieke kracht en wijsheid. Ieder wezen, dat gedurende een eeuwigheid door de vervallen kamers van het kasteel zwerft, heeft deze zelfde attributen. De huidige staat van uw attributen verschijnt als volgt op uw scherm:



Als u een monster tegenkomt, en u besluit de strijd met hem aan te binden dan kunt u slechts met één attribuut tegelijk vechten. U kunt echter zelf kiezen met welke. In het algemeen is het zo, dat u de meeste kans heeft om te winnen, als het verschil in een bepaalde attribuut van u en dat van het monster erg groot is. Bijvoorbeeld, als uw MAGIE 20 is, en dat van het monster slechts 4, en het verschil tussen andere kwaliteiten is minder groot, dan kunt u het beste het monster met uw MAGIE proberen te verslaan.



Vreselijk! U bent de schuil-
plaats van de afschuwelijke
Breinvreter binnengegaan.
U ziet in een oogopslag dat
zijn kracht 12 is, zijn magie
is 18, en zijn wijsheid is 17

Wat wilt U nu doen?
Nee!! U zult moeten vechten

Met welke eigenschap wilt U
vechten?

Het verschil is 1
en hij is in het voordeel

Dit gevecht kost [redacted] punten

U heeft hem verslagen!
[redacted]

Dit is de meest indrukwekkende
kamer in het kasteel, de Grote
Hal, met een massief houten pla-
fond. U kunt de hal verlaten
door de dubbele deuren naar het
noorden of door de deuren in de
oostelijke muur achter U, van-
waar U muziek hoort komen.
Door de ramen in de westelijke
muur ziet U de Golvende Tuinen
en daar achter de ramen van een
kamer, waarin U schilderijen
ziet hangen.

In de muur voor U, ziet U een
deur van een kleine safe en daar
voor ligt een sleutel

Wilt U de safe openen (J of N)?
Nu Wijsheid:
    U: 11 Bewaker: 19
Het verschil is 8
en de Bewaker ligt op kop

```
Aan het einde van deze beslis-
sende strijd heeft U :
    MAGIE: 44
    KRACHT: 1
    WIJSHEID: 2

Het is U gelukt, Oh held van
deze duistere en gevaarlijke
tijden. Ik sla U [redacted] ROB
```

Raak niet in paniek, als het nu al te ingewikkeld dreigt te worden. Het programma doet bijna al het werk voor u, en leidt u door het spel, waarbij verkeerde invoer door het toetsenbord verworpen of genegeerd wordt.
Ieder gevecht geeft u de mogelijkheid om de attribuut, waarmee u vecht te vergroten, of om met die attribuut te betalen, als u gewond raakt tijdens het gevecht. Als u al uw punten verloren heeft, is het spel ten einde. U heeft zo veel mogelijk punten nodig voor het Laatste Gevecht. Als u weet, waar u de Zwarte Lagunge kunt vinden (om het u moeilijk te maken, de Lagune kan zich op twee verschillende plaatsen bevinden, dus uw kaart is niet altijd volledig betrouwbaar!), dan moet u drie gevechten leveren tegen de Bewaker van de Zwarte Lagune, ieder met een ander attribuut. Sommige van de kisten, die in het huis rondzwerven, bevatten 'drakengoud', en het is nuttig om dit te verzamelen, omdat het gebruikt kan worden om nieuwe attribuutpunten te kopen nadat u de Bewaker heeft gevonden, maar voordat u met hem gevochten heeft. U moet uit deze laatste gevechten komen met en totaal van ten minste 10 attribuutpunten om het kasteel te kunnen verlaten. Ieder gevecht met een monster werkt als volgt: Nadat u het attribuut uitgekozen heeft, waarmee u de strijd wilt aanvangen, geeft de computer het verschil aan tussen de waarden van uw attribuut en dat van de tegenstander.

print de computer (regel 5240) 'Het verschil is 13', gevolgd door 'en u heeft het voordeel'. Nu wordt er, net zoals bij en dobbelspel, gegooid met een zes-zijdige dobbelsteen, en het getal, dat bovenkomt, is het 'verlies' voor deze ronde van het gevecht. Er wordt een willekeurig getal gegenereerd tussen 1 en het verschil tussen de twee scores. Dit wordt dan vergeleken met uw score en die van het monster voor het gekozen attribuut. De winnaar is degene, die een aantal attribuutpunten heeft, dat het dichtst bij het gegenereerde getal ligt. U kunt dit eventueel veranderen, zodat de winnaar degene is, wiens score het verst van het gevonden getal vandaan ligt. Als het verschil tussen beide attribuutscores gelijk is aan 1, dan wordt het gegenereerde getal altijd 1, zodat de zwakste speler altijd wint. (In deze regel zit een aanwijzing verscholen, die u in staat kan stellen om meer dan een evenredig gedeelte van de gevechten te winnen. Leest u de voorgaande regels daarom nog eens goed door!). Ik laat het aan u over om uit te rekenen, wie er wint, als de attribuutpunten gelijk aan elkaar zijn. Nu zijn winnen en verliezen in dit spelletje twee totaal verschillende zaken. De kosten en winsten van een gevecht zijn afhankelijk van het door u gekozen attribuut, en uw huidge score voor dat attribuut. U kiest het attribuut, waarmee u wilt vechten, door de eerste letter ervan in te tikken (M voor MAGIE, etc.) en de computer gebruikt deze informatie in de subroutine vanaf regel 5340 om de straf voor verlies, of de winst bij een overwinning vast te stellen. Als u bijvoorbeeld met Magie gevochten heeft, en u verliest, dan verliest u het totale aantal punten, dat met de dobbelsteen gegooid werd, als u ten minste zoveel punten bezit. Als u dus twee punten heeft, en er wordt een drie gegooid in een gevecht, dat u verliest, dan verliest u geen punten.

waarbij u Kracht heeft uitgekozen als attribuut, dan verliest u ook het aantal punten, dat met de dobbelsteen gegooid werd, als u zoveel punten ten minste bezit. Als u verliest, terwijl u uw tegenstander heeft uitgedaagd met uw wijsheid, dan verliest u slechts de helft van het gegooide aantal punten, naar beneden afgerond op het eerste gehele getal. Als uw straf dus 1 zou zijn, en u verliest, dan kost dit u niets, omdat de helft van 1 naar beneden afgerond wordt op nul.
Winnen is natuurlijk veel leuker. U kunt meer winnen dan verliezen, als u bij een gevecht gebruik maakt van uw Magie. Als u met Magie wint dan krijgt u het dubbele van de gegooide punten. Met kracht wint u geen punten. U kunt dus beter niet proberen met Kracht te vechten, omdat u vaak geen partij bent voor de zwaar gespierde monsters. Winnen met Wijsheid geeft u als beloning de helft van de gegooide punten erbij (omdat het verliezen met Wijsheid ook maar de helft van de punten kost).
Het programma volgt een bepaalde, vooropgezette structuur, het is dus relatief eenvoudig om een gedeelte van het programma te vinden, dat u wilt veranderen. (Als u de kaart heeft gevonden, die in het programma gecodeerd is, dan wilt u deze wellicht wijzigen. De manier waarop vertel ik u. U kunt dan een meer uitgebreide kaart gebruiken, waar nog steeds dezelfde monsters zijn en dezelfde regels gelden. Als u dit voor elkaar heeft, dan wilt u waarschijnlijk de basis van het programma gebruiken om een eigen gemaakte kaart te gebruiken.)
De structuur van het programma ziet er als volgt uit:

20 – 170	Laat zien, in welke kamer u bent, waar de uitgangen zijn, etc. Dit is ook de hoofdprogramma-lus.
2000 – 2050	Beschrijvingen van de monsters
4000 – 4950	Uiteindelijke confrontatie (de Zwarte La-

5000 – 5620 Acties. U voert uw richting in, of u besluit te vluchten of te vechten. Alle gevechten worden ook in dit gedeelte van het programma uitgevoerd, en straffen en beloningen worden hier uitgedeeld.

7000 – 7500 Voorwerpen. Dit gedeelte van het programma bevat de vier kisten, de fles met drank etc. Deze routine wordt willekeurig aangeroepen vanaf het einde van de actie-subroutine.

8000 – 8140 Beschrijvingen van de kamers, en de aanroep van de monster-subroutines

9000 – 9670 Vertraging/geluid subroutine (die verschillende malen vanuit het programma aangeroepen wordt, om de kleur toe te voegen, en om u de kans te geven om te lezen, wat er op het scherm staat, voordat dit schoongemaakt wordt)

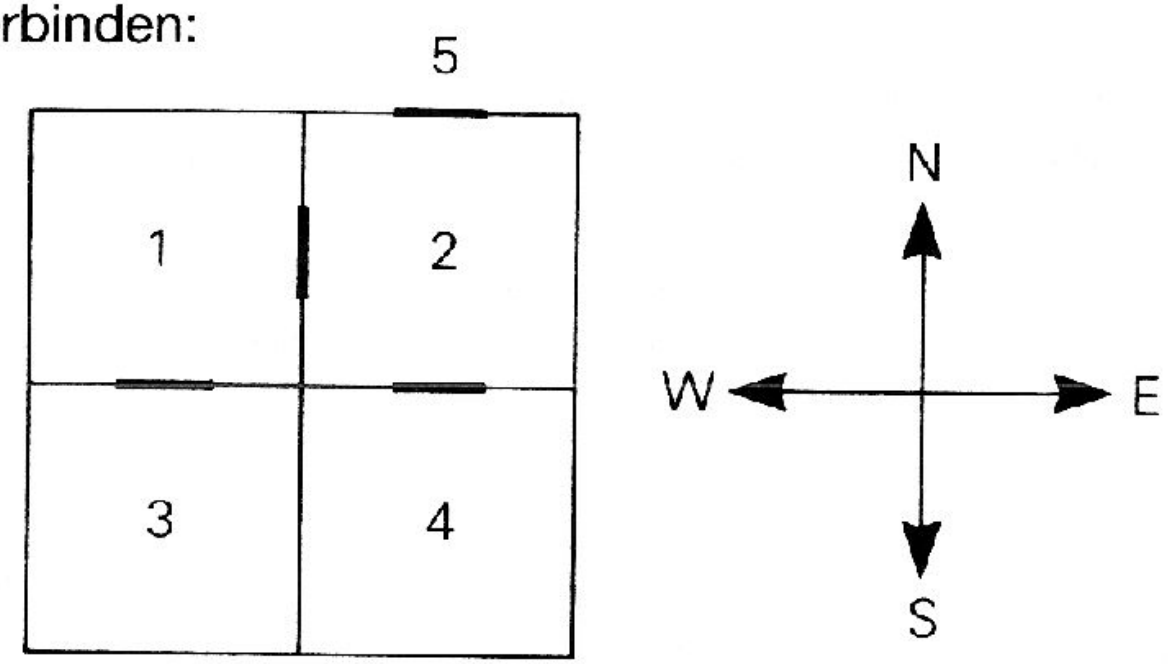
Het meest belangrijke gedeelte van het programma, en het middelpunt van dit avonturenprogramma, is het mechanisme, dat de hele wereld bijeen houdt. Hier worden de relaties tussen de verschillende onderdelen van het systeem bepaald, en worden potentiële bewegingen van de ene naar de andere kamer gecontroleerd, om vast te stellen, of ze legaal zijn. Vanuit twee van de kamers in het spel heeft u uitzicht over de 'Golvende Tuinen', maar u kunt niet in de tuin komen, om zo van de ene kamer naar de andere te lopen. (De tuin is aan het spel toegevoegd, om ten eerste het spel aantrekkelijker te maken, en in de tweede plaats de kaart-tekenaars een hint te geven over de lokatie van bepaalde kamers.)

Ik ga nu uitleggen hoe het programma de verschillende elementen van het systeem bijeenhoudt, maar ik wil

rassingen te ontdekken. Kijk naar de regels 9570 tot 9680. De DATA-opdrachten, die hier staan, bevatten een hoop bruikbare informatie (en er zijn dan ook een aantal dummy-data-opdrachten toegevoegd, om te voorkomen, dat u alleen met deze informatie gewapend zult proberen, uw kaart te tekenen.) Kijk naar de vierde DATA-opdracht (regel 9600). De eerste twee elementen van de string zijn het nummer van de kamer, die zich ten noorden van de kamer bevind, die in deze DATA-opdracht bewaard wordt. Er is ten noorden van deze kamer geen andere kamer, dus er is ook geen uitgang aan de noordzijde van deze kamer. De volgende twee elementen laten zien, dat ten zuiden kamer nummer zeven ligt, de volgende twee betekenen een oost-uitgang naar kamer twee in het kasteel, en de volgende twee geven een westelijke deur naar kamer vijf. De twee nullen daarachter worden gebruikt om de informatie op te slaan over de inhoud van de kamer.

U kunt dit systeem van DATA-opdrachten gebruiken om iedere kaart in op te slaan. Als ik in de kamer zou zijn, die door regel 9600 wordt weergegeven, en ik zou het commando geven om naar het noorden te gaan, dan weet de computer, dat hij alleen maar naar de eerste twee elementen uit de string hoeft te kijken, die deze kamer beschrijven. Als deze eerste twee elementen beide nul zijn, dan weet de computer, dat er geen doorgang naar het noorden bestaat, en vraagt hij u om een andere richting. Als er wel een getal gevonden wordt, dan kan de variabele, die gebruikt wordt om het kamer-nummer in op te slaan (in dit programma Z) gevuld worden met de VAL van die twee elementen. Dit programma maakt uitgebreid gebruik van de zeer eenvoudige string-manipulatie-technieken in Sinclair BASIC. De attributen van de monsters worden hierbij in het array M$ opgeslagen,

kamer verslaat, en zijn MAGIE daarbij terugbrengt tot 9, dan zal zijn Magie later nog steeds 9 zijn. Als u een monster maar vaak genoeg tegenkomt, dan kunt u hem dus eenvoudig van de kaart vegen. Ook dit is een ander voorbeeld van het consistente karakter van dit avonturen-programma: alle monsters blijven dezelfde monsters, ook als u ze later in een andere kamer tegenkomt. Als u een zeer eenvoudig avontuur zou hebben, met slechts 4 kamers, dan kunt u deze als volgt onderling verbinden:



De deuren ziet u in de scheidingswanden. De kamers kunnen dan met het door ons gebruikte systeem beschreven worden met:

Kamer 1: '00030200' Geen weg naar het noorden, kamer 3 naar het zuiden, kamer twee naar het oosten, niets naar het westen
Kamer 2: '05040001' Merk op, dat ook de buitenwereld een nummer moet hebben, dat hebben we hier voor het gemak maar even 5 genoemd.
Kamer 3: '01000000' Niet veel uitgangen hier
Kamer 4: '02000000' Hier ook niet

de buitenwereld, er is verder een deur naar het zuiden en naar het westen. Waar wilt u naar toe gaan?'. Als u nu 'O' invoert, dan kijkt de computer naar het 'oost element' uit de DATA-opdracht, die kamer twee weergeeft. Dit zijn het vijfde en zesde element. Hij ziet dan, dat hier twee nullen staan. Hij weet daardoor, dat hij de opgegeven richting niet kan volgen, en signaleert dit aan de speler: 'U kunt niet door muren lopen'. Hij wacht dan op een nieuwe richting. Als de speler nu 'Z' intikt, waarmee hij bedoelt, dat hij naar het zuiden wil bewegen, dan kijkt de computer naar de 'zuid-elementen' van de string, en vindt daar de cijfers '04'. Dit betekent, dat de speler van kamer twee naar kamer vier gaat bewegen. De computer hoeft nu alleen de VAL van het gevonden getal op te slaan in de variabele, die het kamernummer van de speler bevat, en de speler is nu naar een andere kamer gegaan. Merk op, dat er in de PRINT-opdrachten geen enkele referentie gedaan wordt naar de kamernummers. Deze bestaan uitsluitend voor intern gebruik.

Ik stel voor, dat u het programma nu eerst een aantal malen uitprobeert, om er achter te komen, hoe u ermee om kunt gaan. Dit geeft u een behoorlijk goed idee van de indeling van het kasteel. De kaart kunt u het beste tekenen als u een reeks losse vierkanten op het papier zet, die u daarna vult met de namen van de kamers,die u ontdekt (zoals 'De Hoofdingang'), en daarna kleine 'gangen' tekent, die de verschillende kamers verbinden. Deze gangen kunnen boven en onder elkaar door gaan terwijl u langzamerhand de samenhang tussen de kamers ontdekt, en de kaart na enige tijd opnieuw in het 'net' overgetekend worden, als u denkt dat u een behoorlijk accurate kaart van het kasteel gemaakt hebt. Als u dit gedaan heeft, dan kunt u naar eigen wens 'door

terugkeren. Als we dan de listing bestuderen, zal ik een aantal manieren aangeven om het programma te wijzigen en te verbeteren, en u een meer algemene achtergrond van avonturenprogramma's geven.

```
   10 REM DE WRAAK VAN KASTEEL AN
OST
   20 GO SUB 9000
   30 CLS
   35 IF J$="000000" THEN GO TO 9
0
   40 PRINT INVERSE 1;N$;", Uw ei
genschappen zijn:"
   50 IF VAL (J$(1 TO 2))>0 THEN
PRINT TAB 4; INVERSE 1;"MAGIE: "
;J$(1 TO 2)
   60 IF VAL (J$(3 TO 4))>0 THEN
PRINT TAB 4; INVERSE 1;"KRACHT:
";J$(3 TO 4)
   70 IF VAL (J$(5 TO 6))>0 THEN
PRINT TAB 4; INVERSE 1;"WIJSHEID
: ";J$(5 TO 6)
   80 IF MONEY>0 THEN PRINT TAB 4
; INVERSE 1;"RIJKDOM: $";MONEY
   90 IF J$="000000" THEN PRINT "
Het avontuur is afgelopen","U be
nt aan het eind van Uw ","kracht
en. U heeft dapper en","goed gew
ochten, maar U kon de","moeilijk
heden niet overwinnen","....Vaar
wel...": STOP
  100 GO SUB PAUSE
  110 GO SUB ROOM
  120 LET M=0: IF Z>1 THEN IF RND
>.5 THEN GO SUB 7000: POKE 23692
,-1
  130 GO SUB PAUSE
  140 GO SUB ACTION
  150 GO SUB PAUSE
  170 GO TO 30
  180 REM ****************
2000 REM MONSTER SUBROUTINE
2010 IF Q=1 THEN PRINT "Er is ee
n boze tovenaar in deze","kamer.
 Zijn magische krachten ","zijn
";M$(1,2 TO 3);", zijn kracht is
 ";M$(1,4 TO 5);" en","zijn wijs
heid is ";M$(1,6 TO 7)
2020 IF Q=2 THEN PRINT "Deze kam
```

```
 Zijn magische krachten zijn ";M
$(2,2 TO 3)
2030 IF 0=3 THEN PRINT "Vreselij
k! U bent de schuil-","plaats va
n de afschuwelijke ","Breinvrete
r binnengegaan.","U ziet in een
oogopslag dat","zijn kracht ";M$
(3,4 TO 5);" is","zijn magie","is
 ";M$(3,2 TO 3);", en zijn wijsh
eid is ";M$(3,6 TO 7)
2040 IF 0=4 THEN PRINT "In de du
isternis struikelt U","ergens ov
er. Het wordt wakker","en U staa
t oog in oog met","de waanzinnig
e Bottenbreker.","De magie van d
e Bottenbreker is",M$(4,2 TO 3);
", zijn kracht is ";M$(4,4 TO 5)
;",","en zijn wijsheid is ";M$(4
,6 TO 7)
2050 IF 0=5 THEN PRINT "Nu heeft
 U problemen.": GO SUB PAUSE: PR
INT "In deze kamer huist de gevr
eesdevijand van ieder, die het
      kasteel betreedt, Graaf Lio
n vanPenCe. Met een kracht van "
;M$(5,4 TO 5);",","een wijsheid
van ";M$(5,6 TO 7);" en een magi
evan ";M$(5,2 TO 3);" is hij een
 vijand om      voorzichtig te be
naderen."
2060 GO SUB PAUSE
2070 RETURN
2080 REM ************
4000 REM EINDE SPEL
4010 PRINT "U bent terecht gekom
en in de ","slijmerige modder, d
ie de ZwarteLagune onder het kas
teel ","omgeeft. Om uit het kast
eel te","ontsnappen moet U de Be
waker vande Zwarte Lagune bevech
ten."
4020 PRINT "Het gevecht moet met
 alle ","eigenschappen gevochten
 worden, en U heeft totaal nog 1
0 punten nodig om daarna te onts
nappen."
4030 GO SUB PAUSE
4040 IF MONEY>0 THEN PRINT "U he
eft $";MONEY;" aan goud bij U"
4050 GO SUB PAUSE
4060 PRINT "Eigenschappen van d
```

```
,4 TO 5)
4090 PRINT TAB 3;"WIJSHEID:  ";M$
(6,6 TO 7)
4100 PRINT '"Uw eigenschappen:"
4110 PRINT TAB 3;"MAGIE:  ";J$(1
TO 2)
4120 PRINT TAB 3;"KRACHT:  ";J$(3
 TO 4)
4130 PRINT TAB 3;"WIJSHEID:  ";J$
(5 TO 6)
4140 GO SUB PAUSE
4150 IF MONEY<100 THEN GO TO 427
0
4160 PRINT '"U kunt Uw eigenscha
ppen ver-","beteren voor $100 pe
r punt"
4170 PRINT "Als U wilt kopen tik
 dan de","eerste letter van de e
igenschap","", die U wilt, gevol
d door het","aantal punten. Tik
'N' in, als","U niet wilt kopen.
"
4190 INPUT FLASH 1;"Eigenschap?
";E$
4200 IF E$="N" THEN GO TO 4270
4210 INPUT FLASH 1;"Bedrag?  ";AM
4220 IF MONEY-AM<1 OR AM<100 THE
N GO TO 4210
4230 IF E$="M" THEN LET J$(1 TO
2)=STR$ (VAL (J$(1 TO 2))+INT (A
M/100))
4240 IF E$="K" THEN LET J$(3 TO
4)=STR$ (VAL (J$(3 TO 4))+INT (A
M/100))
4250 IF E$="W" THEN LET J$(5 TO
6)=STR$ (VAL (J$(5 TO 6))+INT (A
M/100))
4252 LET MONEY=MONEY-AM
4255 PRINT "MAGIE:  ";J$(1 TO 2);
"  KRACHT:  ";J$(3 TO 4),"WIJSHEID
:  ";J$(5 TO 6);"  GOUD $";MONEY
4260 IF MONEY>99 THEN GO TO 4190
4270 CLS : PRINT "En nu het Laat
ste Gevecht...": GO SUB PAUSE
4275 INPUT FLASH 1; BRIGHT 1; PA
PER 2;"Toets ENTER als U dapper
genoeg bent om te vechten  ",A$;
 GO SUB PAUSE: CLS
4280 PRINT '"Eerst, Magie:"
4290 LET MH=VAL (J$(1 TO 2)): LE
```

```
4330 IF MH>MG THEN PRINT "in Uw
voordeel"
4340 IF MG>MH THEN PRINT "en de
Bewaker heeft de leiding"
4350 GO SUB PAUSE: LET COST=INT
(RND*((MH+MG)/2))+1
4360 PRINT '"Deze ronde kost ";C
OST;" punten": GO SUB PAUSE
4370 LET RESULT=INT (RND*DIFF)+1
4380 LET RESHU=ABS (RESULT-MH)
4390 LET RESGA=ABS (RESULT-MG)
4400 IF RESHU>RESGA THEN PRINT "
En U wint ";COST;" magie punten"
: LET MH=MH+COST: LET J$(1 TO 2)
=STR$ (MH): GO TO 4480
4410 PRINT "En U verliest ";COST
;" magiepunten": LET MH=MH-COST:
 LET J$(1 TO 2)="00": IF MH>0 TH
EN LET J$(1 TO 2)=STR$ (MH)
4480 INPUT FLASH 1; PAPER 7; INK
 2;"Toets ENTER als U klaar bent
om   deze beslissende strijd voor
t    te zetten",,A$: GO SUB PAUSE
: CLS
4485 PRINT "Nu kracht:"
4490 LET MH=VAL (J$(3 TO 4)): LE
T MG=VAL (M$(6,4 TO 5))
4500 PRINT TAB 3;"U:  ";MH;"  Bew
aker:  ";MG
4510 LET DIFF=ABS (MH-MG)
4520 PRINT "Het veschil is ";DIF
F
4530 IF MH>MG THEN PRINT "met U
op kop"
4540 IF MG>MH THEN PRINT "en de
Bewaker ligt op kop"
4550 GO SUB PAUSE: LET COST=INT
(RND*((MH+MG)/2))+1
4560 PRINT '"Deze ronde kost ";C
OST;" punten": GO SUB PAUSE
4570 LET RESULT=INT (RND*DIFF)+1
4580 LET RESHU=ABS (RESULT-MH)
4590 LET RESGA=ABS (RESULT-MG)
4600 IF RESHU>RESGA THEN PRINT "
En U wint ";COST;" kracht punten
": LET MH=MH+COST: LET J$(3 TO 4
)=STR$ (MH): GO TO 4660
4610 PRINT "En U verliest ";COST
;" kracht punten": LET J$(3 TO 4
)="00": IF MH>0 THEN LET MH=MH-C
OST: LET J$(3 TO 4)=STR$ (MH)
4670 LET RESULT=INT (RND*DIFF)+1
```

```
",A$: GO SUB PAUSE: CLS
4685 PRINT "Nu Wijsheid:"
4690 LET MH=VAL (J$(5 TO 6)): LE
T MG=VAL (M$(6,6 TO 7))
4700 PRINT TAB 3;"U:  ";MH;"  Bew
aker:  ";MG
4710 LET DIFF=ABS (MH-MG)
4720 PRINT "Het veschil is ";DIF
F
4730 IF MH>MG THEN PRINT "en U l
igt op kop"
4740 IF MG>MH THEN PRINT "en de
Bewaker ligt op kop"
4750 GO SUB PAUSE: LET COST=INT
(RND*(MH+MG))+1
4760 PRINT '"Deze ronde kost ";C
OST;" punten": GO SUB PAUSE
4770 LET RESULT=INT (RND*DIFF)+1
4780 LET RESHU=ABS (RESULT-MH)
4790 LET RESGA=ABS (RESULT-MG)
4800 IF RESHU>RESGA THEN PRINT "
En U wint ";COST;" Wijsheid punt
en": LET MH=MH+COST: LET J$(5 TO
 6)=STR$ (MH): GO TO 4850
4810 PRINT "En U verliest ";COST
;" Wijsheid punten": LET MH=MH-C
OST: LET J$(5 TO 6)="00": IF MH>
0 THEN LET J$(5 TO 6)=STR$ (MH)
4850 LET A=VAL (J$(1 TO 2))
4860 LET B=VAL (J$(3 TO 4))
4870 LET C=VAL (J$(5 TO 6))
4880 GO SUB PAUSE
4890 PRINT '"Aan het einde van d
eze beslis-","sende strijd heeft
 U :"
4900 PRINT TAB 3;"MAGIE:  ";A
4910 PRINT TAB 3;"KRACHT:  ";B
4920 PRINT TAB 3;"WIJSHEID:  ";C
4930 IF A+B+C>9 THEN PRINT ' FLA
SH 1; BRIGHT 1; PAPER 7; INK 2;"
Het is U gelukt, Oh held van",'"d
eze duistere en gevaarlijke ","t
ijden. Ik sla U "; INVERSE 1;"Ri
dder"; INVERSE 0;" ";N$: GO SUB
PAUSE: STOP
4940 PRINT "Het is voorbij. U ha
d tenminste 10 punten totaal nod
ig om aan  de greep van de Bewa
ker te ont- snappen."
4950 PRINT '"U heeft dapper gevo
chten, maar wordt nu helaas doo
```

```
5002 PAUSE 1000
5005 LET D=4: IF B$(Z,9)="0" THE
N LET D=1
5010 PRINT '"Wat wilt U nu doen?
5015 INPUT Z$: IF Z$="0" THEN ST
OP
5020 IF Z$="" THEN CLS : LET Z$=
"#"
5022 IF D=4 AND Z$(1)<>"U" THEN
LET D=0: GO TO 5160
5025 IF Z$(1)="U" THEN GO TO 513
0
5030 IF Z$="N" AND B$(Z,1 TO 2)=
"00" THEN PRINT ,"GEEN UITGANG":
GO TO 5010
5040 IF Z$="Z" AND B$(Z,3 TO 4)=
"00" THEN PRINT "ER IS GEEN DEUR
IN DIE RICHTING": GO TO 5010
5050 IF Z$="O" AND B$(Z,5 TO 6)=
"00" THEN PRINT "DAT IS NIET MOG
LIJK": GO TO 5010
5060 IF Z$="W" AND B$(Z,7 TO 8)=
"00" THEN PRINT "U KUNT NIET DOO
R MUREN LOPEN": GO TO 5010
5070 IF Z$="N" THEN LET Z=VAL (B
$(Z,1 TO 2)): RETURN
5080 IF Z$="Z" THEN LET Z=VAL (B
$(Z,3 TO 4)): RETURN
5090 IF Z$="O" THEN LET Z=VAL (B
$(Z,5 TO 6)): RETURN
5100 IF Z$="W" THEN LET Z=VAL (B
$(Z,7 TO 8)): RETURN
5110 IF Z$(1)<>"V" THEN RETURN
5130 IF B$(Z,9)="0" THEN PRINT "
ER IS NIETS OM TEGEN TE VECHTEN"
: GO TO 5010
5140 IF Z$="VL" THEN LET D=INT (
RND*2)
5150 IF D=1 THEN PRINT "Welke ri
chting?": GO TO 5015
5160 IF d=0 THEN PRINT "Nee!! U
vult moeten vechten"
5165 BEEP .1,5: BEEP .05,0: BEEP
1,5
5170 PRINT '"Met welke eigensch
ap wilt U ","vechten?"
5180 LET Q=VAL (B$(Z,9))
5190 INPUT Z$: IF Z$<>"M" AND Z$
<>"K" AND Z$<>"W" THEN GO TO 519
0
5200 IF Z$="M" THEN LET HUM=VAL
```

```
 TO 5))
5220 IF Z$="W" THEN LET HUM=VAL
(J$(5 TO 6)): LET MON=VAL (M$(0,
 TO 7))
5230 LET DIFF=ABS (HUM-MON)
5240 PRINT ,, FLASH 1; BRIGHT 1;
INK 2; PAPER 7;"Het verschil is
";DIFF
5250 IF HUM>MON THEN PRINT FLASH
1; BRIGHT 1; PAPER 2; INK 7;"U
bent in het voordeel!"
5260 IF HUM<MON THEN PRINT FLASH
1; BRIGHT 1; PAPER 2; INK 7;"U
is in het voordeel!"
5270 LET COST=INT (RND*6)+1
5280 PRINT '' FLASH 1; BRIGHT 1;
PAPER 3; INK 9;"Dit gevecht kos
t "; INVERSE 1;COST; INVERSE 0;"
punten"

5290 LET RESULT=INT (RND*DIFF)+1
5300 GO SUB PAUSE
5310 LET RESHU=ABS (RESULT-HUM)
5320 LET RESMO=ABS (RESULT-MON)
5330 IF RESHU<RESMO THEN GO TO 5
390
5335 FOR I=-10 TO -30 STEP -.5:
BEEP .01,I: NEXT I
5340 PRINT '' FLASH 1; PAPER 2;
INK 6;"U bent verslagen!"
5350 IF Z$="M" THEN LET M$(0,2 T
O 3)=STR$ (VAL (M$(0,2 TO 3))+2*
COST): IF VAL (J$(1 TO 2))>=COST
 THEN LET J$(1 TO 2)=STR$ (VAL (
J$(1 TO 2))-COST)
5360 IF Z$="K" AND VAL (J$(3 TO
4))>=COST THEN LET J$(3 TO 4)=ST
R$ (VAL (J$(3 TO 4))-COST)
5370 IF Z$="W" THEN LET M$(0,6 T
O )=STR$ (VAL (M$(0,6 TO 7))+INT
 (COST/2)): IF VAL (J$(5 TO 6))>
=COST THEN LET J$(5 TO 6)=STR$ (
VAL (J$(5 TO 6))-COST)
5380 GO TO 5440
5390 REM MENS WINT
5395 FOR I=10 TO 30 STEP .5: BEE
P .01,I: NEXT I
5400 PRINT '' FLASH 1; BRIGHT 1;
 INK 6; PAPER 2;"U heeft hem ver
slagen!"
5410 IF Z$="M" THEN LET J$(1 TO
```

```
0 5)=STR$ (VAL (M$(Q,4 TO 5))-CO
ST)
5430 IF Z$="W" THEN LET J$(5 TO
6)=STR$ (VAL (J$(5 TO 6))+INT (C
OST/2)): IF VAL (M$(Q,6 TO 7))>=
COST THEN LET M$(Q,6 TO 7)=STR$
(VAL (M$(Q,6 TO 7))-COST)
5440 GO SUB PAUSE
5450 PRINT "Na dit gevecht zijn
Uw eigen-   schappen:"
5460 PRINT TAB 3;"MAGIE:";J$(1 T
O 2)
5470 PRINT TAB 3;"KRACHT:";J$(3
TO 4)
5480 PRINT TAB 3;"WIJSHEID:";J$(
5 TO 6)
5490 PRINT '"En die van de ";
5500 IF Q=1 THEN PRINT "tovenaar
"
5510 IF Q=2 THEN PRINT "Angstbre
nger"
5520 IF Q=3 THEN PRINT "Breinvre
ter"
5530 IF Q=4 THEN PRINT "Bottenbr
eker"
5540 IF Q=5 THEN PRINT "Graaf"
5550 PRINT "zijn:"
5560 PRINT TAB 3;"MAGIE:  ";M$(Q,
2 TO 3)
5570 PRINT TAB 3;"KRACHT:  ";M$(Q
,4 TO 5)
5580 PRINT TAB 3;"WIJSHEID:  ";M$
(Q,6 TO 7)
5590 GO SUB PAUSE
5600 PRINT "Toets ENTER om door
te gaan"
5610 INPUT T$: GO SUB PAUSE: CLS
5620 LET K=2+INT (RND*8)
5630 IF K=Z THEN GO TO 5620
5640 LET B$(K,9)=B$(Z,9)
5650 LET B$(Z,9)="0"
5660 IF RND>.5 THEN RETURN
7000 REM INHOUD
7005 IF B$(Z,9)<>"0" THEN LET M=
1: RETURN : REM GEEN'INHOUD'ALS
MONSTER BEVOCHTEN WORDT
7010 PRINT : GO SUB 7000+100*INT
 (RND*5+1)
7020 GO SUB PAUSE
7030 RETURN
7100 LET CHEST=CHEST+1: IF CHEST
```

```
7120 PRINT '"Wilt U de kist open
en (J of N)?"
7130 GO SUB 7600
7150 IF Z$="N" THEN RETURN
7160 LET J=INT (RND*3): GO SUB P
AUSE
7170 IF J=0 THEN LET CASH=100+IN
T (RND*300): PRINT "De kist bev
t Drakengoud ter      waarde van $
";CASH;"!": BEEP .5,0: BEEP .25,
12: BEEP .125,7: BEEP .625,0: LE
T MONEY=MONEY+CASH: RETURN
7180 IF J=1 THEN PRINT "Een dwer
g springt uit de kist    en steek
t U neer!": BEEP .7,-30: LET LOS
S=INT (RND*6)+1: IF VAL (J$(3 TO
 4))-LOSS>=0 THEN LET J$(3 TO 4)
=STR$ (VAL (J$(3 TO 4))-LOSS): R
ETURN
7190 IF J=2 THEN PRINT "Uit de k
ist komt een vreemde      rook, di
e U slaperig maakt en Uwmagische
 krachten wegzuigt": BEEP .3,5:
BEEP .3,7: BEEP .3,5: BEEP .3,0:
 BEEP 1.5,5: LET LOSS=INT (RND*6
)+1: IF VAL (J$(1 TO 2))-LOSS>=0
 THEN LET J$(1 TO 2)=STR$ (VAL (
J$(1 TO 2))-LOSS): RETURN
7200 IF POTION=1 THEN GO TO 7010
7210 LET POTION=1
7220 BEEP .5,60: BEEP .3,-30: PR
INT "U ziet een kleine fles
      gegraveerd";
7230 PRINT " met vreemde tekens.
"
7240 PRINT "Wilt U het flesje le
eg drinken  (J of N)?"
7250 GO SUB 7600
7260 IF Z$="N" THEN RETURN
7265 FOR X=30 TO 0 STEP -1: BEEP
 X/400,X: NEXT X
7270 GO SUB PAUSE
7280 IF RND>.6 THEN PRINT "De in
houd van het flesje heeft  Uw wi
jsheid vergroot": FOR I=1 TO 3:
BEEP .3,12*I: NEXT I: LET J$(5 T
O 6)=STR$ (VAL (J$(5 TO 6))=INT
(RND*6+1)): RETURN
7290 PRINT "De inhoud van het fl
esje heeft Uverzwakt en U valt i
n slaap.": BEEP 4,-30: GO SUB PA
```

```
$(3 TO 4))-LOSS): RETURN
7300 IF SCROLL=1 THEN RETURN
7310 LET SCROLL=1
7320 PRINT "U ziet een rol perka
ment.          Wilt U de rol lezen
(J of N)?"
7325 GO SUB 7600: IF Z$="N" THEN
 RETURN
7330 IF RND>.5 THEN PRINT "U kun
t de taal niet lezen.": GO SUB P
AUSE: RETURN
7335 PRINT "De rol beschrijft ee
n magische spreuk. Wilt U de sp
reuk lezen (J of N)?"
7340 GO SUB 7600: GO SUB PAUSE:
IF Z$="N" THEN RETURN
7350 IF RND>.5 THEN PRINT FLASH
1; BRIGHT 1; INK 7; PAPER 2;"Het
 was een goede spreuk": GO SUB P
AUSE: LET J$(1 TO 2)=STR$ (VAL (
J$(1 TO 2))+INT (RND*6+1)): RETU
RN
7360 PRINT PAPER 2; INK 7; FLASH
 1; BRIGHT 1;"Het was een kwaada
rdige          spreuk!!!",: GO SU
B PAUSE: IF VAL (J$(3 TO 4))>5 T
HEN LET J$(3 TO 4)=STR$ (VAL (J$
(3 TO 4))-INT (RND*6+1)): RETURN

7400 IF SAFE=1 THEN GO TO 7010
7405 LET SAFE=1
7410 PRINT "In de muur voor U, z
iet U een   deur van een kleine
safe en daarvoor ligt een sleute
l"
7415 PRINT '"Wilt U de safe open
en (J of N)?"
7420 GO SUB 7600
7425 IF Z$="N" THEN RETURN
7430 IF RND>.3 THEN GO TO 7460
7435 GO SUB PAUSE: PRINT '"Uit d
e safe komt een vampier,  die m
et een kreet zijn tanden inUw ke
el laat zinken!!!"
7440 GO SUB PAUSE
7445 PRINT "U vecht wanhopig ...
": GO SUB PAUSE: PRINT "...maar
U ziet kans de vampier  te doden
."
7450 IF VAL (J$(3 TO 4))>5 THEN
LET J$(3 TO 4)=STR$ (VAL (J$(3 T
```

```
U voelt U opgeknapt en          gen
ezen."
7480 LET J$(3 TO 4)=STR$ (VAL (J
$(3 TO 4))+INT (RND*6+1))
7490 RETURN
7500 GO TO 7100
7600 LET Z$=INKEY$
7605 BEEP .05,INT (40+(RND*2))
7610 IF Z$<>"N" AND Z$<>"J" THEN
 GO TO 7600
7620 BEEP .1,1: BEEP .5,6: BEEP
.1,1: RETURN
7630 REM **************
8000 REM INHOUD VAN KAMERS
8010 IF Z=1 THEN PRINT "U staat
voor de ingang van een  eeuwen o
ud, naargeestig kasteel.U bent a
an de noord-zijde van   het kast
eel en als U zuidwaards kijkt, z
iet U, dat de poort in  de bouwv
allige muur open is en  niet bew
aakt wordt."
8020 IF Z=2 THEN PRINT "U bent i
n de ingangshal, die    rijkelijk
 bekleed is met         kostbare
 tapijten. U ziet deurenaan de z
uid- en oostzijde en er is een o
pen poort naar het      westen."

8030 IF Z=3 THEN PRINT "Dit is a
lechts een voorraad-    kamer. Er
 is maar 1 uitgang, dieterug le
idt naar het westen."
8040 IF Z=4 THEN PRINT "Dit is e
en kleine kamer, waarin U een be
werkt beeld van de maan-godin op
 een voetstuk in de     noord-oos
t hoek ziet staan. Er   zijn deu
ren naar het zuiden, hetwesten e
n het oosten."
8050 IF Z=5 THEN PRINT "U bent n
u in de Zaal der        Fluisteri
ngen, waarvan de muren  met hout
en panelen zijn bekleed.De zaal
lijkt gevuld met zachte geluiden
 en er zijn uitgangen   naar het
 oosten en naar het     zuiden,
vanwaar de geur van     zwavel e
n een onaards gezang    U tegemoe
t komen."
```

```
er ruikt sterk naar      brandend
 zwavel en U hoort de    echo van
 oude spreuken.          U kunt n
aar het noorden, het     zuiden o
 het oosten."
0070 IF Z=7 THEN PRINT "U bent i
n een rustige, vredige   hal ter
echt gekomen. Dit is de  schilde
rijen-galleri van het    kasteel.
 U ziet een groot schil-derij w
an de legendarische      Bewaker
van de Zwarte Lagune,    links v
an het raam in de ooste- lijke mu
ur. Door het raam ziet Ude verva
llen ramen van de Grote  Zaal, a
an de overzijde van de   Golvende
 Tuinen.                 U kunt d
e galleri verlaten doordeuren n
aar het noorden en het   westen."
0080 IF Z=8 THEN PRINT "Dit is d
e meest indrukwekkende   kamer in
 het kasteel, de Grote   Hal, met
 een massief houten pla-fond. U
kunt de hal verlaten     door de
dubbele deuren naar het noorden
of door de deuren in de oosteli j
ke muur achter U, van-   waar U m
uziek hoort komen.       Door de
ramen in de westelijke   muur zie
t U de Golvende Tuinen   en daar
achter de ramen van een  kamer, w
aarin U schilderijen     ziet han
gen."
0090 IF Z=9 THEN PRINT "U hoort
het geluid van een       strijkkw
artet en U begrijpt dat U nu in
de muziekkamer bent.     Deze kam
er heeft deuren naar hetwesten e
n naar het zuiden."
0100 IF Z=10 THEN PRINT "U bent
in de Gang der Gedachten,waarvan
 de stilte bedriegelijk  kan zij
n. De gang is vochtig en koud. E
r is een uitgang naar hetnoorden
.": IF B$(10,3 TO 4)="12" THEN P
RINT "Er is een deur naar het zu
iden."
0110 IF Z=11 THEN PRINT "Dit is
de Brug der Zuchten, waarhet don
ker en vochtig is.       Volgens
 de legende kan hier somsde Gews
```

```
 naar het          zuiden."
3120 PRINT : IF Z=12 THEN GO TO
4000
3125 IF Z<>1 THEN IF B$(Z,9)="0"
 AND RND>.55 THEN LET B$(Z,9)=ST
R$ (INT (RND*6)): REM VERWIJDERE
N VOOR MINDER MONSTERS
3130 IF B$(Z,9)<>"0" THEN LET Q=
VAL (B$(Z,9)): GO SUB 2000
3140 RETURN
3150 REM ***************
9000 REM INITIALISATIE
9010 RANDOMIZE
9020 LET Z=1
9030 LET PAUSE=9900
9040 LET ROOM=8000
9050 LET ACTION=5000
9060 LET MONEY=0
9070 LET CHEST=0
9080 LET POTION=0
9090 LET SCROLL=0
9100 LET SAFE=0
9110 DIM B$(12,10)
9120 DIM M$(6,7): DIM J$(6)
9130 POKE 23692,-1: POKE 23658,8
9140 INPUT "Wat is Uw voornaam?
";N$
9150 PAPER 1: INK 7: BORDER 1: C
LS
9160 PRINT "Welkom, ";N$: PRINT
"Even geduld..."
9320 REM VUL KAMERS
9330 FOR T=1 TO 12
9340 READ T$
9350 LET B$(T)=T$
9360 BEEP .1,INT T/2: NEXT T
9365 LET B$((10+INT (RND*2)),3 T
O 4)="12"
9370 REM VERDEEL EN
        MAAK MONSTERS
9380 LET M$(1,1)=STR$ (INT (RND*
6)+1)
9390 FOR A=2 TO 5
9400 BEEP .1,A*5: LET M$(A,1)=ST
R$ (INT (RND*6)+1)
9440 LET B$(VAL (M$(A,1)),9)=M$(
A,1)
9450 NEXT A: LET B$(1,9)="0"
9460 REM EIGENSCHAPPEN
9470 FOR A=1 TO 6
```

```
(RND*11)+10)
  510  NEXT A
  520  REM MENSELIJKE EIGENSCHAPPE
N
  530  LET J$(1 TO 2)=STR$ (INT (R
ND*11)+7)
  540  LET J$(3 TO 4)=STR$ (INT (R
ND*11)+5)
  550  LET J$(5 TO 6)=STR$ (INT (R
ND*11)+3)
  560  RETURN
  570  DATA "0002000000"
  580  DATA "0008030400"
  590  DATA "0000000200"
  600  DATA "0007020500"
  610  DATA "0006040000"
  620  DATA "0510070000"
  630  DATA "0400000600"
  640  DATA "0200090000"
  650  DATA "0011000800"
  660  DATA "0600000000"
  670  DATA "0900000000"
  680  DATA "0000000000"
 9000  REM PAUSE ROUTINE
 9010  POKE 23692,-1: IF RND>.5 TH
EN GO TO 9950
 9020  FOR I=1 TO RND*10+10
 9030  BEEP .05,(Z+RND*10): BEEP
.1,-30+Z*3
 9040  NEXT I
 9050  PRINT
 9060  RETURN
 9930  FOR I=1 TO 30
 9940  IF RND>.7 THEN BORDER RND*7
 9970  NEXT I
 9980  BORDER 1
 9990  GO TO 9930
```

Nu u het in het Kasteel van de Angst (Versie 1) heeft overleefd wilt u misschien de volgende wijzigingen aanbrengen. U krijgt dan een nieuwe versie van Kasteel Angst, met een meer gecompliceerde kaart. Breng de volgende veranderingen aan en u kunt het gehele spel weer opnieuw beginnen. Voor de conservatieven onder ons is het misschien een teleurstelling, dat de Tuin (die eigenlijk meer een soort binnenplaats was) nu naar buiten het kasteel verschoven is. De Tuin kan echter nog

overzijde zien, dus de Tuin kan nu niet meer gebruikt worden om u op een gemakkelijke manier te helpen de kaart te maken van het nieuwe kasteel.
Verander eerst de DATA-opdrachten als volgt:

```
9570 DATA "0000020000"
9580 DATA "0407000000"
9590 DATA "0000000700"
9600 DATA "0002050000"
9610 DATA "0006000400"
9620 DATA "0500010900"
9630 DATA "0200030000"
9640 DATA "0911000000"
9650 DATA "0008060800"
9660 DATA "0012000600"
9670 DATA "0800120000"
9680 DATA "0000000000"
9890 REM PAUSE ROUTINE
9900 POKE 23692,-1: IF RND>.5 TH
EN GO TO 9950
9905 FOR I=1 TO RND*10+10
9910 BEEP 0.05,(Z+RND*10): BEEP
```

Nu moet u enige onderdelen van de verschillende ruimten wijzigen. Om het wijzigen te vereenvoudigen heb ik de te wijzigen woorden in hoofdletters getikt. Merk op, dat u in regel 8030 geen wijzigingen hoeft aan te brengen.

```
8000 REM INHOUD VAN KAMERS
8010 IF Z=1 THEN PRINT "U staat
voor de ingang van een  eeuwen o
ud, naargeestig kasteel.U bent a
an de  WEST-zijde van  het kast
eel en als U OOSTwaards kijkt, z
iet U, dat de poort in  de bouwv
allige muur open is en  niet bew
aakt wordt."
8020 IF Z=2 THEN PRINT "U bent i
n de ingangshal, die  rijkelij
k bekleed is met  kostbare
 tapijten. U ziet deurenaan de Z
uid- en NOORDzijde.": REM LAAT D
E REST VAN DEZE REGEL VERVALLEN
8030 IF Z=3 THEN PRINT "Dit is s
```

```
werkt beeld van de maan-godin op
 een voetstuk in de noord-oo
st hoek ziet staan. Er zijn deu
ren naar het zuiden, het OOSTEN e
n EEN RAAM GEEFT UITZICHT
 OP DE GOLVENDE TUINEN."
3050 IF Z=5 THEN PRINT "U bent n
u in de Zaal der Fluister
ingen, waarvan de muren met hout
en panelen zijn bekleed.De zaal
lijkt gevuld met zachte geluiden
 en er zijn uitgangen naar het
 WESTEN en naar het zuiden,
vanwaar de geur van zwavel e
n een onaards gezang U tegemo
et komen. HET RAAM IN DENOORD-MU
UR GEEFT UITZICHT OP DE GOLVENDE
 TUINEN."
3060 IF z=6 THEN PRINT "U bent b
innen gegaan in de kamervan de T
ovenaar, waar een grote kookpot
staat te borrelen boven een vuur
 met groene vlammen. Deze kam
er ruikt sterk naar brandend
e zwavel en U hoort de echo van
 oude spreuken. U kunt n
aar het noorden, het WESTEN o
f het oosten."
3070 IF Z=7 THEN PRINT "U bent i
n een rustige, vredige hal tere
cht gekomen. Dit is de schilder
ijen-galleri j van het kasteel.
 U ziet een groot schil-derij va
n de legendarische Bewaker
van de Zwarte Lagune, links va
n DE DEUR in de NOORDE- LIJKE mu
ur. U kunt de galleri j verlaten
 door deuren naar het noorden
en het OOSTEN."
3080 IF Z=8 THEN PRINT "Dit is d
e meest indrukwekkende kamer in
 het kasteel, de Grote Hal, met
 een massief houten pla-fond. U
kunt de hal verlaten door de
dubbele deuren naar het ZUIDEN o
f door de deuren in de NOORDELI
JKE muur achter U, van- waar U m
uziek hoort komen."
3090 IF Z=9 THEN PRINT "U hoort
het geluid van een strijkw
artet en U begrijpt dat U nu in
```

```
' UIT DE VOLGENDE REGELS IS VERW
IJDERD
3100 IF Z=10 THEN PRINT "U bent
in de Gang der Gedachten,waarvan
 de stilte bedriegelijk  kan zi
n. De gang is vochtig en koud. E
r is een uitgang naar hetWESTEN
EN een deur naar het      zuiden.
"
3110 IF Z=11 THEN PRINT "Dit is
de Brug der Zuchten, waarhet don
ker en vochtig is.          Volgens
 de legende kan hier somsde Bewa
ker van de Zwarte Lagune gehoord
 worden. Er is een deur  naar he
t WESTEN EN een gang naarhet zui
den."
3120 PRINT : IF Z=12 THEN GO TO
1000
3125 IF Z<>1 THEN IF B$(Z,9)="0"
 AND RND>.55 THEN LET B$(Z,9)=ST
R$ (INT (RND*6)): REM VERWIJDERE
N VOOR MINDER MONSTERS
```

U kunt nu rustig met dit programma verder werken, en zo veel versies van Kasteel Angst maken als u zelf wilt. U kunt dezelfde kamernamen behouden, en hetzelfde aantal kamers (12), en daarbij uw eigen kaart ontwikkelen, met de bijbehorende DATA en PRINT-opdrachten. Daarna kunt u de kaart weggooien en proberen, de details te vergeten, als u het eigengemaakte kasteel binnengaat. Misschien wilt u liever kastelen uitwisselen met vrienden, en proberen de weg te vinden, die zij hebben gemaakt in hun eigen versies van mijn programma.
Hoewel u een wereld moet maken, die op een kaart te tekenen is (anders zou u gewoon PRINT-opdrachten kunnen laten genereren en telkens de ENTER-toets indrukken) kunt u natuurlijk proberen de verbindingen tussen de kamers zo 'onmogelijk' mogelijk te maken, als maar in twee dimensies mogelijk is. Het is ook relatief eenvoudig om meer monsters toe te voegen, of deze

kamers verdubbelen, waarbij u de grootte van het array B$ moet verdubbelen om hiermee rekening te houden. Ook moet u natuurlijk DATA-opdrachten maken, die in het array B$ opgeslagen worden, en de PRINT-opdrachten die iedere kamer beschrijven. U kunt dit echter beter niet proberen totdat u volledig vertrouwd bent met mijn originele kasteel, en de manier, waarop deze is gecodeerd in het programma, en voordat u zelf een aantal variaties op Kasteel Angst heeft gemaakt met 12 kamers. De monsters worden opgeslagen in het array M$. Misschien kunt u het beste de meest eenvoudige wijzigingen in het programma eerst proberen. Dit is het wijzigen van de inhoud van de kisten, die zich in het Kasteel bevinden. Het effect van het drinken van de toverdrank of het resultaat van het lezen van de papyrusrol kan ook gewijzigd worden. Zoals het programma standaard werkt, wordt de rol, de vergulde safe of de toverdrank slechts één maal in het programma gebruikt, en bestaan er slechts vier kisten in het kasteel. Hoewel het programma niet meer dan vier kisten laat zien, is er geen mechanisme in het spel aanwezig, dat er voor zorgt, dat er iedere keer dezelfde dingen uit de kisten komen, als u besluit er een te openen. (Als u reeds vier maal door een vampier in de keel gebeten bent, zodat uw KRACHT flink verminderd is, dan zou u misschien willen, dat er een manier was, om er voor te zorgen, dat de vampiers niet in iedere kist van het kasteel leven.)
U vindt het misschien jammer dat u een computer met enorme grafisch mogelijkheden heeft, zoals de Spectrum, en dat alle uitvoer in geschreven vorm is, hoewel er dan wel gebruik gemaakt wordt van FLASH, INVERSE en BRIGHT. Er is geen twijfel, dat sommige spelers programma's prefereren, die complete plaatjes tekenen van iedere kamer, compleet met monsters, kisten etc. De listing is echter al erg lang, en als hij nog veel langer

kunnen zijn. Programma's, die gedetailleerde plaatjes tekenen, gebruiken daarvoor meestal een grote wir-war van PLOT en DRAW opdrachten, en u heeft dan ook een groot aantal programmaregels nodig om een redelijk plaatjes te tekenen, veel meer dan ik in dit programma zinvol acht. U kunt het programma echter zelf zo mooi maken als u zelf wilt. Zelfs als u geen nieuwe kamers wilt, geen monsters of voorwerpen toe wilt voegen, dan wilt u het programma wellicht verbeteren door het toevoegen van plaatjes. Ik ben erg geïnteresseerd in het zien van versies, die met grafische tekeningen uitgerust zijn, of andere variaties, die u ontwikkeld heeft. Stuurt u ze me via de uitgeverij toe. Wie weet, kunnen de beste versies ervan in een van de volgende uitgaven uit onze serie gebruikt worden.
Zelfs als u niet zo ver wilt gaan, grote blokken opdrachtencode toe te voegen, die complete kamers tekenen, dan kunt u uw kamers natuurlijk bepaalde kleuren geven, of u kunt bepaalde beschrijvingen van monsters telkens in dezelfde kleuren laten printen, wellicht met een bepaalde 'herkenningsmelodie'. Merk daarbij regels 2010 tot 2050 op. U kunt hier eenvoudig kleureninformatie toevoegen, en de muziek voor de herkenningsmelodie in deze regels opnemen, direct na het opdrachtgedeelte 'IF Q = .. THEN ...'. Een ander alternatief naast geheel getekende kamers is gebruik te maken van een blok van vier grafische tekens voor gebruik van de vijf monsters en deze tekens dan te gebruiken om een karakteristieke rand aan het scherm te tekenen als een bepaald monster genoemd wordt. U zou zelfs het hele scherm kunnen vullen met deze blokken grafische tekens, voordat u het scherm schoonmaakt om de relevante PRINT-opdrachten en de herkenningsmelodie uit te voeren.

ANDERE OMGEVINGEN

Als u de kaart van KASTEEL ANGST niet geheel uitgewerkt heeft, maar u wilt uw avonturen-programma meer op een eigen programma laten lijken, dan kunt u verregaande wijzigingen aanbrengen. Als we aannemen dat u de in- en uitgangen van de kamers gelijk houdt, dan kunt u natuurlijk de namen van de kamers zo maken, als u zelf wilt. U zou, bijvoorbeeld, het hele avontuur uit kunnen zetten aan boord van een verlaten ruimteschip, dat achtergelaten is door een vreemd ras in een langzame baan rond een onbekende asteroïde-gordel. In plaats van (of behalve) de 'monsters', kunt u gevaarlijke buitenaardse robots gebruiken, die door het onbekende ras zijn achtergelaten om het schip tegen indringers te beschermen, zelfs nu hun meesters al lang zijn uitgestorven. Het doel, waar u naar zoekt, kan de motor van het schip zijn, waarvan u zou mogen verwachten, dat hij van een radicaal gewijzigde technologie is. Als u de kamer vindt, waar de machines van het schip zich bevinden (de kamer, die in het kastel de Zwarte Lagune is), dat door de Spectrum bestuurd wordt, en die door u verslagen moet worden.
In plaats van MAGIE, KRACHT en WIJSHEID kunnen uw eigenschappen meer aardse zaken zijn (vergeef mij de uitdrukking, hier in een ruimteschip!), zoals Zuurstof, Voedsel en Schoten uit uw laser-pistool. U kunt niet alleen de soort eigenschappen veranderen, die in het spel gebruikt worden, maar ook het aantal vergroten, door nieuwe eigenschappen toe te voegen, zoals 'Uithoudingsvermogen' of 'Kennis van het ruimteschip'. U kunt de speler dan toestaan een maximum van bijvoorbeeld 90 eigenschappunten totaal te verdelen over de eigenschappen, die hij of zij wil hebben, inplaats van de punten uit te delen met een random-getallen-gene-

len, is uw speler de mogelijkheid te geven, om te bepalen, wat voor 'type' speler hij of zij is. U kunt dan gebruik maken van bepaalde, vooropgezette typen avonturiers, die bijvoorbeeld 'STOER' of 'SLUW' zijn, en waarbij bij ieder type speler een bepaalde verdeling van punten hoort.
Ons originele programma kan natuurlijk ook de basis zijn voor een ontdekkingsreis in een piramide, die door u onderzocht wordt, voordat hij opgeblazen wordt om een nieuwe dam in de Nijl te kunnen bouwen. U dwaalt door tunnels, en onderzoekt de kamers in de piramide, totdat u in de kamer van de Farao terecht komt (de Zwarte Lagune in Kasteel Angst), waar u de Vloek van de Farao moet zien te overwinnen.
Uit deze ideëen ziet u, dat het niet zo moeilijk is, om een avonturenprogramma te schrijven, dat weinig of geen overeenkomst meer heeft met het originele programma. Als u een beetje vertrouwd geraakt bent met het manipuleren van gecodeerde 'werkelijkheid' op deze manier, dan zit u waarschijnlijk te springen om zelf een avonturenprogramma te schrijven. U kunt een hoop ideëen opdoen door te kijken naar avonturenprogramma's die commercieel verkrijgbaar zijn voor de Spectrum, of voor andere computers. Sommige van deze programma's zijn, wellicht tot uw verrassing, niet gebaseerd op een 'sluitende werkelijkheid', en zijn meer afhankelijk van 'willekeurige getallen' dan van de werkelijke kundigheid van de speler in het tekenen van de kaart of het onthouden van details uit het programma.
U kunt experimenteren met het door elkaar mengen van verschillende programma's, een dimensionaal doolhof toevoegen, of een reactietest in het programma op te nemen, om te zien, hoe sterk uw gevechten werkelijk zijn.

doelloos in het kasteel rondzwerft, hier en daar schatten vindt of draken doodt, dan is het spelletje niet erg leuk, en wordt uiteindelijk al snel vervelend, tenzij er een bepaald laatste probleem op te lossen is, een Finale Schat veroverd moet worden, of hij uit de Martelkamer moet zien te ontsnappen. Behalve een vastomlijnd doel in het programma, moeten de acties in het spel gerelateerd zijn aan dat doel.
In Kasteel Angst probeert uw speler zo veel mogelijk eigenschappunten te verzamelen (of goud, dat bij het laatste gevecht voor eigenschap-punten ingewisseld kan worden), om zo het laatste gevecht aan te kunnen, en dan nog genoeg punten overhoudt om te kunnen ontsnappen aan de klauwen van de bewaker. Iedere activiteit (het openen van een kist, waarin wellicht goud zit, of het verslaan van een monster, die voor een bepaalde eigenschap slechts zeer weinig punten heeft) kan gezien woren als een actie in relatie tot het uiteindelijke doel van het spel. Het is natuurlijk erg leuk om de kaart van het kasteel te ontrafelen, of gewoon in een 'nep-omgeving' rond te wandelen, maar het spel is veel aantrekkelijker als de speler weet, waarom hij deze ongebruikelijke omgeving doorzoekt.
Als u meer wilt weten over de Rollen-spellen, om ideëen op te doen voor nieuwe ontwikkelingen in uw eigen avontuur, dan kunt u deze vinden in de volgende drie boeken (ik heb zelf het eerste boek gelezen, toen ik over de omgeving van Kasteel Angst dacht):

Fantasy Role Playing Games Holmes, Eric J. (Arms and Armour Press, 1981)

Dicing With Dragons, an Introduction to Role-Playing Games Livingstone, Ian (Routledge & Kegan Paul, 1982)

Joe Kapitalist Junior

U bent Joe. Uw opdracht: leid een limonadekraampje en maak uw fortuin (of ga tenminste niet failliet). Het programma helpt u met gebruik van de mogelijkheden van de Spectrum op het gebied van kleuren en high-resolution grafische mogelijkheden, om zo het programma te verlevendigen, terwijl u voor uw bestaan vecht. U ziet hierbij zelfs een portret van uzelf achter de toonbank van uw limonadekraampje. De grafische afbeeldingen in het programma zijn natuurlijk veel beter dan de voorbeelduitvoer, die u hier ziet.
Als het programma gestart is, dan wordt u verteld, dat u nu de eigenaar bent van een kraampje, waar drankjes langs de weg verkocht worden. U krijgt iedere dag de weersvoorspelling. Warm weer betekent grote verkoop. Een saaie dag geeft aan, dat limonade niet bij de Top Tien van de meest verkochte dranken zal staan. Met het weerbericht en de handelskennis, die u tijdens het spel opdoet, moet u beslissen, hoeveel glazen limonade u aan het begin van de dag klaar maakt. U krijgt dan te horen, hoeveel ieder glas limonade u aan produktiekosten gekost heeft. U moet dan beslissen, hoe hoog de prijs is, die u voor het glas limonade in uw kraampje gaat vragen. Natuurlijk hebben erg hoge prijzen de neiging, de verkopen te drukken. Aan het begin van het spel kost uw kraampje u 50 cent huur per dag. Dit wordt natuurlijk in de loop van de tijd meer door inflatie. Inflatie is ook merkbaar in uw winsten, omdat de grondstoffen voor uw limonade in prijs stijgen.
Het voorbeeld hieronder laat u zien, wat voor informatie u ter beschikking heeft: zacht weer, een grondstofprijs van 2 cent per glas limonade, een huur van 50 cent en uw kapitaal, dat in het voorbeeld ƒ 20,— is. Van dit rapport moet u beslissen, hoeveel glazen limonade u

pleet met het aantal volle glazen, en een klein bordje, waarop de verkoopprijs staat afgedrukt.
Daarna ziet u op het scherm, dat de limonade verkocht wordt (de glazen worden leeggedronken). Aan het einde van de dag rollen de luiken voor het kraampje naar beneden, en u krijgt te zien, wat het resultaat van die dag is geweest. 'U heeft 20 glazen verkocht: U heeft ƒ 1,— ontvangen. U heeft 20 glazen gemaakt, die u ƒ 0,40 hebben gekost; De huur was ƒ 0,50; u heeft een winst van ƒ 0,10.'
De DATA-opdrachten in regel 15 en 20 geven de volle en lege glazen weer. De volgende DATA-opdracht, regel 25, bevat het weerbericht en de aantallen, die het effect van het weer op de verkopen hebben. De INK-kleur wordt op zwart en het scherm op normale helderheid gezet. Vervolgens wordt het scherm schoongemaakt (regel 45), voordat in regel 50 en 60 de grafische tekens definiëren, waarbij in regel 50 een teken gelezen en de bijbehorende DATA in regel 60 gePOKEd wordt. Na de BEEP op regel 70 wordt een FUNCTION gedefinieerd in regel 80, die gebruikt wordt in regel 840 om vast te stellen, hoeveel glazen limonade er per dag verkocht worden.
Regels 100 tot 160 printen de instrukties, en na een pauze (zie PAUSE 900 in regel 160) verschijnen de woorden 'Veel geluk!'. Regel 170 print, met FLASH aan, het bekende 'Druk een toets om door te gaan'. PAUSE 0 wacht, totdat er een toets ingedrukt wordt (op de ZX81 is het equivalent hiervoor PAUSE 4E4 of PAUSE 40000) en daarna wordt het scherm schoongemaakt.
De variabelen worden in regel 160 geïnitialiseerd. Uw startbedrag (variabele-naam CASH) wordt op 2000 gezet; het weer op 4; de grondstof-prijs voor uw limonade op 2; en de huur voor het kraampje op 50. Regels 200

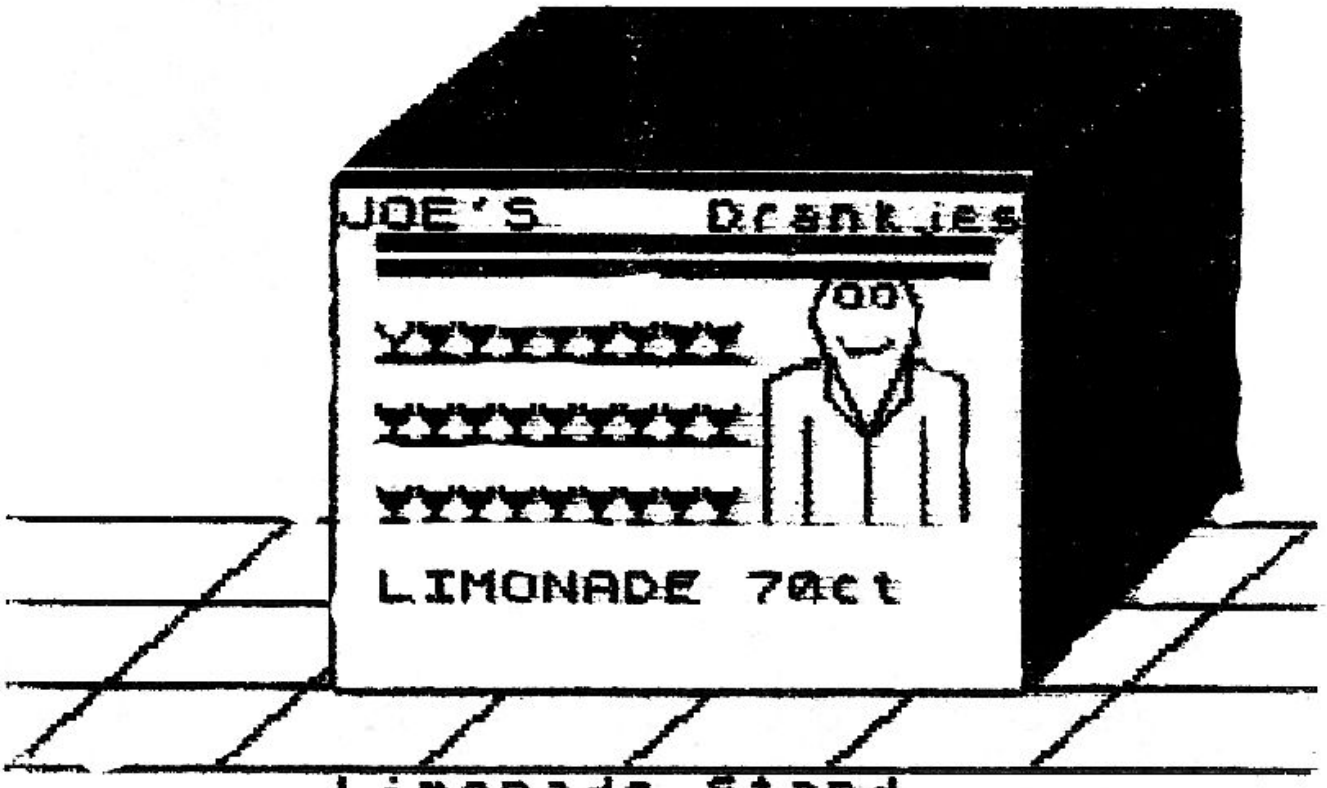
rende getallen bevatten. De getallen (met de namen TO en FROM) worden gebruikt in regel 840, die bekijkt, hoe succesvol u op een bepaalde dag geweest bent.
Uw financiële staat wordt in regel 290 getoond, daarna wordt de vraag gesteld, hoeveel glazen limonade u wilt maken. Als u een negatief getal voorstelt, dan verwerpt regel 302 uw antwoord, en stuurt het programma verder naar regel 300, waar de vraag opnieuw gesteld wordt. Regel 305 stuurt de Spectrum naar de subroutine op regel 9000, die kijkt of u genoeg geld heeft om het aantal glazen limonade te maken, dat u opgegeven heeft. Als u meer glazen limonade wilt maken, dan waar u geld voor heeft, dan wordt regel 9010 uitgevoerd, deze regel wijst u er op, dat dit het geval is ('U heeft niet genoeg geld om zoveel glazen te maken'), en er wordt om een nieuw aantal gevraagd. Dit aantal wordt weer in regel 9000 gecontroleerd, en,als het ingevoerde getal acceptabel is, dan wordt de RETURN op het einde van regel 9000 uitgevoerd. Als het niet acceptabel is, dan gaat het programma weer verder naar regel 9010, waar de zelfde vraag opnieuw gesteld wordt. (Nu we hier toch zijn, kijk eens naar regel 9999. Deze regel werd gebruikt, toen het programma geschreven werd. De regel wordt aangeroepen, als degene, die het programma ontwikkelt, besluit de listing van het prgramma te bekijken. De BACKGROUND, PAPER en INK-kleuren worden normaal ingesteld, daarna doet het programma een CLS om het scherm schoon te maken en een LIST om het programma uit te listen. Aan het einde van de regel ziet u een RETURN-opdracht, die aangeeft, dat de listing aangeroepen kan worden vanuit het programma als en subroutine. U ziet wellicht, dat het erg zinvol kan zijn om zo'n regel aan het einde van een programma op te nemen, als u een ingewikkeld probleem aan de hand

ner is dan één. Als dit het geval is, dan neemt het programma aan, dat er geen limonade verkocht is die dag, en wordt er over het volgende gedeelte van het programma, dat de verkoop van limonade regelt, heengesprongen. Als u echter wel een getal, groter dan nul opgegeven heeft voor het aantal te verkopen glazen limonade, dan geeft u aan, dat u werkelijk zaken wilt doen, en het programma print dan op regel 310 het aantal glazen, dat u aangemaakt heeft. In de loop op regel 320 worden deze volle glazen afgedrukt. Regel 330 vraagt om uw verkoopprijzen voor die dag, waarbij regel 333 prijzen verwerpt, die groter zijn dan 99, of kleiner zijn dan 1. Na een korte BEEP (regel 335) wordt de verkoopprijs afgedrukt, en u moet dan een toets indrukken om het verkoopproces te starten (regel 350). Dan klinkt er een BEEP, en het programma wacht (met een PAUSE 0) totdat u een toets indrukt. Als u dit gedaan heeft, dan klinkt er een tweede BEEP (aan het einde van die regel), en regel 360 maakt het scherm schoon, zet de BRIGHT aan en maakt het scherm leeg met de I-loop. Nu komt het visueel meest interessante gedeelte van het programma. De lange en gecompliceerd uitziende routine (van regel 370 tot 690) maakt het eerste gedeelte van Joe en zijn limonade-kraampje compleet met planken in zijn tentje met glazen limonade. Het volgende gedeelte, dat gebruikt maakt van de variabele COUNT wordt met één verminderd in regel 715, en dit proces van printen en verminderen van COUNT wordt herhaald, totdat COUNT gelijk is aan nul, de computer weet dan, dat alle glazen geprint zijn. Als dit gedaan is, en regel 705 heeft vastgesteld, dat alle glazen op het scherm geprint zijn, zodat de klanten de glazen kunnen komen kopen, maken de regels 730 tot 830 het laatste gedeelte van het limonadekraampje af.

volle glazen door lege glazen op de planken. Als alle glazen verkocht zijn, wordt het bordje 'UITVERKOCHT' opgehangen (regel 900). Na een pauze van een aantal seconden zakken de rolluiken naar beneden. Dan komt er een volgende pauze op regel 940 en dan print de computer het verkoop-overzicht voor die dag. 'U heeft ... glazen verkocht. U heeft ƒ ... ontvangen' .... enz.
De belangrijke variabele PROF, die uw winst bevat, wordt in regel 1000 berekend. Regels 1010 tot 1030 gebruiken de SGN-functie om vast te stellen of de dag in termen van winst een succes of een ramp is geweest. (SGN, afkorting van het Engelse SIGN (teken) geeft –1 als het getal negatief is, +1 als het getal positief is, en 0 als het getal 0 is.)
In regel 1050 slaat de inflatie hard toe, waarbij uw huur opgeslagen wordt met ƒ 0,10. In regel 1060 zorgt een random-getallen-generator er voor, dat ongeveer in één van de 10 keer, dat deze regel uitgevoerd wordt, de grondstofprijs met 1 cent verhoogd wordt (variabele-naam LIMONADE). Regel 1065 wacht, totdat u een toets heeft ingedrukt om door te gaan. Als regel 1067 ontdekt, dat u geen geld meer heeft (als de variabele CASH kleiner is dan 1), dan gaat het programma naar regel 1500, om daar de boel netjes af te sluiten. Als u nog wel solvabel bent, dan krijgt u een volgende kans, om uw geluk te wagen in de handelsmarkt, met een return naar regel 200, waar u een volgende dag vol gruwelijke beslissingen staat te wachten.
De betreurenswaardige boodschap 'GESLOTEN' verschijnt in regel 1500 en 1510, waarbij er gezegd wordt, dat u failliet bent, en dat uw kraampje verkocht is om de verliezen te dekken. Regel 1520 roept daarna zichzelf aan, zodat het programma beëindigd wordt, zonder de Report Code op het scherm te laten zien. U moet dus

Limonade Stand

U bent de eigenaar van een drankjes-stand, U verkoopt limonade. De weersvoor-spellingen worden U iedere dag verteld.
Aan de hand daarvan beslist U hoeveel glazen U maakt.
Ook wordt aan U verteld hoeveel ieder glas U kost en aan de hand daarvan bepaalt U de verkoopprijs. De huur van de grond
kost U 50 ct per dag maar kan hoger worden door de inflatie zo ook de limonade.



Limonade Stand

Verkoop Overzicht

U heeft 1 glazen verkocht

Inkomsten Fl0.7

Gemaakt 30 Kosten Fl0.6

Huurkosten 50ct

Limonade stand

Verslag van vandaag

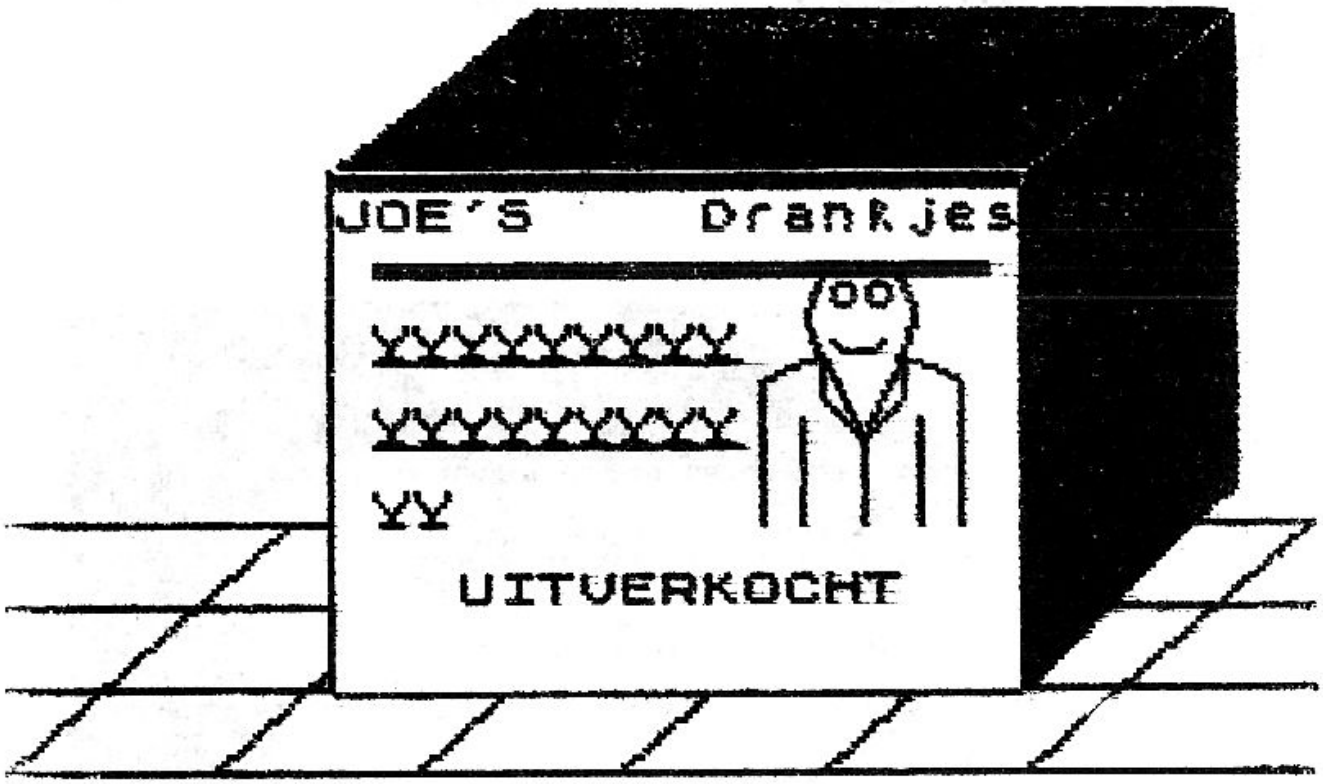
Weer : Betrokken hemel

Ieder glas kost U 2 ct

Huur: 50ct

U heeft Fl20 beschikbaar

Glazen: YYYYYYYYYYYYYYYYYYYY




```
   12 POKE 23609,128
   15   DATA "e",0,BIN 10000010,BI
N 10000010,BIN 01000100,BIN 0010
1000,BIN 00010000,BIN 00010000,B
IN 01111100
   20 DATA "f",0,BIN 10000010,BIN
 11111110,BIN 01111100,BIN 00111
000,BIN 00010000,BIN 00010000,BI
N 01111100
   25 DATA "Storm",10,30,"Nat",25
,40,"Betrokken hemel",35,50,"Zac
ht",45,70,"Warm",65,120,"Heet",1
00,150,"Hittegolf",200,300
   45 INK 0: BRIGHT 0: CLS
```

```
  70 BEEP .5,0
  80 DEF FN p(p,q)=q/(p/100)-100
 100 PRINT PAPER 6;"                Lim
onade Stand                ": PRINT : P
RINT "   U bent de eigenaar van e
en "

 110 PRINT "   drankjes-stand. U
verkoopt      limonade. De weers
voor-         spellingen worden
U iedere dag  verteld."

 120 PRINT "   Aan de hand daarva
n beslist U   hoeveel glazen U m
aakt."

 130 PRINT "   Ook wordt aan U ve
rteld         hoeveel ieder glas
 U kost en"

 140 PRINT "   aan de hand daarva
n bepaalt U   de verkoopprijs. D
e huur van    de grond"

 150 PRINT "   kost U 50 ct per
dag maar kan  hoger worden door
 de inflatie"

 160 PRINT "   zo ook de limonade
.": PAUSE 900: PRINT INK 2;"
     Succes !"
 170 PRINT FLASH 1;'"Druk een wi
llekeurige toets in  om te begin
nen": PAUSE 0: CLS
 180 LET cash=2000: LET weather=
4: LET lemonade=2: LET rent=50
 200 LET weather=weather+INT (RN
D*3-1)
 210 IF weather=0 THEN LET weath
er=1
 220 IF weather=8 THEN LET weath
er=7
 230 PRINT PAPER 6;"                Lim
onade stand                "
 240 PRINT INK 7; PAPER 1;'"
     Verslag van vandaag       "

 250 RESTORE 25: FOR i=1 TO weat
```

```
; INK 1;Lemonade;" ct"
 280 PRINT '"Huur: "; INK 1;rent
;"ct"
 290 PRINT '"U heeft "; INK 1;"F
";cash/100; INK 0;" beschikbaar

 300 BEEP .5,10: INPUT INK 2;"Ho
veel glazen wilt U maken "; FLA
SH 1;"?";cups
 302 IF cups<0 THEN GO TO 300
 305 GO SUB 9000: BEEP .5,5
 307 IF cups<1 THEN LET sold=0:
ET price=1: LET cups=0: GO TO 9
45
 310 INK 6: PAPER 0: PRINT '"Gla
zen: ";
 320 FOR i=1 TO cups: PRINT "Y";
: NEXT i: PAPER 7: PRINT
 330 BEEP .5,10: INPUT INK 2;"Wa
t is Uw verkoopprijs vandaag"; F
LASH 1;"?";price
 333 IF price>99 OR price<1 THEN
 GO TO 330
 335 BEEP .1,5
 340 INK 0: PAPER 7: PRINT '"Lim
onade wordt verkocht voor ";pric
e;"ct per glas"
 350 PRINT FLASH 1; INK 2;'"Toet
s een willekeurige toets om", "te
 verkopen": BEEP .1,10: PAUSE 0:
 BEEP .5,5
 360 CLS : BRIGHT 1: FOR i=16 TO
 21: PRINT AT i,0;"
                    ": NEXT i
 370 FOR i=0 TO 48 STEP 16
 380 PLOT 0,i: DRAW 255,0
 390 NEXT i
 400 FOR i=0 TO 200 STEP 40
 410 PLOT i,0: DRAW 55,48
 420 NEXT i
 430 PRINT PAPER 6;AT 6,8;"JOE'S
    Drankjes"; PAPER 6;AT 9,8;"
                "
 440 FOR i=92 TO 98
 450 PLOT INK 7;72,i: DRAW INK 7
 120,0-PI/12
 460 NEXT i
 470 FOR i=1 TO 6: PRINT PAPER 6
 AT 9+i,8;" ";TAB 24; PAPER 6;"

 480 NEXT i
```

```
 490 INK 6
 500 FOR i=112 TO 115
 510 PLOT 64,i: DRAW 136,0
 520 NEXT i
 530 FOR i=64 TO 199
 540 PLOT i,116: DRAW 40,31
 550 NEXT i
 560 FOR i=115 TO 24 STEP -1
 570 PLOT BRIGHT 0;200,i: DRAW B
RIGHT 0;40,31
 580 NEXT i
 590 FOR i=16 TO 23
 600 PLOT BRIGHT 0,200,i: DRAW B
RIGHT 0,40,40
 610 NEXT i
 615 INVERSE 1
 620 PLOT PAPER 0;64,116: DRAW P
APER 0;134,0
 630 PLOT PAPER 0;64,112: DRAW P
APER 0;134,0
 640 INK 0: INVERSE 0
 650 PLOT 63,16: DRAW 0,100
 660 PLOT 199,16: DRAW 0,96: DRA
W INVERSE 1; INK 6; PAPER 0,0,4
 670 PLOT 64,15: DRAW 134,0
 680 INK 2: PLOT 72,79: DRAW 72,
0
 690 PLOT 72,63: DRAW 72,0
 695 LET count=cups: GO SUB 700:
 GO TO 730
 700 IF count=0 THEN RETURN
 705 FOR i=11 TO 15 STEP 2: FOR
j=1 TO 9
 710 BRIGHT 0: PRINT AT i,8+j;"Y
": BEEP .1,20
 715 LET count=count-1: IF count
=0 THEN RETURN
 720 NEXT j: NEXT i: RETURN
 730 PLOT 160,80: DRAW 15,0,-PI*
3/2
 740 PLOT 148,48: DRAW 0,27: DRA
W 12,5,-PI/6
 750 PLOT 187,48: DRAW 0,27: DRA
W -12,5,PI/6
 760 PLOT 156,48: DRAW 0,20
 770 PLOT 179,48: DRAW 0,20
 780 PLOT 168,48: DRAW 0,15: DRA
W 8,8: DRAW 0,8: DRAW -8,-16
 790 DRAW -8,8: DRAW 0,8: DRAW 8
,-16
 800 PLOT 162,83: DRAW 10,0,PI/2
 810 CIRCLE 164,92,2: CIRCLE 171
```

```
 840 LET sold=(INT (RND*(to-from
+from)+(to-from))*60/FN p(lemon
ade,price)
 844 IF sold<0 THEN LET sold=INT
 (cups+2/2)
 845 LET sold=INT sold: IF sold>
cups THEN LET sold=cups
 850 LET sellc=sold
 860 FOR i=11 TO 15 STEP 2: FOR
j=9 TO 17
 870 PRINT AT i,j;"Y": BEEP .1,1
0
 880 LET sellc=sellc-1: IF sellc
=0 THEN GO TO 900
 890 NEXT j: NEXT i: GO SUB 700:
 GO TO 860
 900 IF sold=cups THEN PRINT INK
 0; PAPER 4;AT 17,9;"  UITVERKOC
HT  "
 910 PAUSE 100: FOR i=103 TO 48
STEP -1
 920 BEEP .02,i-48: PLOT 72,i: D
RAW 120,0
 930 NEXT i
 940 PAUSE 200: BEEP .5,5
 950 CLS : PAPER 7: INK 0: PRINT
 PAPER 6;"              "; BRIGHT 1;"L
imonade Stand"; BRIGHT 0;"
    "
 960 PRINT INK 7; PAPER 1;"  Ver
koop Overzicht
 970 PRINT '"U heeft "; INK 1;so
ld;" glazen verkocht"
 980 PRINT '"Inkomsten  "; INK 1
;"Fl";sold*price/100
 990 PRINT '"Gemaakt     ";cups;"
  Kosten  "; INK 1;"Fl";cups*lem
onade/100
 995 PRINT '"Huurkosten ";rent;"
Ct"
1000 LET prof=sold*price-cups*le
monade-rent
1010 IF SGN prof=-1 THEN PRINT '
"U heeft een verlies gemaakt van
 "; INK 2; INK 0;"Fl";-prof/100
1020 IF SGN prof=0 THEN PRINT '"
U heeft geen verlies of winst
gemaakt"

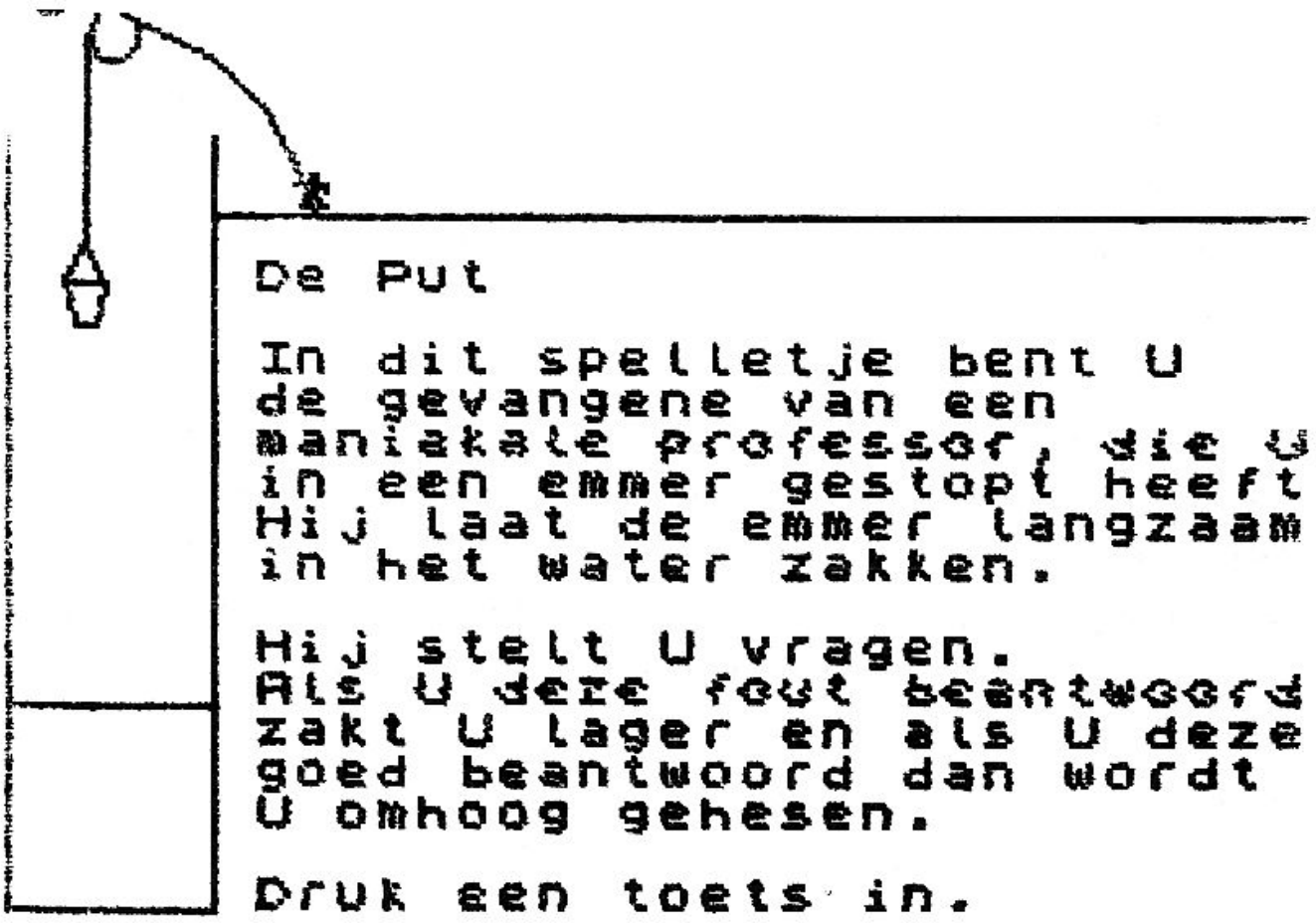
1030 IF SGN prof=1 THEN PRINT '"
U heeft "; INK 4;"een winst beha
```

```
t+INT (RND*10+1)
1060 IF RND>.9 THEN LET lemonade
=lemonade+1
1065 BEEP .5,10: PAUSE 0: BEEP .
5,5
1067 IF cash<1 THEN GO TO 1500
1070 CLS : GO TO 200
1500 CLS : PRINT INK 1; FLASH 1;
"                GESLOTEN
 "
1510 PRINT '" U bent failliet en
 Uw stand is  verkocht om Uw sch
ulden te       betalen."
1520 GO TO 1520
9999 STOP
9000 IF cups*lemonade<=cash THEN
 RETURN
9010 INPUT INK 1;"U heeft niet g
enoeg geld om zo- veel te maken.
"; INK 2;"Hoeveel glazen wilt U
maken "; FLASH 1;"?";cups
9020 GO TO 9000
9998 STOP
9999 INVERSE 0: PAPER 7: BRIGHT
0: INK 0: BORDER 7: CLS : LIST :
 RETURN
```

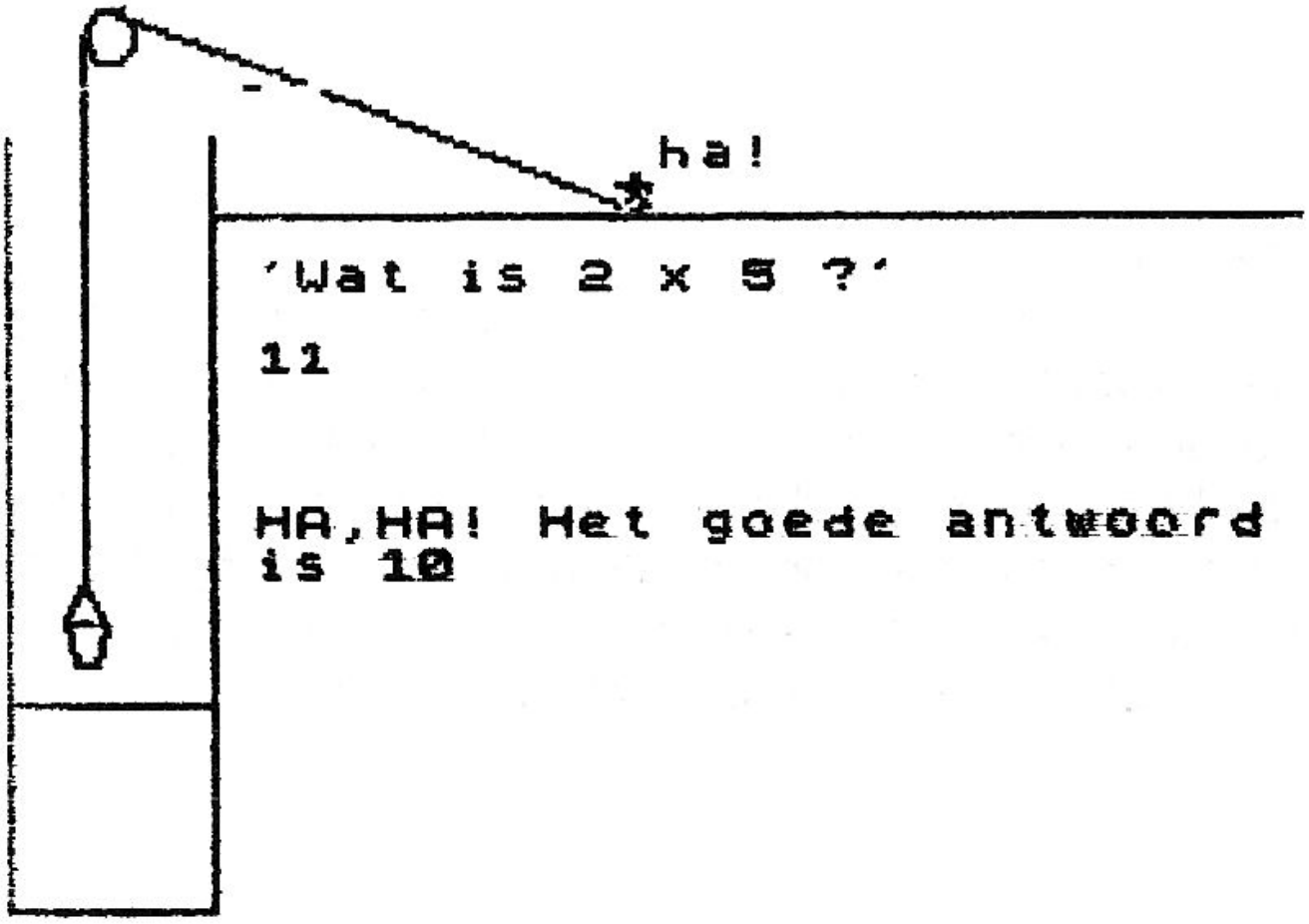
HERSENBREKERS

De Put

In het volgende spelletje wordt een zeer realistisch en geloofwaardig scenario gebruikt, waarin u de hulpeloze gevangene bent van een wilde, krankzinnige oude man, die heeft besloten, dat u voor uw zonden in een put verdronken moet worden, als u niet in staat bent, voldoende rekenkundige kennis ten toon te spreiden.
Hier ziet u, hoe het programma de situatie uitlegt:



En zo ziet het programma er tijdens uw ondervraging uit:



Het programma is recht-toe – recht-aan geschreven, en u kunt er dan ook zelf naar hartelust in wijzigen. De routine vanaf regel 1000 tekent het plaatje. Nadat het plaatje voor de eerste keer getekend is, vraagt het programma, hoe goed u in rekenen bent. Als u een getal kleiner dan 12 intoetst, dan kiest de computer een willekeurige moeilijkheidsgraad. U kunt het beste beginnen met 15, als u het programma de eerste keer gebruikt. Regel 710 zet T$ gelijk aan het vermenigvuldigingsteken (*) of het optel-teken (+), en de variabele A krijgt in regel 720 een waarde tussen 1 en uw moeilijkheidsgraad. Als T$ een '*' is, dan krijgt B in regel 735 een waarde tussen 1 en 10. Als T$ een '+' is, dan krijgt B een waarde tussen 0 en 100 maal uw moeilijkheidsgraad. Regel 740 print de vraag op het scherm, waarbij

mogelijkheid uw antwoord in te tikken. Let hierbij goed op, hoe het programma de INKEY$-functie gebruikt om getallen van het toetsenbord te lezen, en hoe daarna een getal van meer dan één cijfer gemaakt wordt. U kunt deze routine zelf in uw eigen programma's opnemen. De professor reageert op uw antwoord (regels 850 tot 985), en daarna wordt de volgende vraag gesteld. Als u naar de routine aan het einde van het programma kijkt, dan ziet u, dat hier de variabele l belangrijk is. Hij bepaalt, hoe ver de emmer in de put is getekend. Als de emmer aan de bovenzijde van de put komt, dan heeft u het overleefd, en wordt de professor met het touw gestraft voor zijn wandaden, zodat het programma een prachtige afloop krijgt. (Deze afloop ontkent trouwens de wetmatige relatie tussen emmers, touwen en putten). Als u zelf in het water terecht komt, dan valt deze geweldige afloop van het programma u ten beurt.

```
  5 REM  De Put
 10 LET l=40
 20 GO SUB 9000
510 PRINT AT 6,6;"De Put"
520 PRINT ,,TAB 6;"In dit spel
letje bent U"
530 PRINT TAB 6;"de gevangene v
an een"
531 PRINT TAB 6;"maniakale prof
essor, die U"
540 PRINT TAB 6;"in een emmer g
estopt heeft"
550 PRINT TAB 6;"Hij laat de em
mer langzaam"
560 PRINT TAB 6;"in het water z
akken."
580 PRINT ,,TAB 6;"Hij stelt u
vragen."
590 PRINT TAB 6;"Als U deze fou
t beantwoord"
600 PRINT TAB 6;"zakt U lager e
n als U deze"
610 PRINT TAB 6;"goed beantwoor
d dan wordt"
```

```
 640 PRINT AT 21,6;"Druk een toe
ts in."
 650 PAUSE 0
 660 GO SUB 1000
 670 PRINT AT 6,6;"De professor
zegt:"
 680 INPUT "'Hoe goed bent U in
rekenen?(%)';d
 682 IF d<12 THEN LET d=INT (RND
*7)+4
 685 LET l=121-d
 690 IF d<0 OR d>100 THEN PRINT
AT 8,6;"Geef een getal tussen 0-
100": GO TO 680
 700 GO SUB 1000
 710 LET t$=CHR$ (42+INT (RND*2)
)
 720 LET a=INT (RND*d)+1
 730 IF t$="*" THEN LET b=INT (R
ND*10)+1: GO TO 740
 735 LET b=INT (RND*d*100)
 740 PRINT AT 6,6;"'Wat is ";
 750 LET u$=t$
 760 IF t$="*" THEN LET u$="x"
 765 LET c$=STR$ VAL (STR$ a+t$+
STR$ b)
 770 PRINT a;" ";u$;" ";b;" ?'"
 772 LET tt=0
 775 LET w$=""
 777 FOR g=1 TO LEN c$
 780 FOR t=1 TO l*100
 785 LET i$=INKEY$
 790 IF i$="" THEN NEXT t
 800 IF CODE i$<48 OR CODE i$>57
 THEN GO TO 910
 810 LET w$=w$+i$
 815 LET tt=tt+1
 820 IF w$(g)<>c$(g) THEN GO TO
910
 825 PRINT AT 8,6;w$; BEEP .1,20
 830 IF INKEY$<>"" THEN GO TO 83
0
 840 NEXT g
 850 PRINT AT 10,6;"Jammer!! U h
eeft het goed!"
 860 FOR a=1 TO tt/10
 870 BEEP .01,20
 880 NEXT a
 890 LET l=l-(l*10-tt/LEN c$)/10
0
 895 BEEP 1,30
```

```
 930 NEXT a
 940 PRINT AT 12,6;"HA,HA! Het g
ede antwoord"
 950 PRINT AT 13,6;"is ";c$
 952 FOR u=1 TO 3
 955 PAUSE 30
 960 PRINT AT 3,l/8+3;"ha!"
 965 PAUSE 30
 970 PRINT AT 3,l/8+3;"    "
 975 NEXT u
 980 LET l=l+4
 985 GO TO 700
 990 STOP
1000 CLS
1005 PLOT 0,150
1010 DRAW 0,-150
1020 DRAW 40,0
1030 DRAW 0,150
1032 PLOT 40,135: DRAW 215,0
1035 PLOT 1,40: DRAW 39,0
1040 CIRCLE 20,170,5
1050 PLOT 20,175
1060 DRAW l,-40,-(l<85)
1065 PRINT AT 4,l/8+2; OVER 1;"*
"
1070 PLOT 15,170
1080 DRAW 0,-l
1090 DRAW -4,-8
1100 DRAW 8,0
1110 DRAW -4,8
1120 PLOT 11,162-l
1130 DRAW 2,-8
1140 DRAW 4,0
1150 DRAW 2,8
1160 IF l=0 THEN BEEP 1,22: GO T
O 2000
1165 IF l=153 THEN BEEP 1,22: GO
 TO 3000
1170 IF l<20 THEN LET l=0: BEEP
1,20: GO TO 1000
1175 IF l>122 THEN LET l=153: BE
EP 1 20: GO TO 1000
1180 RETURN
2000 PRINT AT 4,2; OVER 1;"*"
2010 FOR a=6 TO 16
2020 PRINT AT a-1,2;" "
2030 PRINT AT a,2;"*"
2035 BEEP .05,a
2040 NEXT a
2050 FOR a=17 TO 21
2055 PRINT AT a,2;"*"
2060 PRINT AT a-1,2;" "
```

```
2100 PRINT AT 16,2;"_"
2110 STOP
3000 STOP
9000 DATA 24,24,254,58,24,36,102
,0
9010 FOR a=0 TO 7
9020 READ b
9030 POKE USR "a"+a,b
9040 NEXT a
9050 RETURN
```

PROGRAMMEERTIPS

Verbeter uw programma's

'Kunstenaars zijn altijd de eersten geweest, die nieuwe technieken gebruiken....' (David Thornburg, Computer and Society, Computer magazine, Maart 1982, p. 16)

Het gebruik van de personal computer heeft veel mensen een uitlaatklep voor hun creativiteit gegeven. Velen hebben ondekt, dat het toegang hebben tot een computer hun een mogelijkheid heeft gegeven voor creatieve activiteiten, waarbij gebruik gemaakt wordt van vaardigheden, die anders verborgen zouden zijn gebleven. Zoals bij alle kunstvormen is het ook hier zo, dat kennis van de technieken het eenvoudiger kan maken, om uw creatieve programmeerwensen uit te drukken. Als u een aantal technieken en trucks 'achter de hand' heeft, kunt u makkelijk verder gaan met het ontwikkelen van het programma, dat u in gedachte heeft, zonder daarbij gehinderd te worden door de angst, dat u de computer niet kunt laten doen, wat u wilt. Bijvoorbeeld, als u weet, hoe u een bepaald object op uw scherm kunt laten bewegen, dan kunt u zich concentreren op andere belangrijke gedeelten van het programma, zonder dat u zich daarna nog druk hoeft te maken over het feit, of u uiteindelijk het probleem van de beweging nog op kunt lossen.

Als u de programma's in dit boek, en andere boeken goed bestudeert, dan heeft u al heel wat over programmeren geleerd. Als u de aantekeningen bij de programmalisting in dit boek leest, dan kunt u een aantal ideeën oppikken, die u eenvoudig in uw eigen programma's

schien het beste grote gedeelten van de programma's in dit boek overnemen in uw eigen programma's, waarbij u dan zaken kunt veranderen zoals de user-defined grafische tekens of de toetsen, die door een INKEY$-routine gelezen worden, en dan later pas verder gaan met het schrijven van volledig originele programma's
Houd altijd de 'menselijke' zijde in de gaten als u een programma schrijft. In veel gevallen worden uw programma's niet alleen door u zelf gebruikt, maar ook door andere mensen, u moet daarom proberen het gebruik van uw programma's voor anderen zo plezierig en gemakkelijk mogelijk te maken. Bijvoorbeeld, veel programma's, die voor de Spectrum geschreven zijn (waaronder enkele programma's in dit boek) gebruiken de toetsen '5', '6', '7' en '8' voor besturing van bewegingen, waarbij een object onder besturing van de speler in de richting verplaatst wordt die aangegeven wordt door de pijlen boven die toetsen. Maar als u even nadenkt, dan ziet u, dat dit niet onder alle omstandigheden de meest ideale toetsen zijn. Ze staan dicht naast elkaar aan de bovenzijde van het toetsenbord, en zijn dus zeker niet eenvoudig te bedienen. Het kan daarom veel handiger zijn, als een programma de toetsen 'Z' en 'M' gebruikt voor het bewegen naar links (Z) en rechts (M). Andere toetsen kunnen gebruikt worden voor naar boven en naar beneden bewegen. U ziet dit in een aantal programma's in dit boek, zoals DEATH RACE 2000.
Het is de moeite waard om iedere programmeergewoonte die u ontwikkelt, bij het gebruik kritisch te beschouwen, om te zien, of deze gewoonte ook werkelijk een toegeving geeft aan een zo gebruikersvriendelijk mogelijk programma (zoals bij het gebruik van de '5' tot '8'-toetsen).
Als u andere toetsen gebruikt dan de '5' tot '8'-toetsen,

sities, die enigszins verband houden met de functie, die ze in het programma hebben (zoals de linkerbenedenhoek van het toetsenbord –'Z'– om links te bewegen). Voor andere besturingen in een programma, zoals het afvuren van uw dodelijke laser geweer, is het handig om een toets te kiezen, die zeer gemakkelijk te bereiken is, of een toets, waarvan de letter een bepaalde relatie heeft tot de functie in het programma, zoals 'V' om uw laser af te Vuren, of 'R' om een dobbelsteen te Rollen. U ziet dit bijvoorbeeld in De Wraak van Kasteel Angst, waar u de 'O' intikt om naar het oosten te gaan, 'VL' om te vluchten, en 'VE' om te vechten. Als u de instrukties voor zo'n spel eenmaal gelezen heeft, zult u waarschijnlijk de functies van deze toetsen niet snel vergeten.

Als het mogelijk is, gebruik dan altijd zeer duidelijke aanwijzingen, op het moment dat de speler iets in moet tikken, zodat hij in iedere situatie precies weet wat er van hem verwacht wordt. Hoewel het niet nodig is om altijd uitgebreid gebruiksaanwijzingen bij en programma in te bouwen (deze gebruiken vaak een hoop typetijd en geheugenruimte), moeten uw INPUT-opdrachten altijd de nodige duidelijkheid verschaffen. Het gebruik van bepaalde aanwijzingen voor de gebruiker, tussen quotes in de INPUT-opdracht op de Spectrum, maken het eenvoudig de gebruiker de nodige aanwijzingen te verstrekken. Zoals u waarschijnlijk wel weet, kunt u alle standaard PRINT-opdracht elementen, zoals FLASH, BRIGHT en INVERSE ook in de INPUT-opdracht gebruiken.

Als uw programma eenmaal loopt, moet u enige tijd besteden aan het zo aantrekkelijk mogelijk maken van de gebruikte plaatjes. Zelfs als de uitvoer van het programma uitsluitend bestaat uit PRINT-opdrachten, dan kunnen deze altijd nog op een aantrekkelijke, interes-

af en toe een verandering van de BORDER-kleur kunnen de boodschappen uit uw programma verduidelijken, en de leesbaarheid van uw programma vergroten. Willekeurige veranderingen van de BORDER-kleur worden bijvoorbeeld gebruikt in het schaakprogramma uit dit boek, om aan te geven, dat de computer een stuk van zijn tegenstander gaat pakken. Geluiden kunnen natuurlijk ook gebruikt worden om het programma te verbeteren. Het is verrassend om te zien hoe veel een klein aantal BEEP-opdrachten toe kunnen voegen aan de uitvoer van een programma, niet allen voor de hand liggende zaken, zoals het geluid van verpulverende Space Invaders, of voor een kaatsende bal, maar ook voor applicaties, zoals Beginners Schaak. In dat programma BEEPt de computer, terwijl hij door een bepaald stuk programma gaat. Dit geeft de zekerheid, dat als het erg lang duurt, voordat de computer een zet gevonden heeft en weer een teken van leven geeft, de speler in ieder geval weet, dat het programma niet in een eindeloze loop is geraakt. Het gebruik van kleur en geluid kan altijd de aantrekkingskracht van een programma verbeteren. Hoewel er mensen zijn, die sterke argumenten aan kunnen voeren tegen het gebruik van kleuren, lijkt het onzinnig als u gebruik maakt van een computer zoals de Spectrum – die als een van de sterke punten kleuren heeft – geen voordeel te trekken uit deze mogelijkheden. Dit zelfde argument geldt ook voor user-defined grafische tekens. U moet deze tekens gebruiken, als u denkt dat ze uw programma verbeteren. In ons schaakprogramma hebben we ze niet gebruikt, omdat de nadruk daar heeft gelegen op de kwaliteit van de speelsterkte van het programma in plaats van de uiterlijke verschijning. Er is echter geen enkel excuus mogelijk (behalve luiheid) om inverse sterretjes te gebruiken,

Terwijl we dus de waarde van het toevoegen van kleuren met de opdrachten INK en PAPER, of het gebruiken van FLASH, INVERSE en BRIGHT al gezien hebben, zijn er een aantal meer subtiele dingen die u kunt doen om uw programma sterker over te doen komen, zoals het afdrukken van bepaalde woorden in een zin in hoofdletters. U ziet dit in één van de aanpassingen in de Wraak van Kasteel Angst, waar ik alle veranderde woorden in HOOFDLETTERS heb afgedrukt, zodat u kunt zien, welke woorden u voor dit spelletje moet veranderen.
Hoewel de woorden hier in hoofdletters geplaats zijn om de conversie te vereenvoudigen, en niet zozeer om deze woorden te benadrukken, geeft de methode wel weer, hoe eenvoudig en effectief dit kan zijn. De positie van de print-uitvoer op uw scherm kan ook van belang zijn, en PRINT AT maakt het eenvoudig om uw uitvoer overal op het scherm te plaatsen, waar het het beste tot zijn recht komt.

Verhogen van de programmasnelheid

In sommige programma's is de kwaliteit, waarmee er gespeeld wordt, belangrijker dan de snelheid, waarop het spel zich afspeelt (zoals in PIRANDELLO en DAMMEN in dit boek). Maar zelfs in dit soort programma's moet u toch trachten een programma te schrijven, dat zo snel mogelijk loopt. Snelheid is meestal een cruciale factor in video-spelletjes. Trage, slordig bewegende objecten maken een spelletje al snel onbevredigend.
In de inleiding tot sommige spelletjes in dit boek heb ik de aandacht al gericht op een aantal aspecten in het programma, die helpen bij het verhogen van de uitvoeringssnelheid. Ik dacht, dat het wel zinvol zou zijn, om deze (en andere) tips nog eens aan het einde van het

doorzoeken, vanaf de allereerste regel, totdat hij het regelnummer tegenkomt, dat in de GOTO of GOSUB genoemd werd. Daarom lopen programma's waarvan de veel aangeroepen subroutines, aan het begin van de listing staan, een beetje sneller, dan andere programma's. Als er een hoofd-loop is, die telkens opnieuw in het programma uitgevoerd wordt, dan is het vaak de moeite waard, om het programma zo te organiseren, dat deze hoofdloop ook aan het begin van de listing staat. U ziet dat de meeste programma's in dit boek zo georganiseerd zijn.
Naast deze indeling van de programma's, moeten bepaalde delen van het programma, die slechts zeer weinig gebruikt worden (zoals instrukties voor de speler aan het begin van het spel, en de initialisatie van variabelen en user-defined grafische tekens) aan het einde van het programma geplaatst worden. Het is dan niet nodig, dat het programma iedere keer deze code moet 'doorworstelen', als hij op zoek is naar een regelnummer voor een GOTO of een GOSUB-opdracht. U zult zien, dat in het algemeen geldt, dat hoe kleiner het aantal regels in een programma is, hoe sneller het uitgevoerd kan worden, die op één regel 'aan elkaar geknoopt' zijn, worden iets sneller uitgevoerd, dan opdrachten, die alle op losse regels van telkens één opdracht staan. Dit is het duidelijkst merkbaar bij programma's met bewegende grafische tekens, u moet dus proberen reeksen PRINT AT-opdrachten altijd met ':' aan elkaar te knopen. In het algemeen is de verlaging van de snelheid door het plaatsen van iedere opdracht op een nieuwe regel, of het toevoegen van vele REM-opdrachten kritisch, noch duidelijk merkbaar. Als de snelheid van het programma echter verhoogd moet worden, dan is het de moeite waard, om twee versies van een programma

mogelijk is tijdens de ontwikkeling en het opsporen van fouten. Als u deze versie foutloos heeft, dan kunt u een tweede versie maken, waarin u geen REM-opdrachten meer heeft, en waarbij u alle opdrachten zo veel mogelijk op één regel plaatst.
GOTO opdrachten kunnen vaak geheel uit het programma weggelaten worden. Bekijk iedere GOTO-opdracht, die niet past in een IF . . . THEN . . . constructie zorgvuldig, om te zien, of deze wellicht verwijderd kan worden. Sommige programmeer-puristen zeggen, dat een onvoorwaardelijke GOTO-opdracht altijd slecht is. Dit is echter geen reden om deze veel gebruikte opdracht nooit meer te gebruiken. U moet echter kritisch door uw programma's heen gaan, en iedere onvoorwaardelijke GOTO met argwaan beschouwen. Wellicht kunt u dan een manier bedenken, om de GOTO te verwijderen.
GOSUB's kunnen ook vaak geheel weggelaten worden. Als er een betrekkelijk korte subroutine in een programma staat, die twee of drie maal gebruikt wordt, dan is het vaak beter, om de code voor die subroutine op te nemen op de plaatsen, waar u anders de GOSUB zou doen, in plaats van het uitvoeren van een tijdverslindende GOSUB en daarna een RETURN.
Als uw programma loopt, dan kunt u vaak een listing van het programma maken, en deze bestuderen. U ziet meestal wel een aantral blokken code, die logisch bij elkaar horen. U zult ontdekken, dat u de meeste programma's op deze manier kunt 'ontrafelen', zelfs als de programma's niet met een gestructureerd idee in het achterhoofd van de programmeur geschreven zijn. Als u uw programma's op deze wijze bekeken heeft, dan ontdekt u vaak, dat de programma's vanuit een logisch standpunt bekeken niet altijd op de meest voor de hand liggende manier in elkaar zitten. U kunt dan proberen

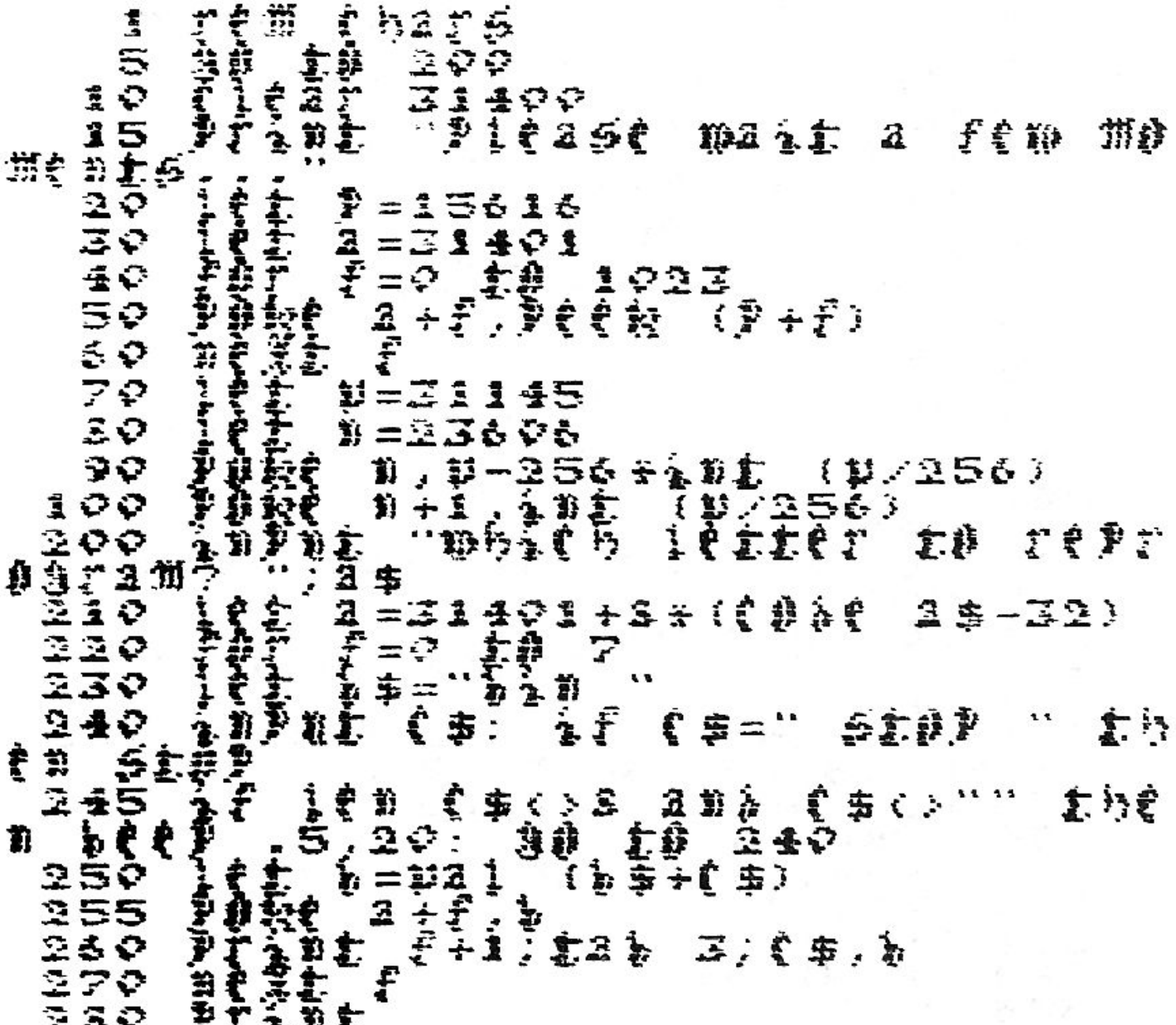
Gebruik geen haakjes, tenzij absoluut noodzakelijk. Haakjes gebruiken zowel geheugen- als uitvoeringstijd. U zult al gemerkt hebben, dat de Spectrum in het algemeen met minder haakjes genoegen neemt dan vele andere computers. Een voorbeeld hiervan is de opdracht PRINT CHR$ 134, die in de meeste andere BASIC's geschreven moet worden als PRINT CHR$(134).
U kunt wat problemen krijgen als u een programma, dat voor een 16K ZX81 geschreven is, in een 16K Spectrum te krijgen. De grafische tekens nemen op de Spectrum meer geheugenruimte in beslag. Als u programma's converteert, laat dan alle REM-opdrachten weg, en probeer de PRINT-opdrachten zo veel mogelijk 'aan elkaar te plakken' met behulp van ';'. Laat de gebruikersinstructies geheel weg. Bekijk daarna de listing om te zien, of deze nog op een of andere manier verkort kan worden, zoals het opnemen van veel-gebruikte woorden of zinnen in strings, of door het opnemen van een subroutine voor stukken code, die verschillend malen vanuit het programma worden aangeroepen. Dit vertraagt de uitvoering van het programma wel enigszins, maar dat is een klein offer, dat we moeten brengen om het programma van de ZX81 ook op de Spectrum te kunnen draaien. Zoals u waarschijnlijk wel weet, is de Spectrum een veel snellere computer dan de ZX81, dus dit offer zal waarschijnlijk geen probleem zijn. Als het programma op de ZX81 op een redelijke snelheid liep, dan kan het op de Spectrum alleen maar beter zijn.

APPENDIX

Gotische Generator

Dit programma is geen spelletje, maar een zeer nuttige programmeerhulp, en bovendien kunt u er veel plezier aan beleven. Het programma geeft u de mogelijkheid om één, twee of zelfs alle tekens, die door de Spectrum geprint worden, als tekens van uw eigen keuze te laten afdrukken. Het programma is slechts 30 regels lang, maar u kunt er werkelijk zeer bijzondere effecten mee produceren.

Hier is bijvoorbeeld de listing van het programma in gotische tekens, die gemaakt zijn door de gehele tekenset opnieuw de definiëren:



```
[illegible]
[illegible]
[illegible]  200
```

Zoals u kunt zien, werkt het programma erg effectief. Meestal wilt u echter niet de gehele tekenset opnieuw definiëren, maar slechts een aantal tekens veranderen. U runt het programma, en wacht een ogenblik, voordat u antwoord geeft op de volgende vraag: 'Welke letter opnieuw programmeren,'.
Als u deze vraag beantwoord heeft, dan geeft u (regel 240) een enkel BIN getal voor C$, de computer verwerkt dit, en vraagt u om het volgende BIN getal. Als u klaar bent met redefiniëren, dan drukt u gewoon op de STOP-toets om het gehele proces te beëindigen. U zult nu zien, dat, als u de letter A opnieuw gedefinieerd had, u nu uw nieuwe teken krijgt, als u op de A-toets drukt. Hier is het programma in het niet-gotisch:

```
  10 CLEAR 31400
  15 PRINT "Even geduld..."
  20 LET p=15616
  30 LET a=31401
  40 FOR f=0 TO 1023
  50 POKE a+f,PEEK (p+f)
  60 NEXT f
  70 LET v=31145
  80 LET n=23606
  90 POKE n,v-256*INT (v/256)
 100 POKE n+1,INT (v/256)
 200 INPUT "Welke letter herprog
rammeren? ";a$
 210 LET a=31401+8*(CODE a$-32)
 220 FOR f=0 TO 7
 230 LET b$="BIN  "
 240 INPUT c$: IF c$=" STOP " TH
EN STOP
 245 IF LEN c$<>8 AND c$<>"" THE
N BEEP .5,20: GO TO 240
 250 LET b=VAL (b$+c$)
 255 POKE a+f,b
 260 PRINT f+1;TAB 3;c$,b
 270 NEXT f
 280 PRINT
 290 PRINT a$
```

Het is gemakkelijker om een heel alfabet opnieuw te programmeren, als u decimale invoer kunt geven, inplaats van binaire invoer. Als u echt een gotisch alfabet wilt gebruiken, dan kunt u het volgende kleine programmaatje gebruiken:

```
1000 FOR a=31400 TO 32190
2000 INPUT (a);"?",p
3000 POKE a,p
4000 NEXT a
```

Als u het uitvoert, dan vraagt het u om een enorm aantal getallen, die het als invoer gebruikt. Als u de volgende lijst getallen intikt, na het getal, waar om gevraagd wordt, dan heeft u uw gehele alfabet gereprogrammeerd tot gotische tekens. Dit is natuurlijk ideaal als u volgend jaar goedkope nieuwjaarskaarten wilt sturen.

```
31400    62          31401    0
31402    0           31403    0
31404    0           31405    0
31406    0           31407    0
31408    0           31409    0
31410    16          31411    16
31412    16          31413    16
31414    0           31415    16
31416    0           31417    0
31418    36          31419    36
31420    0           31421    0
31422    0           31423    0
31424    0           31425    0
31426    36          31427    126
31428    36          31429    36
31430    126         31431    36
31432    0           31433    0
31434    8           31435    62
31436    40          31437    62
31438    16          31439    62
31440    8           31441    0
31442    32          31443    100
31442    32          31443    100
31444    8           31445    16
31446    48          31447    70
```

31552	0	31553	0
31554	126	31555	4
31556	8	31557	28
31558	70	31559	124
31560	0	31561	0
31562	12	31563	28
31564	44	31565	126
31566	12	31567	30
31568	0	31569	0
31570	126	31571	64
31572	124	31573	6
31574	66	31575	60
31576	0	31577	0
31578	24	31579	48
31580	112	31581	76
31582	100	31583	56
31584	0	31585	0
31586	126	31587	66
31588	4	31589	24
31590	16	31591	48
31592	0	31593	0
31594	28	31595	36
31596	24	31597	28
31598	38	31599	60
31600	0	31601	0
31602	28	31603	38
31604	50	31605	14
31606	12	31607	24
31608	0	31609	0
31610	0	31611	0
31610	0	31611	0
31612	16	31613	0
31614	0	31615	16
31616	0	31617	0
31618	0	31619	16
31620	0	31621	0
31622	16	31623	16
31624	32	31625	0
31626	0	31627	4
31628	8	31629	16
31630	8	31631	4
31632	0	31633	0
31634	0	31635	0
31636	62	31637	0
31638	62	31639	0
31640	0	31641	0
31642	0	31643	16
31644	8	31645	4

31652	4	31653	8
31654	0	31655	8
31656	0	31657	0
31658	60	31659	74
31660	36	31661	94
31662	64	31663	60
31664	0	31665	0
31666	56	31667	72
31668	24	31669	40
31670	104	31671	124
31672	0	31673	32
31674	96	31675	32
31676	56	31677	52
31678	52	31679	52
31680	24	31681	16
31682	56	31683	104
31684	96	31685	96
31686	96	31687	48
31688	24	31689	16
31690	32	31691	16
31692	24	31693	44
31694	44	31695	44

31696	16	31697	16
31698	56	31699	104
31700	120	31701	96
31702	96	31703	48
31704	24	31705	14
31706	26	31707	16
31708	60	31709	16
31710	16	31711	48
31712	24	31713	16
31714	40	31715	105
31716	44	31717	28
31718	44	31719	70
31720	60	31721	32
31722	32	31723	44
31724	54	31725	34
31726	34	31727	2
31728	4	31729	16
31730	32	31731	0
31732	48	31733	88
31734	24	31735	26
31736	28	31737	24
31736	28	31737	24
31738	0	31739	56
31740	24	31741	24

31750	105	31751	93
31752	19	31753	16
31754	45	31755	16
31756	16	31757	16
31758	16	31759	28
31760	24	31761	43
31762	126	31763	170
31764	42	31765	42
31766	42	31767	106
31768	0	31769	0
31770	40	31771	116
31772	52	31773	52
31774	52	31775	52
31776	0	31777	24
31778	44	31779	105
31778	44	31779	105
31780	105	31781	105
31782	103	31783	104
31784	43	31785	56
31786	116	31787	52
31788	52	31789	56
31790	45	31791	45
31792	64	31793	0
31794	46	31795	88
31796	88	31797	88
31798	56	31799	12
31800	8	31801	0
31802	44	31803	116
31804	32	31805	32
31806	32	31807	96
31808	48	31809	2
31810	28	31811	32
31812	105	31813	118
31814	6	31815	105
31816	48	31817	16
31818	48	31819	126
31820	48	31821	48
31822	48	31823	114
31824	60	31825	0
31826	40	31827	116
31828	52	31829	52
31830	52	31831	52
31832	10	31833	0
31834	34	31835	113
31836	50	31837	50
31838	34	31839	60
31840	16	31841	0

31848	20	31849	54
31850	54	31851	24
31852	60	31853	24
31854	24	31855	42
31856	68	31857	108
31858	38	31859	38
31860	38	31861	124
31862	64	31863	48
31862	64	31863	48
31864	12	31865	126
31866	68	31867	8
31868	28	31869	6
31870	2	31871	12
31872	48	31873	0
31874	14	31875	8
31876	8	31877	8
31878	3	31879	14
31880	0	31881	0
31882	0	31883	64
31884	32	31885	16
31886	3	31887	4
31888	0	31889	0
31890	56	31891	72
31892	24	31893	40
31894	104	31895	124
31896	0	31897	32
31898	16	31899	56
31900	84	31901	16
31902	16	31903	16
31904	0	31905	0
31904	0	31905	0
31906	0	31907	0
31908	0	31909	0
31910	0	31911	0
31912	255	31913	0
31914	28	31915	34

31916	120	31917	32
31918	32	31919	126
31920	0	31921	0
31922	56	31923	72
31924	24	31925	40
31926	104	31927	124
31928	0	31929	32
31930	96	31931	32
31932	56	31933	52
31934	52	31935	52

1944	24		1945	16	
1946	32		1947	16	
1948	24		1949	44	
1950	44		1951	44	
1952	16		1953	16	
1954	56		1955	104	
1956	120		1957	96	
1958	96		1959	48	
1960	24		1961	14	
1962	26		1963	16	
1964	60		1965	16	
1966	16		1967	48	
1968	24		1969	16	
1970	40		1971	102	
1972	44		1973	28	
1974	44		1975	70	
1976	60		1977	32	
1978	52		1979	44	
1980	54		1981	34	
1982	34		1983	2	
1984	4		1985	16	
1986	52		1987	0	
1988	48		1989	88	
1990	48		1991	88	
1992	24		1993	26	
1994	0	2	1995	24	
1996	24		1997	24	
1998	24		1999	16	
2000	96		2001	32	
2002	96		2003	250	
2004	107		2005	114	
2006	108		2007	98	
2008	19		2009	16	
2010	48		2011	16	
2012	16		2013	16	
2014	16		2015	28	
2016	24		2017	43	
2018	126		2019	170	
2020	42		2021	42	
2022	42		2023	106	
2024	0		2025	0	
2026	40		2027	116	
2028	52		2029	52	
2030	52		2031	52	
2032	0		2033	24	

32040	4
32042	16
32044	8
32046	4
32048	64
32050	48
32052	8
32054	6
32056	8
32058	44
32060	32
32062	32
32064	48
32066	24
32068	10 8
32070	6
32072	48
32074	48
32076	48
32078	48
32080	60
32082	40
32084	52
32086	52
32088	10
32090	34
32092	50
32094	54
32096	16
32098	84
32100	107
32102	107
32104	20
32106	84
32108	60
32110	24
32112	68
32114	38
32114	38
32116	38
32118	64
32120	12
32122	68
32124	28
32126	2
32128	48

32041	56
32043	52
32045	60
32047	42
32049	40
32051	88
32053	88
32055	12
32057	0
32059	116
32061	32
32063	96
32065	2
32067	32
32069	118
32071	108
32073	16
32075	126
32077	48
32079	114
32081	0
32083	116
32085	52
32087	52
32089	0
32091	118
32093	50
32095	60
32097	0
32099	234
32101	107
32103	106
32105	54
32107	24
32109	24
32111	42
32113	102
32115	38
32115	38
32117	124
32119	48
32121	126
32123	8
32125	6
32127	12
32129	0

```
5CD0 21 38 5C 22 F9 41 21 40 5C 22 FB 41 21 42 5C 22
5CE0 FD 41 21 44 5C 22 FF 41 21 46 5C 22 01 42 21 48
5CF0 5C 22 03 42 21 50 5C 22 05 42 21 52 5C 22 07 42
5D00 21 54 5C 22 09 42 21 56 5C 22 0B 42 21 58 5C 22
5D10 0D 42 21 60 5C 22 0F 42 21 62 5C 22 11 42 21 64
5D20 5C 22 13 42 21 66 5C 22 15 42 21 68 5C 22 17 42
5D30 21 70 5C 22 19 42 21 72 5C 22 1B 42 21 74 5C 22
5D40 1D 42 21 76 5C 22 1F 42 21 78 5C 22 21 42 21 80
5D50 5C 22 23 42 21 82 5C 22 25 42 21 84 5C 22 27 42
5D60 21 86 5C 22 29 42 21 88 5C 22 2B 42 21 90 5C 22
5D70 2D 42 21 92 5C 22 2F 42 21 94 5C 22 31 42 21 96
5D80 5C 22 33 42 21 98 5C 22 35 42 C9 00 00 00 00 00
```

```
5CD0 21 39 5C 22 F9 41 21 41 5C 22 FB 41 21 43 5C 22
5CE0 FD 41 21 45 5C 22 FF 41 21 47 5C 22 01 42 21 49
5CF0 5C 22 03 42 21 51 5C 22 05 42 21 53 5C 22 07 42
5D00 21 55 5C 22 09 42 21 57 5C 22 0B 42 21 59 5C 22
5D10 0D 42 21 61 5C 22 0F 42 21 63 5C 22 11 42 21 65
5D20 5C 22 13 42 21 67 5C 22 15 42 21 69 5C 22 17 42
5D30 21 71 5C 22 19 42 21 73 5C 22 1B 42 21 75 5C 22
5D40 1D 42 21 77 5C 22 1F 42 21 79 5C 22 21 42 21 81
5D50 5C 22 23 42 21 83 5C 22 25 42 21 85 5C 22 27 42
5D60 21 87 5C 22 29 42 21 89 5C 22 2B 42 21 91 5C 22
5D70 2D 42 21 93 5C 22 2F 42 21 95 5C 22 31 42 21 97
5D80 5C 22 33 42 21 99 5C 22 35 42 C9 00 00 00 00 00
```

Machine Code Renumber

Dit is een code-programma, dat een BASIC-programma opnieuw regelnummers geeft, en dat deze regelnummers ook aanpast in GOTO, GOSUB, RESTORE, LINE en RUN-opdrachten. Het is geschreven op adres 5CDO, dit is de eerste regel van een programma. Het adres kan veranderd worden (de te veranderen adressen zijn onderstreept), maar dit kan ik u niet aanbevelen, omdat het wel erg veel werk is, en de kans op fouten niet gering.

SUB en RESTORE-bytes door het gehele programma, en let daarbij op, dat ze niet behoren tot een REM-opdracht, een PRINT-opdracht, een regel-nummer, of de lengte van een regel. Als het einde van het programma gevonden wordt, dan is het opnieuw genummerd. Als een GOTO of een GOSUB of een van de andere bytes gevonden wordt, dan kijkt het programma, of er in de volgende 5 bytes een CHR$ voorkomt. Als dit niet het geval is, dan begint het programma weer vanaf het begin. Het programma laadt nu het nummer in van variabele A.

2. Het programma POKEd het getal terug in het tweede gedeelte van het programma – nu in hexadecimaal. Als het de positie vindt van regelnummers na bytes zoals GOTO of GOSUB, en deze regelnummers zijn groter dan het laatst-geconverteerde regel-nummer, dan gaat het programma weer verder bij stap 1. Het converteert het regelnummer na de GOTO of GOSUB in de nieuwe bestemming, en dit getal wordt in een string opgeslagen in A$. Nu laadt het de volgende vijf bytes na de CHR$ 14.

3. Nu schrijft het programma de string in de positie direkt volgende op de GOSUB of GOTO. Het verwijdert spaties, of voegt deze toe, zodat de string 'netjes' gepositioneerd wordt, en tegelijkertijd worden de systeemvariabelen en de lengte van de regel aangepast. Dan gaat het programma weer verder met stap 1. Om het programma te gebruiken, moet u eerst de BASIC Hex-loader van de cassette laden. Als u dit programma uitvoert, dan tikt u de paren Hex-code in, zoals deze in de hierna volgende lijst gegeven wordt. De hex-lader accepteert ieder aantal bytes tegelijk, en de code moet ingetikt worden zonder spaties tussen de paren data. De code maakt gebruik van kleine letters, en geeft u de mogelijkheid om onjuist ingetikte data opnieuw op te

Controleer het volgende:

PEEK 23765 moet geven: 33
PEEK 23891 moet geven: 11
PEEK 23993 moet geven: 33
PEEK 24072 moet geven: 42

Verwijder nu alle regels uit het originele programma, behalve regel 1. Tik listing 2 in. Merk op dat regel 5 de stap-grootte en het eerste regelnummer bevat. In dit programma zijn deze beide op 10 gezet, maar u kunt deze instellen zoals u zelf wilt. Ze moeten echter beide 4 tekens lang zijn, en daarom voorloopnullen bevatten. Als u deze waarde probeert te veranderen, dan moet u adres 23901 POKEn met het eerste regelnummer MIN de stapgrootte, en 23909 met de stapgrootte. Geef daarna RUN om een programma opnieuw te nummeren. U kunt het RENUMBER-programma het beste op een aparte cassette bewaren, en het met het programma, dat u in de computer heeft, MERGEn op het moment, dat u het nodig heeft. Listing 1 is de Hexloader, die op de cassette staat:

```
 1 REM

 2 LET b$="0123456789abcdef"
 3 FOR a=23760 TO 24131
 4 INPUT a$
 5 PRINT a$( TO 2);"   ";
 6 LET c$=a$(1)
 7 GO SUB 19
 8 IF b=16 THEN GO TO 4
 9 LET c=b*16
10 LET c$=a$(2)
11 GO SUB 19
12 IF b=16 THEN GO TO 4
13 POKE a,c+b
14 LET a$=a$(3 TO )
15 LET a=a+NOT NOT LEN a$
16 IF LEN a$ THEN GO TO 5
17 NEXT a
18 STOP
```

Hier volgen de bytes Hex code die u in moet tikken:

```
48 59 61 63 65 21 19 5F
03 23 23 23 22 E0 5C C9
1D 5F 18 60 FE EC C8 FE
ED C8 FE E5 C8 FE CA C8
FE F7 C9 E8 2A 48 5C 37
ED 52 EB C9 E1 7E E5 06
06 21 CF 5C 23 E5 6E 26
5C A7 28 07 34 20 02 2C
34 18 05 35 20 02 2C 35
E1 10 E9 C9 D5 CD 0C 5E
00 2A 65 5C A7 ED 52 44
4D 62 6B 23 ED B0 CD FC
5C 00 D1 C9 C5 D5 E5 2A
65 5C E5 23 A7 ED 52 44
4D 19 D1 E6 ED B8 CD FC
5C 40 CD 0C 5E 40 E1 23
D1 C1 C9 08 2A E0 5C 7E
CD F3 5C 30 1D 01 00 00
2A D6 5C E5 21 0A 00 09
44 4D E1 CD F3 5C D8 70
23 71 23 5E 23 56 23 19
18 E9 23 11 04 00 FE EA
20 04 3E 0D 18 04 FE 22
```

```
0D 20 05 FE 0E 20 04 13
10 18 00 CD E4 5D 20 B7
00 E0 5C 3E 0E 01 05 00
ED B1 20 A7 22 E2 5C ED
5B 4B 5C 13 0E 05 ED B0
C9 21 D1 5D 36 D8 2A D6
5C 01 00 00 03 56 23 5E
03 E5 21 0A 00 37 ED 52
E1 D8 5E 23 56 23 19 CD
F3 5D 30 E8 01 00 00 C9
0A 4B 5C 23 ED 5B E2 5C
01 05 00 ED B0 ED 5B E0
5C 03 4E 23 46 23 1A FE
0E CC 34 5D ED A0 78 B1
00 F4 1A FE 0E C8 CD 1C
5D 18 F7 C9 21 D1 5D 36
C8 2A E0 5C 3E 0D ED B9
E5 D5 23 23 46 23 4E 03
E0 40 CB 5D CD BE 5D D1
E1 78 B1 28 E7 01 04 00
09 01 0A C5 A7 28 07 34
00 00 23 34 18 05 35 20
02 23 35 C9
```

En dit is het renumber-programma:

```
6: POKE 24012,INT (a/256): LET a
=USR 23683: IF NOT a THEN GO TO
3
    5 LET a=(a-1)*0010+0010: LET
a$=STR$ a: GO TO 3 AND NOT USR 2
4032
    6 REM
```

Programma auteurs

Met dank aan de volgende programmeurs, die een enorme hulp zijn geweest bij het samenstellen van dit boek met interessante spelletjes:

LEVEL 5 - Paul Toland
CIRCUS - Paul Toland
JOGGER - Tim Hartnell
MIJNEN - Paul Toland
COPTER - Tim Rogers
MAZURKA - Paul Toland
ALPHABATTLE - Tim Rogers
SMASHOUT - Ant Hurrion (verbeterd door Tim Hartnell)
3-D DRIVE - Tim Rogers
NORWAY PANIC - Tim Rogers
DEATHRACE 2000 - Tim Hartnell en Tim Rogers
CLONE CRAZY - Mike O'Neill
PIRANDELLO - Graham Charlton (veranderd door Tim Hartnell)
DAMMEN - Tim Hartnell
BOTER, KAAS EN EIEREN - Tim Hartnell
BEGINNERS SCHAAK - Tim Hartnell
DE WRAAK VAN KASTEEL ANGST - Tim Hartnell
JOE KAPITALIST JUNIOR - Paul Holmes
DE PUT - Tim Rogers
DE GOTISCHE GENERATOR - Tim Rogers

Bibliografie

Er zijn veel boeken, waaruit u ideeën op kunt doen om op de Spectrum spelletjes te schrijven. Deze lijst bevat enkele van de beste boeken, die ik heb kunnen vinden:

Lezers, die meer van programmeren op de Spectrum willen weten, kunnen wellicht veel nuttige informatie vinden in het zakboek Sinclair Spectrum, ook uitgebracht door Delfia Press.

Computerspellenboeken

*40 Computer Games from kilobaud microcomputing** Gibbs, E. A. en Perry, J. (eds.) (Wayne Green Inc., USA 1980)

*The Softside** *Sampler* Witham, J. (ed.) (Hayden Book Co., Inc., USA 1982)

BASIC Computer Games Ahl, D. (ed.) (Workman Publishing Co., USA 1980)

More BASIC Computer Games Ahl, D. (ed.) (Workman Publishing Co., USA 1980)

Algemene computerboeken

The Way To Play the Diagram Group (eds.) (Corgi, 1977)

The Indoor Games Book Pennycook, A. (Faber & Faber Ltd., 1973)

Games for Two Wasley, J. (Proteus Publishing Ltd., 1981)

Everyman's Indoor Games Brandreth, G. (J. M. Dent & Sons Ltd., Everyman Paperback, 1982)

Dice Games, New and Old Tredd, W. E. (The Oleander Press, Cambridge, 1981)

Discovering Old Board Games Bell, R. C. (Shire Publications Ltd., 1980)

DELFIA PRESS BV

In dit boekje beschrijft Tim Hartnell twintig boeiende spellen voor de ZX Spectrum, die variëren van bordspelletjes tot aktiespelletjes en worden afgesloten met een fantastisch avontuur.
Elk spel wordt gedetailleerd beschreven, waarbij het programma regel voor regel wordt toegelicht.
De toegepaste trucs en technieken worden afzonderlijk uitgelegd, zodat u ze zelf in uw eigen programma's kunt toepassen.
In het laatste hoofdstuk volgen suggesties hoe u de op de cassette bijgeleverde programma's kunt verbeteren en aanpassen aan uw eigen omstandigheden.
(ook geschikt voor de Spectrum+)

PROGRAMMA'S

- LEVEL-5
- Circus
- Jogger
- Mijnenveger
- Copter
- Mazurka
- Alphabattle
- Smashout
- 3-D Drive
- Norway Panic
- Death Race 2000
- Clone Crazy
- Pirandello
- Dammen
- Boter, kaas en eieren
- Beginners schaken
- De wraak van Kasteel Angst
- Joe Kapitalist Junior
- De Put
- Verbeter Uw programma's
- Verhogen van de programma-snelheid
- Gotische Generator
- Machine Code Renumber

ISBN 906449 0376

Tim Hartnell

20 dynamische spellen voor de zx spectrum

DELFIA PRESS BV

20 dynamische spellen voor de zx spectrum

Tim Hartnell