

Computer Games Series

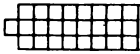
GIOCHIAMO CON ZX SPECTRUM

GREMESE EDITORE

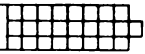


Computer Games Series

GIOCHIAMO
CON
ZX SPECTRUM



GIOCHIAMO CON ZX SPECTRUM



GIOCHIAMO CON ZX SPECTRUM

Peter Shaw

GREMESE EDITORE

Computer Games Series

Periodico mensile

N. 3 - Maggio 1984

Registrazione Trib. di Roma N. 138/84
del 24 marzo 1984

Direttore responsabile: Grazia Valci

Titolo originale

Games for your ZX Spectrum

Traduzione dall'inglese

Lucia Perri

Edizione italiana a cura di

Giancarlo Zagarese

Design

Ray Hyden

Illustrazioni

Sue Walliker

Fotocomposizione

Typo-centro s.n.c. - Roma

Stampa

Litopat - Verona

© 1983 Interface/Virgin Books

© 1984 GREMESE EDITORE s.r.l.

Via Virginia Agnelli, 88 - 00151 Roma

Tutti i diritti riservati. Nessuna parte di questo libro può essere riprodotta, registrata o trasmessa, in qualsiasi modo o con qualsiasi mezzo, senza il preventivo consenso dell'Editore.

ISBN 88-7605-118-1

PETER SHAW - AUTORE

Peter Shaw è uno studente sedicenne che ha collaborato con la rivista Interface e con ZX Computing.

**TIM HARTNELL
CURATORE DELL'EDIZIONE INGLESE**

Tim Hartnell è un eminente giornalista la cui esperienza computeristica ha contribuito a determinare il successo della Technical Consumer Press. È anche l'autore di diversi libri, tra i quali: *Getting Acquainted With Your ZX81*, *Let Your BBC Micro Teach You to Program* e *Programming Your ZX Spectrum*.

**GIANCARLO ZAGARESE
CURATORE DELL'EDIZIONE ITALIANA**

Giancarlo Zagarese, insegnante di discipline scientifiche, è autore di oltre 250 articoli e di vari volumi nei settori dell'elettronica e dell'attività subacquea. Per GREMESE EDITORE ha già collaborato in *Il Sub per tutti* e *Il Sub in apnea* della serie "gli Abbicci".

SUE WALLIKER-ILLUSTRATRICE

Sue Walliker è un'illustratrice free-lance.

RINGRAZIAMENTI

L'autore desidera ringraziare Maureen Gates per l'aiuto prestato nella stesura del libro; Alan Dennis, Steven Gunning e Michael Merrifield per il contributo di idee portato ai programmi; e Tim Hartnell e Clive Gifford per l'incoraggiamento. Un grazie particolare a Mark e David Palmer che, nel loro piccolo, hanno reso possibile questo libro. Per l'edizione italiana si ringrazia Marco Perri dell'ITC "Michelangiolo" di Roma.

INDICE

Introduzione	9
Introduzione dell'autore	10
Prefazione all'edizione italiana	11
Abilità e tecniche di programmazione	13
Egg Saviour (Il salvatore di uova)	15
Box Bounce (Muro di rimbalzo)	21
Nessie	24
Spider'n'Fly (Il ragno e le mosche)	28
Ascot	31
Munchie Man (Il divoratore)	36
Track Trap	41
Pontoon (Il gioco del "Ventuno")	43
Breakout	47
Excalibur Against the Enemy (Excalibur contro il nemico)	50
Leaky Roof (Il tetto che pende)	53
Touchdown on Mars (Atterraggio su Marte)	55
Orchard Thief (Il ladro del frutteto)	58
Charge of the Teddy (La carica degli orsacchiotti)	61
Snake in the Triangles (Il serpente nei triangoli)	65
Death Race (La corsa della morte)	67
Draughts (La dama)	70
Simon Said	76
3D Driver (Il guidatore in 3D)	78
Drop Out	80
M4	82
Hangman (L'impiccato)	84
Video Salesman (Il venditore di video)	88
Dam Busters	90
Come scrivere programmi migliori	93
Glossario	101
Traduzioni	116

Introduzione

Il vostro computer è in attesa per sfidarvi. Rapidi «games» di grafica, giochi di concentrazione, di parole e di enigmistica sono tutti qui pronti a farvi divertire.

Nel libro vi sono una notevole varietà di giochi i cui programmi sono stati scritti da alcuni fra i più giovani e abili programmatori che lavorano attualmente in Gran Bretagna. L'esaminare i programmi dei giochi vi dà la possibilità di apprendere raffinate tecniche e sottili metodi di programmazione che voi stessi potete poi applicare. Inoltre, una volta che avete conosciuto a fondo i programmi presentati dal libro, potreste senz'altro provare a migliorarli — un programma non è mai "perfetto" — arricchendo le vostre capacità di programmazione. Adesso voltate pagina e incominciate a "battere" i programmi. Sono certo che il vostro divertimento sarà pari a quello che abbiamo provato durante la stesura di questo volume.

Introduzione dell'autore

Lo Spectrum è un elaboratore eccezionale. Al suo confronto, lo ZX 81 è lento e poco evoluto, poiché l'unico modo di giocare un game tipo "Space Invader" è quello di inserirsi nel codice macchina. Lo Spectrum offre la possibilità di programmare diversi giochi di qualità "Arcade" in BASIC. I programmi di questo libro sono essenzialmente giochi. Ho preferito evitare programmi del tipo "Bioritmi" o "Disegni", non per avversione contro questa categoria, ma semplicemente perché si possono trovare, ormai, in qualsiasi rivista.

Quando avrete «battuto» questi programmi, non fermatevi, ma cercate di migliorarli. Se trovate un carattere definito per l'utente, e voi pensate di poterlo fare con un bit in meno, allora cambiatelo!

Prefazione all'edizione italiana

È con piacere che ho accolto l'invito dell'editore Gremese per curare una serie, finalmente in italiano, di volumetti sui videogiochi. Molti acquirenti di piccoli e medi personal computer, sia che lo utilizzino personalmente sia che ne abbiamo fatto oggetto di regalo ai propri figli si sono fatti trascinare dalla pubblicità che precisava «al prezzo di un semplice video-gioco acquistate un intero computer». È vero ed è stato un buon acquisto. Per imparare ad utilizzare un computer in modo semplice e divertente non c'è però niente di meglio della via ludica. Non giochi comprati e fruiti passivamente, però, ma "creati" e vissuti, istruzione dopo istruzione prima compiendo e poi personalizzando o "inventando", in modo da accedere gradualmente nel nuovo mondo dell'informatica.

g.z.

Abilità e tecniche di programmazione

La maggior parte dei miei programmi inizia con questo schema tipico:

```
10 REM Titolo del programma
20 GOSUB 9000: REM UDGs
30 GOSUB 8000: REM Variabili
40 GOSUB 7000: REM Tracciato schermo
```

Anche se si può rendere tutto ancor più facile aggiungendo delle istruzioni quando al computer viene dato il comando GO TO, comunque il computer inizia dalla istruzione uno, finché trova quella giusta. Se si scrivono giochi, come quelli di azione, tipo "Arcade", in cui la velocità è essenziale, è preferibile usare questo metodo di subroutine frazionate per accelerare il movimento.

Consigli e avvertimenti

Lo Spectrum ha una certa quantità di segreti, che non sempre sono ben evidenziati nel manuale di istruzione; alcuni di essi non sono neanche menzionati. Il simbolo # può essere usato nelle istruzioni PRINT per stampare in diverse posizioni sullo schermo:

```
PRINT # 1; "È stampato nella parte inferiore dello schermo"; PAUSE 0
```

Come si vede # 1 stampa nella parte inferiore dello schermo, rendendo possibile una visualizzazione di 24 righe.

I computer ZX sono microelaboratori che accettano gli INPUT solo nella parte bassa dello schermo; tuttavia, usando INPUT AT, si possono inserire informazioni in qualsiasi parte dello schermo.

```
INPUT AT 22,0; AT 0,0; "Come ti chiami";
LINE a$; AT 10,0; "Quanti anni hai"; (a$);
""; a; AT 15,0; (a$); "L'età di...è..."; a; AT 20,0;
"Premi ENTER per continuare"; b$.
```

Con questo metodo, però, ci sono dei problemi. Innanzitutto la lunghezza delle righe e, inoltre, il colore del BORDER

deve essere lo stesso del PAPER, perché la metà inferiore dello schermo si prolunga quasi fino a tutta la parte alta. (Dico quasi, perché rimane una striscia della parte superiore che è chiaramente visibile, a meno che i colori del fondo e del bordo non siano gli stessi).

In tutto il libro ho cercato di usare UDGs, ogni volta che mi è stato possibile. Ho avuto grande aiuto dal "Print 'n' Plotter Jotter". È un "modello" con due griglie di coordinate su ogni pagina: la parte quadrettata in basso è 32x22, ed è utile per disegnare la visualizzazione. La parte quadrettata in alto è 64x44, ed è quella che ho usato per definire i miei caratteri. Potete usare anche la carta millimetrata, ma i quadrati sono così piccoli che UDGs diventa troppo piccolo e difficile da definire.

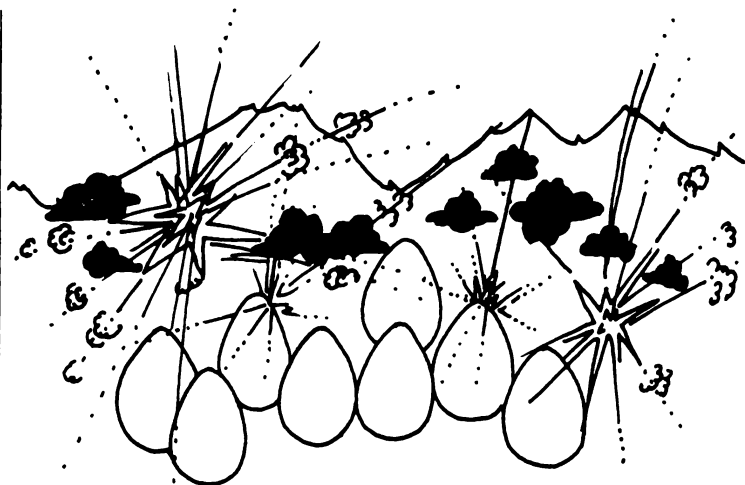
EGG SAVIOUR

In questo gioco siete "Salvatori di uova"; il vostro compito è quello di salvare le povere, indifese, uova dalle grinfie di poteri tirannici e maligni, che limitano i vostri movimenti sparando raggi laser. Raccogliete le uova toccando la 'X' sotto la scatola di uova che si trova in alto nello schermo; quando le avrete raccolte, il vostro uovo diventerà verde. A questo punto dovete superare i raggi laser e mettere l'uovo nella vostra scatola correndo sopra '-'. Quando avrete raccolto una mezza dozzina di uova, guadagnerete 100 punti e un nuovo carico di uova.

Regole del gioco:

- 1) Non si può passare sopra qualcosa che non sia 'X' o '-', altrimenti si perde una "vita".
- 2) Se si è colpiti da un laser, o si perde una "vita" in qualche altro modo, si perde anche l'uovo che si porta in quel momento; se non si porta alcun uovo, si perde solo una "vita".
- 3) Si può portare un solo uovo per volta; se state portando un uovo e cercate di prenderne un altro, la 'X' sparirà, e renderà così impossibile la raccolta dell'uovo nella scatola sotto la 'X'.
- 4) Avete tre "vite", della durata di circa dieci secondi, quando giocate per primi; con un po' di pratica potrete totalizzare un punteggio ragionevole.

Usate i tasti del cursore per controllare i vostri movimenti.



```

10 REM Egg saviour
20 REM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
30 REM Peter Shaw
40 REM Original idea by
   Micheal Merrifield
50 REM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
55 LET hi=0
60 GO SUB 9000
70 GO SUB 8000
80 GO SUB 7000
150 PRINT AT h,v;" "
160 LET h=h+(INKEY$="6" AND h<2
1)- (INKEY$="7" AND h>0): LET v=v
+(INKEY$="8" AND v<31)- (INKEY$="
5" AND v>0)
170 IF SCREEN$ (h,v)<>" " THEN
GO SUB 6000
180 IF AND>.5 THEN GO SUB 6500
190 PRINT AT h,v; INK 5+eggs;"X"
"
191 BEEP .008,0
200 IF lives<1 THEN PRINT AT 21
,0;" You Have been killed
": GO TO 1500
210 PRINT AT 21,0;"Lives left "
;lives
;230 PRINT AT 21,15;"Eggs collec
ted ":tot
998 GO TO 150
999 STOP

```

EGG SAVIOUR

```

1500 BEEP 1,4: BEEP 1,4: BEEP .3
.4: BEEP 1.2,4: BEEP .75,7: BEEP
.5,6: BEEP 1,6: BEEP .3,4: BEEP
.7,4: BEEP .5,3: BEEP 1,4
1501 IF sc>hi THEN LET hi=sc
1510 INPUT "Press ENTER to play
again "; LINE a$: CLS : GO TO 70
2000 IF eggs=1 THEN RETURN
2005 PRINT AT 1,v;" " : AT 2,v; I
NK 6;" "
2010 LET eggs=1
2015 BEEP .1,10
2020 RETURN
2500 IF eggs=3 THEN PRINT AT h,v
: INK 1;"-": LET h=h+1: RETURN
2510 PRINT AT 17,v; INK 6;"▲": A
T 18,v; OVER 1; INK 6;"●"
2515 LET eggs=0
2516 PRINT AT 19,v;" "
2517 BEEP .1,20: BEEP .15,15
2520 LET tot=tot+1: IF tot=5 THE
N GO TO 9500
2530 RETURN
6000 LET b$=SCREEN$(h,v)
6010 IF b$="X" THEN GO TO 2000
6020 IF b$="-" THEN GO TO 2500
6030 LET lives=lives-1
6031 LET eggs=0
6040 PRINT AT h,v; FLASH 1; INK
3;"X": BEEP .1,-10: FOR p=1 TO 2
0: NEXT p: RETURN
6500 GO TO (INT (AND*3)+1)*100+5
500
6600 PLOT 9,#1: DRAW INK 2; BRIG
HT 1;245,0
6605 GO SUB 6900
6606 PAUSE 2
6610 DRAW OVER 1;-248,0: PLOT DV
ER 1;254,#1: RETURN
6700 PLOT #2,167: DRAW INK 2; BR
IGHT 1;0,-120
6705 GO SUB 6900
6706 PAUSE 2
6710 DRAW OVER 1;0,120: RETURN
6800 PLOT #3,167: DRAW INK 2; BR
IGHT 1;0,-120
6805 GO SUB 6900
6806 PAUSE 2
6810 DRAW OVER 1;0,120: RETURN
6900 IF SCREEN$(h,v)<>" " THEN
GO TO 6000
6910 RETURN
7000 LET n=0: PRINT AT 0,0;"Score
e ";sc;TAB 23;"High ";hi
7005 FOR a=1 TO 4

```

```

7010 PRINT AT 1,n;" ";
7020 FOR b=1 TO e(a)
7030 PRINT INK 6;"▲";
7040 NEXT b
7050 PRINT AT 2,n;" ";
7060 FOR b=1 TO e(a)
7070 PRINT INK 6;"▼";
7080 NEXT b
7090 OVER 1
7095 PRINT AT 2,n;" ";
7100 FOR b=1 TO e(a)
7110 PRINT INK 6;"┌";
7120 NEXT b
7122 PRINT AT 3,n;" ";
7125 FOR b=1 TO e(a): PRINT INK
6;"X ";: NEXT b
7130 LET n=n+1e(a)*2)+2
7140 NEXT a
7150 PRINT AT 0,9; INK 2;"▼";AT
0,21;"▼";AT 4,0;"▲"
7160 PRINT AT 18,8; INK 1;"┌┌┌┌
┌┌┌┌"
7165 PRINT AT 19,8; INK 1;"- -
- -"
7170 OVER 0
7990 RETURN
8000 BORDER 0: INK 7: PAPER 0: C
LS
8010 DIM e(4)
8020 RESTORE 9120: FOR z=1 TO 4:
READ e(z): NEXT z
8030 LET h=10: LET v=15
8040 LET sc=0: RANDOMIZE
8060 LET #1=17*8+4: LET #2=9*8+4
: LET #3=21*8+4
8070 LET eggs=0
8080 LET lives=3
8090 LET tot=0
8100 RETURN
8989 REM *****
8990 REM Graphics characters
8991 REM      ● Graphic ab
8992 REM      ● Graphic cd
8993 REM      ┌ Graphic ef
8994 REM      ‡ Graphic g
8995 REM      ▶ Graphic h
8996 REM      ▼ Graphic i
8997 REM *****




```

EGG SAVIOUR

```

9000 FOR a=USR "a" TO USR "j"-1
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 1,3,7,7,15,15,15,31
9040 DATA 128,192,224,224,248,24
0,248,248
9050 DATA 31,31,31,31,15,15,7,3
9060 DATA 248,248,248,248,248,24
0,224,192
9070 DATA 192,192,192,192,192,22
4,224,255
9080 DATA 3,3,3,3,3,7,7,255
9090 DATA 56,56,16,254,16,16,48,
68
9100 DATA 128,192,240,255,255,24
0,192,128
9110 DATA 255,126,60,60,60,24,24
,24
9111 REM *****
9120 DATA 3,2,2,3
9500 LET sc=sc+100
9505 LET h=10: LET v=15: LET tot
=0: LET eggs=0
9510 PRINT AT 0,0;"Score ";sc
9515 IF sc>hi THEN LET hi=sc
9520 FOR p=1 TO 31: BEEP .01,p:
BEEP .006,-p: NEXT p
9530 CLS
9540 GO TO 80

```

Score 0  Y  High 0  0

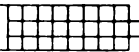
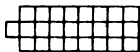
X X X X X X X



+



Lives left 1 Eggs collected 2



Score @ Y High @

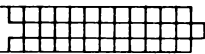
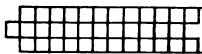


Lives left 1 Eggs collected 2

Score @ Y High @

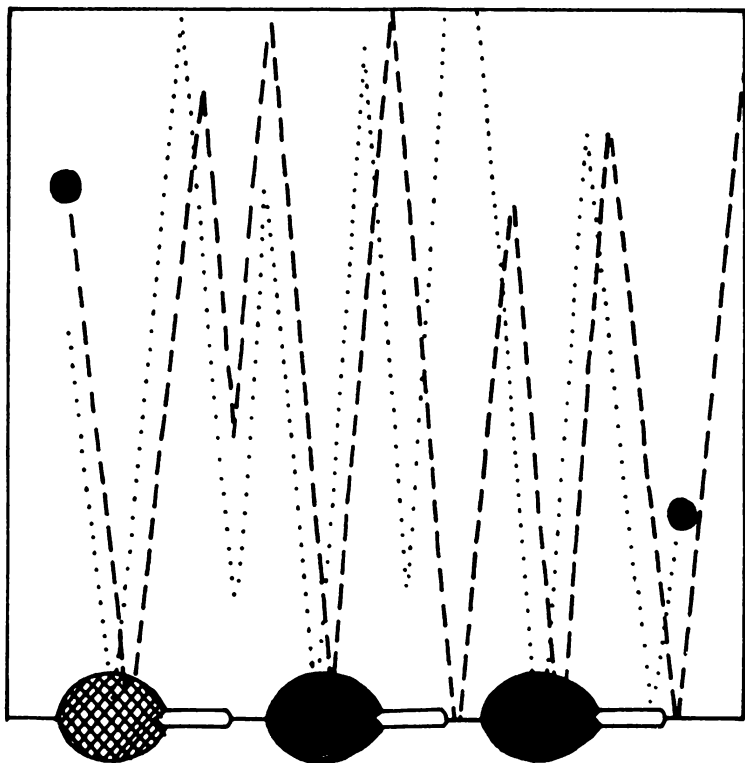


You Have been killed



BOX BOUNCE

Lo scopo del gioco è prendere la palla che rimbalza qua e là sullo schermo; totalizzate dei punti quando riuscite a respingere la palla dal vostro campo (il quadrato nella parte bassa dello schermo). Se però la palla tocca la parte laterale, perdete una *vita*. Avete tre *vite*; vi sfido a battere il mio record personale di 20 palline. Usate il tasto 5 per muovere a sinistra, l'8 per muovere a destra.



Score 2



Lives left 3

Score 6



Lives left 1

Score 2



Lives left 3

```

10 REM box-bounce
20 REM Peter Shaw
30 REM Original idea by
   Alan Dennis
40 LET hi=0
50 GO SUB 9000
60 GO SUB 8000
65 PRINT AT 21,15;"Lives left
"; lives
70 PRINT AT 20,v;" "
80 LET v=v+(INKEY$="8" AND v<0)
1) -(INKEY$="5" AND v>0)
85 LET v=v+2*(INKEY$=CHR$ 9 AN
D v<31)-2*(INKEY$=CHR$ 8 AND v>0)
)
90 PRINT AT 20,v; INK 5; BRIGHT
T 1;"□"
100 PRINT AT a,b;" "
```


BOX BOUNCE

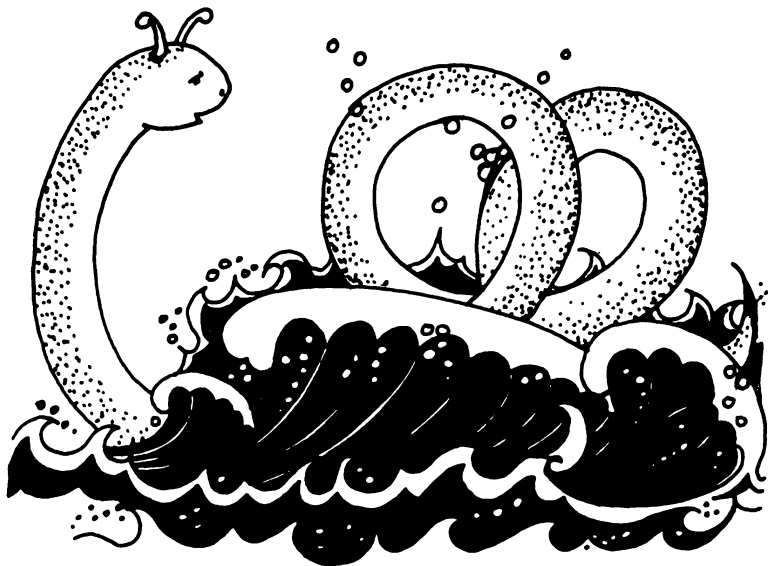
```

110 LET a=a+m: IF a>19 OR a<1 T
HEN BEEP .05,10: LET m=-m
120 LET b=b+n: IF b>30 OR b<1 T
HEN BEEP .05,15: LET n=-n
130 PRINT AT a,b: INK c;"●"
140 IF a=0 OR a=20 THEN PRINT A
T a,b: INK c;" "
150 IF b=0 OR b=31 THEN PRINT A
T a,b: INK c;" "
160 IF a=19 AND b=v THEN PRINT
AT a,b: INK c;"●": BEEP .05,15:
LET sc=sc+2: LET m=-m
170 IF a=20 AND b=v THEN GO SUB
500
180 PRINT AT 21,0:"Score ";sc
190 GO TO 70
500 PRINT AT 20,v: FLASH 1: INK
2:"□"
510 LET lives=lives-1
515 PRINT AT 21,15:"Lives left
";lives
520 FOR p=1 TO 20: BEEP .005,-p
: NEXT p
525 IF lives=0 THEN GO TO 600
526 LET a=INT (RAND*5)+2: LET b=
INT (RAND*27)+2
530 RETURN
600 PRINT AT 2,10: FLASH 1: BRI
GHT 1:"GAME OVER"
610 PRINT " " " You scor
ed ";sc
620 IF sc>hi THEN LET hi=sc
630 PRINT " " " Highest score
today ";hi
640 INPUT "Press ENTER to play
again "; LINE a$: GO TO 50
8000 BORDER 1: PAPER 0: INK 7: C
Ls
8005 LET v=15: LET a=INT (RAND*5)
+2: LET b=INT (RAND*27)+2
8010 LET sc=0: LET m=1: LET n=1
8020 LET c=6: LET lives=3
8090 RETURN
9000 FOR a=USR "a" TO USR "d"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 60,126,255,255,255,255
,126,60
9040 DATA 0,0,0,60,126,255,126,6
0
9050 DATA 6,26,62,62,62,62,26,6
9060 DATA 255,129,129,129,129,129,12
9,129,255

```

NESSIE

È una variante del Duckshoot (Caccia all'anatra), in cui i vostri bersagli attraversano a nuoto lo schermo, sotto di voi. Per totalizzare punti, dovete uccidere i bersagli; più ne colpite, più punti guadagnate. C'è un premio se riuscite ad abbattere l'intera fila. Usare il tasto 5 per andare a sinistra, l'8 per andare a destra, lo 0 per sparare.



BONUS 2000 POINTS

SCORE 3160

MISSILES 0

GAME OVER

You scored 3160

Highest score today 3160

```

10 REM NESSIE
20 LET hi=0: GO SUB 9000
30 GO SUB 8000
40 LET s=0: LET mis=20: GO SUB
7000
50 PRINT AT 5,0: a$;'b$
60 LET h=h+(INKEY$="8" AND h<6
0)-(INKEY$="5" AND h>0)
70 IF INKEY$="0" THEN LET mis=
mis-1: GO SUB 1000
75 IF mis<1 THEN GO TO 500
80 PRINT AT 20,h: INK 6;"
90 LET b$=b$(2 TO )+b$(1)
100 LET a$=a$(LEN a$)+a$( TO L
EN a$)-1)
101 PRINT AT 0,0: PAPER 5: INK
0;"SCORE ";s,"MISSILES ";mis;" "
110 IF b$="
120 IF a$="
140 GO TO 50
    
```

```

500 PRINT AT 10,12; FLASH 1;"GAME OVER"
501 PRINT AT 0,0; PAPER 5; INK 0;"SCORE ";s,"MISSILES ";#i;" "
510 PRINT AT 12,10;"You scored ";s
520 IF s>hi THEN LET hi=s
525 PRINT "TAB 4;"Highest score today ";hi
530 INPUT "PRESS ENTER TO PLAY AGAIN "; LINE a$: GO TO 30
1000 LET b=h+2
1010 FOR f=19 TO 4 STEP -1
1012 PRINT AT f,b;"*"
1019 BEEP .001,-(f-30)
1020 PRINT AT 5,0;a$'b$
1030 LET h=h+(INKEY$="8" AND h<0)-(INKEY$="5" AND h>0)
1040 PRINT AT 20,h; INK 5;"
1050 LET b$=b$(2 TO )+b$(1)
1060 LET a$=a$(LEN a$)+a$(TO (LEN a$)-1)
1070 IF f=5 THEN IF a$(b)<>" " THEN GO TO 4000
1080 IF f=7 THEN IF b$(b)<>" " THEN GO TO 5000
1090 PRINT AT f,b;" "
1100 NEXT f: RETURN
2000 PRINT AT 10,7; FLASH 1;"BONUS 1000 POINTS"
2010 LET b$="
2011 LET s=s+1000
2015 BEEP .1,20; BEEP .2,15
2020 FOR p=1 TO 150: NEXT p
2040 PRINT AT 10,7;"
": RETURN
3000 PRINT AT 10,7; FLASH 1;"BONUS 2000 POINTS"
3010 LET a$="
3011 LET s=s+2000
3015 BEEP .1,20; BEEP .2,15
3020 FOR p=1 TO 150: NEXT p
3030 PRINT AT 10,7;"
": RETURN
4000 LET a$(b-4 TO b+4)="
4010 LET s=s+20
4011 BEEP .05,-5
4020 RETURN
5000 LET b$(b-4 TO b+4)="
5010 LET s=s+10

```

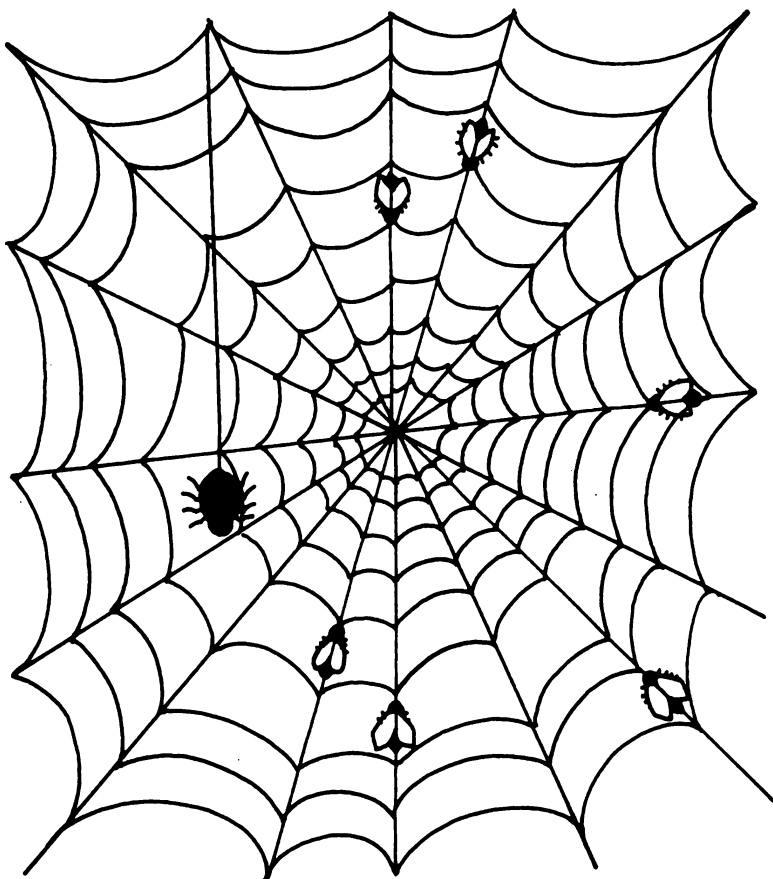
```

5011 BEEP .05,-5
5020 RETURN
7000 BORDER 0: PAPER 1: INK 4: C
LS
7010 FOR a=0 TO 3: PRINT PAPER 5
;
;: NEXT a
7020 RETURN
8000 LET a$="~~~~~"
8010 LET b$="~~~~~"
8020 LET h=15
8022 LET s=0
8030 RETURN
8900 REM *****
8910 REM
8920 REM Graphics
8930 REM
8940 REM ~~~~~ - abc
8950 REM
8960 REM ~~~~~ - defg
8970 REM
8980 REM * - h
8990 REM
8999 REM *****
9000 FOR a=USR "a" TO USR "h"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 0,0,0,127,255,255,255,
255
9040 DATA 0,24,60,255,255,255,25
5,255
9050 DATA 0,0,0,254,255,255,255,
255
9060 DATA 0,96,248,252,127,63,15
,7
9070 DATA 1,15,31,63,255,252,240
,192
9080 DATA 128,240,248,255,255,63
,31,7
9090 DATA 2,15,31,255,255,252,24
8,192
9100 DATA 24,60,60,24,60,36,0,0

```

SPIDER 'N' FLY

Siete un ragno molto affamato che pende da un filo sulla sinistra dello schermo, e le sei mosche grassottelle che si vedono nel disegno sono i vostri bersagli. Avete 99 secondi per mangiare quante più mosche potete. Usate il tasto 6 per andare in alto, il 7 per andare in basso.



Score 8 _____ Time left 43

```

10 REM Spider 'n fly
20 LET hi=0: GO SUB 9000
30 GO SUB 8000
40 PRINT AT 0,0;"Score ";sc
50 PRINT AT v,0;" "
60 LET v=v+(INKEY$="6" AND v<2
1)- (INKEY$="7" AND v>2)
70 IF SCREEN$(v,2)="" THEN
GO SUB 1000
80 PRINT AT v-1,0;"|"
90 PRINT AT v,0; INK 5;"●"
100 FOR a=1 TO 5
110 LET f(a)=f(a)-INT (AND#2):
IF f(a)<1 THEN PRINT AT a*3,1;"
": LET f(a)=28
120 PRINT AT a*3,f(a); INK 4;"#
"
130 NEXT a
140 PRINT AT 0,19;"Time left ";
INT time;" ": LET time=time-.5:
IF time<0 THEN GO TO 2000
150 GO TO 50
990 STOP
1000 LET sc=sc+2
1010 PRINT AT v,2;" "
1020 LET f(v/3)=28
1030 PRINT AT 0,0; INK 7;"Score
";sc
1040 RETURN
2000 PRINT AT 5,9;"GAME OVER!"
2010 PRINT AT TAB 6;"You scored "
;sc

```

```

2020 IF sc>hi THEN LET hi=sc
2030 PRINT "" highest score t
oday ";hi
2040 INPUT "Press "; FLASH 1;"EN
TER"; FLASH 0;" to play again ";
LINE a$: GO TO 30
8000 BORDER 1: PAPER 1: INK 7: C
LS
9010 LET v=10
9020 DIM f(6)
9030 FOR a=1 TO 6: LET f(a)=28:
NEXT a
9040 LET sc=0
9050 RANDOMIZE
9060 FOR a=1 TO v-1: PRINT AT a,
0;"|": NEXT a
9070 PLOT 0,168: DRAW 255,0
9080 LET time=99
9090 RETURN
9091 REM
9092 REM Graphics
9093 REM
9094 REM graphics c
9095 REM graphics ab
9096 REM
9097 REM graphics de
9098 REM
9099 REM *****
9000 FOR a=USR "a" TO USR "e"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 60,126,255,255,127,127
,149,148
9040 DATA 0,108,248,220,252,246,
95,0
9050 DATA 16,16,16,16,16,16,16,1
9060 DATA 0,17,63,94,111,23,45,0
9070 DATA 0,252,2,2,252,224,0,0

```


ASCOT

Con questo programma potete possedere un cavallo da corsa, senza l'inconveniente di doverlo nutrire o di dover pulire la scuderia. Avete a disposizione un cavallo, 50 sterline e la possibilità di far fortuna sulla pista proprio con il vostro capitale. Potete giocare con altre quattro persone al massimo: vince chi, al termine, possiede la cifra più alta. I soldi si vincono scommettendo sul cavallo che si è scelto per le cinque corse, e le puntate vanno dichiarate prima di ogni corsa. Se finite i soldi, ma il vostro cavallo vince una corsa, vi sarà data in premio una certa somma di denaro che vi permetterà di giocare nella corsa successiva.





Race number 1

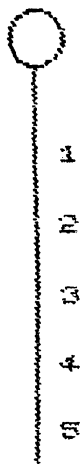
The odds on Red Gin
to win are 1:1

The odds on Sumley Gilds
to win are 1:1

The odds on Sparkle
to win are 5:1

The odds on Danny Boy
to win are 8:1

The odds on Shergart
to win are 3:1



Horse: - Red Gin

Owner: - Peter

You have £50

The winner is Danny Boy
 Owned by Micheal
 who wins 200

Horse:- Shergart

Owner:- Bill

You have £0

You have no money so you cannot
 place a bet

```

10 REM Ascot
20 GO SUB 9000: REM UDG's
30 GO SUB 8000
40 FOR r=1 TO 5
50 CLS
60 PRINT AT 0,0;"Race number "
;
70 DIM d(p)
80 FOR a=1 TO p: LET d(a)=INT
(RND*10)+1
90 PRINT "The odds on ";a$;"a"
100 PRINT "to win are ";d(a);"
1"
110 NEXT a
120 GO SUB 5000
130 PRINT AT 0,0;"#1;" Press any
y key to continue"
140 PAUSE 0
150 CLS : CIRCLE INK 2;240,155,
10: PLOT 240,155: DRAW INK 2;0,-
140: FOR a=1 TO p: PRINT AT a#3+
3,31;a: NEXT a
155 DIM c(p)
160 FOR a=1 TO p
170 PRINT AT a#3+2,c(a);a$;"AT a
#3+3,c(a);b$;"AT a#3+4,c(a);c$
180 LET c(a)=c(a)+(1/d(a)+INT (
RND*2))
185 IF c(a)>25 THEN GO TO 250
190 NEXT a
200 FOR a=1 TO p: BEEP .008,r/a
;

```

```

210 PRINT AT a#3+2,c(a);a$;AT #
#3+3,c(a);d$;AT a#3+4,c(a);e$
220 LET c(a)=c(a)+f1/d(a)+INT (
RND#2)
225 IF c(a)>25 THEN GO TO 250
230 NEXT a
240 GO TO 160
250 LET ##=h$(a)
260 PAUSE 100:CLS
270 PRINT "The winner is ";#$
280 PRINT "Owned by ";n$(a)
290 LET #s=d(a)/#(a)
300 PRINT "Who wins ";#s. LET
m(a)=m(a)+#s+5(a)
310 FOR z=1 TO 50: BEEP .000,z:
BEEP .000,-z: NEXT z
320 NEXT s
330 CLS
340 LET tot=0: PRINT "At the en
d of the game"
350 FOR a=1 TO p
360 PRINT n$(a); " has ";m(a)
370 IF m(a)>tot THEN LET tot=#(
a): LET win=a
380 NEXT a
390 PRINT "So the winner is ";
n$(win)
400 PRINT "on ";h$(win)
410 PRINT "With £";m(win)
420 INPUT "Press enter for ano
her game "; LINE a$: GO TO 30
990 STOP
5000 FOR z=1 TO 50: BEEP .002#z,
z: BEEP .000,-z: NEXT z
5010 FOR a=1 TO p
5020 CLS
5030 PRINT "Horse. - ";h$(a)
5040 PRINT "Owner: - ";n$(a)
5050 PRINT "You have £";m(a)
5055>IF m(a)<1 THEN PRINT "You
have no money so you cannot plac
e a bet": PAUSE 150: NEXT a: RET
URN
5060 INPUT "How much will you be
t on the next race £";s(a): I
F s(a)>m(a) THEN GO TO 5060
5070 LET m(a)=m(a)-s(a)
5080 NEXT a
5090 RETURN
8000 BORDER 0: PAPER 0: INK 7: C
LS : BEEP .1,10: BEEP .2,15

```

```

8010 LET a$="" : REM abcd
8020 LET b$="" : REM efgh
8030 LET c$="" : REM ijkl
8040 LET d$="" : REM mnop
8050 LET e$="" : REM qr
8060 PRINT AT 1,12;"ASCOT"
8070 INPUT "How many players (Max
x 5)"; LINE p$: IF p$<"1" OR p$>
"5" THEN GO TO 8070
8080 LET p=VAL p$: DIM s(p): DIM
m(p): DIM n$(p,10): DIM h$(p,15
)
8090 FOR a=1 TO p
8100 PRINT AT 5,3;"Type in name
of player #";a
8110 INPUT "Max 10 letters "; LI
NE n$(a)
8120 LET m(a)=50: NEXT a
8125 CLS
8130 RESTORE 9200: FOR a=1 TO p
8140 READ h$(a): PRINT n$(a);" B
ow owns "h$(a)"
8150 NEXT a
8160 PRINT AT 0,0;#1;" Press a
ny key to continue": PAUSE 0
8990 RETURN
9000 FOR a=USR "a" TO USR "q"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 0,0,0,0,0,0,0,0
9040 DATA 0,0,0,0,0,1,113,207
9050 DATA 0,20,10,42,07,132,15,2
38
9060 DATA 0,0,16,240,60,124,100,
131
9070 DATA 5,13,13,0,0,0,0,0
9080 DATA 9,16,0,144,157,170,190
,192
9090 DATA 221,209,33,1,194,242,1
1,10
9100 DATA 120,0,0,0,0,0,0,120
9110 DATA 3,2,2,2,2,1,0,0
9120 DATA 192,192,96,32,16,32,0,
0
9130 DATA 13,20,25,19,10,0,0,0
9140 DATA 120,120,120,0,0,0,0,0
9150 DATA 16,0,16,144,177,157,04
,120
9160 DATA 69,1,1,1,194,246,20,20
9170 DATA 120,0,0,0,0,0,0,0
9180 DATA 120,120,120,120,120,04
,0,0
9190 DATA 20,20,24,24,20,12,10,0
9200 DATA "Red Gin","Suntley Bird
s","Sparkle","Danny Boy","Sherge
rt"

```

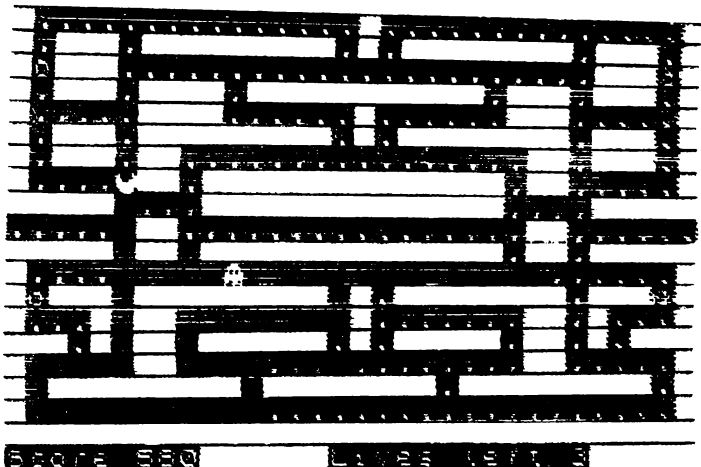
MUNCHIE MAN

È probabilmente il gioco clou di questa raccolta. C'è un fantasma che voi dovete cercare di mangiare. Negli angoli potete mangiare i "vantaggi" del fantasma, ma otterrete solo dei punti, e non la possibilità di mangiare il fantasma stesso.

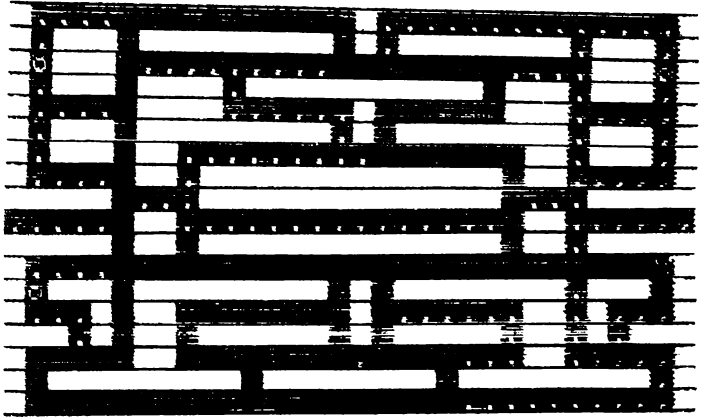
Usate i tasti del cursore per controllare i vostri movimenti.

Nota

I grafici nelle istruzioni 7010 e 7020 sono ottenuti scambiando "Inv. video" e "True video", per ottenere un punto in negativo e la messa in evidenza del true video (spostamento del simbolo Ø)

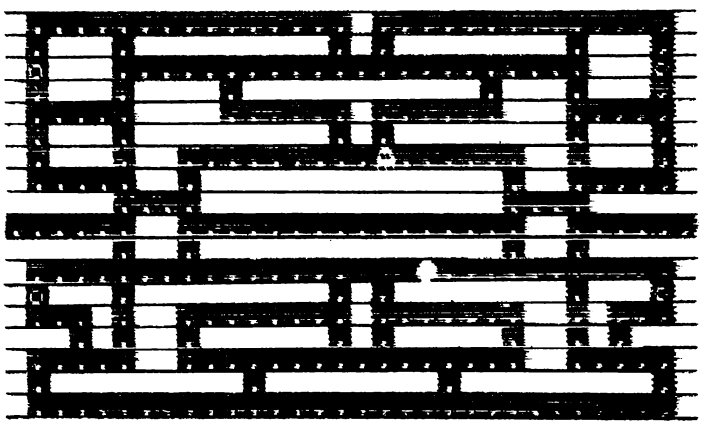


MUNCHIE MAN



Score 1528

LIVES LEFT 5



Score 1630

LIVES LEFT 5

```

10 REM Munchie Man
20 LET hi=0: GO SUB 2000
30 GO SUB 8000
40 LET v=12: LET h=16: GO SUB
7000: INVERSE 1: PRINT AT 20,15:
"Lives left "; lives
    
```

```

45 PRINT AT v,h; INVERSE 1; a$
50 LET a$=INKEY$: IF a$<"5" OR
a$>"8" THEN GO TO 80
60 IF a$<>" " THEN LET c1=VAL a
$
70 LET m$=("●" AND c1=7)+("◐"
AND c1=8)+("◑" AND c1=5)+("◒" AN
D c1=5)
80 PRINT AT v,h; m$
85 IF h<1 THEN LET h=30
90 IF h>30 THEN LET h=0
90 LET v1=v: LET h1=h
100 LET v=v+(c1=6)-(c1=7)
110 LET h=h+(c1=8)-(c1=5)
120 IF SCREEN$(v,h)="_" THEN L
ET v=v1: LET h=h1: PRINT AT v,h;
m$: GO TO 150
130 IF SCREEN$(v,h)="." THEN L
ET sc=sc+10: BEEP .008,10: LET c
ount=count+1
135 IF SCREEN$(v,h)="o" THEN G
O SUB 6000
140 IF SCREEN$(v,h)=" " THEN GO
TO 5000
145 PRINT AT v,h; "●"; AT 20,0; "5
core ";sc
150 PRINT AT z,x; " "
160 LET z1=z: LET x1=x
170 LET z=z+(m1=6)-(m1=7): LET
x=x+(m1=8)-(m1=5)
180 IF SCREEN$(z,x)="_" THEN L
ET m1=INT (RAND*4)+5: LET z=z1: L
ET x=x1: GO TO 150
185 PRINT AT v,h;m$
190 IF SCREEN$(z,x)="." THEN P
RINT AT z1,x1; "."
195 IF SCREEN$(z,x)="o" THEN P
RINT AT z1,x1; "o"
200 IF SCREEN$(z,x)=" " THEN GO
TO 5000
210 PRINT AT z,x; PAPER 4; "●"
230 IF lives<1 THEN GO TO 2000
240 IF count=tot THEN LET count
=0: GO TO 40
250 GO TO 50
990 STOP
2000 CLS
2010 INVERSE 0: PRINT "TAB 5; "G
AME OVER"
2020 PRINT " " " You scor
ed ";sc
2030 IF sc>hi THEN LET hi=sc
2040 PRINT " " " Highest score t
oday ";hi
2050 INPUT "Press enter to play
again "; LINE a$: GO TO 30

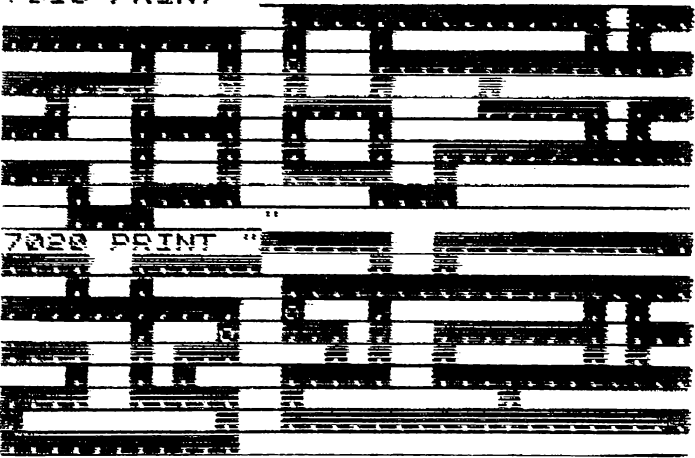
```



```

5000 INVERSE 1
5005 PRINT AT z,x;" "
5010 PRINT AT v,h;"●"
5020 BEEP .3,15
5030 PRINT AT v,h;"▲"
5040 BEEP .3,11
5050 PRINT AT v,h;"●"
5060 BEEP .3,7
5070 PRINT AT v,h;"▲"
5080 BEEP .3,3
5090 PRINT AT v,h;"X"
5100 BEEP .3,0
5110 PRINT AT v,h;" "
5120 BEEP .5,-5
5130 LET count=0: LET v=12: LET
h=16: LET lives=lives-1
5140 INVERSE 0: GO TO 40
6000 BEEP .006,15: BEEP .005,-15
: LET sc=sc+INT (RND*100)+200
6010 RETURN
7000 PAPER 0: CLS : INK 0: PAPER
6: BORDER 0
7010 PRINT "

```



```

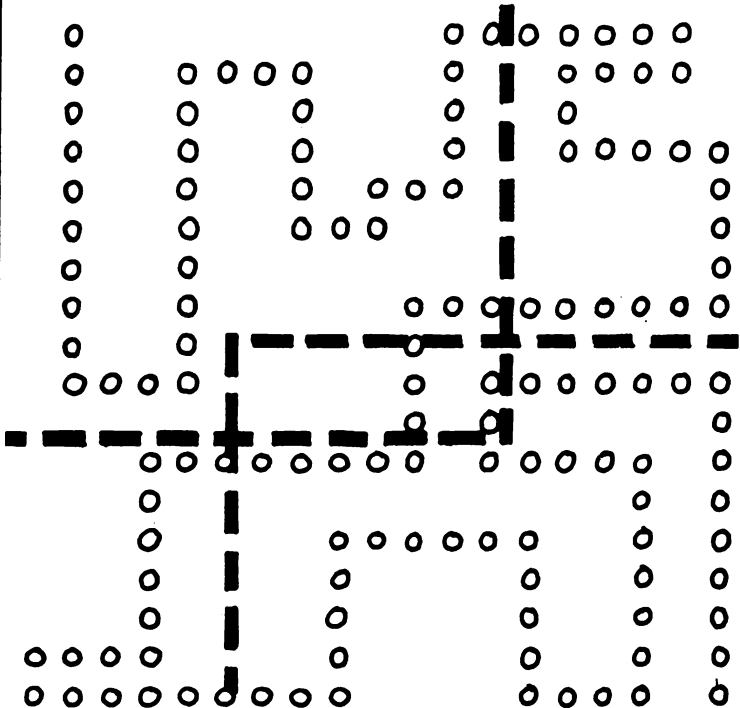
7020 PRINT "
7030 INK 0
7040 PAPER 6
7100 RETURN
8000 LET ##=" "
8020 LET c1=6: LET m1=5
8030 LET sc=0
8040 LET z=7: LET x=15
8050 LET lives=3
8060 LET count=0
8070 LET tot=266
8990 RETURN

```

```

9000 FOR a=USR "a" TO USR "i"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 60,126,255,255,255,255
,126,60
9040 DATA 0,66,195,231,255,255,1
26,60
9050 DATA 60,126,248,240,240,248
,126,60
9060 DATA 60,126,255,255,231,195
,66,0
9070 DATA 60,126,31,15,15,31,126
,60
9080 DATA 0,0,126,255,255,255,12
6,60
9090 DATA 0,0,0,0,16,24,60,60
9100 DATA 129,66,36,0,0,36,66,12
6
9110 DATA 28,62,42,107,127,127,1
09,73
9000 REM a b c d e f g h i
9000 REM ● ● ● ● ● ● ● × ●

```



TRACK TRAP

In questo gioco voi e il computer vi muovete sullo schermo, lasciando una traccia. Garegiate per conquistare lo spazio. Il computer guadagna un punto se toccate la sua pista, se finite sul lato dello schermo o se ritornate in un punto su cui siete già passati.

Il computer, però, non è imbattibile e, con un po' di pratica, riuscirete a batterlo; con molta soddisfazione!

Usate i tasti del cursore per controllare i vostri movimenti.

```

10 REM Track trap
20 LET hi=0
30 GO SUB 8000
40 LET a$=INKEY$
50 IF a$="3" OR a$="5" THEN GO
TO 70
60 LET m=VAL a$: LET j=0
70 LET h1=h1+(m=6)-(m=7)
80 LET h2=h2+(m=8)-(m=5)
90 LET b$=SCREEN$(h1,h2)
100 BEEP .005,10
110 IF b$="_" THEN GO SUB 2000
120 LET j=j+1: PRINT AT h1,h2; IN
VERSE 1;" "
130 PRINT AT h1,h2; INVERSE 1;
INK 6;" "
140 LET u=0
150 IF u=5 THEN GO SUB 1000
160 LET c3=c1: LET c4=c2
170 LET c1=c1+(n=6)-(n=7)
180 LET c2=c2+(n=8)-(n=5)
190 LET c$=SCREEN$(c1,c2)
200 IF c$="_" THEN LET n=u+5: L
ET c1=c3: LET c2=c4: LET u=u+1:
GO TO 150
220 PRINT AT c1,c2; INVERSE 1;"
"
240 GO TO 40
1000 PRINT AT c1,c2; FLASH 1;"_"
1010 LET hs=hs+1: FOR p=1 TO 30:
BEEP .008,p: NEXT p
1020 PRINT AT 0,0;"SCORE COMP ";
c$;" HUMAN ";hs
1030 PRINT INVERSE 1;"_____
"
1040 FOR l=1 TO 10
1050 PRINT INVERSE 1;"_"; INVER
SE 0;"

```

```

"; INVERSE 1;"_"
1060 NEXT I
1070 PRINT INVERSE 1;"_____
"
1080 LET c1=10: LET c2=10
1090 LET h1=10: LET h2=20
1100 IF hs=10 THEN GO TO 3000
1110 RETURN
2000 PRINT AT h1,h2; FLASH 1;"_"
2010 LET cs=cs+1: FOR p=1 TO 30:
  BEEP .008,p: NEXT p
2020 PRINT AT 0,0;"SCORE COMP ";
cs;" HUMAN ";hs
2030 PRINT INVERSE 1;"_____
"
2040 FOR l=1 TO 10
2050 PRINT INVERSE 1;"_"; INVERSE
E 0;"
"; INVERSE 1;"_"
2060 NEXT l
2070 PRINT INVERSE 1;"_____
"
2080 LET c1=10: LET c2=10
2090 LET h1=10: LET h2=20
2100 IF cs=10 THEN GO TO 3000
2110 RETURN
3000 PRINT AT 5,8;("COMPUTER " &
ND cs>hs)+("HUMAN " AND hs>cs);
WINS!!"
3010 INPUT "Press enter to play
again "; LINE a$
3020 LET j=1: LET c1=10: LET c2=
10
3010 LET h1=10: LET h2=20
3020 LET hs=0: LET cs=0
3030 BORDER 0: PAPER INT (AND#6)
: INK 9: CLS
3040 PRINT "SCORE ";
3050 PRINT "COMP 00 HUMAN 00"
3060 PRINT INVERSE 1;"_____
"
3070 FOR l=1 TO 10
3080 PRINT INVERSE 1;"_";TAB 31;
INVERSE 1;"_"
3090 NEXT l
3100 PRINT INVERSE 1;"_____
"
3110 LET m=5: LET n=8: RETURN

```

PONTOON

È la versione, un po' semplificata, di un popolare gioco di carte, il ventuno. Il fine del gioco è quello di arrivare il più possibile vicino a 21 punti, senza però superarli.

Voi cominciate per primi. Il computer vi mostra le vostre prime due carte: dovete decidere se prenderne un'altra (Twist) o rimanere con i punti che avete (Stick). Forse dovrete decidere due, tre o più volte. Se il totale supera 21, il computer annuncia "Bust" e voi perdete 10 sterline (la posta per ogni partita). Se decidete "Stick" con un punteggio inferiore a 21, il computer gioca a sua volta (le sue carte non vengono mostrate). A questo punto il computer decide per "Bust" o "Stick", prima di proclamare il vincitore. Un asso può valere solo un punto. Quando ne avrete abbastanza, interrompete il programma mentre il computer sta distribuendo le carte.

D - Deal
T - twist
S - Stick



Cash 50

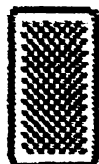


BUST

I win that game!

T - twist

S - Stick



Cash 50



I have 21

You have 19

PONTOON!

```

10 REM Pontoon
20 GO SUB 9000
30 LET money=50
40 GO SUB 7000
50 LET c1=INT (RND*10)+1
60 LET c2=INT (RND*10)+1
70 PRINT AT 11,2; " "
80 LET b$=CHR$ (144+(INT (RND*
4)))
90 PRINT AT 12,0; " "; b$; " |"
100 PRINT AT 13,0; " |"; (CHR$ 5
110 PRINT AT 14,0; " |"; (CHR$ 5
AND c1=10); c1; " |"; "
120 PRINT AT 15,0; " |"; "
130 PRINT AT 16,0; " |"; b$; " |"
140 PRINT AT 17,2; " "
150 LET b$=CHR$ (144+(INT (RND*
4)))
160 PRINT AT 11,6; " "
170 PRINT AT 12,6; " "; b$; " |"
180 PRINT AT 13,6; " |"; (CHR$ 5
190 PRINT AT 14,6; " |"; (CHR$ 5
AND c2=10); c2; " |"; "
200 PRINT AT 15,6; " |"; "
210 PRINT AT 16,6; " |"; b$; " |"
220 PRINT AT 17,6; " "

```

PONTOON

```

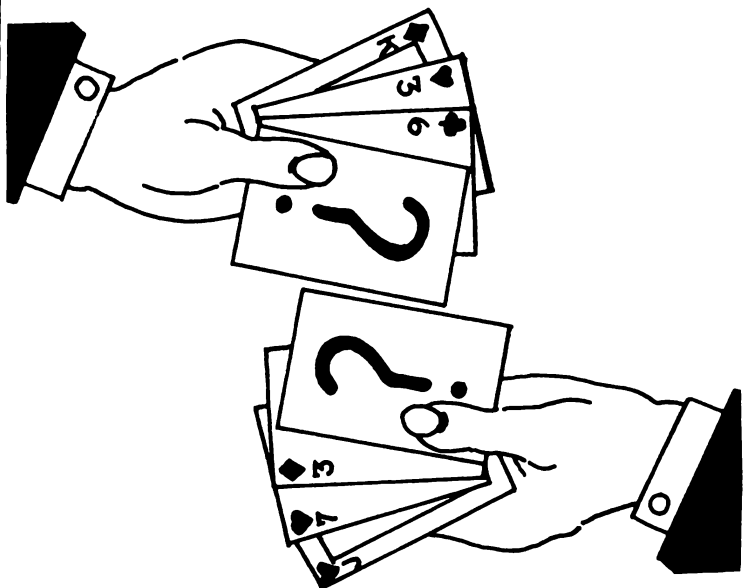
225 LET tot=c1+c2: LET c3=INT (
RND*10)+1
230 PRINT AT 5,5:"T - twist"
235 PRINT AT 7,5:"S - Stick"
240 IF INKEY$="t" OR INKEY$="s"
THEN GO TO 250
245 GO TO 240
250 IF INKEY$="s" THEN GO TO 10
00
250 LET tot=tot+c3
270 LET b$=CHR$ (144+(INT (RND*
4)))
275 IF p>26 THEN LET p=2: CLS
280 PRINT AT 11,p:" "
290 PRINT AT 12,p:" ";b$;" |"
300 PRINT AT 13,p:" "
310 PRINT AT 14,p:" ";CHR$ 8
AND c3=10);c3:" |"
320 PRINT AT 15,p:" |"
330 PRINT AT 16,p:" ";b$;" |"
340 PRINT AT 17,p:" "
350 IF tot>21 THEN PRINT AT 1,1
;"BUST": GO TO 2000
360 LET p=p+6
365 LET c3=INT (RND*10)+1
370 GO TO 240
990 STOP
1000 LET cs=INT (RND*13)+10
1010 CLS
1020 PRINT "I have ";cs;" "You h
ave ";tot".
1030 IF cs>21 THEN PRINT "I have
bust, You win ": GO TO 1070
1040 IF cs=21 THEN PRINT "PONTOO
N!": GO TO 1070
1050 IF cs>=tot THEN PRINT "I wi
n": GO TO 1070
1060 PRINT "You win $50!": LET m
oney=money+50
1070 FOR k=1 TO 60: BEEP .005,k:
BEEP .008,-k: NEXT k: GO TO 40
2000 PRINT "I win that game!"
2010 GO TO 1070
3000 CLS
3010 PRINT AT 10,7;"You're out o
f money"
3020 INPUT "Press enter to play
again "; LINE a$: RUN
7000 BORDER 0: PAPER 4: INK 9: C
LS
7005 LET p=14: LET money=money-1
0: IF money<0 THEN GO TO 3000
7010 PRINT AT 1,24; INK 1;"
7020 FOR a=1 TO 6
7030 PRINT TAB 24; INK 1;"|////|"

```

```

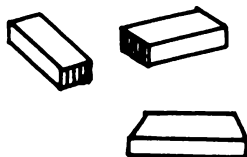
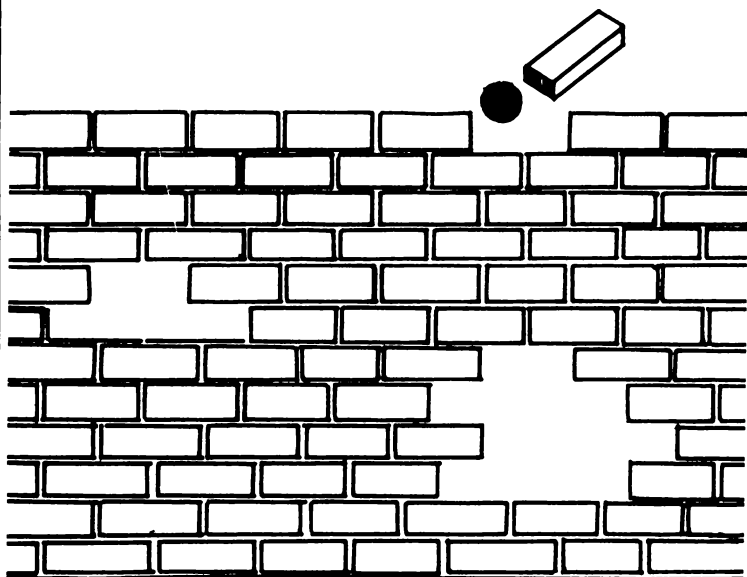
7040 NEXT a
7050 PRINT TAB 24; INK 1; "  "
:AT 9,23;"Cash ";money
7060 PRINT AT 3,5;"D - Deal"
7070 IF INKEY$<>"d" THEN GO TO ?
070
7080 RETURN
7990 RETURN
0000 FOR a=USR "a" TO USR "z"+7
0010 READ user7, PORE a,user7
0020 NEXT a: RETURN
0030 DATA 24,60,126,255,255,126,
00,24
0040 DATA 24,60,90,255,255,90,24
,60
0050 DATA 16,56,124,254,254,54,1
0,56
0060 DATA 68,236,254,254,254,124
,56,16
0070 DATA 204,204,51,51,204,204,
51,51
0080 DATA 0,0,0,7,15,12,24,24
0090 DATA 0,0,0,192,240,40,24,24
0100 DATA 24,24,12,15,7,0,0,0
0110 DATA 24,24,40,240,192,0,0,0
0120 DATA 0,0,0,255,255,0,0,0
0130 DATA 24,24,24,24,24,24,24,2
4
0090 REM a b c d e f g h i j k
0091 REM ♠ ♣ ♡ ♠ ♣ ♡ ♠ ♣ ♡ ♠

```



BREAKOUT

È una variante di un vecchio, famoso, gioco per computer. Scopo del gioco è realizzare punti buttando giù dei mattoni da un muro, usando una pallina spinta dal vostro campo verso il muro. Avete a disposizione 10 palline. Il gioco termina quando la decima scompare dallo schermo. Il tasto 5 vi fa spostare a sinistra, l'8 a destra. C'è anche la possibilità di scegliere la velocità, digitando 1 per andare veloce e un numero oltre il 10 per andare lentamente.



```

10 REM Breakout
20 GO SUB 9000
30 GO SUB 8000
35 GO SUB 7000
40 PRINT AT 20,v;" "
50 LET v=v+2*(INKEY$="8" AND v
(27)-2*(INKEY$="5" AND v>0): IF
INKEY$="5" AND v<0 THEN PRINT AT
20,1;" "
55 PRINT AT 0,0;"SCORE ";sc: F
OR a=1 TO 3: NEXT a
60 PRINT AT e,f;" "
70 LET e=e+c: IF e<2 THEN LET
c=-c: BEEP .005,20
80 LET f=f+d: IF f<1 OR f>30 T
HEN LET d=-d: BEEP .005,10
90 IF SCREEN$(e,f)=" " THEN L
ET sc=sc+1: LET c=-c: BEEP .005,
15
95 IF SCREEN$(e,f)="_" THEN L
ET count=count+1: IF count=(32+5
) THEN LET count=0: LET s=s-1s+1
): GO TO 35
100 IF SCREEN$(e,f)=" " THEN LE
T c=-c: BEEP .005,15
110 PRINT AT e,f: INK 6;"●": IF
e>20 THEN LET b=b+1: GO SUB 100
0
115 IF b=10 THEN GO TO 2000
130 GO TO 40
200 STOP
1000 PRINT AT e,f;" "
1010 LET e=20: LET f=INT (RND*15
)+10: LET c=-1
1020 PRINT AT 0,25;"BALL ";b
1030 RETURN
2000 PRINT AT 10,12;"GAME OVER"
2010 PRINT "" You sco
red ";sc
2020 INPUT "Press Enter to play
again "; LINE a$: GO TO 30
7000 PRINT AT 0,0;"SCORE 0";AT 0
,25;"BALL 1": FOR a=1 TO 3
7010 PRINT " INK a+2; INVERSE 1;
"
"
7020 NEXT a
7030 RETURN
8000 BORDER 0: PAPER 1: INK 7: C
LS
8010 LET v=15
8020 LET c=-1: LET d=1
8030 LET sc=0
8040 LET e=20: LET f=INT (RND*15
)+10

```

BREAKOUT

```
8050 LET b=1
8060 INPUT "What speed (1-Fast 2
0-Extra slow)";s
8070 LET count=0
8990 RETURN
9000 FOR a=USR "a" TO USR "b"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 60,126,255,255,255,255
,126,60
9040 DATA 0,0,0,0,0,0,255,255
9990 REM a b
9991 REM ● -
```

EXCALIBUR AGAINST THE ENEMY

Si gioca su un quadrato di 7x7. Voi siete il comandante della navicella spaziale Excalibur e dovete difendervi da parecchie astronavi straniere. Se queste vi circondano, e non potete più muovervi, il computer ha vinto. Se catturate almeno 17 astronavi, prima di aver fatto 50 mosse, avete vinto voi. Potete muovere in alto e in basso, a destra e a sinistra, ma non in diagonale. Saltate nella posizione secondo la direzione che avete scelto. Per muovere, immettete il numero del quadrato verso il quale volete andare. Scrivete il numero giusto preso dalla colonna laterale, seguito dal numero dell'altra coordinata, senza porre spazi o virgole fra l'uno e l'altro. Il computer vi avvertirà immediatamente se fate una mossa sbagliata. Potete andare nei settori con le stelle, ma non in quelli che contengono navicelle o quadrati nemici.

```

10 REM Excalibur Against the
      enemy
20 RANDOMIZE : GO SUB 9000
30 GO SUB 600
40 GO SUB 80
50 GO SUB 380
60 GO SUB 80
70 GO TO 50
80 REM -Computer Moves-
90 LET y=0: LET p=p+1
100 LET k=INT (RND*63)+13
110 LET y=y+1
120 IF h(k)<>144 THEN GO TO 100

```

EXCALIBUR AGAINST THE ENEMY

```

130 LET J=1
140 IF h(k+z(j))=146 THEN GO TO
190
150 LET J=J+1
160 IF J<9 THEN GO TO 140
170 IF y<100 THEN GO TO 100
180 PRINT "Excalibur wins!!"
STOP
190 LET h(k)=146: LET h(k+z(j))
=144
200 RETURN
210 REM -Accept player move-
220 PRINT
230 REM Player enters 99 for q
if no move
240 LET q=99
250 REM Input at top of screen-
260 INPUT AT 4,0;AT 0,0;"Excali
bur moves to ";l
270 IF l=99 THEN GO TO 750
280 REM -Mugtraps-
290 LET r=ABS(l-q)
300 IF h(q)<>69 OR h(l)<>146 TH
EN PRINT "Illegal move": GO TO 2
30
310 IF r=9 OR r=18 OR r=18 OR r
=22 THEN PRINT "Diagonal moves a
re illegal.": GO TO 230
320 LET h(q)=146: LET h(l)=69
340 IF r=20 OR r=2 THEN LET h((
l+q)/2)=146: LET s=s+1
350 LET qm=l
360 RETURN
370 REM -Print board-
380 PRINT
390 PRINT : PRINT TAB 3;"Move n
umber ";p
400 IF k=0 OR k+z(j)=10 OR k=k+
z(j) THEN GO TO 420
410 PRINT
420 PRINT 50-p;" Moves Left ":
PRINT
430 PRINT TAB 4;"1234567"
440 FOR J=70 TO 10 STEP -10
450 LET a=h(j+1): LET b=h(j+2)
: LET c=h(j+3): LET d=h(j+4): LE
T e=h(j+5): LET f=h(j+6): LET g=
h(j+7)
470 PRINT TAB 3;j/10;
480 PRINT CHR$(a);CHR$(b);CH
R$(c);CHR$(d);CHR$(e);CHR$(f
);CHR$(g);j/10
490 NEXT J
500 PRINT TAB 4;"1234567"
510 PRINT

```

```

520 IF p=50 THEN GO TO 740
530 PRINT " Enemy tally: ";17-s;
" Ships to go!"
540 IF s=17 THEN GO TO 130
550 GO SUB 210
560 IF k=0 THEN GO TO 580
570 RETURN
580 FOR j=1 TO 750: NEXT j
590 RETURN
600 BORDER 0: PAPER 0: INK 5: B
RIGHT 1: CLS
610 DIM h(87): DIM z(5): LET k=
0: LET s=0: LET p=0
620 RESTORE 630: FOR j=1 TO 3:
READ q: LET z(j)=q: NEXT j
630 DATA 10,1,-1
640 LET j=1: LET l=100
650 FOR a=1 TO 87: LET h(a)=145
660 IF a>72 AND a<76 OR a>62 AN
D a<66 OR a>51 AND a<57 OR a>41
AND a<47 THEN LET h(a)=145
680 IF a=51 OR a=41 OR a=57 OR
a=47 THEN LET h(a)=144
690 IF a>30 AND a<38 OR a>22 AN
D a<26 OR a>12 AND a<16 THEN LET
h(a)=144
700 NEXT a
710 LET h(44)=69
720 LET gm=44
730 RETURN
740 PRINT " Time is up!"
750 PRINT "Enemy Tally is ";s
760 PRINT
770 INPUT AT 20,0;AT 0,0;"Anoth
er game (y or n)",#s
780 IF #s="y" THEN GO TO 30
790 PRINT
800 PRINT "I'm glad you know #t:
en you're beaten,"
810 PRINT
820 PRINT "Excalibur"
830 STOP
9000 RESTORE 9000: FOR a=USR "a"
TO USR "c"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 24,24,60,90,153,169,19
5,129
9040 DATA 255,129,129,129,129,12
9,129,255
9050 DATA 0,0,1,0,4,0,0,16

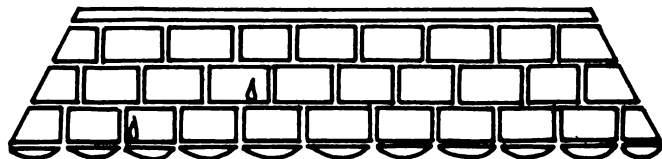
```



```

7035 PRINT INK 2; "██████████████████████
██████████████████████"
7090 RETURN
8030 BORDER 0: PAPER 0: INK 7: C
8050 : BRIGHT 0
9010 LET y1=0
9020 LET y=15
9030 LET d=0
9090 RETURN
9000 FOR a=USR "a" TO USR "j"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 255,255,255,255,255,25
57,0,0
9040 DATA 252,252,252,252,252,25
52,0,0
9050 DATA 0,0,0,0,0,1,0,7
9060 DATA 0,0,0,0,0,120,192,224
9070 DATA 7,7,15,15,15,7,7,7
9080 DATA 224,224,240,240,240,22
4,224,192
9090 DATA 0,0,127,63,95,111,119,
59
9100 DATA 0,0,254,254,253,251,25
1,246
9110 DATA 62,63,63,63,31,31,31,3
1
9120 DATA 14,254,254,254,252,252
,250,252
9130 REM a b c d e f g h i j
9140 REM ██████

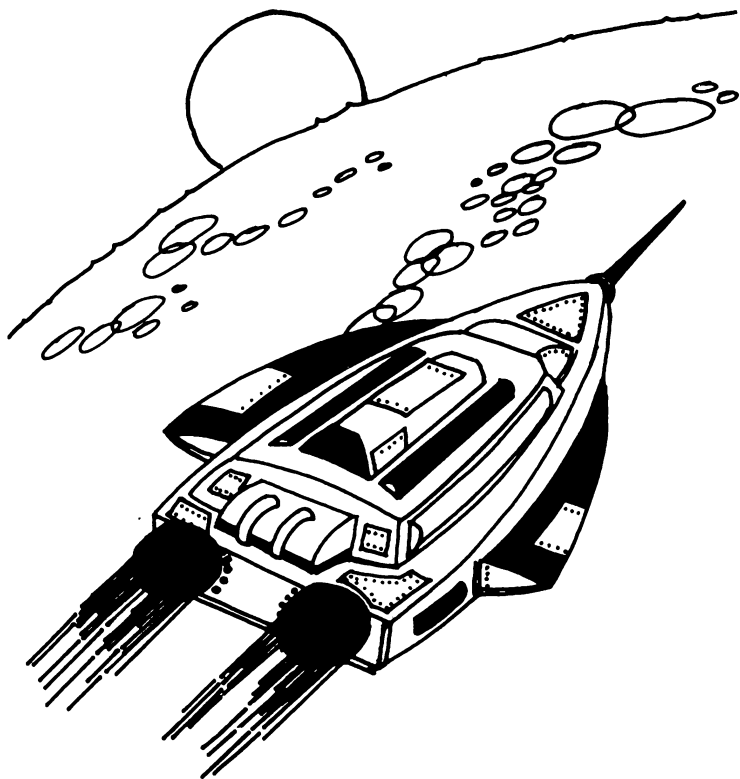
```



TOUCHDOWN ON MARS

Siete in grado di atterrare con il vostro modulo sulla pista di atterraggio? L'ingresso è stretto, e c'è bisogno di un occhio molto acuto per atterrare sani e salvi. Dopo il primo atterraggio, avrete un altro modulo, più veloce del primo, per cui bisognerà fare molta attenzione a prendere bene la mira, altrimenti vi schianterete al suolo e correrete il pericolo di rimanere su Marte per sempre.

Usate il tasto 5 per andare a sinistra, l'8 per andare a destra.



```

10 REM Touchdown on Mars
20 GO SUB 9000: LET hi=0
30 GO SUB 8000
40 GO SUB 7000
50 PRINT AT v,h;" ";AT v+1,h;
" "
60 LET v=v+(((sc/5)+1)/5): LET
h=h+(INKEY$="B" AND h<31)-(INKE
Y$="5" AND h>0)
70 IF SCREEN$ (v+1,h)(">" " THE
N GO TO 1000
80 PRINT AT v,h; INK 6;" ";AT
v+1,h;" "
90 IF v>20 THEN GO TO 2000
100 PRINT AT 21,24;"SCORE ";sc
110 LET t=t+.1
120 PRINT AT 21,0;"TIME ";INT t
150 GO TO 50
990 STOP
1000 PRINT AT 2,5;"GAME OVER IN
";t;" SECONDS"
1010 PRINT AT v,h; FLASH 1; INK
5;" ";AT v+1,h;" "
1020 FOR p=1 TO 100: NEXT p
1030 IF sc>hi THEN LET hi=sc
1040 PRINT AT 7,9;"You scored ";
sc
1050 PRINT AT 9,5;"Highest score
today ";hi
1060 INPUT "Press ENTER to play
again "; LINE a$: GO TO 30
2000 LET v=0: LET h=0: LET sc=sc
+1
2010 CLS
2020 GO TO 40
7000 LET b=INT (RND*16)+10: LET
b=b*2
7010 FOR a=0 TO b-10: PLOT a,0.
DRAW 0,LINE (RND*50)+50)
7020 NEXT a
7040 FOR a=b+10 TO 255: PLOT a,0
: DRAW 0,INT (RND*50)+50
7050 NEXT a
7090 RETURN
9000 BORDER 0: PAPER 0: INK 2: C
LS
9010 LET h=0: LET v=0
9020 LET t=0: LET sc=0
9090 RETURN
9000 FOR a=USR "a" TO USR "d"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 0,0,7,15,31,53,53,53
9040 DATA 0,0,224,240,248,172,17
2,172

```

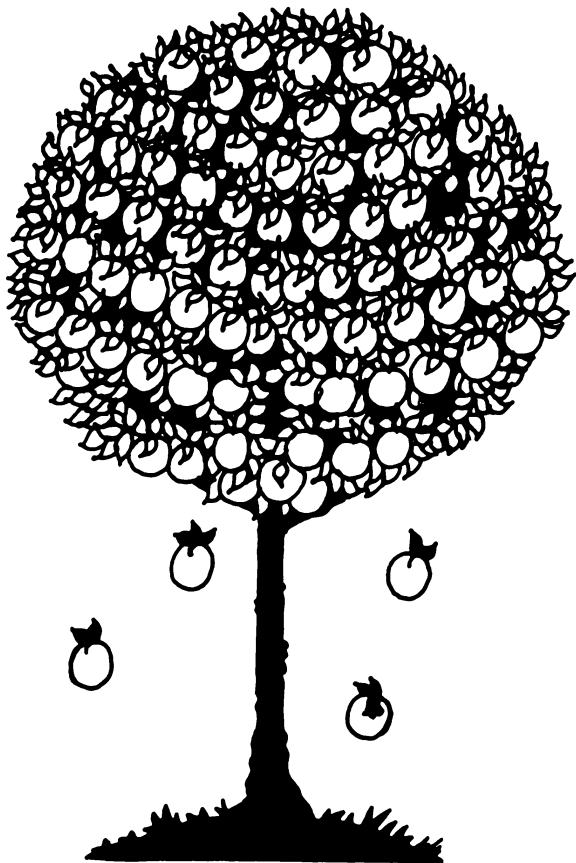
```

9050 DATA 31,13,18,16,8,4,14,8
9060 DATA 248,176,72,8,16,32,112
9070 REM a b c d
9080 REM @ A ? ?
7000 LET b=INT (RND*16)+10: LET
b=b*8
7010 FOR a=0 TO b-10: PLOT a,@:
DRAW @,(INT (RND*50)+50)
7020 NEXT a
7040 FOR a=b+10 TO 255: PLOT a,@
: DRAW @,INT (RND*50)+50
7050 NEXT a
7990 RETURN
8000 BORDER @: PAPER @: INK @: C
LS
8010 LET h=@: LET v=@
8020 LET t=@: LET sc=@
8990 RETURN
9000 FOR a=USR "a" TO USR "d"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN

```

ORCHARD THIEF

Raccogliete quante più mele potete, prima che arrivi il contadino; a quel punto, infatti, il gioco finisce. Avrete 3 punti per ogni mela mangiata; se riuscite a mangiare tutte le 50 mele nel tempo di 30 secondi, avrete un punteggio extra.



ORCHARD THIEF

```

10 REM Orchard Thief
20 GO SUB 9000: LET hi=0
30 GO SUB 8000: GO SUB 7000
40 PRINT AT v,h: " "
45 LET v1=v: LET h1=h
50 LET v=v+(INKEY$="6" AND v<2
0)-(INKEY$="7" AND v>1)
60 LET h=h+(INKEY$="8" AND h<3
1)-(INKEY$="5" AND v>0)
70 IF SCREEN$(v,h)=" " THEN L
ET v=v1: LET h=h1: GO TO 45
80 IF SCREEN$(v,h)="" THEN LE
T sc=sc+3: LET co=co+1
90 PRINT AT v,h: INK 6; "x"
100 LET ti=ti+.1: PRINT AT 0,0;
PAPER 2; "SCORE ";sc; "TIME ";INT
ti
105 BEEP .008,50
110 IF ti>.30 THEN GO TO 1000
120 IF co=50 THEN GO TO 2000
130 GO TO 40
1000 PRINT AT v,h: FLASH 1; INK
2; "x"
1010 PRINT AT 1,12: PAPER 1; "GAM
E OVER"
1020 PRINT AT 5,7: PAPER 2; "YOU
RAN OUT OF TIME"
1030 PRINT AT 12,10; "YOU SCORED
";sc
1040 IF sc>hi THEN LET hi=sc
1050 PRINT AT 20,6; "Highest scor
e today ";hi
1060 INPUT " Press enter to pla
y again "; LINE a$: GO TO 30
2000 PRINT AT v,h: FLASH 1; INK
6; "x"
2010 PRINT AT 2,10: PAPER 1; "GAM
E OVER"
2020 PRINT AT 5,0: PAPER 2; "YOU
DID IT IN ";INT ti; " TIME UNITS"
2030 PRINT AT 7,10: PAPER 4; FLA
SH 1; "BONUS ";INT (100-ti)*3
2040 LET sc=sc+INT (100-ti)*3
2050 GO TO 1000
7040 PRINT AT 0,0; PAPER 2; "SCOR
E ";sc; "TIME ";ti;
7050 PRINT INVERSE 1; INK 6; "____
"
7060 FOR a=1 TO 18
7070 PRINT INVERSE 1; INK 6; " ";
INVERSE 0; TAB 31; INVERSE 1; "____"
7080 NEXT a
7090 PRINT INVERSE 1; INK 6; "____
"
7095 FOR a=1 TO 50

```

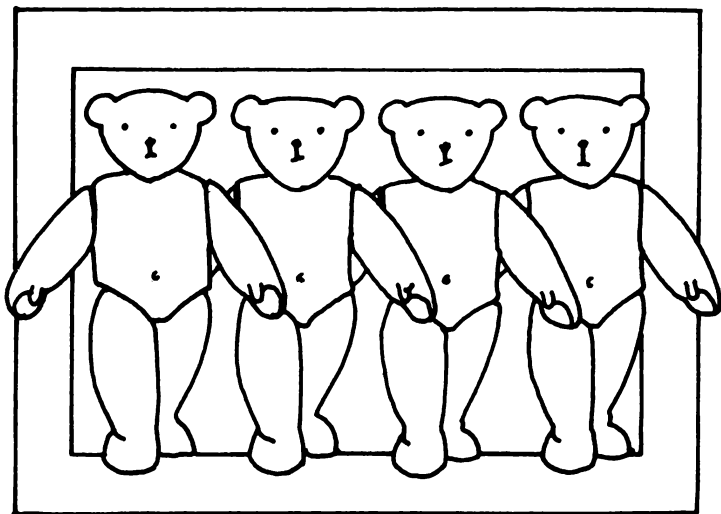
```

7100 LET J=INT (RND*16)+2: LET K
=INT (RND*29)+2
7110 IF J=V AND K=h THEN GO TO 7
100
7115 IF SCREEN# (J,K)="" THEN GO
TO 7100
7120 PRINT AT J,K: INK 4: "2": NE
XT a
7130 RETURN
8000 BORDER 0. PAPER 0. INK 9: C
L5
8010 LET V=INT (RND*16)+2: LET h
=INT (RND*29)+2
8020 LET CO=0: LET ti=0
8030 LET SC=0
8040 RETURN
9000 FOR a=USA "a" TO USA "g"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 12,24,62,127,127,127,1
27,62
9040 DATA 28,28,6,63,6,28,34,65
9050 REM a b
9060 REM 2 X

```

CHARGE OF THE TEDDY

L'orsacchiotto vuole attraversare lo schermo ma, purtroppo, ci sono diversi raggi laser che sparano sul suo percorso. Se riuscite a evitarli, l'orsacchiotto sarà salvo. Per ogni orsacchiotto portato in salvo ci sono tre punti. Cominciate a giocare con cinque orsacchiotti. Usate i tasti del cursore per controllare il movimento.



```

10 REM Charge of the Teddybear
20 REM Peter Shaw
30 REM Original idea by
   Steven Gunning
40 GO SUB 9000: LET h:=6
50 GO SUB 8000: GO SUB 7000
60 PRINT AT v,h: OVER 1: PAPER
8: INK 6;"███":AT v+1,h:"███"
70 LET v=v+2*(INKEY$="6" AND v
<10)-2*(INKEY$="7" AND v>0)
80 LET h=h+2*(INKEY$="8" AND h
<20)-2*(INKEY$="5" AND h>1)
90 PRINT AT v,h: OVER 1: PAPER
8: INK 6;"███":AT v+1,h:"███"
100 GO SUB 1000
110 IF v=0 THEN GO SUB 2000
115 PRINT AT 21,0: PAPER 2:"SCO
RE ":sc,"TEDDIES LEFT ":td,
120 BEEP .008,sc
130 GO TO 60
990 STOP

1000 LET r=INT (RAND%6)+1
1010 PLOT 8,m(r): DRAW INK 0;240
0
1020 LET p=(21-((m(r)-4)/8))
1030 IF v=p OR v=p-1 THEN GO TO
3000
1035 PLOT 8,m(r). DRAW INK 0;240
0
1040 PLOT 8,m(r): DRAW OVER 1;24
0,0
1050 RETURN
2000 PRINT AT v,h: PAPER 6;" ";
AT v+1,h;" "
2005 BEEP .1,20
2010 LET sc=sc+3
2020 LET v=18: LET h=15
2030 PRINT AT v,h: INK 6: PAPER
8: OVER 1;"███":AT v+1,h;"███"
2040 RETURN
3000 PRINT AT v,h: INK 2: FLASH
1;"███":AT v+1,h;"███"
3001 BEEP .5,-10
3005 LET td=td-1: IF td=-1 THEN
GO TO 4000
3010 PRINT AT v,h;" " :AT v+1,h;
" "
3020 LET v=18: LET h=15
3030 PRINT AT v,h: INK 6: PAPER
8: OVER 1;"███":AT v+1,h;"███"
3040 GO TO 1035
4000 PRINT AT 2,12: PAPER 6:"GAM
E OVER"
4010 PRINT "TAB 10: PAPER 1:"YO

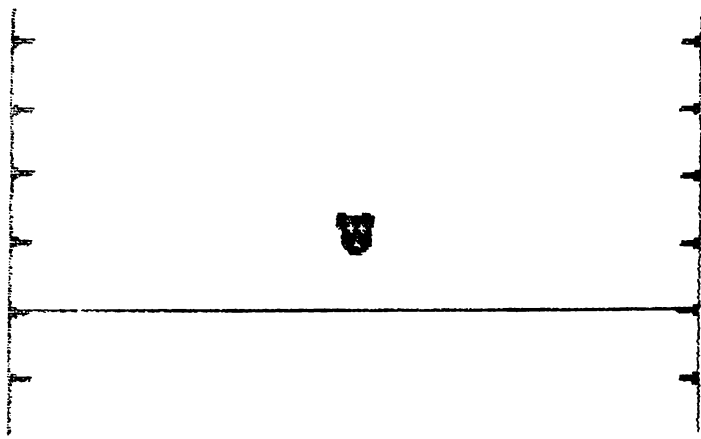
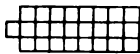
```


CHARGE OF THE TEDDY

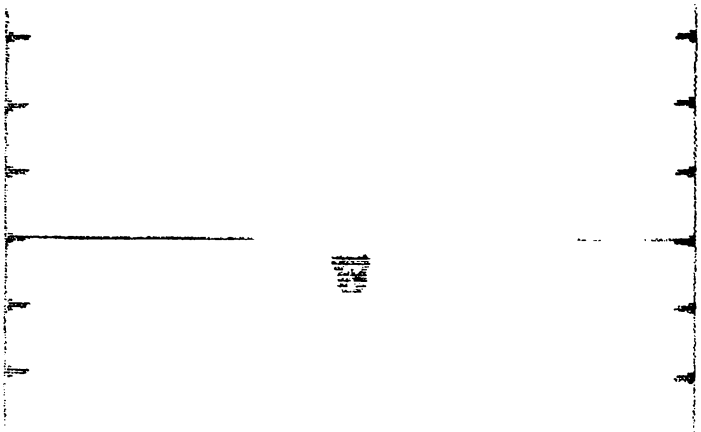
```

U SCORED ";sc
4020 IF sc>hi THEN LET hi=sc
4030 PRINT AT 10,0: PAPER 0;"Hi9
hest score today ";hi
4040 INPUT "Press "; PAPER 2;"EN
TER"; PAPER 0;" to play again ";
LINE a$
4050 GO TO 50
4990 STOP
7000 CLS
7010 PRINT PAPER 1,,
7020 PLOT 0,14: DRAW 0,152
7030 PLOT 255,14: DRAW 0,152
7040 FOR a=2 TO 17 STEP 3
7050 PRINT AT a,0:"1":AT a,31:"4
"
7060 NEXT a
7070 PRINT AT 20,0: PAPER 1,,
7080 PLOT 0,8: DRAW 255,0
7090 PRINT AT 21,0:"SCORE ";sc,"
TEDDIES LEFT ";td
7100 PRINT AT v,h: OVER 1: PAPER
8: INK 6:"";AT v+1,h;"
7990 RETURN
8000 BORDER 0: PAPER 0: INK 9: C
LS
8010 LET sc=0: DIM # (5): LET v=1
8: LET h=15
8020 RESTORE 9090: FOR a=1 TO 6
8030 READ # (a): NEXT a
8040 LET co=0
8050 LET td=5
8090 RETURN
9000 FOR a=USR "a" TO USR "f"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 0,56,127,127,127,125,5
0,29
9040 DATA 0,26,254,254,254,222,1
40,220
9050 DATA 31,31,31,31,36,15,7,3
9060 DATA 252,252,124,124,60,248
,248,224
9070 DATA 128,128,224,255,255,22
4,128
9080 DATA 1,1,7,255,255,7,1,1
9090 DATA 36,60,84,108,108,156
9095 REM
9099 REM

```



SCORE 3 TEDDIES LEFT 5



SCORE 6 TEDDIES LEFT 0

SNAKE IN THE TRIANGLES

Voi siete il serpente che cammina per la propria strada, angolata e sinuosa, verso il basso dello schermo. Alcuni triangoli blu, mortalmente velenosi, si dirigono rapidamente verso di voi. Dovete evitarli a tutti i costi: se ne urtate uno, infatti, il gioco sarà finito. Andate a sinistra automaticamente, per cui dovrete usare il tasto 8 per muovervi verso destra.

```

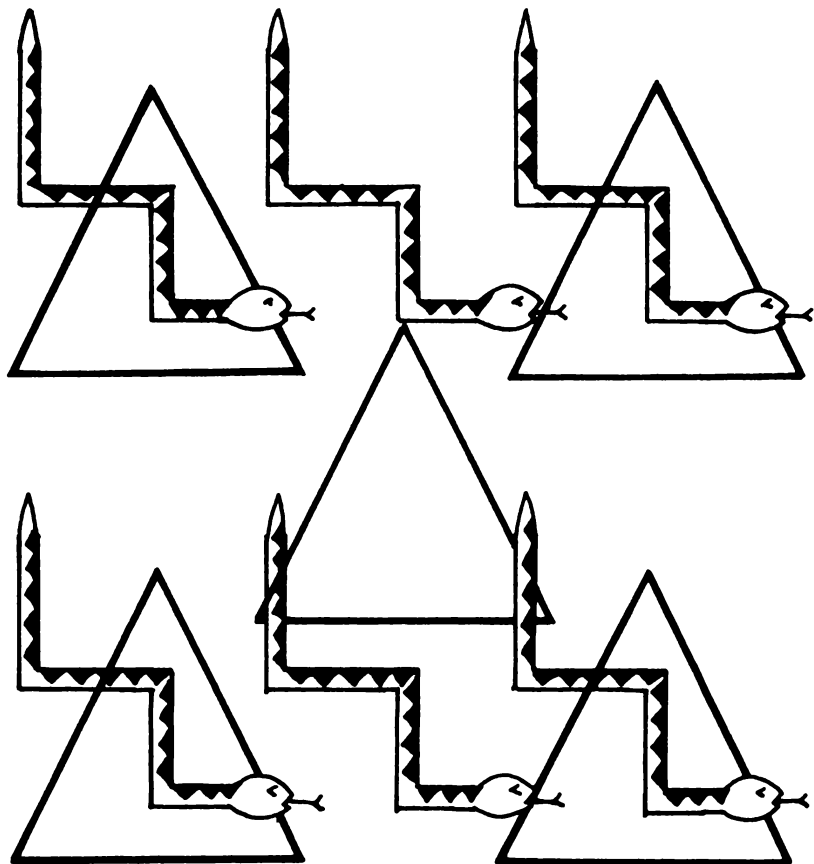
10 REM Snake in the Triangles
20 GO SUB 9000: LET hi=0
30 GO SUB 8000
40 PRINT AT 9,h;" "
50 LET h=h+2*(INKEY$="6" AND v
(29): IF h>0 THEN LET h=h-1
54 IF SCREEN$(10,h)="" THEN G
0 TO 1000
55 PRINT AT 9,h; INK 6;"▼";AT
10,h;"0"
60 PRINT AT 21,0;
70 PRINT TAB INT (RAND*31); INK
1;"▼"
80 POKE 23692,255
90 LET sc=sc+1: PRINT '
100 GO TO 40
1000 PRINT AT 0,0; OVER 1; PAPER
6; INK 2;v$(1)
1010 PRINT AT 1,10; FLASH 1;"GAM
E OVER"
1020 PRINT AT 5,7;"You scored ";
sc
1030 IF sc>hi THEN LET hi=sc
1040 PRINT AT 10,5;"Highest scor
e today ";hi
1050 INPUT "Press "; PAPER 1;"EN
TER"; PAPER 0;" to play again ";
LINE a$: GO TO 30
1090 STOP

```

```

0000 BORDER 0: PAPER 0: INK 9: 0
0010 : LET v=10: LET h=15
0020 LET sc=0: RANDOMIZE
0030 DIM v$(1,704)
0040 RETURN
0050 FOR a=USR "a" TO USR "c"+7
0060 READ user: POKE a,user
0070 NEXT a: RETURN
0080 DATA 255,126,126,60,60,24,2
0090 DATA 0,99,119,127,62,28,0,0
0100 DATA 99,119,73,73,127,62,28
0110 ,0

```



DEATH RACE

Scopo del gioco è quello di "atterrare" quanti più pedoni è possibile, come nel film omonimo. Per ogni pedone abbattuto totalizzerete dei punti. Quando colpite il bersaglio, appare una lapide funeraria. C'è un tempo limite di 60 secondi a partita, quindi cercate di non sprecarlo. Usate i tasti del cursore per controllare il movimento della vostra automobile.

```

10 REM Death Race
20 GO SUB 9000: LET hi=0
30 GO SUB 8000
40 GO SUB 7000
50 PRINT AT v,h:" "
51 IF INKEY$<"5" OR INKEY$>"8"
THEN GO TO 55
53 LET a$=INKEY$
55 LET v1=v: LET h1=h
60 LET v=v+(a$="6")-(a$="7")
70 LET h=h+(a$="8")-(a$="5")
80 IF SCREEN$(v,h)="" THEN GO
TO 1000
90 IF SCREEN$(v,h)="_" THEN L
ET v=v1: LET h=h1: LET a$="": GO
TO 55
100 PRINT AT v,h; INK 3;c$
105 LET d$=c$
110 LET c$=(" " AND a$="7")+(" "
AND a$="8")+(" " AND a$="5")+
(" " AND a$="5")
120 IF c$="" THEN LET c$=d$
125 PRINT AT 21,16;"TIME ";INT
ti;" ": LET ti=ti-.2: IF ti<0 TH
EN GO TO 2000
130 BEEP .008,50
140 PRINT AT j,k:" "
150 LET j1=j: LET k1=k
160 LET j=j+INT (RND*3)-1: LET
k=k+INT (RND*3)-1
170 IF SCREEN$(j,k)="_" THEN L
ET j=j1: LET k=k1: GO TO 150
175 IF SCREEN$(j,k)="" THEN GO
TO 1000
180 PRINT AT j,k; INK 5:"f"
200 GO TO 50

```



```

1000 PRINT AT j,k;"@
1010 FOR a=1 TO 3
1020 LET sc=sc+1
1030 PRINT AT 21,0;"SCORE ";sc
1040 BEEP .7,sc
1050 NEXT a
1060 LET j=4: LET k=20: LET v=10
: LET k=16: GO TO 40
2000 PRINT AT 21,16; PAPER 2;"TI
ME 0": BEEP 2,-10: CLS
2010 PRINT AT 2,12; PAPER 1;"GAM
E OVER"
2020 PRINT AT 5,10; PAPER 2;"YOU
SCORED ";sc
2030 IF sc>hi THEN LET hi=sc
2040 PRINT AT 18,6; PAPER 6;"HIG
HEST SCORE TODAY ";hi
2050 INPUT PAPER 1;"PRESS "; PAP
ER 2;"ENTER"; PAPER 1;" TO PLAY
AGAIN "; LINE a$: GO TO 30
7000 CLS : PRINT INVERSE 1; INK
6; "

```

```

7010 FOR a=1 TO 19
7020 PRINT INVERSE 1; INK 6;"_";
AT a,31;"_

```

DEATH RACE

```

7030 NEXT a
7040 PRINT INVERSE 1; INK 6; "
-----
7050 PRINT AT 21,0;"SCORE " : SC
7060 RETURN
8000 BORDER 0: PAPER 0: INK 9: C
LS
8010 LET v=10: LET h=16
8020 LET j=4: LET k=20
8030 LET sc=0: LET ti=60
8040 LET c$=" "
8050 LET a$=" "
8090 RETURN
9000 FOR a=USR "a" TO USR "f"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 60,153,255,153,24,169,
255,169
9040 DATA 238,68,229,255,255,229,
68,238
9050 DATA 169,255,169,24,153,255,
153,60
9060 DATA 119,34,167,255,255,167,
34,119
9070 DATA 56,124,238,198,238,238,
254,254
9080 DATA 56,56,16,124,16,16,40,
68
9090 REM a b c d e f
9099 REM 漢 語 漢 語 漢 語

```

DRAUGHTS

Si gioca secondo le regole convenzionali. Ogni giocatore parte con 12 pezzi; voi giocate su una superficie quadrata di 8x8. Il gioco è limitato ai quadrati neri, e le mosse sono in diagonale. Voi potete muovere verso l'alto e il computer verso il basso, finché non raggiungete il lato opposto del quadrato e diventate "King" (re), e potete muovere in tutte le direzioni.

VOI SIETE IL BIANCO E IL COMPUTER È IL NERO.

Per fare una mossa, dovete immettere la lettera e il numero del quadrato dal quale volete muovere (per esempio C6); poi premere ENTER e immettere il numero e la lettera del quadrato nel quale volete andare (per esempio E4). Se catturate una pedina del computer, questa sparirà automaticamente dalla scacchiera e vi sarà chiesto (apparirà il punto interrogativo) se volete muovere di nuovo. Immettete Y e premete ENTER per fare la mossa successiva.

PER GIOCARE, TUTTE LE LETTERE DEVONO ESSERE IMMESSE IN "UPPER CASE"; È PREFERIBILE EVITARE CHE IL COMPUTER SIA MESSO IN "CAPS-LOCK" PRIMA DI MANDARE IN ESECUZIONE IL PROGRAMMA.

DRAUGHTS

```

10 REM Draughts/Checkers
20 GO SUB 9000
30 GO SUB 1620
40 GO TO 1340
50 CLS
60 PRINT PAPER 1;TAB 12;"CHECK
ERS";
70 PRINT
80 PRINT TAB 2; PAPER 6;"ABCDE
FGH"
90 GO SUB 460
100 PRINT "YOUR score ";sm;"Mac
hines score ";si;" "
110 PRINT
120 IF u$="N" THEN GO TO 650
130 IF si=12 THEN PRINT "I win
": STOP
140 IF sm=12 THEN PRINT "You wi
n": STOP
150 REM
160 IF q=2 THEN GO TO 410
170 PRINT "Last to ";f$;
180 INPUT " From (Let.,No.):";c$
": to ";b$
220 LET f$=b$
230 LET d$=c$
240 GO SUB 1550
250 LET c=m(i)
260 IF i=0 THEN GO TO 180
270 LET d$=b$
280 GO SUB 1550
290 LET b=m(i)
300 IF i=0 THEN GO TO 180
310 IF ABS (c-b)=10 OR ABS (c-b
)=8 THEN LET sm=sm+1
320 LET u$=""
330 IF b-c=10 THEN LET a(b-5)=0
340 IF b-c=8 THEN LET a(b-4)=0
350 IF c-b=10 THEN LET a(c-5)=0
360 IF c-b=8 THEN LET a(c-4)=0
370 LET a(b)=a(c)
380 LET a(c)=0
390 LET q=2
400 GO TO 50
410 LET u$="": LET q=0
420 IF ABS (c-b)=10 OR ABS (c-b
)=8 THEN PRINT : PRINT : PRINT :
PRINT : INPUT u$
440 IF u$(0)="Y" THEN GO TO 650
450 GO TO 50
470 LET #1=0: LET k=1
480 LET j=-1
490 FOR i=40 TO 6 STEP -1
500 IF a(i)=1 AND i>37 THEN LET
a(i)=2
510 IF a(i)=-1 AND i<10 THEN LE

```

```

T a(i)=-2
520 IF i=14 OR i=32 OR i=23 THE
N GO TO 580
530 IF m1=0 THEN PRINT k;" ";
LET m1=0: LET k=k+1: LET j=-1#i:
IF j=1 THEN PRINT INK 2;"■";
540 LET a=a(i)
550 GO SUB 610
560 IF m1<>3 OR j=-1 THEN PRINT
INK 2;"■";
570 LET m1=m1+1: IF m1>3 THEN L
ET m1=0: PRINT
580 NEXT i
590 PRINT
600 RETURN
610 REM Print pieces
620 IF a=0 THEN PRINT " ";
630 IF a=1 THEN PRINT "●";
640 IF a=-1 THEN PRINT "0";
650 IF a=-2 THEN PRINT INVERSE
1;"0";
660 IF a=2 THEN PRINT INVERSE 1
;"●";
670 RETURN
680 LET U$="" LET q=0
690 LET z=6
700 IF z<9 THEN GO TO 740
710 IF a(z)<0 AND (a(z-4)=1 OR
a(z-4)=2) AND a(z-8)=0 THEN GO T
O 930
720 IF z<11 THEN GO TO 740
730 IF a(z)<0 AND (a(z-5)=1 OR
a(z-5)=2) AND a(z-10)=0 THEN GO
TO 1030
740 IF z>25 THEN GO TO 770
750 IF a(z)=-2 AND (a(z+4)=1 OR
a(z+4)=2) AND a(z+8)=0 THEN GO
TO 1140
760 IF a(z)=-2 AND (a(z+5)=1 OR
a(z+5)=2) AND a(z+10)=0 THEN GO
TO 1250
770 LET z=z+1: IF z<=40 THEN GO
TO 700
780 REM * RANDOM *
790 LET u=0
800 LET z=6+JNT (RND*34)+1
810 LET k=0
820 LET u=u+1
830 IF a(z)<0 AND a(z-4)=0 THEN
LET k=1
840 IF a(z)<0 AND a(z-5)=0 AND
k=0 THEN LET k=2
850 IF k=0 AND z<26 AND a(z)=-2
AND a(z+4)=0 THEN LET k=-7
860 IF z<10 THEN GO TO 880

```

DRAUGHTS

```

870 IF (k=1 OR k=2) AND v<200 AND (a(z-(10 AND z>10))=1 OR a(z-
8) =1) THEN GO TO 800
880 IF k=0 AND v<400 THEN GO TO
800
890 IF k=0 THEN LET sm=12: GO T
0 50
900 LET a(z-(3+k))=a(z)
910 LET a(z)=0
920 GO TO 50
930 LET a(z-8)=a(z)
940 LET a(z)=0
950 LET a(z-4)=0
960 LET si=si+1
970 IF z<24 THEN GO TO 50
980 IF (a(z-13)=1 OR a(z-13)=2)
AND a(z-18)=0 THEN LET p=2
990 IF p=1 THEN LET a(z-18)=a(z
-8): LET a(z-13)=00
1000 IF p=2 THEN LET a(z-12)=0:
LET a(z-12)=0
1010 IF p>0 THEN LET a(z-8)=0
1020 GO TO 50
1030 LET a(z-10)=a(z)
1040 LET a(z)=0
1050 LET a(z-5)=0
1060 LET si=si+1
1070 IF z<25 THEN GO TO 50
1080 IF (a(z-15)=1 OR a(z-15)=2)
AND a(z-20)=0 THEN LET p=1
1090 IF (a(z-14)=1 OR a(z-14)=2)
AND a(z-18)=0 THEN LET p=2
1100 IF p=1 THEN LET a(z-15)=0:
LET a(z-20)=a(z-10)
1110 IF p=2 THEN LET a(z-14)=0:
LET a(z-18)=a(z-10)
1120 IF p>0 THEN LET a(z-10)=0
1130 GO TO 50
1140 LET a(z+8)=-2
1150 LET a(z+4)=0
1160 LET a(z)=0
1170 LET si=si+1
1180 IF z<32 AND (a(z+3)=1 OR a(
z+3)=2) AND a(z-2)=0 THEN LET p=
1
1190 IF z<33 AND (a(z+14)=1 OR a
(z+14)=2) AND a(z+16)=0 THEN LET
p=2
1200 IF z<33 AND (a(z+13)=1 OR a
(z+13)=2) AND a(z+18)=0 THEN LET
p=3
1210 IF p=1 THEN LET a(z+3)=0: L
ET a(z-2)=-2
1220 IF p=2 THEN LET a(z+14)=0:
LET a(z+16)=0

```

```

1230 IF p=3 THEN LET a(z+13)=0:
LET a(z+18)=-2
1240 IF p>0 THEN LET a(z+8)=0
1250 LET a(z+10)=-2
1260 LET a(z+5)=0
1270 LET a(z)=0
1280 LET si=si+1
1290 GO TO 50
1300 PRINT : PRINT
1310 PRINT : PRINT
1320 RETURN
1330 REM * INITIALISE *
1340 DIM a(45)
1350 PRINT
1360 FOR z=1 TO 45
1370 IF z<6 THEN LET a(z)=9
1380 IF z>5 AND z<19 THEN LET a(z)=1
1390 IF z>18 AND z<33 THEN LET a(z)=0
1400 IF z>27 AND z<41 THEN LET a(z)=-1
1410 IF z>40 THEN LET a(z)=9
1420 NEXT z
1430 LET a(14)=9: LET a(23)=9: LET a(32)=9
1440 LET f$="--"
1450 LET p=0: LET q=0
1460 LET si=0: LET sm=0
1470 PRINT : PRINT
1480 INPUT "Do you want the first move y/n"; LINE q$
1490 LET u$=""
1500 PRINT
1510 IF q$(<>"Y") THEN GO TO 550
1520 LET u$=""
1530 PRINT
1540 GO TO 50
1550 REM Decode move
1560 LET i=1
1570 IF m$(i)=d$ THEN GO TO 1610
1580 LET i=i+1
1590 IF i=33 THEN LET i=0: GO TO 1610
1600 GO TO 1570
1610 RETURN
1620 DIM m$(32,2): DIM n(32)
1630 DATA "B1", "D1", "F1", "H1", "A2", "C2", "E2", "G2", "B3", "D3", "F3", "H3", "A4", "C4", "E4", "G4", "B5", "D5", "F5", "H5", "A6", "C6", "E6", "G6"
1640 DATA "G6", "B7", "D7", "F7", "H7", "A8", "C8", "E8", "G8"
1650 DATA 40,39,38,37,36,35,34,33,31,30

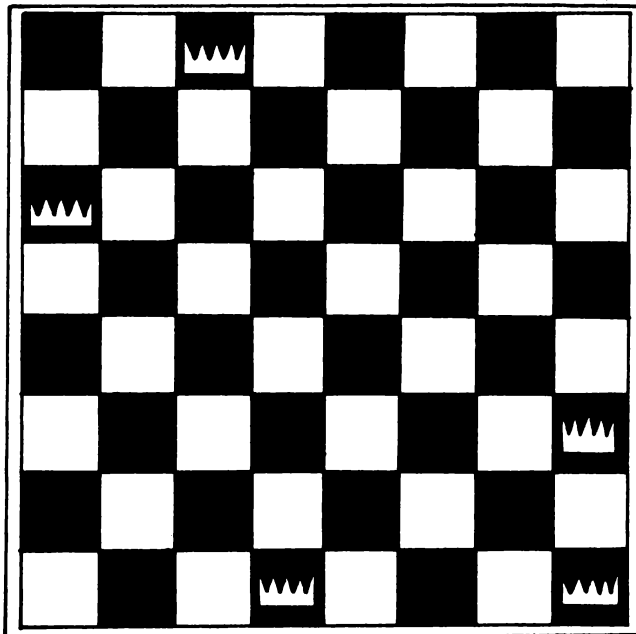
```

DRAUGHTS

```

1660 DATA 29,28,27,26,25,24,22,2
1,20,19
1670 DATA 18,17,16,15,13,12,11,1
0,9,8,7,6
1675 RESTORE 1630
1680 FOR i=1 TO 32
1690 READ M$(i)
1700 NEXT i
1710 FOR i=1 TO 32
1720 READ M(i)
1730 NEXT i
1740 RETURN
9000 BORDER 0: PAPER 0: INK 9: C
L5
9005 RESTORE 9030: FOR a=USR "a"
TO USR "a"+7
9010 READ User: POKE a,User
9020 NEXT a: RETURN
9030 DATA 0,00,126,126,126,126,6
8,0
0000 DIM a
0000 DIM i

```



SIMON SAID

È una versione del gioco omonimo. Il computer seleziona una sequenza di colori, accompagnati da suoni, che voi dovete ripetere esattamente. Ci sono quattro colori diversi. La sequenza inizia con un colore scelto a caso, e ogni volta che la sequenza viene ripetuta, viene aggiunto un colore. La lunghezza massima della sequenza è da 1 a 10.

```

10 REM Simon Said...
20 REM Tim Hartnell
   Peter Shaw
30 GO SUB 9000: GO SUB 8000
40 FOR a=1 TO 10
50 LET c$=c$+STR$ (INT (RND#4)
+1)
60 NEXT a
70 FOR q=1 TO x
80 LET l=4*(CODE c$(q)-48)
90 LET t=VAL c$(q)
100 BEEP .05,10*t
110 PRINT AT l,7: INK 6;" ■■■ ";Z
NT l/4;AT l-1,7;" ■■■ ";AT l+1,7;"
120 PAUSE 4
130 CLS
140 NEXT q
150 FOR b=1 TO x
160 IF INKEY$<>" " THEN GO TO 15
@
170 LET t$=INKEY$
180 IF CODE t$=0 THEN GO TO 170
190 CLS
200 LET y=4*(CODE t$-48)
210 PRINT AT y,7: INK y/4;" ■■■ "
; INK 6;t$;AT y-1,7: INK y/4;" ■
";AT y+1,7;"
220 BEEP .01,2.5*y
230 IF CODE t$<>CODE (c$(b)) TH
EN GO TO 1000
240 PAUSE 7
250 CLS
260 NEXT b
270 IF x=10 THEN PRINT "You def
eat the machine!": INPUT "Press

```

```

Enter to play again "; LINE a$:
RUN
280 LET x=x+1
290 PAUSE 50
300 GO TO 70
1000 PRINT "You scored ";x-1
1010 BEEP .02,RND*30
1020 INPUT "Press Enter to play
again "; LINE a$: RUN
0000 BORDER 0: PAPER 0: INK 7: C
LS : BRIGHT 1
0010 LET c$=""
0020 LET x=1
0090 RETURN
0000 FOR a=USR "a" TO USR "d"+7
0010 READ user: POKE a,user
0020 NEXT a: RETURN
0030 DATA 0,0,0,0,3,7,15,15
0040 DATA 0,0,0,0,192,224,240,24
0
0050 DATA 15,15,7,3,0,0,0,0
0060 DATA 240,240,224,192,0,0,0,
0

```



BLUEGREEN
REDYELLOW

3D DRIVER

È un adattamento particolare di alcuni giochi di guida di stile "Arcade". Vi viene data una visione tridimensionale della vostra automobile e la prospettiva della strada, che è in continuo movimento da un lato all'altro. Si deve assolutamente evitare di urtare contro i bordi della strada. Usate i tasti 5 e 8 del cursore per controllare l'automobile.

```

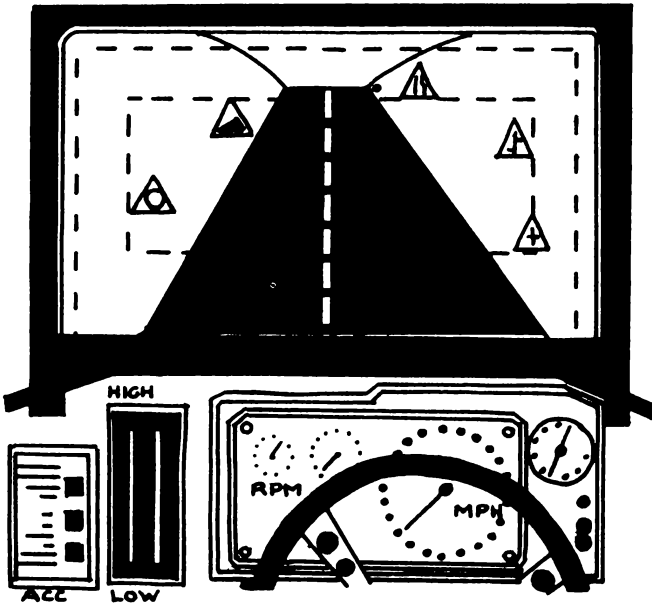
10 REM 3D Driver
20 GO SUB 9000
30 GO SUB 8000
40 PRINT AT 15,15: INK 4: "  "
AT 16,15: " "
50 IF p<100 OR p>150 THEN GO T
O 1000
60 PLOT p-10,159: DRAW -170+(p
-10),-159
70 PLOT p+10,159: DRAW -70+(p-
10),-159
80 PLOT OVER 1:p-10,159: DRAW
OVER 1:-170+(p-10),-159
90 PLOT OVER 1:p+10,159: DRAW
OVER 1:-70+(p-10),-159
100 LET p=p+4+(INKEY#="5")-4+(I
NKEY#="8")
110 LET p=p+(INT (RND*20)-10)
115 LET sc=sc+1
120 GO TO 50
990 STOP
1000 PLOT p-10,159: DRAW -170+(p
-10),-159
1010 PLOT p+10,159: DRAW -70+(p-
10),-159
1020 PRINT AT 15,15: FLASH 1: IN
K 2: "  ";AT 16,15: " "
1030 FOR a=0 TO 50: BEEP .05,-a:
BEEP .008,a: NEXT a
1040 PRINT AT 1,10: PAPER 1:"You
scored ";sc
1050 INPUT "Press "; INK 6;"ENTE
R"; INK 7;" to play again "; LIN
E a$: GO TO 30
8000 BORDER 0: PAPER 0: INK 7: C
LS
8010 LET p=127
8020 LET sc=0

```



```
8040 PRINT AT 0,0: PAPER 1;"
```

```
8990 RETURN
9000 FOR a=USR "a" TO USR "d"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 0,0,0,0,13,11,10,9
9040 DATA 0,0,0,0,176,208,176,19
9050 DATA 6,117,207,223,208,240,
12,3
9060 DATA 102,174,240,251,19,15,
48,192
9998 REM ab
      cd
9999 REM
```



DROP OUT

È un gioco relativamente semplice. Una pallina si trova in cima a una superficie, in alto sullo schermo. Dovete valutare la forza necessaria per farla cadere in una buca posta in basso sullo schermo. Per far ciò, inserite nel computer un numero da 1 a 9.

Questo programma dovrebbe aiutarvi a valutare le distanze sullo schermo, e ciò vi servirà quando arriverete a scrivere programmi in cui il testo dev'essere allineato al centro.

```

10 REM Drop out
20 GO SUB 2000
30 GO SUB 5000
40 GO SUB 7000
50 INPUT "Force ? "; LINE a$:
IF a$>"9" OR a$<"1" THEN GO TO 5
0
60 FOR a=0 TO 8
70 PRINT AT 3,a;"●"
80 BEEP .1,a
110 PRINT AT 3,a;" " : NEXT a
120 FOR a=4 TO 15
130 PRINT AT a,8;"●"
140 BEEP .1,a
170 LET b=b+.5
180 PRINT AT a,8;" "
190 LET a=a+b
200 NEXT a
210 PRINT AT 15,8;"●"
220 BEEP .2,5
230 PRINT AT 15,8; INK 1;"L"
240 FOR a=1 TO VAL a$*2
250 PRINT AT 15,8+a;"●"
260 BEEP .1,a
290 PRINT AT 15,8+a;" "
300 NEXT a
310 IF VAL a$=h THEN PRINT AT 1
5,8+(h*2);"●": LET sc=sc+1; PRIN
T AT 21,20; PAPER 8; INK 9;"SCOR
E ";sc; BEEP 1,30
315 BEEP 1,10
320 GO TO 40
7000 REM


```

DROP OUT

```

7010 FOR a=4 TO 15
7020 PRINT AT a,0: INK 1; "
7030 NEXT a
7040 FOR a=16 TO 21
7050 PRINT INK 1; "
7060 NEXT a
7070 PRINT AT 15,8: INK 1; "L"
7080 LET h=INT (RND*9)+1
7090 PRINT AT 16,8+(h*2); " "
7100 RETURN
8000 LET sc=0
8010 BORDER 0: PAPER 0: INK 6: C
LS
8020 LET b=.5
8090 RETURN
9000 FOR a=USR "a" TO USR "b"←7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 60,126,255,255,255,255
,126,60
9050 DATA 126,126,126,126,126,12
0,224,255
9080 REM a b
9090 REM ● L

```

DROP OUT


M4

Gettarsi nella mischia dell'autostrada M4 nell'ora di punta non è un'idea del tutto geniale, ma tale è lo scopo di questo gioco. Dovete evitare due corsie di automobili che procedono in direzioni opposte; se siete urtati da un veicolo, perdete una delle vostre cinque "vite". Per ogni uomo portato in salvo guadagnate 10 punti. C'è la possibilità di realizzare un punteggio alto.

Usate i tasti del cursore per controllare il movimento.

```

10 REM M4
20 GO SUB 9000: LET hi=0
30 GO SUB 8000
40 LET v=14: LET h=16: GO SUB
7000
50 PRINT AT v,h; PAPER 8;" "
60 LET v=v+(INKEY$="6" AND v<2
1)- (INKEY$="7" AND v>0)
70 LET h=h+(INKEY$="8" AND h<0
1)- (INKEY$="5" AND h>0)
80 PRINT AT 4,0; INK 6;a$;AT 2
1,0;b$;AT 5,0;b$;AT 12,0;a$
90 IF SCREEN$(v,h)="" THEN GO
TO 1000
100 PRINT AT v,h; PAPER 8;"I"
110 LET b#=b$(32)+b$( TO 32)
120 LET a#=a$(2 TO )+a$(1)
130 IF v=2 THEN GO TO 2000
150 GO TO 50
1000 PRINT AT v,h; FLASH 1;"I"
1010 DIM s$(1,704)
1020 PRINT AT 0,0; FLASH 8; OVER
1; PAPER 8; INK 2; s$(1)
1025 IF m1>0 THEN LET m1=m1-1: G
O TO 40
1025 IF sc=h1 THEN LET h1=sc
1030 INPUT "Press ENTER to play
again "; LINE b$
1040 GO TO 30
2000 LET sc=sc+10
2010 GO TO 40
7000 CLS
7010 PRINT (TAB 12;"HIGH ";hi;TA
B 23;"SCORE ";sc); PAPER 1;" "
7020 PRINT INK 6'a$'b$
7030 PRINT ' PAPER 4;"

```

```

7050 PRINT INK 6;b$a$
7060 PRINT " PAPER 1;"
7070 PRINT AT v,h; PAPER 3;"*"
7080 FOR a=1 TO m1: PRINT AT 1,a
: INK 5;"*"; NEXT a
7090 RETURN
8000 BORDER 0: PAPER 0: INK 9: 0
LS
8010 LET sc=0
8030 LET a$=""
8040 LET b$=""
8050 LET m1=5
8090 RETURN
9000 FOR a=USR "a" TO USR "z"
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 0,1,2,127,235,253,25,5
9040 DATA 0,240,16,252,215,167,5
6,16
9050 DATA 0,15,8,63,235,221,25,5
9060 DATA 0,128,64,254,215,191,0
8,8
9070 DATA 28,28,8,8,62,8,28,34

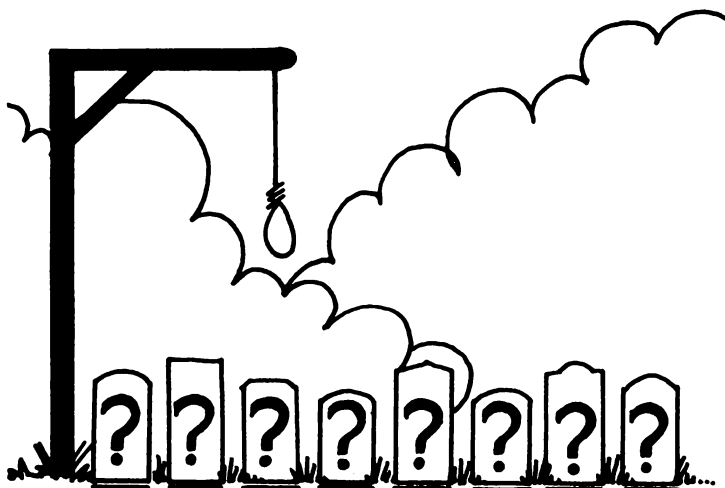
```



HANGMAN

È una versione del celebre gioco di parole. Il computer mostra un numero di tratti corrispondenti al numero di lettere della parola. Voi immettete una lettera; se si trova nella parola, appare sullo schermo e voi potete di nuovo indovinare. Se sbagliate, il computer comincia a dare l'immagine del "poveretto", pronto ad essere impiccato. Avete 13 possibilità prima che l'uomo sia messo a morte. Non vi conviene enumerare tutte le lettere dell'alfabeto in ordine: in questo modo è molto difficile arrivare alla soluzione.

Quando scegliete una lettera e la inserite nel computer, appariranno tutti gli esempi di quella lettera: per esempio "carrot" ha due "r", ed entrambe devono essere inserite nel computer. Come si può vedere dal tabulato, questo programma ha un dizionario piuttosto esteso; cosicché chi ha "battuto" il listato può continuare a giocare senza conoscere volta per volta la parola scelta.



HANGMAN

```

10 REM Hangman
20 GO SUB 9000
30 GO SUB 9000
40 GO SUB 7000
50 FOR a=1 TO LEN w$
60 PRINT "- ";
70 NEXT a
80 INPUT "GUESS (ONLY 1 LETTER
); LINE n$
90 LET r=0: FOR a=1 TO LEN w$
100 IF w$(a)=n$ THEN LET r=r+1
110 NEXT a
120 IF r<>0 THEN GO TO 1000
130 LET t$=t$+n$+" "
140 GO SUB 6000
150 LET c$="": FOR a=1 TO LEN w$
$: LET d$=SCREEN$(0,(a*2)-2): L
ET c$=c$+d$: NEXT a: IF c$=w$ TH
EN GO TO 3000
160 GO TO 30
1000 FOR a=1 TO LEN w$
1010 IF w$(a)=n$ THEN PRINT AT 0
,(a*2)-2;n$
1020 NEXT a
1030 GO TO 150
2000 PRINT AT 5,19;"!": BEEP 1,-
10
2010 PRINT AT 5,0;"YOU KILLED hi
m!"
2011 PRINT AT 7,0;"the word was.
";AT 8,0;w$
2020 INPUT "Press enter to play
again "; LINE j$: GO TO 30
5000 PRINT AT 5,0;"Got it!"
5010 INPUT "Press enter to play
again "; LINE j$: GO TO 30
6000 LET d=d+2: LET h=h+2: IF h>
31 THEN LET v=v+1: LET h=18
6010 PRINT AT v,h; INK 4;n$
6020 PRINT AT INT (d/2)+2-5,17;
INK 6;b$(INT ((d-16)/2))
6030 IF INT ((d-16)/2)=13 THEN G
O TO 2000
6040 RETURN
7000 RESTORE 7000
7005 FOR a=1 TO INT (RAND*193)+1
7010 READ w$
7020 NEXT a
7030 DATA "airship","aircraft","
ape","anvil","apply","artic","ar
my","arthropod","barrow","binary
","birch","bizzare","bite","btis
ter","blood","bobtail","bradawi
","brain","camera","canoe","cap"
,"care","castanets","catch","cell

```

```

" , "chapel", "chariot", "classic", "
coach"
7040 DATA "doublet", "duckling", "e
dune", "dual", "dwarf", "embank", "e
mploy", "encounter", "engine", "ent
iven", "envelope", "equinox", "fade
", "feel", "fence", "fill", "finish"
, "fish", "flag"
7050 DATA "gauge", "gear", "geomet
ry", "glassy", "gnu", "gold", "grade
s", "grudge", "heart", "head", "heli
copter", "heraldry", "hospital", "h
overcraft", "ice", "imperial", "in
low", "jib", "jersey", "joiner", "ju
venile", "kale", "kennel", "kneel",
"labarum", "landau", "lathe", "lead
er", "leghorn", "light", "lip"
7060 DATA "mass", "match", "maze",
"medium", "mellow", "melt", "mill",
"moon", "motor", "mount", "mystify",
ject", "nerve", "niche", "nostalgia", "ob
ject", "once", "optical", "orange", "o
orb", "ordinary", "orphan", "obtus
e", "oversee", "ozone"
7070 DATA "pall", "pancake", "pant
ograph", "parcel", "parallax", "par
alimint", "particle", "phaeton", "pi
ck", "pillory", "plaise", "plane",
pleat", "poetry", "postage", "pound
", "project", "propeller", "quadrit
le", "quilt", "queue", "quote", "rab
bit", "raffle", "reciprocal"
7080 DATA "remand", "remain", "res
idence", "revenue", "reach", "savag
e", "sage", "scarp", "scare", "scurr
y", "sea", "second", "semaphore", "s
entence", "shampoo"
7090 DATA "tassel", "tatty", "taz
a", "teeter", "teller", "tennis", "t
ension", "tetanus", "thick", "tick"
, "tiger", "titanic", "toast", "toni
c", "torrid", "tribe", "umpire", "un
couth", "understate", "unruly", "un
seat", "urge"
7100 DATA "vacuum", "valentine", "v
agrant", "valve", "variable", "van
lt", "vein", "visit", "wage", "walte
t", "warn", "warning", "water", "wax
", "weapon", "wells", "whale", "whir
l", "whippet", "whistle", "xylophon
e", "year", "youth", "yolk", "zany",
"zebra", "zoom", "zoologist"
7110 RETURN
8000 BORDER 0: PAPER 0: INK 0: C
LS

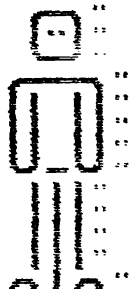
```


HANGMAN

```

0010 DIM b# (10,5)
0020 LET b#(1) = ""
0030 LET b#(2) = ""
0040 LET b#(3) = ""
0050 LET b#(4) = ""
0060 LET b#(5) = ""
0070 LET b#(6) = ""
0080 LET b#(7) = ""
0090 LET b#(8) = ""
0100 LET b#(9) = ""
0110 LET b#(10) = ""
0120 LET b#(11) = ""
0130 LET b#(12) = ""
0140 LET b#(13) = ""
0150 LET t# = ""
0160 LET h=15: LET v=0
0170 LET d=16
0990 RETURN
9000 RESTORE 9000: FOR a=USR "a"
  TO USR "z"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 0,0,0,7,15,23,24,24
9040 DATA 24,24,23,16,7,0,0,0
9050 DATA 0,0,0,224,240,55,24,0
9060 DATA 24,24,55,240,204,0,0,0
9070 DATA 0,0,0,255,255,0,0,0
9080 DATA 24,24,24,24,24,24,24,24
4
9090 DATA 0,0,102,102,0,0,0,0

```



VIDEO SALESMAN

Quanto potete resistere nell'arduo mondo del commercio dei videoregistratori? In questo gioco comprate e vendete videoregistratori; li comprate a 200 sterline ognuno, poi scegliete un prezzo a cui rivenderli. Il numero di apparecchi che voi vendete in una settimana dipende dal prezzo di vendita e dalla situazione di mercato (visualizzata dal computer prima di ogni gioco).

Bisogna stare attenti, quando si compra, perché se non si riesce a vendere in una settimana, non si può vendere la settimana successiva. Il gioco dura 5 settimane e all'inizio avete un capitale di 1000 sterline.

```

10 REM Video Salesman
20 REM
30 GO SUB 8000
40 INPUT AT 22,8;AT 8,6;"How many
any players ";LINE a$;AT 12,6;"
Press ENTER to play";AT 22,30;LINE a$
50 LET p=VAL p$: IF p<1 THEN G
O TO 40
55 DIM s(p); DIM c(p); FOR a=1
TO p: LET c(a)=1000: NEXT a
60 FOR w=1 TO 5
70 FOR l=1 TO p
75 IF c(l)<200 THEN PRINT "Sales
esman ";l,,"You have not got eno
ugh money to buy any stock": NEXT
l: NEXT w: GO TO 270
80 CLS
90 PRINT "Salesman ";l,"Sales
";s(l)
95 PRINT "Cash in hand ";c(l)
100 GO SUB 2000
110 PRINT "Local & National Re
ws :-"
120 PRINT h$

```

VIDEO SALESMAN

```

130 INPUT AT 15,0;AT 0,0;"How #
any recorders will you stock
this week (£200 ea) ";re;AT 5,0
;"How much will you charge per
recorder ";ch;AT 15,5;"Press E
NTER to continue"; LINE a$
135 IF re<200>c(1) THEN GO TO 1
30
140 CLS
150 PRINT "Salesman ";l
160 LET sa=(h*(10)/(ch/10))
170 IF sa>re THEN LET sa=re
180 PRINT "Cash in hand ";c(1)
190 LET s(1)=INT sa
200 PRINT "Sales this week ";s
(1)
210 LET pr=(s(1)*ch)-(re*200)
220 PRINT "Profit ";pr
230 LET c(1)=c(1)+pr
240 PRINT "New balance ";c(1)
250 INPUT "Press enter to conti
nue "; LINE a$: NEXT l
260 NEXT w
990 STOP
2000 LET h=INT (RND*20)+1
2010 IF h>10 THEN GO TO 2500
2020 RESTORE 2000
2030 FOR a=1 TO INT (RND*6)+1: R
EAD h$: NEXT a
2040 RETURN
2050 DATA "This is World Cup wee
k!","E.T. has just been release
d on video officially","Video rec
orders go down in price","Specia
l offer on Blank tapes","It's ge
tting near Christmas","New video
tape hire shop opens in town"
2500 RESTORE 2500
2510 FOR a=1 TO INT (RND*5)+1: R
EAD h$: NEXT a
2520 RETURN
2530 DATA "Inflation hits video
industry","Cheap video discs bec
ome popular","Cable television c
uts sales","Cinema has increased
popularity","Slump in video ind
ustry"
8000 BORDER 0: PAPER 0: INK 0: C
LS
9990 RETURN

```

DAM BUSTERS

Volando di ritorno da una missione in Europa, avete noie al motore, e cominciate a perdere quota. L'unica possibilità di salvezza è quella di tentare di ammarare su un fiume. Sfortunatamente c'è una diga e voi dovete farla saltare.

Usare il tasto Ø per sganciare le bombe.

```

10 REM Dam Busters
20 GO SUB 9000: LET hi=0
30 GO SUB 8000
40 GO SUB 7000
50 FOR v=1 TO 15
55 PRINT AT v-1,0; PAPER 8;"
"
60 FOR h=0 TO 31
70 PRINT AT v,h; INK 6; PAPER
8: " ████████ "
90 IF SCREEN$(v,h+6) <> " " THEN
N GO TO 1000
90 IF INKEY$="Ø" THEN GO SUB 2
00
95 LET sc=sc+1
100 BEEP (.008 AND ch=1)+(.01 AND
ND ch=0),0
105 IF ch=0 THEN GO TO 120
110 IF (SCREEN$(13,15)=" " AND
SCREEN$(13,17)=" " AND SCREEN$(
113,18)=" ") THEN LET ch=0: P
NT AT 13,0;" " ;AT
14,20; INK 2; PAPER 1;"
";AT 13,16; PAPER 0;" "
120 NEXT h
130 NEXT v
140 GO TO 1000
150 PRINT AT v+1,0; INK 5; PAPE
R 8;" * "
210 PRINT AT v,h; PAPER 6; INK
5: " ████████ "
220 LET h=h+1; IF h>30 THEN LET
v=v+1; LET h=0
240 PRINT AT v+1,h; INK 5; PAPE
R 8;" * "

```

DAM BUSTERS

```

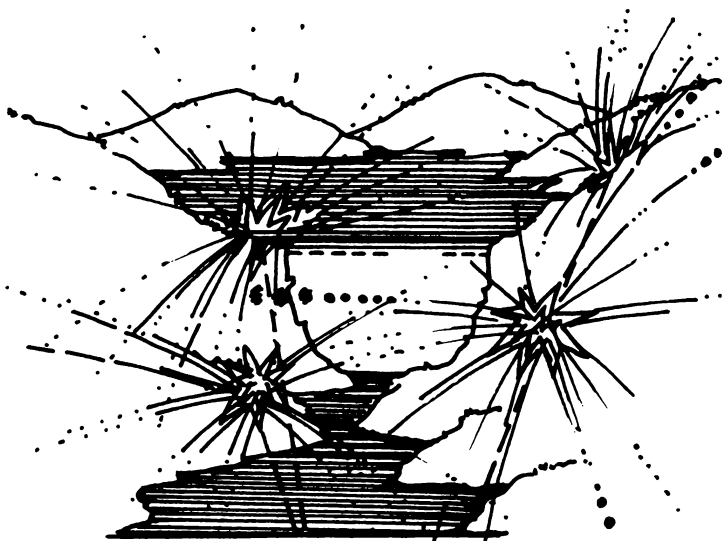
250 PRINT AT v,h; PAPER 8; INK
6;" "
260 LET h=h+1: IF h>31 THEN LET
v=v+1: LET h=0
275 PRINT AT v+2,h; PAPER 8;" "
280 LET b1=h
290 FOR b=v+1 TO 14
295 BEEP .01,2
300 PRINT AT b,b1; PAPER 8; INK
5;" ";AT b+1,b1;"$"
310 PRINT AT v,h; INK 6; PAPER
8;" "
320 LET h=h+1: IF h>30 THEN LET
v=v+1: LET h=0
330 IF SCREEN$(b+2,b1)<>" " TH
EN GO TO 500
335 PRINT AT v-1,32;" "
340 NEXT b
350 PRINT AT b,b1; PAPER 8;" "
360 LET h=h-1
380 RETURN
500 FOR b=b TO b+1
510 PRINT AT b,b1; PAPER 8;" ";
AT b+1,b1; INK 5;"$"
520 NEXT b
524 BEEP .005,-b
525 LET h=h-1: PRINT AT b,b1; P
APER 8;" "
530 RETURN
1000 FOR a=v TO 15
1010 PRINT AT a,h+1; PAPER 1;"
";AT a+1,h+1; INK 6;" "
1020 BEEP .5,-a: NEXT a
1030 GO TO 1510
1500 LET sc=sc+100: PRINT AT 0,1
2;"GAME OVER"
1510 PRINT AT 5,10;"YOU SCORED "
:sc
1520 IF sc>hi THEN LET hi=sc
1530 PRINT AT 10,6;"HIGHEST SCORE
a today ";hi
1540 INPUT "Press ENTER to play
again "; LINE a$: GO TO 30
7000 CLS
7010 PRINT AT 13,0; PAPER 1;"
7020 PRINT AT 14,0; PAPER 1;"
7030 PRINT AT 15,0; PAPER 1;"
7040 PRINT AT 12,16; INK 2;" "
AT 13,16; PAPER 1;" "; PAPER 8;
";AT 14,16; PAPER 1;" "; PA
PER 8;" ";AT 15,16; PAPER 1;"

```

```

7050 FOR a=1 TO 4
7060 PRINT PAPER a;
7070 NEXT a
7080 RETURN
8000 BORDER 0: PAPER 0: INK 0: 0
GOTO
9010 LET sc=0
9020 LET sb=1
9030 RETURN
9040 FOR a=USR "a" TO USR "z"+7
9050 READ user: POKE a,user
9060 NEXT a: RETURN
9070 DATA 50,50,60,60,60,70,80,80
9080 DATA 9,9,100,100,100,100,100
9090 DATA 0,0,100,100,100,100,100
9100 DATA 0,0,204,240,240,240,24
9110 DATA 0,0,200,100,100,100,0
9120 DATA 0,0,100,100,100,100,0
9130 DATA 100,0,100,100,100,100,0
9140 DATA 100,0,100,100,100,100,0
9150 DATA 100,0,100,100,100,100,0
9160 DATA 100,0,100,100,100,100,0
9170 DATA 100,0,100,100,100,100,0
9180 DATA 100,0,100,100,100,100,0
9190 DATA 100,0,100,100,100,100,0
9200 DATA 100,0,100,100,100,100,0
9210 DATA 100,0,100,100,100,100,0
9220 DATA 100,0,100,100,100,100,0
9230 DATA 100,0,100,100,100,100,0
9240 DATA 100,0,100,100,100,100,0
9250 DATA 100,0,100,100,100,100,0
9260 DATA 100,0,100,100,100,100,0
9270 DATA 100,0,100,100,100,100,0
9280 DATA 100,0,100,100,100,100,0
9290 DATA 100,0,100,100,100,100,0
9300 DATA 100,0,100,100,100,100,0
9310 DATA 100,0,100,100,100,100,0
9320 DATA 100,0,100,100,100,100,0
9330 DATA 100,0,100,100,100,100,0
9340 DATA 100,0,100,100,100,100,0
9350 DATA 100,0,100,100,100,100,0
9360 DATA 100,0,100,100,100,100,0
9370 DATA 100,0,100,100,100,100,0
9380 DATA 100,0,100,100,100,100,0
9390 DATA 100,0,100,100,100,100,0
9400 DATA 100,0,100,100,100,100,0
9410 DATA 100,0,100,100,100,100,0
9420 DATA 100,0,100,100,100,100,0
9430 DATA 100,0,100,100,100,100,0
9440 DATA 100,0,100,100,100,100,0
9450 DATA 100,0,100,100,100,100,0
9460 DATA 100,0,100,100,100,100,0
9470 DATA 100,0,100,100,100,100,0
9480 DATA 100,0,100,100,100,100,0
9490 DATA 100,0,100,100,100,100,0
9500 DATA 100,0,100,100,100,100,0
9510 DATA 100,0,100,100,100,100,0
9520 DATA 100,0,100,100,100,100,0
9530 DATA 100,0,100,100,100,100,0
9540 DATA 100,0,100,100,100,100,0
9550 DATA 100,0,100,100,100,100,0
9560 DATA 100,0,100,100,100,100,0
9570 DATA 100,0,100,100,100,100,0
9580 DATA 100,0,100,100,100,100,0
9590 DATA 100,0,100,100,100,100,0
9600 DATA 100,0,100,100,100,100,0
9610 DATA 100,0,100,100,100,100,0
9620 DATA 100,0,100,100,100,100,0
9630 DATA 100,0,100,100,100,100,0
9640 DATA 100,0,100,100,100,100,0
9650 DATA 100,0,100,100,100,100,0
9660 DATA 100,0,100,100,100,100,0
9670 DATA 100,0,100,100,100,100,0
9680 DATA 100,0,100,100,100,100,0
9690 DATA 100,0,100,100,100,100,0
9700 DATA 100,0,100,100,100,100,0
9710 DATA 100,0,100,100,100,100,0
9720 DATA 100,0,100,100,100,100,0
9730 DATA 100,0,100,100,100,100,0
9740 DATA 100,0,100,100,100,100,0
9750 DATA 100,0,100,100,100,100,0
9760 DATA 100,0,100,100,100,100,0
9770 DATA 100,0,100,100,100,100,0
9780 DATA 100,0,100,100,100,100,0
9790 DATA 100,0,100,100,100,100,0
9800 DATA 100,0,100,100,100,100,0
9810 DATA 100,0,100,100,100,100,0
9820 DATA 100,0,100,100,100,100,0
9830 DATA 100,0,100,100,100,100,0
9840 DATA 100,0,100,100,100,100,0
9850 DATA 100,0,100,100,100,100,0
9860 DATA 100,0,100,100,100,100,0
9870 DATA 100,0,100,100,100,100,0
9880 DATA 100,0,100,100,100,100,0
9890 DATA 100,0,100,100,100,100,0
9900 DATA 100,0,100,100,100,100,0
9910 DATA 100,0,100,100,100,100,0
9920 DATA 100,0,100,100,100,100,0
9930 DATA 100,0,100,100,100,100,0
9940 DATA 100,0,100,100,100,100,0
9950 DATA 100,0,100,100,100,100,0
9960 DATA 100,0,100,100,100,100,0
9970 DATA 100,0,100,100,100,100,0
9980 DATA 100,0,100,100,100,100,0
9990 DATA 100,0,100,100,100,100,0

```



Come scrivere programmi migliori

di Tim Hartnell

Questo libro contiene molti, ottimi programmi, come del resto la maggior parte delle riviste specializzate in computer. D'altra parte, per quanto siano buoni i programmi proposti dalle pubblicazioni, è certo che far "girare" un programma parzialmente e completamente scritto da te ti offre un piacere maggiore. Variando i programmi, dai il tuo tocco personale, rifletti i tuoi desideri, la tua creatività. Questo è un modo eccellente per migliorare i programmi e alla fine, naturalmente, diverrai un programmatore migliore e la tua immaginazione sarà potenziata.

I programmi, presentati dai periodici o da libri come questo, sono un ideale trampolino di lancio per i tuoi progressi in questo settore. Scoprirai che perfino la pubblicità sul software può ispirarti delle idee. Devi solo leggere la descrizione dei programmi commercialmente disponibili e farai il primo passo verso la creazione del "tuo" programma. Naturalmente, non ti è consentito di violare la legge sul diritto d'autore per quanto riguarda i dati e le informazioni illustrati sul video, il nome del programma, o i nomi dei "caratteri" all'interno del programma. Probabilmente scoprirai che il programma, a mano a mano che si sviluppa, prenderà vita propria, crescerà e si evolverà staccandosi dallo schema originale finché il tuo gioco risulterà concepito e realizzato in un modo completamente nuovo.

Qualsiasi cosa tu faccia, non far passare mai come tuo il lavoro di un altro. Adatta e migliora i programmi pubblicati con ogni mezzo, ma poi non presentarli ai periodici come se fossero originali. Ho perduto il conto delle volte in cui mi è stata proposta la pubblicazione di un programma che era stato tratto da uno dei miei libri. Ricordati che i libri e i periodici specializzati in giochi e computer sono tutti ottimi spunti per le tue idee. Troverai sulla tua strada sempre nuove idee per giochi, forme di caratteri, suoni, conclusioni a sfondo drammatico, e così via. È opportuno che tu ne prenda nota. In questo modo non ti troverai mai a corto di idee e sarai in grado di unire il materiale per produrre programmi migliori e capaci di tener desta l'attenzione del giocatore più a lungo.

I giochi si dividono in tre categorie. È utile quindi che tu individui la categoria alla quale appartiene il programma che proponi **prima** di iniziarlo, dato che la categoria del gioco altera materialmente il modo di dare inizio alla programmazione. Ciò non significa che a mano a mano che lo sviluppi, il programma non possa passare da una categoria all'altra, né che un particolare gioco non possa svilupparsi tra due categorie, ma è tuttavia utile considerare i vari gruppi separati per avere idee chiare. Le tre categorie sono:

- 1) Board games (scacchi, dama, e simili)
- 2) Arcade games (giochi di movimento veloce, di fantasia, pieni di rumori, in tempo reale)
- 3) Giochi di fortuna (come Roulette e Snap)

Nei "Board games" la qualità del gioco è più importante della velocità della risposta. Negli "Arcade games" il movimento deve essere mantenuto ad ogni costo, anche se talune "intelligenze" tra i tuoi invasori Marziani devono essere, a tale scopo, sacrificate. I giochi di fortuna si basano sulla facilità di gioco e sulla casualità vera e propria in modo maggiore rispetto alle altre due categorie.

Scoprirai che i programmi dei giochi si dividono in generi, i quali a loro volta sono suddivisioni delle tre categorie di cui sopra. Molti "board" games sono varianti della dama o degli scacchi, molti "arcade" games erano originariamente giochi del tipo "Invasori dello spazio" mentre i giochi di fortuna sono nati nel "mondo reale" dei dati e delle carte. Prestare attenzione alla descrizione di un programma o a una macchina da gioco, per provare ad indovinare la categoria a cui appartiene il gioco che stai osservando, può aiu-

tarti e ispirarti nuove idee che si adattino a quel particolare genere di giochi. Esiste un concetto nell'ambito della programmazione — chiamato generalmente "programmazione strutturata" — secondo il quale la disciplina è essenziale all'inizio del processo di scrittura dei programmi. Pur essendo meno interessante del fatto di sedersi subito davanti al computer, alla fine il programma prodotto sarà migliore. Una volta scrissi un programma chiamato Dome Dweller (Abitante di cupole) un programma a simulazione nel quale il giocatore ha il controllo di una "cupola lunare" e deve decidere quali prodotti produrre e vendere al fine di comprare ossigeno e cibo per gli abitanti del posto (vedi *The book of Listings*, scritto da Jeremy Ruston e edito dalla BBC). Dopo aver deciso lo schema generale da adottare, elaborai i dati e le informazioni da scrivere sul video, nel modo seguente:

Le scorte di ossigeno sono scarse

All'interno della tua cupola vivono 96 persone nell'anno 3

Il credito in denaro è di \$ 5.693

Le spese di manutenzione annuali ammontano a \$ 226

Le cisterne di ossigeno sono di 811 unità

L'ossigeno costa \$ 8 per unità

Ogni abitante di cupola necessita di 5 unità all'anno

Le riserve di cibo ammontano a 2122

Ogni abitante necessita di 3 unità all'anno (\$ 6 ognuno, \$ 576 per la cupola. Ciò basterà per 7 anni alla popolazione attuale)

Tu puoi commerciare le tue straordinarie sculture lunari con la gente che vive nelle altre cupole. Consumi 2 unità di ossigeno per la costruzione di un'unica scultura, e le vendi a \$ 30.

Come hai probabilmente indovinato da questo "printout" l'idea del programma è quella di decidere quante "straordinarie sculture lunari" devi costruire e vendere al fine di comprare ossigeno e cibo e pagare il "mantenimento annuale". Il problema in questo particolare programma è che per costruire ogni scultura si consuma ossigeno cosicché il giocatore si trova combattuto fra il desiderio di guadagnare e la necessità di usare l'ossigeno intelligentemente.

Puoi provare a scrivere un programma come questo da solo. In tal caso potresti riuscire a scrivere un programma divertente e ciò svilupperebbe notevolmente la tua capacità

di programmazione. La prima cosa da fare è un elenco di ciò che il programma deve compiere:

- Stabilire le variabili necessarie
- Comunicare al giocatore in quali condizioni si trova la cupola
- Chiedere quanto ossigeno bisogna comprare
- Controllare se si ha il denaro sufficiente. In caso positivo comprarlo, altrimenti ritornare e chiedere di nuovo
- Chiedere quanto cibo deve essere comprato
- Controllare se il denaro è sufficiente. In caso positivo comprarlo, altrimenti ritornare e chiedere di nuovo
- Rinnovare la quantità di ossigeno
- Rinnovare la quantità di cibo
- Ridurre il totale del denaro lasciato
- Domandare quante sculture devono essere costruite
- Controllare se c'è l'ossigeno sufficiente per la costruzione di quel numero di statue. In caso negativo ritornare indietro e chiedere di nuovo
- Ridurre la quantità di ossigeno per l'ammontare necessario alla costruzione del numero di sculture specificato. Incrementare il totale del denaro per considerare il prezzo delle sculture costruite
- Incrementare leggermente l'intera popolazione, aggiungere uno all'"anno attuale"
- Controllare se le riserve di cibo sono sufficienti per alimentare l'intera popolazione
- Verificare se l'ossigeno è sufficiente per l'intera popolazione
- Verificare se c'è un po' di denaro
- Se qualcuna fra queste condizioni è negativa (per esempio, il cibo non è sufficiente) trasmettere l'azione ad un sottoprogramma "fine-gioco"
- Se tutte le condizioni sono positive, ripetere di nuovo le istruzioni per comunicare al giocatore le condizioni in cui si trova la cupola, e continuare a ripetere le istruzioni.

Probabilmente saresti capace di scrivere un programma "Dome Dweller" usando le voci qui sopra elencate e l'informazione campione "printout". C'è comunque un segreto che vorrei svelarti e che risolve quasi istantaneamente i problemi legati alla programmazione. In realtà, sei in grado

di scrivere tutte le parti vitali di un programma come questo quando ne hai elaborato soltanto un rozzo schema generale, prima quindi di passare ai dettagli. E una volta che hai la struttura, puoi elaborarla per tutto il tempo che vuoi, sapendo che, mentre fai ciò, il programma che stai sviluppando è in ogni momento operante. Non devi aspettare la fine per metterlo in funzione e vedere come tu stesso ti comporti. Il "segreto" è di mantenere l'intero programma all'interno di una serie di sottoprogrammi a disposizione, ognuno dei quali mantenuto in un circuito di istruzioni continuo. Ecco come ciò che ho riferito può adattarsi a questo programma. Le primissime righe che inserirai nel tuo computer saranno le seguenti:

```

10 REM DOME DWELLER
20 GOSUB 1000: REM ASSEGNAZIONE DELLE VARIABILI
30 GOSUB 2000: REM VISUALIZZAZIONE DELLA SITUAZIONE DELLA CUPOLA
40 GOSUB 3000: REM OSSIGENO
50 GOSUB 4000: REM CIBO
60 GOSUB 5000: REM SCULTURA
70 GOSUB 6000: REM AUMENTO DELLA POPOLAZIONE
80 GOSUB 7000: REM CONTROLLO DELLA SITUAZIONE DELLA CUPOLA
90 IF (tutte le condizioni sono positive, da GOSUB 7000) THEN GOTO 30
100 REM Fine del gioco...
```

Come tu stesso puoi vedere, una volta che hai stabilito il "loop principale delle istruzioni", è relativamente facile compilare uno ad uno i sottoprogrammi, e allo stesso tempo sperimentarli ed elaborarli. In questo modo otterrai un ottimo programma. La sola cosa che adesso ti serve è un elenco delle variabili che userai insieme al programma. Sono dell'opinione che il modo migliore per far ciò sia quello di utilizzare nomi espliciti per le variabili. In questo modo, mentre stai programmando, non devi perder tempo nel verificare, per esempio, se AA indica la popolazione o il numero delle unità di ossigeno consumate per costruire ognuna delle sculture. Per facilitare il più possibile i programmi, nella trasmissione di questi fra differenti computer, puoi re-

star fedele ai nomi variabili a due lettere, o puoi usufruire (se il tuo computer lo permette) di nomi lunghi (come OXYUSE per la quantità di ossigeno usata) per le variabili. Di conseguenza non avrai dubbi sul significato di ogni nome di variabile. Per dimostrarti l'efficacia di questo accorgimento e per illustrarti un ulteriore vantaggio dei nomi espliciti per le variabili, ecco le variabili usate nel Dome Dweller:

- FOLK — la popolazione della cupola
- CASH — denaro in cassa
- FOOD — le riserve di cibo a disposizione
- FOODCOST — quanto costa ogni unità di cibo
- FOODNEED — quante unità di cibo sono consumate da ogni persona nell'arco di un anno
- ARTCOST — quanto ossigeno è consumato per costruire ogni pezzo di scultura
- ARTPAY — a quanti dollari ogni pezzo è venduto
- OXY — le riserve di ossigeno a disposizione
- OXYNEED — quante unità di ossigeno sono consumate da ogni persona nell'arco di un anno
- OXYCOST — quanto costa ogni unità di ossigeno che deve essere comprata
- REPAIR — il costo annuale delle riparazioni concernenti la cupola
- YEAR — l'anno di vita della cupola

Sebbene i nomi espliciti di variabile consumino maggiore memoria rispetto ai nomi di variabile a una o a due lettere, il loro uso permette di seguire un programma in modo molto facile, controllando ciò che compie in realtà ogni sezione del programma. Inoltre, e questo è l'ulteriore vantaggio che ho menzionato, al momento di scrivere il programma, è oltremodo facile inserire le formule richieste per operazioni di calcolo. Con questo voglio dire che se, per esempio, desideri includere (come faccio io in questo programma) un'indicazione di quanto ossigeno è necessario ogni anno, moltiplichi semplicemente il numero delle persone presenti nella cupola (FOLK) per il numero di unità di ossigeno di cui ogni persona necessita in un anno (OXYNEED). A questo punto puoi includere entro le "print" le condizioni riguardanti la cupola come, ad esempio:

```

PRINT "CI SONO"; FOLK; "NELLA CUPOLA"
PRINT "NELL'ANNO"; YEAR
PRINT "OGNI PERSONA ABBISOGNA DI"; OXY-
  NEED; "UNITÀ DI"
PRINT "OSSIGENO OGNI ANNO,"; OXYNEED * FOLK;
  "FABBISOGNO ANNUALE DI OSSIGENO"
PRINT "PER L'INTERA CUPOLA"

```

In questo modo è anche più facile verificare se è possibile comprare. Per esempio, per comprare cibo, potresti dire:

```

PRINT "QUANTO CIBO VUOI COMPRARE?"
INPUT A
IF A * FOODCOST > CASH THEN GOTO (un'altra A)

```

I consigli che seguono possono migliorare i tuoi programmi facendo uso di una "programmazione strutturata":

- redigere un campione di "printout" o un modello dimostrativo dei dati e delle informazioni illustrati sul video
- redigere una lista di ciò che il programma deve compiere ogni volta che attraversa il "loop principale di istruzioni"
- trasformare questa lista in una serie di sottoprogrammi a disposizione
- se possibile, usare espliciti nomi per le variabili

Se stai progettando dei programmi che poi saranno usati da altri, assicurati che sia estremamente chiaro ciò che il giocatore dovrebbe fare mentre conduce il programma. Includere una lunga serie di istruzioni non è di molta utilità, specialmente quando la memoria è limitata, ma è in ogni caso opportuno prenderne nota. Inoltre, i suggerimenti destinati all'utente dovrebbero essere espliciti (come INSERISCI IL NUMERO DELLE MOSSE CHE VUOI EFFETTUARE) e dovrebbero avvertire riguardo i limiti che saranno posti sull'input (CON QUANTE CARTE VUOI INCOMINCIARE: 1, 2, o 3 ?, per esempio).

Non puoi supporre che sarai sempre presente ogniqualvolta un programma è in funzione, quindi, dovresti fare del tuo meglio per renderne il funzionamento il più possibile facile e sicuro. Se è possibile, aggiungi dei sottoprogrammi che eliminino gli errori.

In tal modo se si è precedentemente inserito un sottoprogramma alternativo e c'è stato un errore, questo non rovinerà il programma e non lo renderà incapace di dare buoni risultati in seguito.

Se leggi questa parte del libro da cima a fondo più di una volta e tenterai di applicare le idee qui proposte al tuo lavoro di programmazione, la qualità del tuo lavoro migliorerà in modo significativo. Ciò ti permetterà anche di dedicare più tempo al miglioramento e al perfezionamento di un programma e minor tempo al semplice lavoro meccanico di renderlo operativo.

GLOSSARIO

A

Accumulatore — Il posto all'interno del computer entro il quale si compiono calcoli aritmetici e dove i risultati di tali calcoli sono immagazzinati.

Algoritmo — La serie di passi compiuti dal computer per risolvere un particolare problema.

Alfanumerico — Questo termine è generalmente usato in relazione ad una tastiera. Esempio: "è una tastiera alfanumerica". Ciò significa che la tastiera ha sia lettere sia numeri. Indica anche "la serie di caratteri" del computer e comprende i numeri e le lettere che il computer può riprodurre sul video.

ALU (Arithmetic/Logic Unit) — La parte del computer adibita ad operazioni aritmetiche (come l'addizione e la sottrazione) e in cui vengono prese le decisioni.

AND — Un'operazione logica "booleana" che il computer usa nei suoi processi decisionali. È basata sull'algebra di "Boole", un sistema sviluppato dal matematico George Boole (1815-64). Nell'algebra di Boole le variabili di un'espressione sono soggette alle operazioni logiche come AND, OR, NOR, ecc.

ASCII — Sta per American Standard Code for Information Exchange (Codice Americano Standard per lo Scambio di Informazioni) ed è il sistema in codice più largamente usato per la lingua inglese con caratteri alfanumerici. Ci sono 128 lettere maiuscole e minuscole, cifre e alcuni caratteri speciali. Il codice ASCII trasforma i simboli e le istruzioni di controllo in combinazioni binarie a sette "bit".

ASSEMBLER — Un programma che trasforma i programmi scritti nel linguaggio Assembly in codice macchina (che il computer può comprendere in modo diretto). Il linguaggio Assembler è un linguaggio di programmazione a basso livello che usa le combinazioni facilmente memorizzabili di due o tre lettere per rappresentare una particolare istruzione che l'Assembler poi trasforma affinché il computer possa comprenderlo. Esempio: ADD (aggiungere) e SUB (sottrarre). Un computer programmatico nel linguaggio Assem-

bler tende a lavorare più velocemente di un altro program-
mato in un linguaggio di alto livello come il BASIC.

B

BASIC — È un acronimo per Beginners All-Purpose Symbolic Instruction Code. È il linguaggio per computer più largamente usato nel settore dei micro-computer. Sebbene sia stato molto criticato, ha il vantaggio di essere molto facile da imparare. Molte espressioni BASIC sono simili all'inglese comune.

BAUD — Viene da Baudot, un pioniere della comunicazione telegrafica. Baud misura il livello di trasmissione delle informazioni ed è approssimativamente uguale a un bit per secondo.

BCD — Un'abbreviazione per Binary Coded Decimal (Binario Codificato Decimale).

Benchmark — Un test con cui possono essere misurate certe funzioni del computer. Ci sono molti "test standard Benchmark", che generalmente sperimentano soltanto la velocità. Questo è un aspetto raramente importante in un microcomputer, il tipo di computer più interessante per un potenziale compratore.

Binario — Un sistema numerico che usa soltanto i numeri zero e uno.

Bit — Un'abbreviazione per Binary Digit (cifra binaria). È la più piccola unità di informazione che un circuito di computer può riconoscere.

Boole, Algebra di — Il sistema algebrico sviluppato dal matematico George Boole che usa numerazioni algebriche per esprimere relazioni logiche (vedi AND).

Bootstrap — Un breve programma o sottoprogramma che viene letto all'interno del computer quando è acceso per la prima volta. Orienta il computer ad accettare i programmi successivi, di lunghezza maggiore.

Bug — Un errore che impedisce il funzionamento del programma. Sebbene sia generalmente usato per indicare solo un difetto o un errore nel programma, il termine bug può anche essere usato per un difetto nell'hardware del computer.

Bus — Un numero di conduttori usati per trasmettere segnali, dati e istruzioni.

Byts — Un gruppo di cifre binarie che compone una parola di computer. Generalmente i bits presenti in un byte sono otto.

C

CAI — Computer Assisted Instruction

CAL — Computer Assisted Learning. Il termine è generalmente usato per descrivere quei programmi che coinvolgono lo studente in processi di apprendimento.

CHIP — Il termine che si usa per indicare l'intero circuito, che è inciso su un piccolo pezzo di silicio. Il chip è, naturalmente, il cuore del computer.

CLOCK — Il congegno di sincronizzazione all'interno del computer che sincronizza le sue operazioni.

COBOL — Un linguaggio ad alto livello che deriva dalle parole Common Business Orientated Language. Il COBOL è usato principalmente per la schedatura e il mantenimento di ciò che è stato registrato.

Comparator — Un congegno che mette a confronto due cose e produce un segnale collegato con la differenza tra i due.

Compiler — Un programma che trasforma i linguaggi di programmazione ad alto livello in codici binari da macchina. In questo modo i programmi scritti in linguaggi ad alto livello possono essere utilizzati dal computer.

Complement — Un numero che è derivato da un altro secondo regole stabilite.

Computer — Un congegno con tre principali capacità o funzioni:

- 1) accettare i dati
- 2) risolvere i problemi
- 3) fornire risultati.

CPU — Sta per Central Processing Unit. È il cuore dell'intelligenza del computer dove si gestiscono i dati e si sviluppano le istruzioni.

Cursore — Un carattere che appare sul video quando il computer sta compiendo le sue operazioni. Esso rivela do-

ve il prossimo carattere sarà stampato. Su un computer ci sono generalmente "tasti di controllo cursore" per permettere all'utente di spostare il cursore sul video.

D

Dati — Informazioni presentate in una forma che il computer può elaborare.

Debug — Il termine che si usa quando si esamina un programma e si correggono eventuali errori, cioè, trovare e rimuovere i bugs, i difetti.

Digital Computer — Un computer che opera su quelle informazioni che si presentano come digitali.

Disk/Disc — È un disco di plastica sensibilizzato magneticamente, un poco più piccolo di un "45 giri". È usato per l'immagazzinamento dei programmi e per ottenere dati. I dischi sono notevolmente più veloci da caricare rispetto ad una cassetta della stessa lunghezza di programma. Si può accedere molto velocemente su un disco mentre un programma sta operando per ottenere ulteriori dati.

Display — L'output visivo del computer, generalmente su un video, o su uno schermo di monitor.

Dot Matrix Printer — Una stampante che stampa i listati di un programma o di ciò che appare sul video. Ogni lettera e ogni carattere sono composti da un certo numero di "dots", cioè di punti. Più alto è il numero di punti per ogni carattere e migliore sarà la qualità operativa della stampante.

Dynamic Memory — Un'unità di memoria all'interno del computer che "dimentica" ciò che contiene quando viene tolta l'alimentazione elettrica.

E

Editor — Questo termine è generalmente usato per designare quella sezione del computer che permette al programmatore di cambiare le istruzioni di un programma mentre lo sta scrivendo.

EPROM — Sta per Erasable Programmable Read-Only Memory. È come il ROM nel computer, con la sola differenza che è abbastanza facile inserire materiale all'interno di una EPROM che non sparisce quando si toglie l'alimentazione. Le EPROM devono essere esposte a forti raggi ultravioletti se si vuole cancellarle.

Error Messages (Messaggi di errore) — L'informazione data da un computer che indica dove è stato commesso un errore nella codificazione durante una parte del programma. L'informazione è trasmessa dal computer che si ferma e stampa una parola, o una parola e dei numeri, o soltanto una combinazione di numeri, in fondo al video. Questo rivela quale errore è stato fatto. Gli errori più comuni includono l'uso della lettera O invece dello zero in una linea, o l'omissione in una espressione di entrambe o di una delle parentesi, o l'errore nella definizione di una variabile.

F

File — Una serie di item d'informazione collegati fra loro e organizzati in modo sistematico.

Floppy Disk — Un disco magnetico relativamente poco costoso, usato per immagazzinare le informazioni del computer, e così chiamato perché molto flessibile (vedi Disk/Disc).

Flow Chart — Un diagramma tracciato prima di scrivere un programma nel quale le principali operazioni sono racchiuse entro rettangoli o altre forme e connesse a frecce attraverso linee per rappresentare "loop" di istruzioni, e le decisioni scritte fra parentesi. Ciò ti aiuta a scrivere un programma in modo molto più semplice perché trappole come loop infiniti o variabili non definite possono essere scoperte in ogni fase. Può risultare utile scrivere questo diagramma per programmi molto brevi, ma è senz'altro conveniente se si vuole creare un programma più lungo.

Firmware — Ci sono tre tipi di "ware" nei computer: cioè programmi software "temporanei"; hardware, come quelli contenuti nelle ROM permanenti; e firmware, nel quale l'informazione è relativamente permanente, come in una EPROM (vedi EPROM).

Flip-Flop — Un circuito che mantiene in memoria una condizione elettrica finché questa non è cambiata nella condizione opposta da un segnale.

FORTRAN — Sta per FORMula TRANslation (traduzione di formula). È un linguaggio di computer ad alto livello, orientato verso problemi matematico-scientifici.

G

Gate — Un circuito elettrico che, sebbene possa captare uno o più segnali in arrivo, manda in uscita soltanto un singolo segnale.

Graphics — Informazione grafica, in opposizione alle informazioni fornite da lettere e numeri.

H

Hard Copy — Uscita di computer il cui supporto è permanente.

Hardware — Le parti fisiche di un computer (vedi anche software e firmware).

Hexadecimal (Hex) — Un sistema numerico con base sedici. Sono usate le cifre da zero a nove e le lettere A, B, C, D, E, F per la rappresentazione dei numeri. A è uguale a 10, B è uguale a 11, C è uguale a 12, e così via. Hex è spesso usato dagli utenti di microcomputer.

Hex Pad — Una tastiera specificamente progettata per inserire numerazioni esadecimali.

High Level Language (Linguaggio ad alto livello) — Un linguaggio di programmazione che permette all'utente di parlare con il computer più o meno in lingua inglese. In generale più è alto il livello del linguaggio (cioè, più è simile all'inglese) più lungo sarà il tempo impiegato dal computer per tradurlo in un linguaggio che esso può utilizzare. Linguaggi a più basso livello sono molto più difficili per l'operatore umano ma generalmente offrono una esecuzione più veloce.

I

Input — L'informazione inserita nel computer attraverso una tastiera, un microfono, una cassetta o un disco.

Input/Output (I/O Device) — Un congegno che accetta le informazioni o le istruzioni dal mondo esterno, le trasmette al computer e, dopo l'elaborazione, le rinvia, o sotto una forma adattabile alla memorizzazione, o una forma comprensibile all'essere umano.

Instruction — Il dato che comanda una sola azione nell'elaborazione delle informazioni operate dal computer (noto anche come comando).

Integrated Circuit (Circuito Integrato) — Un completo circuito elettronico residente sulla superficie di un semiconduttore.

Interface — Il confine fra il computer e un periferico come la stampante.

Interpreter — Un programma che traduce, istruzione per istruzione, il linguaggio ad alto livello inserito da un operatore umano in un linguaggio che la macchina può capire.

Inverter (Invertitore) — Un "gate" logico che cambia nell'opposto il segnale inserito.

Interactive Routine (Sottoprogramma Interattivo)— Parte di un programma che è ripetuto più volte finché non si raggiunge una data condizione.

J

Jump Instruction — Un'istruzione che dice al computer di muoversi verso un'altra parte del programma, quando la destinazione di questo spostamento dipende dal risultato di un calcolo appena compiuto.

K

K — Questa lettera riporta la misura della memoria. La memoria è generalmente misurata in blocchi di "K". Un K contiene 1.024 bytes.

Keyword (parola-chiave) — La parola di inizio in una linea di programmazione, generalmente la prima parola dopo il numero della istruzione. Parole-chiavi sono STOP, PRINT e GOTO.

L

Language — I linguaggi di computer sono divisi in tre gruppi: i linguaggi ad alto livello, come il BASIC, che sono relativamente vicini all'inglese ed abbastanza facili da usare per l'uomo; i linguaggi a basso livello, come l'ASSEMBLER, in cui compaiono brevi frasi che hanno qualche collegamento con l'inglese (ADD per add e RET per return, ad esempio); e il codice macchina, che comunica più o meno direttamente con la macchina.

LCD — Sta per Liquid Crystal Diode. Alcuni computer come il TRS-80 Pocket Computer usano un "display" LCD.

LED — Sta per Light Emitting Diode. I numeri luminosi rossi che sono spesso usati negli orologi da polso o da muro sono composti da LED.

Logic — La formula matematica di uno studio di relazioni fra eventi.

Loop — Una serie di istruzioni all'interno di un programma che sono ripetute finché una particolare condizione viene soddisfatta.

M

Machine Language o Machine Code (Codice macchina) — Un codice che può essere capito e messo in pratica direttamente dal computer.

Magnetic Disk — vedi Disk e Floppy Disk.

Mainframe — I computer sono generalmente divisi in tre gruppi e il fatto di appartenere ad un certo gruppo dipende dalla grandezza del computer. Il computer più venduto è il microcomputer. I computer di media grandezza sono i minicomputer, e i computer giganti che qualche volta si vedono nei film di fantascienza sono computer "mainframe". Fino a 15 anni fa tali computer erano, in termini pratici, i so-

li disponibili.

Memory — Dentro un computer ci sono due tipi di memoria. La prima chiamata ROM, è la memoria che arriva già programmata sul computer e che dice al computer come prendere decisioni e come compiere operazioni aritmetiche. Questa memoria non è cancellata se si spegne il computer. Il secondo tipo è la RAM. Questa memoria mantiene il programma che è stato scritto sulla tastiera o che è trasmesso all'interno tramite una cassetta o un disco. La maggior parte dei computer "dimenticano" ciò che è in RAM quando vengono spenti.

Microprocessor — Il cuore di qualsiasi computer. Richiede le interfacce con le unità periferiche, l'alimentazione di energia e i congegni di input e output. In tal modo può operare come un microcomputer.

MODEM — Sta per Modulatore/Demodulatore. È un apparato che permette a due computer di parlare fra loro per telefono. Generalmente i computer usano un supporto nel quale è posto un ricevitore telefonico.

Monitor — Nel linguaggio dei computer ha due significati. Un significato è uno schermo simile a quello televisivo. Un monitor ha serie difficoltà ad adattarsi ai programmi televisivi e generalmente l'immagine prodotta su un monitor è migliore di quella prodotta da una comune Tv. Il secondo significato di un monitor si rapporta alla ROM. Il monitor di un computer è descritto come il complesso di informazioni incorporate nel computer all'atto dell'acquisto. Queste informazioni permettono di prendere decisioni e di compiere calcoli aritmetici.

Motherboard — Una struttura alla quale possono essere aggiunti circuiti extra. Questi circuiti spesso offrono facilitazioni che non sono incorporate nel computer, come quella di produrre suoni o di controllare una penna ottica.

MPU — Abbreviazione per Microprocessor Unit.

N

Nano-secondo — Un nano-secondo è un millimiliardesimo di secondo, l'unità di tempo con la quale si misura la velocità di un computer o di un microcircuito di memoria.

Non-Volatile Memory — La memoria che non si perde quando il computer è spento. Alcuni computer più piccoli

come il TRS-80 Pocket Computer hanno "non volatile memory". Le batterie mantengono il programma inserito per settecento ore.

Not - Un'operazione booleana che trasforma una cifra binaria nel suo opposto.

Null String — Una "stringa" che non contiene caratteri. Nel programma compare sotto forma di due doppie virgolette, senza niente fra di loro.

Numerico — Concerne i numeri quando sono opposti alle lettere (cioè alfabetico). Molte tastiere sono alfanumeriche, sono cioè provviste sia di numeri sia di lettere.

O

Octal — Un sistema numerico che usa otto come base e quindi le cifre da 0 a 7. Tale sistema non è oggi molto usato nel settore dei microcomputer. Il sistema esadecimale è più comune (vedi Hexadecimale)

Operating System — (Sistema operativo) — Il software e il firmware, generalmente previsti su una macchina che permette di far girare altri programmi.

OR — Un'operazione booleana che ritorna a 1 se uno o più input sono 1.

Oracle (Oracolo) — Un metodo di messaggi inviati test tramite un segnale di trasmissione televisiva. Un set di tele-test è richiesto per decodificare i messaggi.

Output — Informazioni o dati trasmessi dal computer a congegni quali uno schermo come quello televisivo, una stampante o una cassetta. L'output generalmente consiste in un'informazione che il computer ha prodotto come risultato della elaborazione di un programma.

Overflow — Un numero troppo grande o troppo piccolo per essere elaborato dal computer.

P

Pad — Vedi Keypad

Pagina — Spesso usata per indicare la quantità di informazioni necessaria per riempire uno schermo televisivo. Così, vedendo una pagina del programma, è possibile analizzare la quantità di informazioni che appaiono sul video tutte in una volta.

PASCAL — Un linguaggio ad alto livello.

Periferico — Qualsiasi cosa che è collegata e controllata dal computer, come un'unità a disco, una stampante o un sintetizzatore vocale.

Port — Un connettore attraverso il quale le informazioni sono trasmesse o inserite nel computer.

Prestel — Il nome inglese per un sistema basato sulla trasmissione di informazioni via telefono da un computer centrale e sulla loro visualizzazione su uno schermo televisivo. Negli Stati Uniti una versione commerciale simile è nota come "The Source".

Program — Nel linguaggio computer può essere una lista di istruzioni che si inseriscono nel computer, oppure può essere un verbo, cioè "programmare un computer".

PROM — Sta per Programmable Read Only Memory. È un sistema che può essere programmato e, una volta che lo è stato, il programma è permanente (vedi anche EPROM e ROM).

R

Random Access Memory (RAM) — La zona di memoria entro il computer che può essere cambiata a comando dalla persona che usa il computer. Il contenuto della RAM è di solito perduto quando un computer è spento. La RAM è l'integrato che memorizza ciò che viene scritto e anche i risultati di calcoli in atto.

Read-Only Memory (ROM) — In contrasto alla RAM, l'informazione qui non può essere cambiata dall'utente e non va perduta quando si spegne il computer. I dati della ROM sono collocati dal produttore e dicono al computer il modo con cui deve prendere decisioni e come compiere calcoli aritmetici. La misura di capacità della ROM e RAM è data in unità K (vedi K).

Recursion — La ripetizione continua di una parte del programma.

Registro — Una specifica sezione della memoria nella quale uno o più parole di computer sono memorizzate nel corso delle operazioni.

Parola Riservata — Una parola che non può essere usata per una variabile in un programma perché il computer la leggerà in modo errato. Un esempio è la parola TO. Poiché TO ha uno specifico significato nel linguaggio dei computer, la maggior parte dei calcolatori respinge questa parola come nome per una variabile. Lo stesso vale per parole come FOR, GOTO e STOP.

Routine — Questa parola può essere usata come sinonimo di programma o può riferirsi a una specifica sezione all'interno del programma (vedi anche Subroutine).

S

Seconda Generazione — Ha due significati. Il primo si applica nei confronti dei computer che usano transistor, in opposizione alla prima generazione di computer che usavano valvole. La seconda generazione può anche indicare la seconda copia di un particolare programma. Susseguenti generazioni sono danneggiate da un disturbo crescente.

Semiconduttore — Un materiale che è generalmente un isolante elettrico, ma sotto specifiche condizioni può diventare un conduttore.

Serial — Informazione che è memorizzata o inviata in una sequenza, un bit alla volta.

Segnale — Un impulso elettrico che trasmette dati.

Silicon Valley — Il nome popolare dato a una zona in California dove si trovano molti produttori di semiconduttori.

SNOBOL — Un linguaggio ad alto livello.

Software — Il programma inserito nel computer dall'utente. Questo programma dice al computer cosa fare.

Software Compatible — Si riferisce a due differenti computer che possono accettare i programmi scritti per l'altro.

Static Memory — Un congegno di memoria non volatile che trattiene le informazioni per tutto il tempo che il computer è acceso. Tuttavia, non richiede ulteriori consumi di energia per mantenere in ordine la memoria.

Subroutine (Sottoprogramma) — Parte di un programma che è spesso inserita molte volte durante l'esecuzione del programma principale. Una subroutine finisce con un'istru-

zione che comanda di ritornare indietro alla istruzione successiva a quella che aveva inviato lo svolgimento del programma alla subroutine.

T

Teletext — Informazione trasmessa nella sezione superiore di un'immagine che appartiene a una trasmissione televisiva. Richiede una struttura speciale per essere decodificata e per riempire il video di informazioni riguardo il test. I messaggi teletext possono anche essere trasmessi tramite un cavo, per esempio, il servizio Prestel in Gran Bretagna o The Sources negli Stati Uniti.

Teletype — Un apparecchio simile ad una macchina da scrivere che può mandare, ricevere e stampare informazioni.

Terminale — Un'unità indipendente dell'unità centrale di elaborazione. Generalmente consiste di una tastiera e di un visore.

Time Sharing — Un processo attraverso il quale molti utenti possono aver accesso a un grande computer che si sposta rapidamente da un utente all'altro in sequenza, cosicché ogni utente ha l'impressione di essere il solo utente del computer.

Truth Table (Tavola della verità) — Una tavola matematica che elenca tutti i possibili risultati di un'operazione booleana. I risultati dipendono dalle varie combinazioni di input.

U

UHF — Ultra High Frequency (300-3000 MegaHertz).

Ultra Violet Erasing — La luce ultravioletta deve essere usata per cancellare le EPROM (vedi EPROM).

V

Variabile — Una lettera o combinazione di lettere e simbo-

li a cui il computer può assegnare un valore o una parola durante il funzionamento di un programma.

VDU — Abbreviazione per Visual Display Unit.

Volatile — Indica la memoria che "dimentica" le informazioni in essa contenute quando il computer è spento.

W

Word (Parola) — Un gruppo di caratteri o una serie di cifre binarie che rappresentano un'unità d'informazione e occupano una singola posizione di memoria. Il computer elabora una parola come singola istruzione.

Word-Processor (Elaboratore di testi) — Una macchina da scrivere altamente intelligente che permette a chi scrive di manipolare il testo, di spostarlo, per giustificare margini e per spostare interi paragrafi, se necessario, su un video prima di mandare l'informazione sulla stampante. Questi elaboratori hanno generalmente memorie, cosicché modelli di lettere e testi di lettere, scritti precedentemente, possono essere nuovamente stampati.

Traduzioni

EGG SAVIOUR

- 10** — Salvatori di uova
- 200** — Vite a disposizione — Sei stato ucciso
- 230** — Uova raccolte
- 1510** — Premi enter per giocare ancora

Vite a disposizione	Uova raccolte
Punteggio	Alto
Sei stato ucciso	

BOX BOUNCE

Punti... Vite a disposizione

- 65** — Vite a disposizione
- 180** — Punti
- 515** — Vite a disposizione
- 600** — Fine gioco
- 610** — Il tuo punteggio è
- 630** — Il miglior punteggio è
- 640** — Premere enter per giocare ancora

NESSIE

- 101** — Punti — Missili
- 500** — Fine gioco
- 501** — Punti — Missili
- 510** — Il tuo punteggio è
- 520** — Il miglior punteggio è
- 530** — Premi enter per giocare ancora
- 2000** — Bonus 1000 punti
- 3000** — Bonus 2000 punti

SPIDER'N'FLY

- 140** — Tempo a disposizione
1030 — Punti
2000 — Fine gioco
2010 — Il tuo punteggio è
2030 — Il miglior punteggio è
2040 — Premi enter per giocare ancora

ASCOT

1^a corsa

Le probabilità su Red Gin per vincere sono 1:1

Le probabilità su Sumley Gilds sono 1:1

su Sparkle 5:1

su Danny Boy 8:1

su Shergart 3:1

Cavallo: Red Gin

Proprietario: Peter

Capitale: 50 sterline

Il vincitore è Danny Boy di proprietà di Michael che vince
 200 sterline

Cavallo: Shergart

Proprietario: Bill

Capitale: zero sterline

Non hai soldi, non puoi scommettere

- 60** — Numero corsa
90 — Le probabilità su
100 — di vincere sono
130 — Premi un tasto per continuare
270 — Il vincitore è
280 — montato da
340 — Alla fine del gioco
390 — Quando il vincitore è
400 — su

- 410** — con sterline
- 420** — Premi enter per un'altra corsa
- 5030** — Cavallo
- 5040** — Fantino
- 5050** — Hai sterline
- 5055** — Non hai soldi, non puoi scommettere
- 5060** — Quanto scommetti alla prossima corsa
- 8070** — Quanti giocatori (max - 5)
- 8100** — Componi il nome del giocatore
- 8140** — Ora corre
- 8160** — Premi un tasto per continuare

MUNCHIE MAN

- 40** — Vite a disposizione
- 145** — Punti
- 2010** — Fine gioco
- 2020** — Il tuo punteggio è
- 2040** — Il miglior punteggio è
- 2050** — Premi enter per giocare ancora

TRACK TRAP

- 1020** — Punti computer — Umano
- 2020** — Punti computer — Umano
- 3000** — Computer — Umano — Vince!
- 3010** — Premi enter per giocare ancora
- 8040** — Punti
- 8050** — Computer 00 Umano 00

PONTOON

Ho vinto questo gioco!

Io ho 21

Tu hai 19

1020 — Io ho — tu hai

- 1030** — lo ho perso, tu hai vinto
- 1050** — Hai vinto
- 1060** — Hai vinto 50 sterline
- 2000** — Ho vinto questa partita
- 3010** — Non hai più denaro
- 3020** — Premi enter per giocare ancora
- 7050** — Denaro

BREAKOUT

- 55** — Punti
- 1020** — Palle
- 2000** — Fine gioco
- 2010** — Premi enter per giocare ancora
- 7000** — Punti — Palle
- 8060** — Che velocità

EXCALIBUR AGAINST THE ENEMY

- 80** — Muove il computer
- 180** — Excalibur ha vinto!!
- 210** — Il giocatore accetta la mossa
- 230** — Il giocatore immette 99 per... se non ci sono mosse
- 250** — Input in alto nello schermo
- 260** — Excalibur si muove verso
- 300** — Mossa errata
- 310** — Non si possono fare mosse diagonali
- 390** — Numero mosse
- 420** — Mosse a disposizione
- 530** — Totale nemico — Navi al via
- 740** — Il tempo è scaduto
- 750** — Totale nemico
- 770** — Un'altra partita (sì/no)
- 800** — Son contento quando riconosci di aver perso

LEAKY ROOF

- 50** — Gocciolamento
- 90** — Punti
- 150** — Fine gioco
- 160** — Il tuo punteggio è
- 180** — Il miglior punteggio è
- 190** — Premi enter per giocare ancora

TOUCHDOWN ON MARS

- 100** — Punti
- 120** — Tempo
- 1000** — Fine gioco — Secondi
- 1040** — Il tuo punteggio è
- 1050** — Il miglior punteggio è
- 1060** — Premi enter per giocare ancora

ORCHARD THIEF

- 100** — Punti — Tempo
- 1010** — Fine gioco
- 1020** — Sei andato fuori tempo
- 1030** — Il tuo punteggio è
- 1050** — Il miglior punteggio è
- 2010** — Fine gioco
- 2020** — Hai finito in — unità di tempo
- 7040** — Punti — Tempo

CHARGE OF THE TEDDY

- 115** — Orsacchiotti a disposizione
- 4000** — Fine gioco
- 4010** — Il tuo punteggio è
- 4030** — Il miglior punteggio è
- 4040** — Premi enter per giocare ancora
- 7090** — Punteggio — Orsacchiotto a disposizione

Snake in the Triangles

- 1010** — Fine gioco
1020 — Il tuo punteggio è
1040 — Il miglior punteggio è
1050 — Premi enter per giocare ancora

Death Race

- 125** — Tempo
1030 — Punti
2000 — Tempo
2010 — Fine gioco
2020 — Il tuo punteggio è
2040 — Il miglior punteggio è
2050 — Premi enter per giocare ancora

Draughts

- 60** — Scacchiera
100 — Il tuo punteggio è — Il punteggio della macchina è
130 — Ho vinto
140 — Hai vinto tu
170 — Ultimo a
180 — Da — a
1480 — Vuoi essere il primo a muovere? (sì/no)

Simon Said

- 270** — Hai sconfitto la macchina — Premi enter per giocare ancora
1000 — Il tuo punteggio è
1020 — Premi enter per giocare ancora

3D DRIVER

- 1040** — Il tuo punteggio è
1050 — Premi enter per giocare ancora

DROP OUT

- 50** — Forza
310 — Punti

M4

- 1030** — Premi enter per giocare ancora
7010 — Record — Punti

HANGMAN

- 80** — Indovina una sola lettera
2010 — L'ho ucciso
2011 — La parola era
2020 — Premi enter per giocare ancora
5008 — Indovinato!
7030 — Aeronave - aeroplano - scimmia - incudine - applicare - artico - esercito - artropod - carrettino - binario - betulla - bizzarro - morso - bolla - sangue - coda - punteruolo - cervello - macchina fotografica - canoa - berretto - attenzione - nacchere - presa - cella - cappella - carro - classico - carrozza
7040 — Doppione - anatroccolo - duna - doppio - nano - argine - uso - incontro - motore - rallegrare - busta - equinozio - dissolvere - sentire - recinto - sazietà - finito - pesce - bandiera

- 7050** — Misura - congegno - geometria - limpido - oro -
 prontuario di prosodia classica - rancore - cuore -
 testa - elicottero - araldica - ospedale - aeronave -
 ghiaccio - imperiale - afflusso - cavallo recalcitrante
 (opp. fiocco di neve) - jersey - falegname - giovane -
 orto - canile - rintocco funebre - labaro - landò -
 tornio - capo - Livorno (opp. cappello di paglia di
 Firenze) - luce - labbro
- 7060** — Massa - combattimenti - labirinto - mezzo - maturo
 - mescolare - mulino - luna - motore - monte - con-
 fondere - nervi - nicchia - nostalgia - oggetto - una
 volta - attico - arancio - globo - ordinario - orfano -
 ottuso - straniero - ozono
- 7070** — Drappo - frittella - pantografo - parcella - parallas-
 se - parlamento - particella - Fetonte - piccone - go-
 gna - passera di mare - piatto - piega - poesia - ta-
 riffa postale - libbra - progetto - propulsore - quadri-
 glija - piumino - coda - quota - coniglio - lotteria -
 reciproco
- 7080** — Rinvio in carcere - rimanere - residenza - reddito -
 lasca - selvaggio - saggio - pendio - spavento - corsa -
 mare - secondo - semaforo - sentenza - shampoo
- 7090** — Fiocco - scadente - tarra - oscillare - narratore -
 tennis - tensione - tetano - grosso - battito - tigre -
 titanico - crostino - tonico - torrido - tribù - arbitro -
 goffo - attenuare - sregolato - privare del posto a
 sedere - impulso
- 7100** — Vuoto - lettera d'amore - vagabondo - valvola - va-
 riabile - volta - vena - visita - paga - portafogli - av-
 vertire - ammonimento - acque - cera - arma - pozzi
 - balena - turbine - whippet (levriero da corsa) - fi-
 schio - xilofono - anno - gioventù - rosso d'uovo -
 buffone - zebra - rombo - zoologo

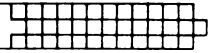
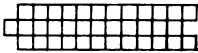
VIDEO SALESMAN

- 40** — Quanti giocatori — Premi enter per iniziare
- 75** — Venditori — Non hai abbastanza denaro per com-
 prare altra merce
- 90** — Venditore — Venditore

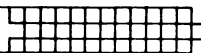
- 95** —Contanti a disposizione
- 110** —Notizie locali e nazionali
- 130** —Di quanti registratori ti vuoi rifornire questa settimana? Premi enter per continuare
- 180** —Denaro a disposizione
- 200** —Vendite di questa settimana
- 220** —Guadagno
- 240** —Nuovo bilancio
- 250** —Premi enter per continuare
- 2050** —È la settimana della coppa del mondo - E.T. è stato proposto ufficialmente sul video - Il video registra la diminuzione dei prezzi - Speciale offerta sui registratori "wxz" - Si sta avvicinando Natale - Un nuovo negozio video apre in città.
- 2530** —L'inflazione batte l'industria del video - I video dischi diventano popolari - La televisione stronca le vendite - Il cinema ha aumentato la popolarità - Calo dei prezzi nell'industria video.

DAM BUSTERS

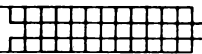
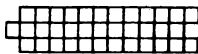
- 1500** — Fine gioco
- 1510** — Il tuo punteggio è
- 1530** — Il miglior punteggio è
- 1540** — Premi enter per giocare ancora



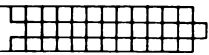
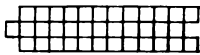
Empty space for annotations, bounded by vertical lines on the left and right.



Large empty rectangular area for annotations, bounded by vertical lines on the left and right sides.



A large empty rectangular area bounded by vertical lines on the left and right, intended for annotations.



A large empty rectangular area bounded by a thin black line, intended for handwritten notes.

Computer Games

GIOCHIAMO CON ZX SPECTRUM

Tanti fantastici programmi, con la traduzione in italiano, appositamente ideati per questa collana e in grado di garantirvi ore e ore di svago istruttivo e divertente.

Tra i giochi spettacolari di questo libro: EXCALIBUR CONTRO IL NEMICO (come difendersi contro le astronavi degli Alieni); ASCOT (fatevi una fortuna alle corse dei cavalli); ATTERRAGGIO SU MARTE (manovrate la vostra astronave in direzione di Marte); LA GUIDA A TRE DIMENSIONI (ci vuole abilità per guidare un'automobile); IL GIOCO DELL'IMPICCATO (indovinate le parole lettera per lettera prima di essere "impiccati"); LA CORSA DELLA MORTE ("stendere" quanti più pedoni possibile).

"GIOCHIAMO CON ZX SPECTRUM" vi aiuterà moltissimo, giocando, a migliorare la vostra abilità fornendovi non solo tutte le istruzioni per inserire correttamente i programmi nel vostro computer, ma anche un utilissimo glossario dei termini essenziali e preziosi consigli e indicazioni su come modificare e migliorare i programmi del libro o realizzarne di nuovi.

Tanti fantastici giochi per tutta la famiglia!

CL 006-0118-7 ISBN 88-7605-118-X

L. 9.500 (i.i.)

Computer Games Are Fun