

77 PROGRAMMI PER SPECTRUM

GRAFICA - BUSINESS GRAFICA - UTILITY - ANIMAZIONI - MUSICA - GIOCHI



GRUPPO
EDITORIALE
JACKSON

di Gaetano Marano

77 PROGRAMMI PER **SPECTRUM**

di
Gaetano Marano



GRUPPO
EDITORIALE
JACKSON
Via Rosellini, 12
20124 Milano

© Copyright 1983 Gruppo Editoriale Jackson

Il Gruppo Editoriale Jackson ringrazia per il prezioso lavoro svolto nella stesura del volume le signore Francesca Di Fiore, Marta Menegardo e l'ing. Roberto Pancaldi.

Tutti i diritti sono riservati. Nessuna parte di questo libro può essere riprodotta, posta in sistemi di archiviazione, trasmessa in qualsiasi forma o mezzo, elettronico, meccanico, fotocopiatura ecc., senza l'autorizzazione scritta.

I contenuti di questo libro sono scrupolosamente controllati. Tuttavia, non si assume alcuna responsabilità per eventuali errori od omissioni. Le caratteristiche tecniche dei prodotti descritti, possono essere cambiate in ogni momento senza alcun preavviso. Non si assume alcuna responsabilità per eventuali danni risultanti dall'utilizzo di informazioni contenute nel testo.

Prima edizione: 1983

Stampato in Italia da:
S.p.A. Alberto Matarelli - Milano - Stabilimento Grafico

Fotocomposizione:
CorpoNove s.n.c. - Bergamo - tel. 035/22.33.63 - 22.33.65

PREFAZIONE

Nel realizzare i programmi di questo libro si è cercato di sfruttare al massimo le nuove interessanti prestazioni dello Spectrum ed in particolare le nuove istruzioni BASIC, il Suono, il Colore e soprattutto la Grafica ad alta risoluzione.

Anche la scelta dei programmi è molto varia, essendoci infatti programmi di Grafica (30), Utility (8), Musica (5), Animazione (6), Giochi (7), Effetti Sonori (12), Elettronica e Circuiti Logici (4) ed altro (5).

Fiore all'occhiello di questo libro è senz'altro la 'BUSINESS GRAFICA' (9 programmi) realizzata, con una certa efficacia visiva e pratica superando alcune limitazioni di colore e di risoluzione che relegavano tali programmi solo a computer molto più costosi.

In tutti i programmi si è sfruttata anche la maggiore velocità dello Spectrum rispetto ai precedenti computer Sinclair ed in particolare nelle Animazioni che sono ora molto più veloci e realistiche. Altra caratteristica molto utilizzata è la possibilità sullo Spectrum di definire dei Caratteri Speciali, inseriti ovunque possibile per rendere i programmi molto curati anche esteticamente o per rendere più realistiche alcune situazioni di programma, come nei giochi dove si creano, per esempio, un sommergibile (nella Caccia al sommergibile), o degli omini (nel Tiro alla Fune o nel Campo Minato), ecc... Spero quindi che troverete molte cose utili da utilizzare sul vostro meraviglioso Spectrum.

Prima di terminare permettetemi di ringraziare, per l'apporto di collaborazione da loro ricevuto, i Sigg.ri Arenai e Harms della Easy Byte di Roma e la Sinclair Research (per il tramite dell'importatore, la GBC Italiana) per i due Spectrum utilizzati nello scrivere il libro, ed i Sigg.ri Dentale e Ferrari della TELAV di Roma per il Monitor a Colori (un Barco CM33) utilizzato per ottenere delle immagini ad alta qualità di cui le foto che ho realizzato spero diano buona testimonianza.

Gaetano Marano

INDICE

PREFAZIONE	III
NOTE IMPORTANTI SUI PROGRAMMI	VIII
GRAFICA	1
1 Disegnatore Caratteri Speciali	2
2 Disegnatore a Colori	4
3 Plotter	7
4 Curve Matematiche	9
5 Disegno Poligoni	11
6 Plotaggio Ellissi	14
7 Disegno Quadrati	16
8 Spirali	17
9 Cono Tridimensionale	18
10 Cilindro Tridimensionale	19
11 Sfera Terrestre	20
Programmi di 'Business Grafica'	22
12 Diagramma a Torta	22
13 Cilindri 3 D	25
14 Istogrammi Orizzontali	27
15 Istogrammi Verticali	29
16 Istogrammi Verticali a Colori Affiancati	32
17 Istogrammi Verticali a Colori Sovrapposti	35
18 Istogrammi Verticali Tridimensionali	38
19 Diagramma	42
20 Diagramma a Superficie	45
21 NEW YORK	49
22 Big Bang	51
23 Disegno di Linee	51
24 Alveare	52
25 Piramide	52
26 Cielo Stellato	53
27 Montagne	54
28 Bandiera Regno Unito	55
29 Bandiera Stati Uniti	57
30 Bandiera Giapponese	58

PROGRAMMI VARI	59
31 Fisica dei Reattori Nucleari	59
32 Lettura Veloce	62
33 Cronometro	65
34 Scritte Rotanti	67
35 Contapezzi	70
36 Simulatore di Porte Logiche	72
37 Progettazione di Multivibratore Monostabile con 555	76
38 Progettazione di Oscillatore con 555	81
39 Legge di OHM	86
UTILITY	89
40 Routine 'PAINT' in BASIC	89
Note sulle routines 'SCROLL'	91
41 Scroll a Sinistra	92
42 Scroll a Destra	93
43 Scroll in Alto	94
44 Scroll in Basso	95
45 Renumber	96
46 Lettore Memoria RAM/ROM	98
47 Conversione Decimale/Esadecimale ed inverso	100
ANIMAZIONI	103
48 Viaggio Spaziale	103
49 Lancio SPACE SHUTTLE	105
50 Animazione Esplosione	107
51 Caleidoscopio	108
52 Rombospirale	110
53 Programma Universale Animazione con PRINT AT	112
MUSICA ED EFFETTI SONORI	115
54 Miniorgano	115
55 Generatore di Musica Casuale – 130 Note	116
56 Metronomo	117
57 Scale Musicali	117
58 Programma Universale per Musica	118
59 Red Alert	120
60 Orologio	121
61 Mitra	121
62 Sirena Americana	121
63 Sirena a 2 Toni	122
64 Sirena Multitono	122
65 Squillo del telefono	122
66 Segnale telefonico di linea libera	123

67	Segnale telefonico di linea occupata	123
68	Generatore di suoni casuali	123
69	Din – Don	123
70	Grilli	124
GIOCHI		125
71	Caccia al Sommergibile	125
72	Tiro alla Fune	127
73	Roulette	130
74	Campo Minato	132
75	Flipper Giapponese	136
76	Tavola di Numeri e Lettere	138
77	1 – 40 DADI	140

NOTE IMPORTANTI SUI PROGRAMMI

Dopo aver inserito i programmi, e PRIMA di premere RUN, è sempre bene registrare su cassetta il programma e controllare la avvenuta registrazione con 'VERIFY', ma ciò è particolarmente IMPORTANTE in tutti i programmi che contengono delle routines in linguaggio macchina ed in genere delle POKE. Queste perché se si aggiunge o si toglie per errore una istruzione della routine o se si sbaglia l'indirizzo della POKE, può capitare che il computer si blocchi o addirittura che il programma si cancelli.

Registrando il programma si può evitare quindi di doverlo riscrivere e sarà più facile trovare l'istruzione che si è inserita in modo errato.

Tutti i programmi sono perfettamente funzionanti in quanto sono stati stampati direttamente dal computer, dopo aver girato con successo, e riprodotti esattamente in questo libro per cui occorre soltanto inserirli con esattezza per vederli funzionare. A tale proposito è bene tenere presente di inserire con particolare cura gli spazi sia come numero che come posizione esattamente come nei programmi, poiché in molti casi sono determinanti per un funzionamento corretto. Per contare gli spazi brevi basta riferirsi alle linee superiori o inferiori, mentre per gli spazi più lunghi, questi sono stati indicati in alcuni programmi, con dei REM subito dopo la fine.

In alcuni programmi, che usano routines in linguaggio macchina per leggere la tastiera, è indicato di non inserire nel programma l'istruzione BORDER altrimenti il programma si blocca. Se comunque si desidera un dato colore per il bordo lo si può inserire con un comando diretto prima di scrivere il programma.

Infine i programmi sono stati scritti in modo da funzionare senza modifiche sia su Spectrum da 16K che da 48K.

GRAFICA

DISEGNATORE CARATTERI SPECIALI

Lo Spectrum ha la possibilità di definire 21 caratteri speciali memorizzabili tramite la funzione `USR` o tramite il comando `POKE`, e utilizzabili in un programma tramite l'istruzione `CHR$` o inseribili direttamente da tastiera premendo i tasti da A a U con il Computer in modo grafico.

Il programma di figura 1 è molto utile perchè permette di disegnare e modificare tali caratteri speciali in grandi dimensioni e quindi di memorizzarli sotto il carattere alfabetico prescelto.

Il programma permette inoltre (premendo il tasto '1' alla fine o anche durante il disegno) di visualizzare i codici delle `POKE` da effettuare nel caso si vogliano caricare molti caratteri speciali nel modo utilizzato in molti programmi di questo libro; vale a dire inserendo tutti i codici in istruzioni `DATA` e caricandoli con delle `POKE` direttamente nella RAM a partire dall'indirizzo 32600 (nello Spectrum 16K) o dall'indirizzo 65367 (nel 48K).

Le funzioni degli otto tasti utilizzati dal programma sono le seguenti:

tasto	Funzione
1	visualizza i codici POKEs
2	carica il carattere speciale in memoria
3	modo 'disegno'
4	modo 'modifiche e spostamenti'
5	a sinistra
6	in basso
7	in alto
8	a destra

La figura 2 mostra alcuni dei caratteri speciali realizzabili con il programma appena descritto, assieme ai relativi codici delle `POKE`. In particolare vi è un piccolo E.T. (2a), un PAC-MAN (2b), un'auto da corsa (2c), un omino (2d) ed il simbolo di microFarad (2e).

NOTA: Premendo il tasto '1' o il tasto '2' il cursore quadrato lampeggiante viene incluso nel disegno o nel calcolo dei codici `POKE` solo se si è in modo 'disegno'.


```

100000 IF C=11 THEN RETURN
100000 LET C=C-1: RETURN
100000 IF C=1 THEN RETURN
100000 LET C=C+1: RETURN
100000 IF C=4 THEN RETURN
100000 LET C=C-1: RETURN
100000 IF C=10 THEN RETURN
100000 LET C=C+1: RETURN

```

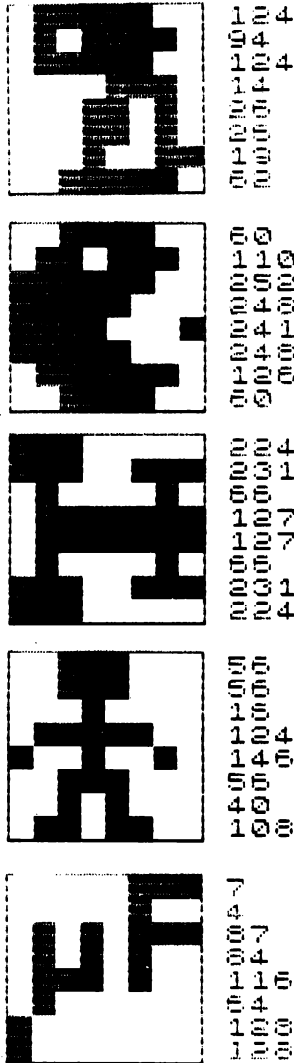


Figura 2

DISEGNAZIONE A COLORI

Con il programma di figura 3 si possono realizzare sullo schermo tramite i tasti 5,6,7,8 dei disegni ad otto colori formati con sedici caratteri grafici dello Spectrum (vedere a pag. 186 del manuale).

All'inizio occorre inserire il colore scelto per il bordo ed il colore del fondo.

Il simbolo (che all'inizio è lo spazio inverso) ed il colore (che all'inizio è bianco o nero) possono essere continuamente scelti e modificati durante il disegno premendo (senza SHIFT) i tasti indicati nella Tabella 1.

Come già detto per disegnare con il carattere prescelto occorre premere i tasti 5,6,7,8 INSIEME al tasto CAPS SHIFT.

Infine per spostarsi o per modificare il disegno basta premere il tasto 'A' che seleziona lo spazio vuoto per poi cambiarlo di nuovo quando si vuol ritornare a disegnare.

La figura 4 mostra un esempio di disegno realizzato con il Disegnatore a colori, tale disegno (come qualsiasi altro) può essere registrato su cassetta premendo BREAK e quindi eseguendo in modo diretto il comando 'SAVE "nome" SCREENS' per continuare si può premere subito dopo CONTINUE ed ENTER.

NOTA: Se si vuole disegnare meno velocemente si può aggiungere la linea
233 PAUSE n
dove n può essere compreso tra 1 e 50.

```
10 REM DISEGNAZIONE A COLORI
20 REM -----
100 INPUT "Inserire colore bord
o (0--7) e premere ENTER "
;b
110 IF b>7 OR b<0 THEN GO TO 10
120 INPUT "Inserire colore fond
o (0--7) e premere ENTER "
;p
130 IF p>7 OR p<0 THEN GO TO 10
140 IF p>0 THEN LET i=0
150 IF p<4 THEN LET i=7
160 LET m=i: BORDER b: PAPER p:
INK i: CLS
170 LET l=0: LET c=0: LET x=140
180 LET k=CODE INKEY$-7
190 IF k>4 THEN LET k=k-36
200 IF k>12 THEN LET k=k-9
210 IF k>20 THEN LET k=k-36
220 IF k>0 AND k<29 THEN GO SUB
k*100+1000
```

Figura 3

```

1030 PRINT AT L,C; INK B;" ";AT
L,C: INK I;CHR# X
11240 GO TO 100
11100 IF C=0 THEN RETURN
11111 LET C=C-1: RETURN
10200 IF C=01 THEN RETURN
10201 LET C=C+1: RETURN
10300 IF L=01 THEN RETURN
10301 LET L=L+1: RETURN
14000 IF L=0 THEN RETURN
14001 LET L=L-1: RETURN
00000 LET I=K-5: RETURN
44004 LET X=K+15: RETURN

```

Figura 3 (continua)



Figura 4

tasto	colore
0	BLACK
1	BLUE
2	RED
3	MAGENTA
4	GREEN
5	CYAN
6	YELLOW
7	WHITE

tasto	carattere
A	
B	.
C	.
D	.
E	.
F	.
G	,"
H	,"
I	,"
J	,"
K	,"
L	,"
M	,"
N	,"
O	,"
P	,"

Tabella 1

PLOTTER

L'alta risoluzione grafica disponibile sullo Spectrum (256 × 176 pixels) permette di realizzare tramite il programma 'Plotter' (Figura 5) dei disegni molto definiti, tali da essere utilizzati in applicazioni di tipo scientifico e tecnico.

La Figura 6 mostra appunto un disegno tecnico realizzato con 'Plotter'.

Plotter chiede all'inizio di inserire i colori prescelti per il bordo, il fondo e la penna scrivente; dopodichè appare nell'angolo in basso a sinistra un puntino che può essere spostato nelle otto direzioni tramite i tasti 3,4,5,6,7,8,9,0.

I tasti 1 e 2 servono invece per selezionare i modi 'disegno' e 'modifiche e spostamenti'.

Il tasto 'C' permette infine di stampare con la ZX Printer il disegno fatto fino a quel momento.

Fare attenzione che il punto, all'inizio, si vede appena, specialmente se si è selezionato un colore 'penna' molto vicino al colore 'fondo' o al colore 'bordo'.

La funzione dei tasti usati da 'Plotter' è la seguente:

Tasto	Funzione
1	modo 'disegno'
2	modo 'modifiche e spostamenti'
3	in alto a sinistra
4	in basso a sinistra
5	a sinistra
6	in basso
7	in alto
8	a destra
9	in basso a destra
0	in alto a destra
C	'COPY'

```
10 REM PLLOTTER
20 REM -----
100 INPUT "Inserire colore bord
0 (0--7) e premere ENTER "
;b
110 IF b>7 OR b<0 THEN GO TO 10
120 INPUT "Inserire colore fond
0 (0--7) e premere ENTER "
;p
130 IF p>7 OR p<0 THEN GO TO 12
140 INPUT "Inserire colore penn
a (0--7) e premere ENTER "
;i
```

Figura 5

```

0 150 IF I=P OR I>7 OR I<0 THEN G
0 1700 140
100 LET X=X0: LET Y=Y0: LET d=1
120 LET HZ=X0: INKEY#-47
130 IF HZ=X0 AND Y#0 THEN COPY
140 IF X#X0 AND Y#0 THEN GO SU
B 1000+1
1000 LET d=x/y
1100 LET d=INT d
1200 LET d=INT POINT (X,Y)=1 TH
1300 LET X=X+1: Y=Y
1400 IF X=0 OR Y=175 THEN RETU
R 1111 LET X=X+1: LET Y=Y+1: RETUR
Z 100 LET d=1: RETURN
1000 LET d=0: RETURN
1400 IF X=0 OR Y=175 THEN RETURN
1444 LET X=X-1: LET Y=Y+1: RETUR
Z 1000 IF X=0 OR Y=0 THEN RETURN
1000 LET X=X-1: LET Y=Y-1: RETUR
Z 1000 IF X=0 THEN RETURN
1000 LET X=X-1: RETURN
1100 IF Y=0 THEN RETURN
1200 LET Y=Y-1: RETURN
1300 LET Y=Y+1: RETURN
1400 LET X=X+1: RETURN
1500 LET X=X+1: RETURN
1600 IF X=0 OR Y=0 THEN RETURN
2000 LET X=X+1: LET Y=Y-1: RETUR
Z

```

Figura 5 (continua)

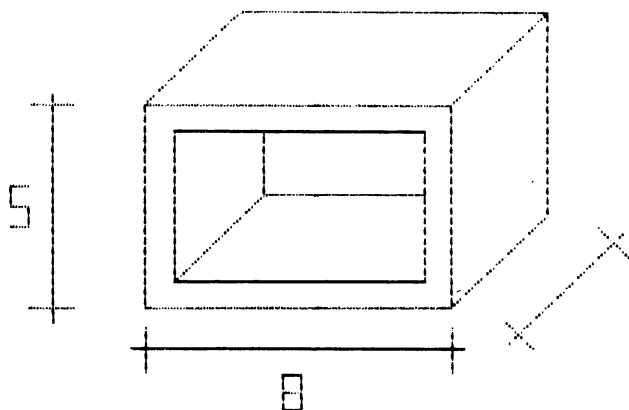


Figura 6

CURVE MATEMATICHE

Sempre sfruttando l'alta risoluzione grafica, 'Curve Matematiche' (Figura 7) disegna sullo schermo le curve di alcune funzioni matematiche. La Figura 8 mostra una delle quattro curve create dal programma; oltre a queste se ne possono comunque aggiungere molte altre.

```

10000  DIM CURVE MATEMATICHE
11000  REM
12000  REM
13000  REM
14000  REM
15000  REM
16000  REM
17000  REM
18000  REM
19000  REM
20000  REM
21000  REM
22000  REM
23000  REM
24000  REM
25000  REM
26000  REM
27000  REM
28000  REM
29000  REM
30000  REM
31000  REM
32000  REM
33000  REM
34000  REM
35000  REM
36000  REM
37000  REM
38000  REM
39000  REM
40000  REM
41000  REM
42000  REM
43000  REM
44000  REM
45000  REM
46000  REM
47000  REM
48000  REM
49000  REM
50000  REM
51000  REM
52000  REM
53000  REM
54000  REM
55000  REM
56000  REM
57000  REM
58000  REM
59000  REM
60000  REM
61000  REM
62000  REM
63000  REM
64000  REM
65000  REM
66000  REM
67000  REM
68000  REM
69000  REM
70000  REM
71000  REM
72000  REM
73000  REM
74000  REM
75000  REM
76000  REM
77000  REM
78000  REM
79000  REM
80000  REM
81000  REM
82000  REM
83000  REM
84000  REM
85000  REM
86000  REM
87000  REM
88000  REM
89000  REM
90000  REM
91000  REM
92000  REM
93000  REM
94000  REM
95000  REM
96000  REM
97000  REM
98000  REM
99000  REM

```

Figura 7

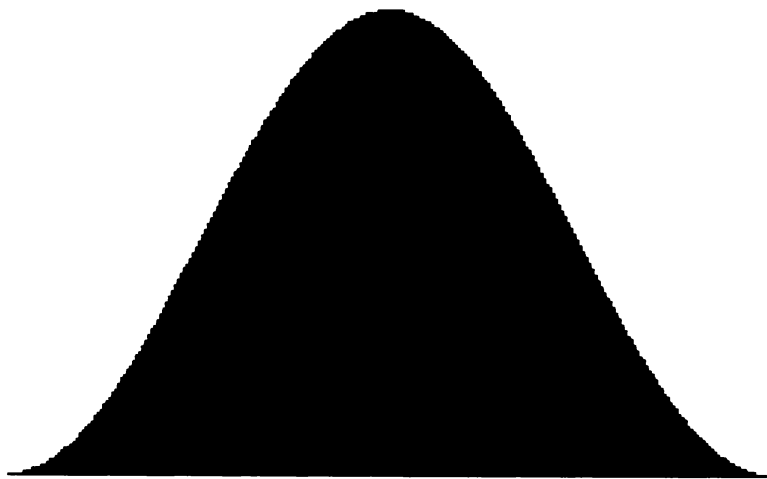


Figura 8

DISEGNO POLIGONI

Il disegnatore di poligoni (Figura 9) è una routine grafica che può essere utilizzata appunto per creare uno o più poligoni a partire da quattro valori numerici, vale a dire, le coordinate X e Y del centro, il raggio R e il numero dei lati NL di ciascun poligono (Figura 10).

Un esempio di poligono a nove lati è in Figura 11.

Sostituendo al programma di Figura 9 le linee da 99 a 999 con le nuove linee di Figura 12 si può creare sul video un'immagine dimostrativa di sei poligoni concentrici di vari colori e con un numero di lati da tre a otto (Figura 13).

```
L 99 BORDER 1: PAPER 0: INK 7: C
LS
100 REM X,Y = coordinate centro
110 INPUT "Inserire X ";X
120 INPUT "Inserire Y ";Y
130 REM R = raggio
140 INPUT "Inserire R ";R
150 REM NL = numero lati
160 INPUT "Inserire NL ";NL
170 GO SUB 40000
180 STOP
999 REM -----

4000 REM DISEGNO POLIGONI
4010 LET CX=X: LET CY=Y+R
4030 FOR P=90 TO 451 STEP (360/NL)
L)
4040 LET J=PI/180*P
4050 LET TX=R*COS J+CX
4060 LET TY=R*SIN J+CY
4070 PLOT CX,CY
4080 DRAW TX-CX,TY-CY
4090 LET CX=TX: LET CY=TY
4100 NEXT P: RETURN
```

Figura 9

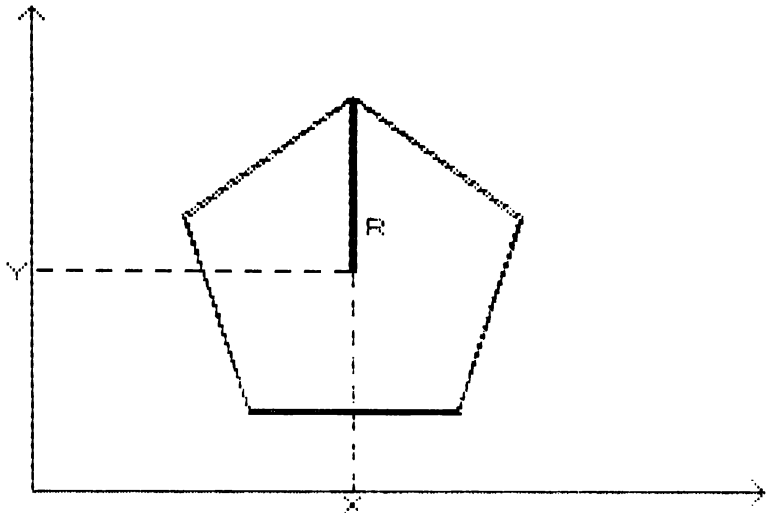


Figura 10

X=128 Y=65 R=60 @ lati

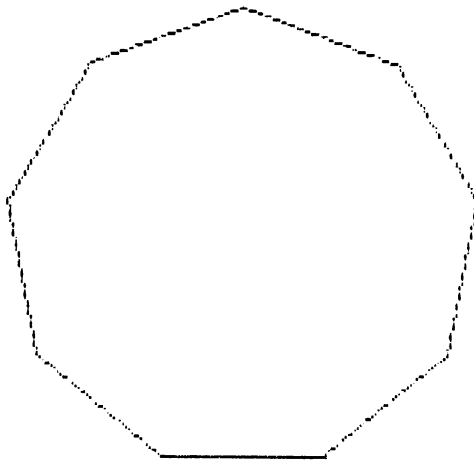


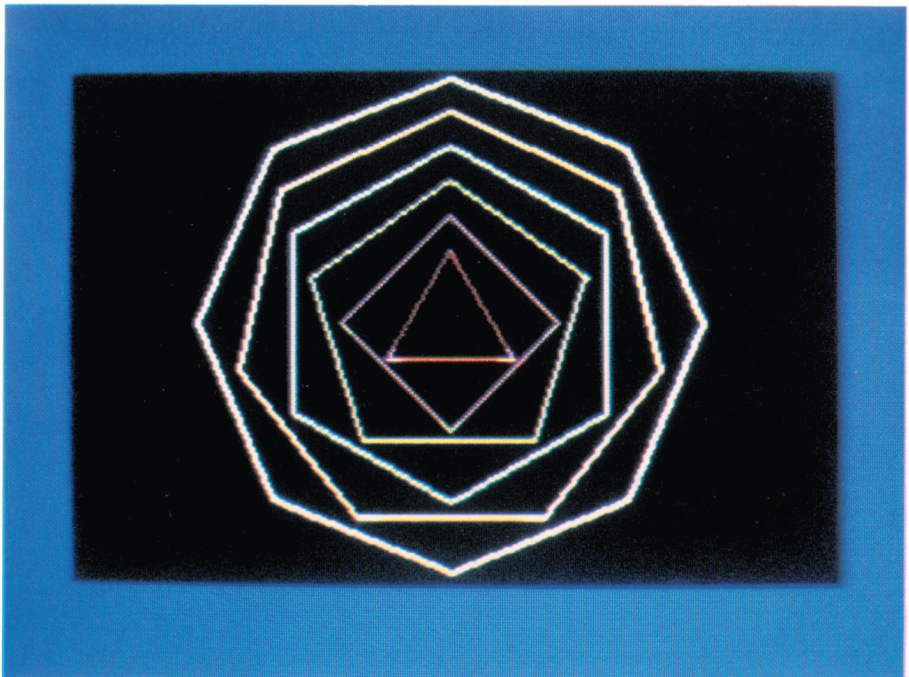
Figura 11


```

1000 REM PROGRAMA demonstrativo
1010 BORN DEF PROC 1: PAPER 0: CLS
1100 LET x=100: LET y=00
1200 FOR i=0 TO 7
1300 INK i
1400 LET rL=i+1
1500 LET rR=i*10
1600 GOTO SUB 4000: NEXT i
1700 STOP
1800 REM -----

```

Figura 12



Figura_13

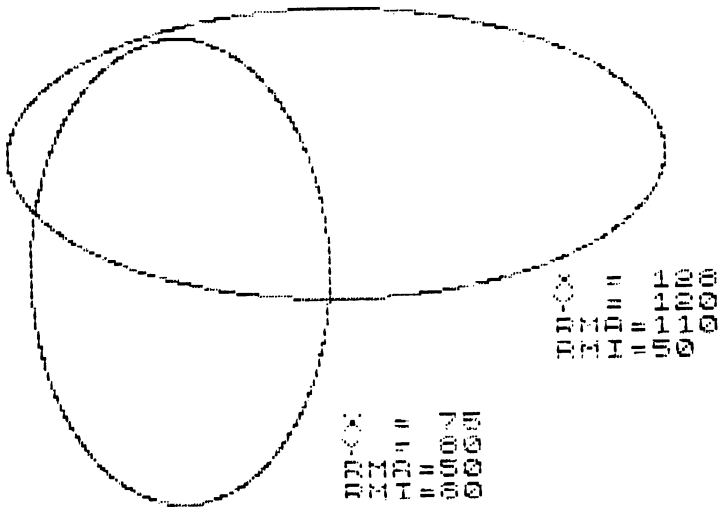


Figura 16

DISEGNO QUADRATI

Ultima routine grafica di uso generale è il 'Disegno Quadrati' (Figura 17) per disegnare dei quadrati a partire dalle coordinate X e Y dello spigolo in basso a sinistra e dalla lunghezza del lato L (Figura 18).

```

L 5 99 BORDER 1: PAPER 0: INK 4: C
100 REM X,Y = coordinate angolo
1100 INPUT "Inserire X ";X
1200 INPUT "Inserire Y ";Y
1300 REM L = lunghezza lato
1400 INPUT "Inserire L ";L
1500 GOTO PLOT
1600 PLOT
1700 RETURN
-----
000000 REM Disegno QUADRATI
000010 PLOT X,Y
000020 DRAW L-1
000030 DRAW L-1,0
000040 DRAW 0,-(L-1)
000050 DRAW -(L-1),0
000060 RETURN
```

Figura 17

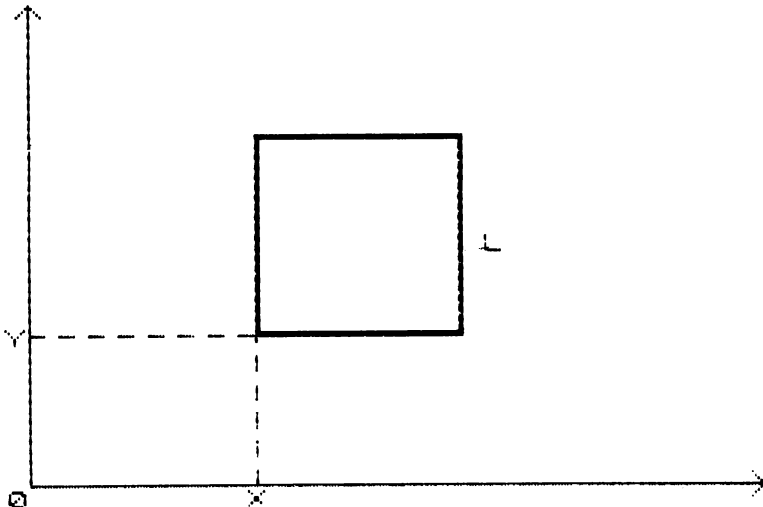


Figura 18

SPIRALI

Il programma di Figura 19 disegna un numero prestabilito (ns) di spirali a partire dal centro di coordinate x, y e con una determinata larghezza (l). Un esempio di otto spirali è in Figura 20.

```

      10  P  SPIRALI
      20  P  -----
      30  P  ORDER 1: PAPER 0: INK 0: C
L 00  P
L 10  P  X = 120 = coordinate centro
L 110 P  Y = 60  = LAT  Y = 60
L 120 P  N = 8  = numero spirali
L 130 P  L = 10 = larghezza spirale
L 140 P  C = 10
L 150 P  S = C / 100
L 160 P  P = 10 TO 30 STEP 2
L 170 P  C = P / 10 * P
L 180 P  W = C * S / 100 * P
L 190 P  W = W + C / 100
L 200 P  W = W + C * SIN J + Y
L 210 P  W = W + C / 100
L 220 P  XT

```

Figura 19

x = 120 y = 60 l = 10 ns = 8

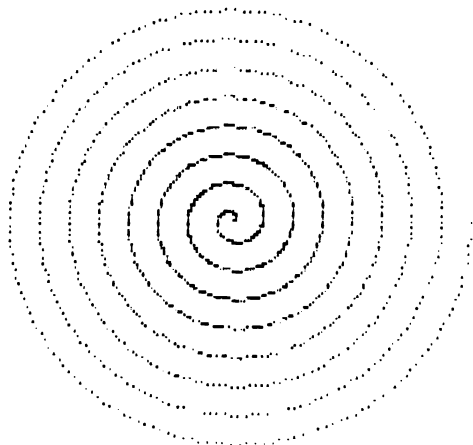


Figura 20

CONO TRIDIMENSIONALE

I dati X,Y,RMA,RMI,H all'interno del programma di Figura 21 che disegna un cono tridimensionale (Figura 22), possono essere variati a piacere per disegnare coni con basi e/o altezze diverse.

```
10 REM CONO TRIDIMENSIONALE
20 REM -----
99 BORDER 1: PAPER 0: INK 4: C
L6
100 REM X,Y = coordinate base
110 LET X=100: LET Y=40
120 REM RMA = raggio maggiore
130 REM RMI = raggio minore
140 LET RMA=60: LET RMI=20
150 REM H = altezza
160 LET H=120
170 GO SUB 1000
180 STOP
1000 IF RMA > RMI THEN LET ST=RMA
1010 IF RMI > RMA THEN LET ST=RMI
```

Figura 21 (continua)

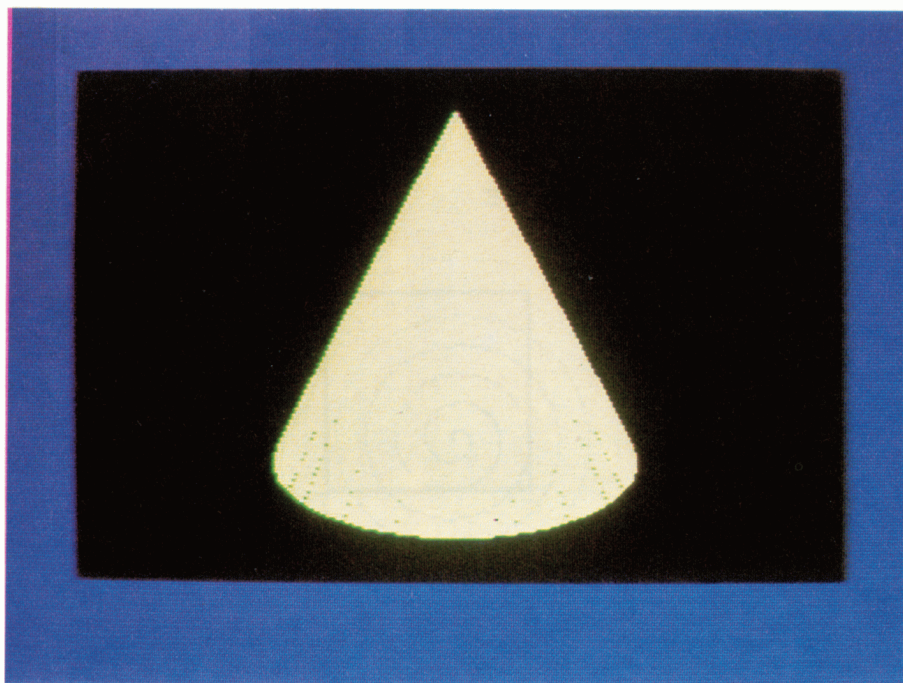


Figura 22


```

1020 FOR p=0 TO 360 STEP 0.2+INT
(300/st)
1030 LET rd=p*PI/180
1040 LET a=rma*(COS rd)+x
1050 LET b=rmi*(SIN rd)+y
1060 PLOT a,b
1070 IF p <=30 OR p >=150 THEN GO
TO 1090
1080 NEXT p: RETURN
1090 DRAW -(a-x),b-1-b+y
1100 GO TO 1030

```

Figura 21

CILINDRO TRIDIMENSIONALE

Anche nel programma di Figura 23 che disegna un cilindro tridimensionale (Figura 24), modificando i dati X,Y,RMA,RMI,H, si possono disegnare dei cilindri di diverse grandezze. Una variante di questo programma è usata in uno dei programmi di Business Grafica per visualizzare dei dati.

```

10 REM CILINDRO
20 REM TRIDIMENSIONALE
30 REM -----
90 BORDER 1: PAPER 0: INK 6: C
L5
100 REM X,Y = coordinate base ↑
110 LET x=120: LET y=120
120 REM rma = raggio maggiore
130 REM rmi = raggio minore
140 LET rma=80: LET rmi=40
150 REM h = altezza
160 LET h=80
170 GO TO SUB 1000
180 STOP
1000 IF rma > rmi THEN LET st=rma
1010 IF rmi > rma THEN LET st=rmi
1020 FOR p=0 TO 360 STEP 0.4+INT
(60/st)
1030 LET rd=p*PI/180
1040 LET a=rma*(COS rd)+x
1050 LET b=rmi*(SIN rd)+y
1060 PLOT a,b
1070 IF p >=180 THEN GO TO 1090
1080 NEXT p: RETURN
1090 DRAW 0, -(y-1)
1100 GO TO 1030

```

Figura 23

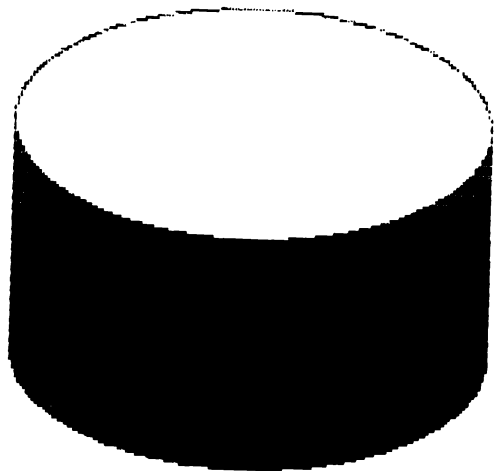


Figura 24

SFERA TERRESTRE

Il programma di Figura 25 disegna, in modo molto bello a vedersi, una sfera terrestre (Figura 26) intrecciando varie ellissi per ottenere le principali latitudini e longitudini della terra.

```
10 REM SFERA TERRESTRE
20 REM -----
99 BORDER 0: PAPER 0: INK 5: C
Ls
100 LET X=120: LET Y=80
110 LET rma=85: LET rmi=85
120 GO SUB 1000
122 LET rmi=20: GO SUB 1000
124 LET rma=74: LET rmi=85
126 GO SUB 1000
128 LET rma=40: GO SUB 1000
130 LET Y=120
```

Figura 25 (continua)

```

140 LET rma=77: LET rmi=18
150 GO SUB 1000
152 LET y=55
154 GO SUB 1000
160 LET y=160
170 LET rma=41: LET rmi=8
180 GO SUB 1000
190 LET y=16: GO SUB 1000
200 PLOT 128,3: DRAW 0.170
2000 STOP
1000 FOR p=0 TO 360 STEP 0.4
1010 LET rd=p*PI/180
1020 PLOT rma*COS rd+x,rmi*SIN r
d+c
1030 NEXT p: RETURN

```

Figura 25

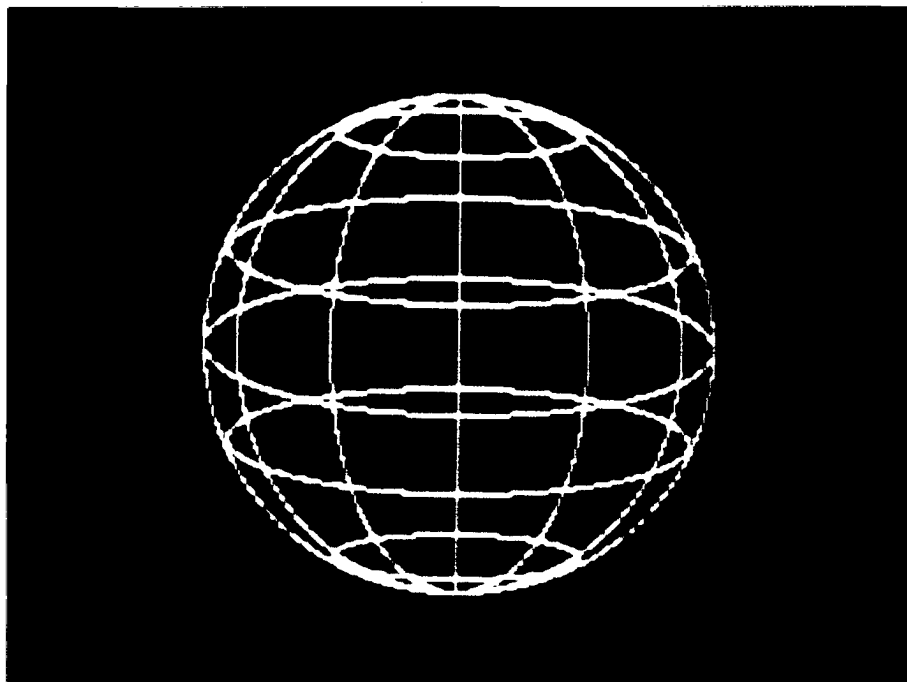


Figura 26

PROGRAMMI DI BUSINESS GRAFICA

I prossimi nove programmi introducono alla cosiddetta 'Business Grafica' vale a dire la rappresentazione in forme grafiche diverse di dati economici, statistici o anche scientifici, tecnici, matematici, ecc.

Nonostante lo Spectrum ponga alcune limitazioni di carattere grafico il risultato è comunque visivamente molto efficace.

Tutti i programmi sono divisi in due parti fondamentali: la parte 'Dati' che contiene appunto tutti i dati numerici ed alfanumerici relativi al problema trattato, con i valori massimi consentiti e la parte 'Programma' che controlla la correttezza dei dati e li trasforma nella figura grafica scelta. È importante tenere conto delle note indicate per i vari programmi.

TUTTI I DATI, I NOMI, LE CIFRE, ECC.. USATI NEGLI ESEMPI SONO PARZIALMENTE O COMPLETAMENTE INVENTATI E SERVONO A MOSTRARE IL FUNZIONAMENTO DEI VARI PROGRAMMI.

Naturalmente è possibile utilizzare i grafici per visualizzare qualsiasi tipo di dati reali, come anche è possibile cambiare i colori impiegati negli esempi.

DIAGRAMMA A TORTA

È certo il più classico modo di rappresentare i dati nella Business Grafica. Il programma è in Figura 27.

I dati vengono rappresentati in forma di fette (da 2 a 5) di una torta (Figura 28) che costituisce il 100%.

Le percentuali delle varie fette sono contenute nelle variabili $f(1) \dots f(5)$ il cui totale deve essere sempre 100, la natura di tali percentuali è invece contenuta nelle variabili stringa $P\$(1) \dots p\(5) .

A causa delle limitazioni di cui si è detto nell'introduzione, in questo programma le fette adiacenti devono essere minimo di 2-3%.

```
10  REM      Business Grafica
20  REM      ██████████ DIAGRAMMA A TORTA ██████████
30  REM      -----
```

Figura 27 (continua)

CILINDRI 3 D

Molto bella è in questo programma (Figura 29) la rappresentazione di quattro dati in forma di cilindri tridimensionali di diversa altezza e colore (Figura 30).

L'altezza dei cilindri è inserita nelle variabili h(1)...h(4).

```

100 REM          Business Grafica
200 REM          CILINDRI 3 D
300 REM          -----
300 BORDER 0: PAPER 0: INK 7: C
L8
44 LET rmi=11: LET rma=23
77 FOR a=10 TO 19
90 FOR t=0 TO 31
90 PRINT AT a,b: PAPER 1;" ":
NEXT t: NEXT a
100 REM          Dati
110 REM          a=# Max 04 caratteri
120 LET a#="Proiezioni vendite
008p) uters      in Italia (pezzi X1
000)
1200 REM          h(4)
1300 REM          h(1)
1400 REM          h(2)
1500 REM          h(3)
1600 REM          h(4)
1700 REM          c=# 1-4 Max 4 caratteri
1800 REM          c(4)
1900 REM          c(1)
2000 REM          c(2)
2100 REM          c(3)
2200 REM          c(4)
2300 REM          c(1)-4 Max 0 numeri
2400 REM          c(2)
2500 REM          c(3)
2600 REM          c(4)
2700 REM          c(1)
2800 REM          c(2)
2900 REM          c(3)
3000 REM          c(4)
3100 REM          #="00744000"
3200 REM          Programma
3300 PRINT AT 0,0: INK 7: INVERS
E
3400 #=#
3500 FOR c=0 TO 3
3600 PRINT AT 00,c*7+3: INK 3: U#
(4+1)
3700 NEXT c
3800 FOR t=0 TO 3
3900 GO TO 40+1*0 OR h(c+1)>100 T
H
4000 LET t#400+c*80
4100 LET t#100-(100-h(c+1))
4200 LET t#t(c+1)

```

Figura 29 (continua)

```

000000 LEFT I=VAL 1$(C+1)
000000 RIGHT SUB 10000
000000 ZORX+
000000 FOR C=C
000000 IF C=0 TO 3
000000 IF I(C+1)>90 THEN PRINT AT
000000 *7; INK VAL I$(C+5);D$(C+1)
000000 IF I(C+1)<94 THEN PRINT AT
000000 (C+1)/8+7;D+C*7; INK VAL 1
000000 :D$(C+1)
000000 NEXT C
000000 STOP
000000 FOR P=0 TO 360 STEP 2
000000 LEFT P=PI/180
000000 PRINT @$(C000 P)+X
000000 PRINT @$(C000 P)+Y
000000 IF P>=180 THEN GO TO 1000
000000 NEXT P: RETURN
000000 DRAW INK 1;0;-(R-1)
000000 GO TO 1000
000000 PRINT AT 5,5; " VALORI ERRE
I

```

Figura 29

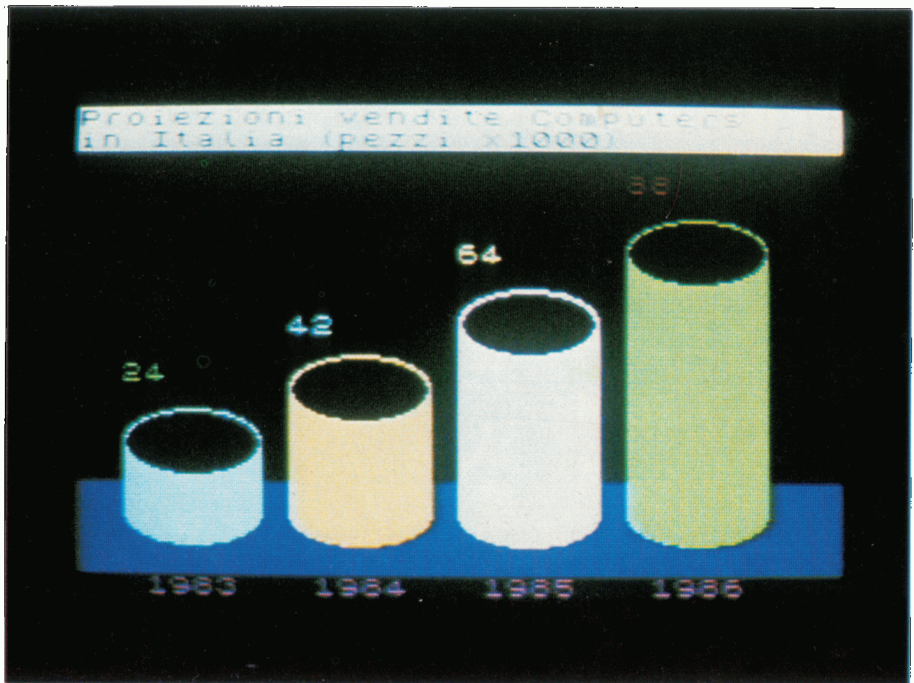


Figura 30


```

0300 NEXT c
0300 LET d=(d(k)/(r/20))-INT (d
(k)/(r/20))*8
040 IF d>=1 THEN PRINT AT L,c)
INX 5; BRIGHT 1,CHR$(143+d)
044 LET X$=STR$ d(k); IF LEN X$
>4 THEN LET X$=X$(1 TO 4)
050 IF c>15 THEN PRINT AT L,10;
BRIGHT 1; PAPER 5; INK 0;X$
060 IF c<10 THEN PRINT AT L,16;
BRIGHT 1; PAPER 0; INK 3;X$
0700 NEXT L
1000 STOP
2000 PRINT AT 5,5;" VALORI ERRE
TI"

```

Figura 31

ISTOGRAMMI VERTICALI

La sistemazione verticale degli istogrammi permette di visualizzare con questo programma (Figura 33) dodici dati diversi (nel modo indicato in Figura 34), dati i cui valori vanno inseriti (dopo aver stabilito 'r') nelle variabili d(1)....d(12).

```

10 REM Business Grafica
200 REM ISTOGRAMMI VERTICALI
300 REM -----
400 BORDER 0: PAPER 0: CLS
100 INK 2: PLOT 47,24
110 DRAW 0,127
120 FOR a=23 TO 153 STEP 8
130 PLOT 40,a: DRAW 199,0
140 NEXT a
150 LET U=PEEK 23675+256*PEEK 2
0676
155 FOR a=0 TO 5
160 FOR p=0 TO 7
165 POKE U+p+a*8,0
166 IF a<p THEN POKE U+p+a*8,25
07
170 NEXT p: NEXT a
180 FOR a=19 TO 4 STEP -6
182 PRINT AT a,4, INVERSE 1,CHR
# 144
184 NEXT a
190 INK 7
200 REM Dati

```

Figura 33 (continua)


```

600 PRINT AT 19,c: INK 5;c$(k,1
)
602 PRINT AT 20,c: INK 5;c$(k,2
)
604 PRINT AT 21,c: INK 5;c$(k,3
)
610 FOR L=18 TO 19-INT (d(k)/(r
/15))
STEP -1
6000 PRINT AT L,c: INK 6;"■"
6030 NEXT L
6040 LET n=((d(k)/(r/15))-INT (d
(k)/(r/15)))*10
6040 IF n>=1 THEN PRINT AT L,c:
INK 6;CHR$(143+(8-n))
6050 NEXT c
1000 STOP
2000 PRINT AT 5,5;" VALORI ERRO
RI"

```

Figura 33

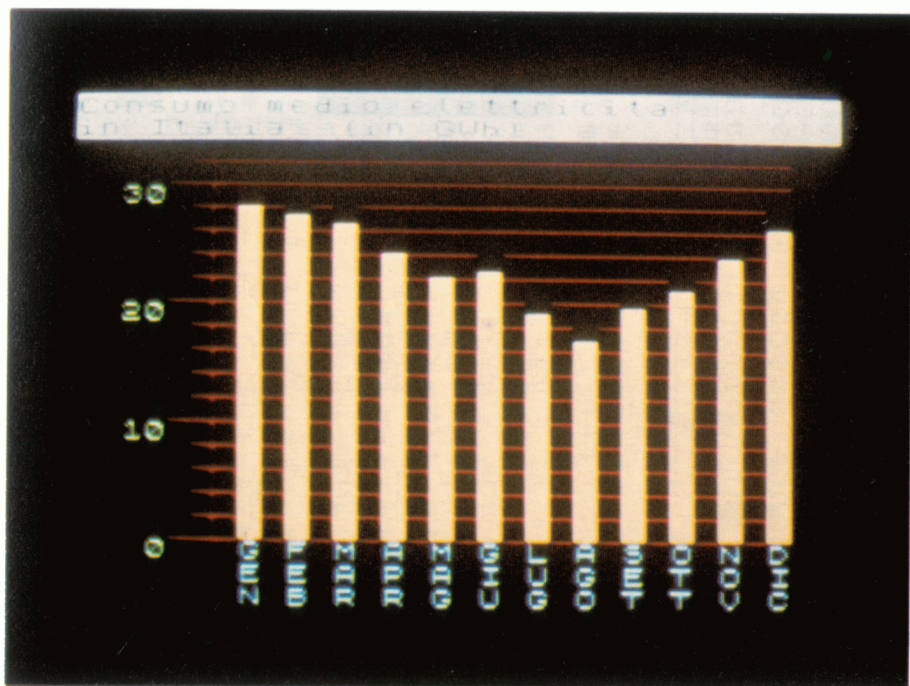


Figura 34

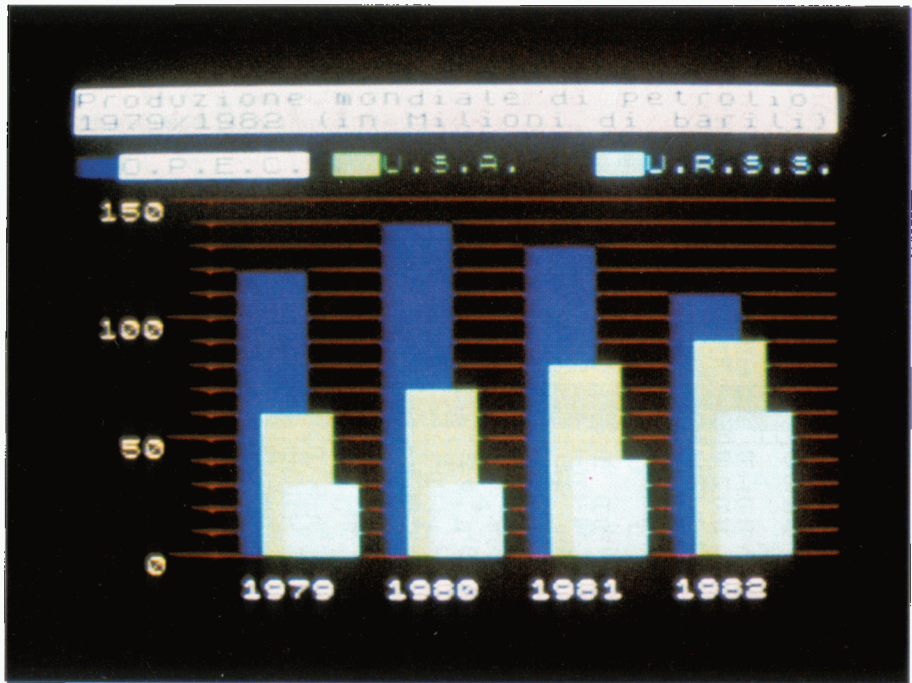


Figura 36

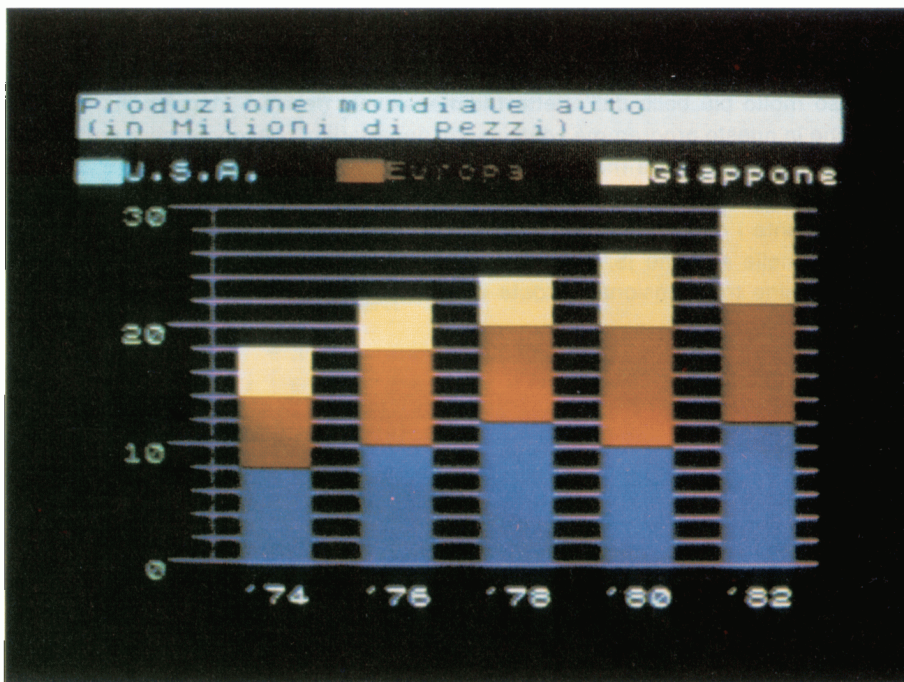


Figura 38

ISTOGRAMMI VERTICALI TRIDIMENSIONALI

Questo programma (Figura 39) è sostanzialmente simile a quello di Figura 35-36 ma reso molto più bello dall'effetto tridimensionale dei vari istogrammi.

Anche in questo programma i dati inseriti nelle variabili devono essere arrotondati e non superare il valore massimo 'r'.

Questa volta gli istogrammi sono verdi, azzurri e gialli e ad essi corrispondono rispettivamente i dati delle variabili d(1)...(4), e (1)...(4), f(1)...(4) e gli argomenti contenuti nelle stringhe j\$, k\$, w\$.

L'immagine tridimensionale creata con i dati esempio è in figura 40.

```

1000 REM              BUSINESS Grafica
2000 REM              ISTOGRAMMI VERTICALI
3000 REM              TRIDIMENSIONALI
4000 REM
5000 REM              -----
6000 HZKX ER : PAPER 0: CLS
7000 DRAW 0: PLOT 99,24
8000 CRT U=PAGEK 23675+256*PAGEK 2
9000
10000 DAT A 0 0 5 5 , 10 7 , 6 3 , 3 1 , 1 5 , 7 , 3 , 1
11000 PPT 0 0 4 0 , 2 4 0 , 0 4 0 , 2 4 0 , 2 5 0 , 2 5 0 , 2 5 0 , 2 5 0
12000 PTT 0 0 0 0 , 1 0 0 , 4 0 0 , 5 5 , 2 2 3 , 2 5 4 , 2 5 0
13000 PTC 0 0 1 0 0 0 , 1 0 0 0 , 1 0 0 0 , 1 0 4 0 , 1 0 0 , 1 0 0 , 1 0 0
14000 PTO a:=U TO U+30
15000 TPOD w:=0 TO U w,d: NEXT a
16000 TPTD w:=0 TO U w,d: NEXT a
17000 THZXT AT 14:14+a,a: INK 0: PAP
18000 CHR# 145:144
19000 ZEXXT w# 144
20000 PFCPT a#0 0 0: DRAW 7,0
21000 PTCPT a#0 1 0 0 TO 145 STEP 0
22000 ZFCPT a#0 2 0 0 TO 145 STEP 0
23000 PFCPT a#0 2 3 TO 145 STEP 40
24000 ZFCPT a#0 4 , a: DRAW 0,0
25000 HZKX 7
26000 REM
27000 REM              Dati
28000 REM              a# Max 30 caratteri
29000 REM              a#M$ Vendite programmi 6
30000 REM              a#M$
31000 REM              k# Profes..
32000 REM              k#M$ Uciendit..
33000 REM              f#M$ Uciendit..
34000 REM              f#M$ M$ Max 6 caratt.
35000 REM              w#M$ M$
36000 REM              w#M$ M$ Max 6 caratt.

```

Figura 39 (continua)


```

600 IF d(k)/(r/15) >= 1 THEN PRIN
T AT 19,c) INK 4) PAPER 2,CHR$ 1
47; "██████";
610 FOR l=18 TO 19-INT (d(k)/(r
/15)) STEP -1
620 PRINT AT l,c) INK 4;CHR$ 14
630 "██████";
640 NEXT l
650 IF d(k)/(r/15) >= 1 THEN PRIN
T AT l+1,c) INK 4;CHR$, 146;"██████";
CHR$, 145;
660 NEXT c
670 FOR c=7 TO 20 STEP 5
680 LET k=(c-2)/5
690 IF e(k)/(r/15) >= 1 THEN PRIN
T AT 20,c+1) INK 5, PAPER 2;CHR$
147; "██████";
700 FOR l=19 TO 20-INT (e(k)/(r
/15)) STEP -1
710 PRINT AT l,c+1) INK 5;CHR$
148;"██████";
720 NEXT l

```

Figura 39 (continua)

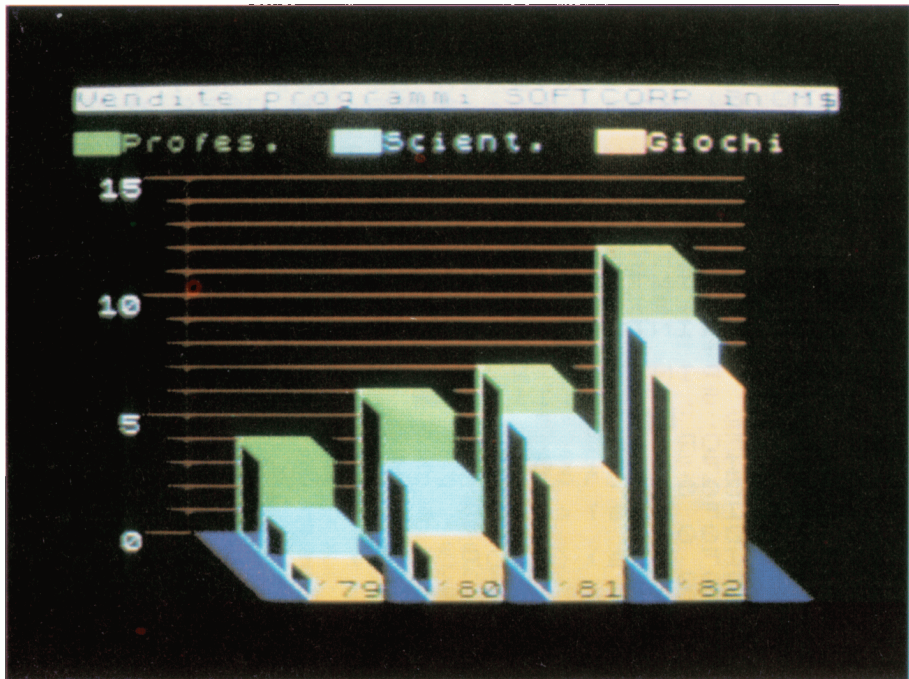


Figura 40

```

T 0000 IF s(k)/(r/15) >= 1 THEN PRIN
;: AT L+1,c+1; INK 5;CHR$ 146;"
0000 CHR$ 145
00004 TZX NEXT c
00000 TZX NEXT c
00000 LIMIT k=(c-2)/5
T 00000 IF f(k)/(r/15) >= 1 THEN PRIN
147;: AT 21,c+2; INK 0; PAPER 2;CHR$
00000 FOR L=20 TO 21-INT (f(k)/(r
/15))
00000 PRINT AT L,c+2; INK 6;CHR$
148;"
00000 NEXT L
T 00000 IF f(k)/(r/15) >= 1 THEN PRIN
;: AT L+1,c+2; INK 6;CHR$ 146;"
00000 CHR$ 145
00000 PRINT AT 21,c+3; INK 0; PAP
ER 2;CHR$ (k)
00000 NEXT c
10000 STOP
20000 PRINT AT 5,5;" VALORI ERRE
TI"

```

Figura 39


```

3370 LET C$(0) = "74"
3380 LET C$(1) = "77"
3390 LET C$(2) = "77"
3400 LET C$(3) = "77"
3410 LET C$(4) = "77"
3420 LET C$(5) = "77"
3430 LET C$(6) = "77"
3440 LET C$(7) = "77"
3450 LET C$(8) = "77"
3460 LET C$(9) = "77"
3470 LET C$(10) = "77"
3480 LET C$(11) = "77"
3490 LET C$(12) = "77"
3500 LET C$(13) = "77"
3510 IF LEN a$ > 64 THEN LET a$ = a$
(1 TO 64)
3520 PRINT AT 0,0; INVERSE 1; a$
3530 INK 7; PRINT AT 10,3; "0"
3540 PRINT AT 4,4-LEN STR$ (r)
3550 PRINT AT 9,4-LEN STR$ (r/3*
2); (r/3*2)
3560 PRINT AT 14,4-LEN STR$ (r/3
); (r/3)
3570 INK 4
3580 FOR a=1 TO 6
3590 IF d(a) < 0 OR d(a) > r THEN GO
TO 2000
3600 NEXT a
3610 FOR a=4 TO 29 STEP 5

```

Figura 41 (continua)

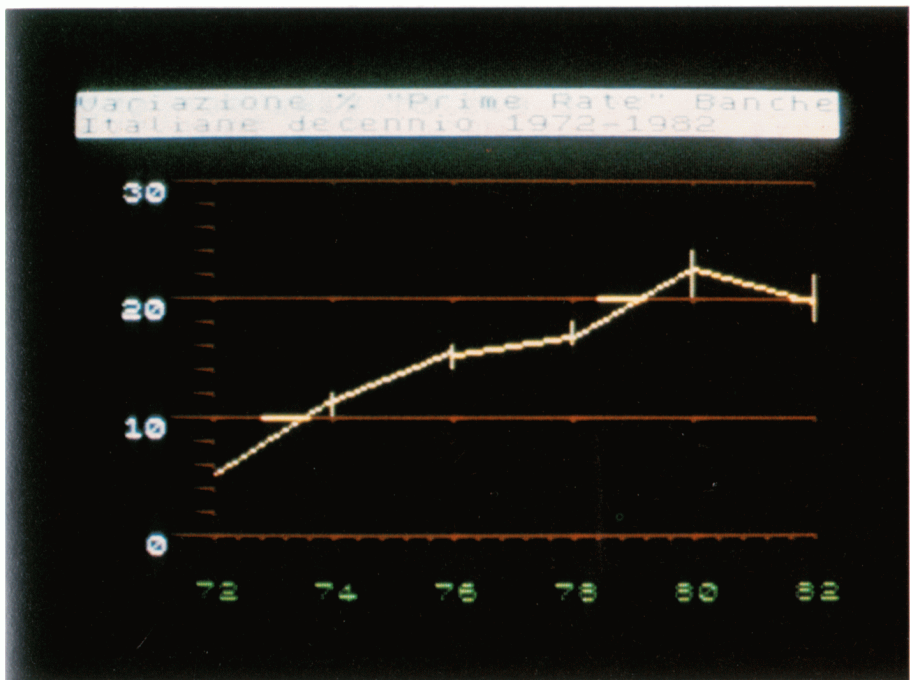


Figura 42

```

0000 PRINT AT 21,a) c#((a+1)/5)
0000 NEXT a
0000 IF X#0
0000 LET X#40 TO 240 STEP 40
0000 LET C#(X-6)/40
0000 LET L#60+a(X)/100
0000 LET U#60+a(X+1)/100
0000 PLOT X,U: DRAW 39,L-Y
0000 NEXT X
0000 INKEY J
0000 STOP
0000 PRINT AT 5,5: " VALORI ERRA
0000 H

```

Figura 41

DIAGRAMMA A SUPERFICIE

Ultimo esempio di Business Grafica è il 'Diagramma a Superficie' utilizzato per confrontare l'andamento di due curve riferite a due argomenti legati tra loro.

Nel programma (Figura 43) i dodici valori che determinano l'andamento della curva blu e della curva gialla vanno inseriti rispettivamente nelle variabili d(1)...(6) e i(1)...(6).

Questo programma può essere utilizzato solo se la distanza tra le due curve è, in tutti i punti, superiore a 16 pixels.

Con i dati esempio si ottiene l'immagine di Figura 44.

```

100 REM          Business Grafica
200 REM          DIAGRAMMA A SUPERFICIE
300 REM          -----
400 BORDER 0: PAPER 0: CLS
1000 INK 4
1020 FOR a=20 TO 143 STEP 6
1040 PLOT 40,a: DRAW 7,0
1060 NEXT a
1080 FOR a=47 TO 248 STEP 6
1100 PLOT a,16: DRAW 0,7
1120 NEXT a
1140 FOR a=20 TO 143 STEP 40
1160 PLOT 30,a: DRAW 15,0: IF a=
20 THEN DRAW 200,0
1180 NEXT a
1200 FOR a=47 TO 248 STEP 40
1220 PLOT a,8: DRAW 0,15: IF a=4
7 THEN DRAW 0,120
1240 NEXT a
1260 FOR l=4 TO 18
1280 PRINT AT l,6: PAPER 6:"
                                ": REM 25
spazi
1300 NEXT l
1320 INK 7
1340 REM          Dati
1360 REM          a#=# Max 04 caratteri
1380 REM          QUOTAZIONI medie #
1400 REM          # Inglese (giallo)
1420 REM          #
1440 REM          # valore massimo scala
1460 REM          # valore per r:
1480 REM          # 10000 10000 30000 50000
1500 REM          #
1520 REM          d(1) -- (6) tra
1540 REM          i(1) -- (6) tra
1560 REM          # il valore Max "r"

```

Figura 43 (continua)


```

7400 FOR X=40 TO 240 STEP 40
7500 LET K=(X-8)/40
7600 LET Y=20+i(K)/(r/120)
7700 LET L=20+i(K+1)/(r/120)
7800 PLOT X,Y: DRAW 39,L-Y
7900 NZRXT X
8000 FOR X=40 TO 240
8100 FOR L=143 TO 24 STEP -1
8200 IF POINT(X,Y)=1 THEN GO TO
8300 PLOT X,Y
8400 NZRXT X
8500 HZRXT X
8600 STOPK 7: PAPER 0
8700 PRINT AT 5,5:" VALORI ERR
8800 ..

```

Figura 43

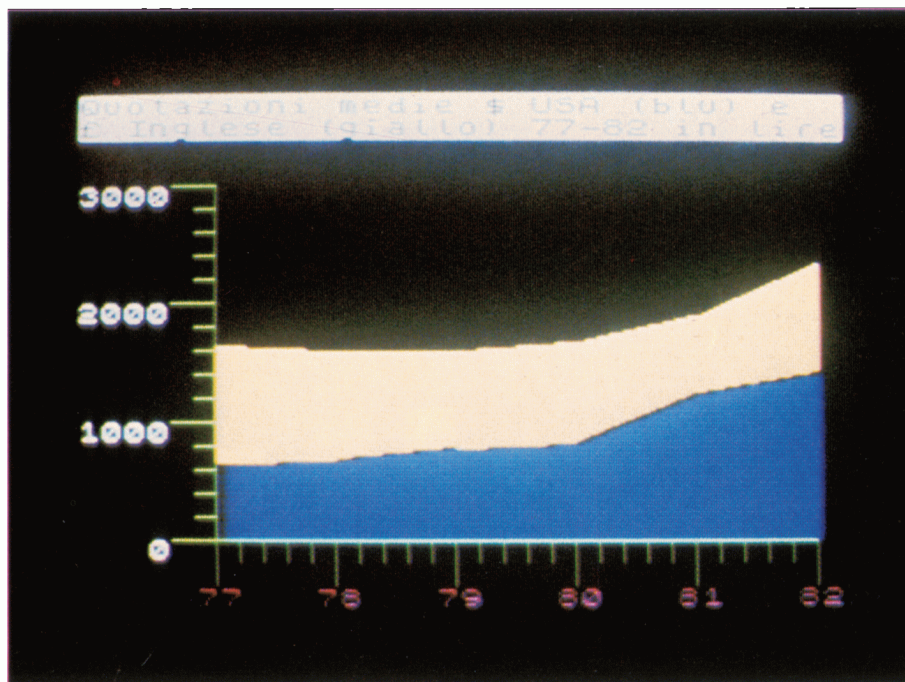


Figura 44

NEW YORK

'New York' (Figura 45) disegna sullo schermo il panorama di una città con due file di grattacieli. Ogni volta che si fa girare, il programma crea un panorama diverso e casuale; un esempio è in Figura 46.

```
10 REM NEW YORK
20 REM -----
3000 BORDER 0: PAPER 0: CLS
1000 FOR a=1 TO 200
1002 INK a+INT (RND*2)
1004 PLOT RND*255,75+RND*100
1006 NEXT a
1008 LET k=4
110 DIM l(15)
120 DIM r(15)
130 DIM i(15)
140 FOR a=1 TO 15
150 LET l(a)=(2+INT (RND*2))*8
160 LET r(a)=(k+INT (RND*(2+k))
) *8+INT (RND*k) -1
```

Figura 45 (continua)

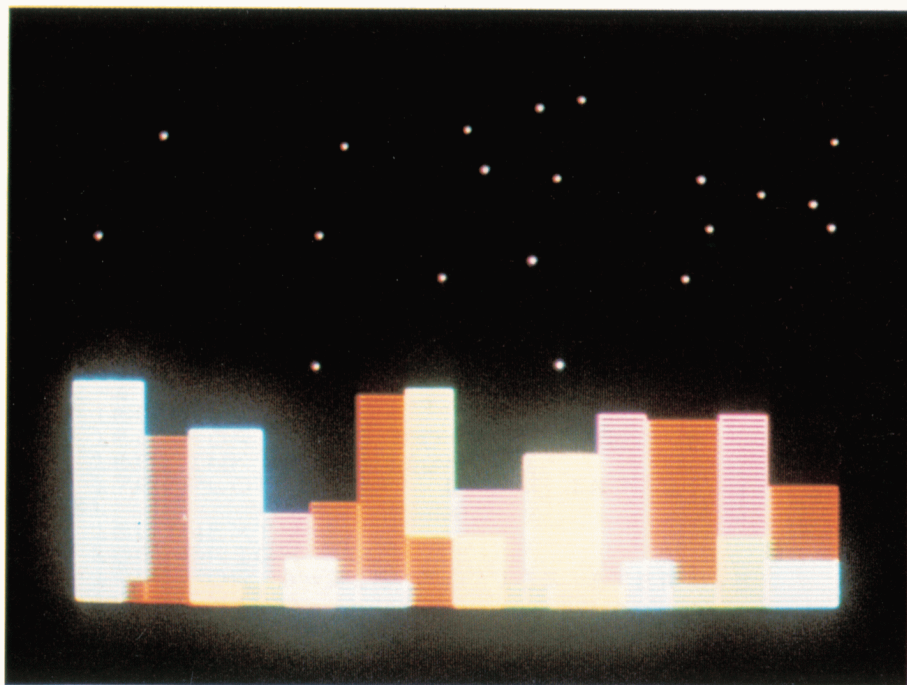


Figura 46

```

1170 LET i(a)=2+INT (RND*6)
1180 FNEXT a
1190 LET x=0
1200 FOR a=1 TO 15
1210 LET x=x+l(a)
1220 IF x<255 THEN GO SUB 1000
1230 IF x>255 THEN GO TO 250
1240 FNEXT a
1250 LET l(a)=255-(x-l(a))
1260 LET x=255
1270 GO SUB 1000
1280 INK 7
1290 IF k=1 THEN GO TO 320
1300 LET k=1
1310 GO TO 110
1320 STOP
1000 INK i(a)
1010 FOR y=0 TO h(a) STEP 2
1020 PLOT x-l(a),y: DRAW l(a)-1,
0
1030 NEXT y
1040 PLOT x-l(a),0: DRAW 0,h(a)-
1
1050 PLOT x-1,0: DRAW 0,h(a)-1
1111 RETURN

```

Figura 45

BIG BANG

'Big Bang' (o 'Grande Scoppio') è chiamata la teoria, oggi ritenuta più probabile, sulla nascita dell'universo. Il programma di Figura 47 ne dà un'immagine simulata.

```
10 REM BIG BANG
20 REM -----
30 BORDER 0: PAPER 0: INK 2: C
L3)
110 FOR a=0 TO 360 STEP 2
120 LET k=5+10*RND
130 LET rd=PI/180*a
140 PLOT 128+k*COS rd,88+k*SIN rd
r d)
150 LET h=30+40*RND
160 DRAW h*COS rd,h*SIN rd
170 NEXT a
180 INK 7
```

Figura 47

DISEGNO LINEE

Un disegno molto bello in quattro colori è creato dal programma di Figura 48.

```
10 REM DISEGNO DI LINEE
20 REM -----
30 BORDER 0: PAPER 0: INK 7
40 FOR a=0 TO 360 STEP 2
100 FOR a=40 TO 120 STEP 8
110 PLOT a,88: DRAW 127-a,a-33
120 NEXT a
130 FOR a=215 TO 135 STEP -8
130 INK 4
140 PLOT a,88: DRAW -(a-128),22
2) - a)
150 NEXT a
160 FOR a=40 TO 120 STEP 8
160 INK 0: BRIGHT 1
170 PLOT a,87: DRAW 127-a,-(a-3
3)
180 NEXT a
180 BRIGHT 0
190 FOR a=215 TO 135 STEP -8
190 INK 6
200 PLOT a,87: DRAW -(a-128),-(
2) a)
210 NEXT a
220 INK 7
```

Figura 48

ALVEARE (Figura 49)

```

1000  REM
1001  REM
1002  BORDER 0: PAPER 0: INK 0: C
L6
100  LET r=10: LET n1=6
110  FOR u=164 TO 0 STEP -30
120  FOR x=0 TO 240 STEP 10
140  GO SUB 4010
150  NEXT x: NEXT u
160  NEXT r
170  FOR a=164 TO 0 STEP -30
180  IF POINT (a,b)=1 THEN PLOT
190  DRAW a,-a
200  NEXT a
210  STOP
40010 LET cx=x: LET cy=y+r
40020 PLOT cx,cy
40030 FOR p=90 TO 451 STEP (360/n1)
40040 LET j=PI/180*p
40050 LET jx=7*COS j+x
40060 LET jy=7*SIN j+y
40070 PLOT jx,jy
40080 DRAW cx,cy:jx,jy
40090 LET cx=jx: LET cy=jy
41000 NEXT p: RETURN

```

Figura 49

PIRAMIDE (Figura 50)

```

10  REM
100  REM
101  BORDER 0: PAPER 0: INK 0: C
L6
100  LET x=122
110  LET k=1
120  FOR u=164 TO 0 STEP -5
130  FOR k=1 TO k
140  PLOT x+(u-1)*6,u: DRAW 7,0:
DRAW u,-4: DRAW 17,0: DRAW 0,4
150  NEXT k
160  LET x=x-4
170  LET k=k+1
180  NEXT u

```

Figura 50

MONTAGNE

Ogni volta che si fa girare, il programma 'Montagne' (Figura 52) crea un panorama diverso e casuale.

```
10 REM MONTAGNE
20 REM -----
100 BORDER 0: PAPER 1: INK 4: C
L5
110 DIM h(20)
120 DIM l(20)
130 FOR a=1 TO 20
140 LET l(a)=12+INT (RND*30)
150 LET h(a)=10+INT (RND*50)
160 NEXT a
170 LET x=0: LET y=h(1)
180 PLOT x,y
190 FOR a=2 TO 20
200 LET x=x+l(a)
210 IF x<255 THEN DRAW l(a),h(a)
220 IF x>255 THEN GO TO 250
230 LET y=h(a)
240 NEXT a
250 FOR W=0 TO 255-(x-l(a)),h(a)-y
260 FOR R=0 TO 175
270 IF POINT (a,R)=1 THEN GO TO
280 PLOT a,R
290 NEXT R
300 NEXT a
```

Figura 52

BANDIERA REGNO UNITO

Quello di Figura 53 è il primo di tre programmi che disegnano delle bandiere, a colori e molto vicine alle vere. Della prima viene anche dato il risultato finale (Figura 54).

```

000100 DD      BANDIERA REGNO UNITO
000200 DD      -----
000300 BORDER 0: PAPER 0: INK 7: C
000400
000500 FOR    a = 1 TO 840
000600 PRINT a; PAPER 7; " ";
000700 NEXT a; PRINT
000800 FOR    a = 7: INK 0
000900 PRINT AT C, 14; PAPER 7; INK
001000
001100 NEXT C
001200 FOR    c = 0 TO 31
001300 PRINT AT 0, 0; " "
001400 PRINT AT 10, 0; " "
001500 PRINT AT 10, 0; " "
001600 PRINT AT 11, 14; " "
001700 PRINT AT 11, 14; " "
001800 FOR    x = 0 TO 100 STEP -1
001900 PRINT AT 0, 0; " "
002000 NEXT x
002100 FOR    x = 4 TO 66
002200 PRINT AT 0, 4; DRAW 66-x, -8
002300 NEXT x
002400 FOR    x = 15 TO 163
002500 PRINT AT 0, 15; DRAW 163-x, -8
002600 NEXT x
002700 FOR    x = 148 TO 241
002800 PRINT AT 0, 148; DRAW 241-x, -8
002900 NEXT x
003000 FOR    y = 112 TO 164
003100 PRINT AT 0, 112; DRAW 72 - (y - 112) * 1
003200 NEXT y
003300 FOR    y = 112 TO 164
003400 PRINT AT 0, 112; DRAW 164 - (164 - y) * 1
003500 NEXT y
003600 FOR    y = 76 TO 86 STEP -1

```

Figura 53 (continua)

```

4470 PLOT 0,y: DRAW 72-(76-y)*1.
4500 NEXT y
4600 FOR y=76 TO 34 STEP -1
0000 PLOT 255,y: DRAW -(y-34)*1.
0010 NEXT y
0020 FOR x=16 TO 103
0030 PLOT x,175: DRAW 0,-(x-16)*
0040 NEXT x
0050 FOR x=238 TO 148 STEP -1
0060 PLOT x,16: DRAW 0,(238-x)*0
0070 NEXT x
0080 FOR x=29 TO 103
0090 PLOT x,16: DRAW 0,(x-29)*0.
0100 NEXT x
0110 FOR x=225 TO 148 STEP -1
0120 PLOT x,175: DRAW 0,-(225-x)
*0130? NEXT x
0140

```

Figura 53

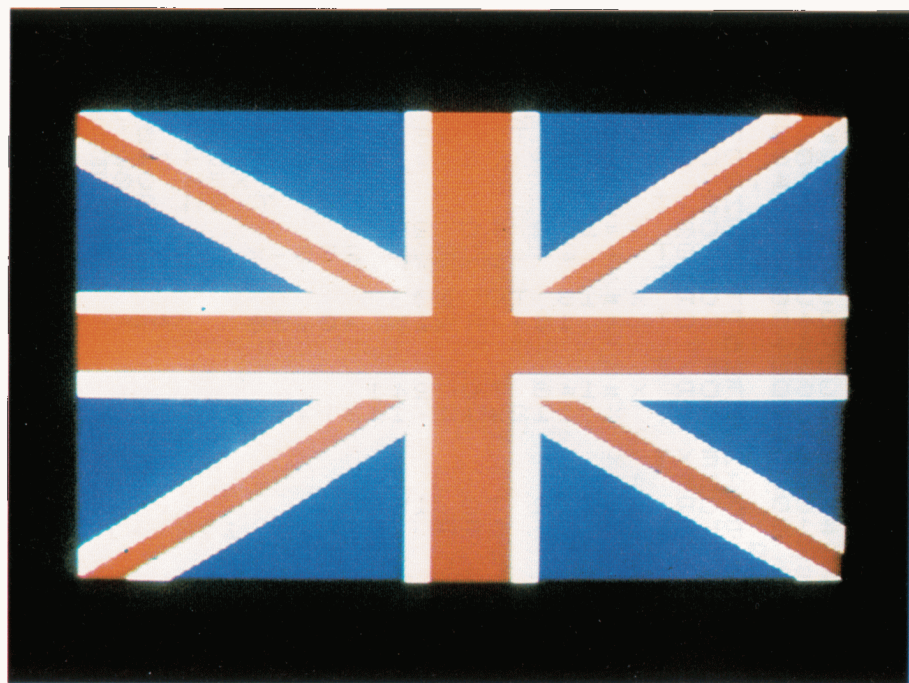


Figura 54

BANDIERA STATI UNITI (Figura 55)

```

10 REM bandiera USA
20 REM -----
30 BORDER 0: PAPER 0: INK 7: C
L
110 LET U=PEEK 23675+256*PEEK 2
3070
120 DATA 16,16,166,124,56,40,68
,0
130 FOR a=U TO U+7
140 READ d: POKE a,d: NEXT a
150 LET a$=""
160 LET b$=""
170 LET c$=""
180 REM 30 spazi
180 FOR l=0 TO 17 STEP 3
190 PRINT AT l,0: PAPER 7: INK
N
200 PRINT AT l+1,0: PAPER 7: INK
X
210 PRINT AT l+2,0: PAPER 7: c$
220 NEXT l
230 PRINT AT 18,0: PAPER 7: INK
240 PRINT AT 19,0: INK 0: b$
+
250 LET p$="" +CHR$ 144+" +CHR$ 144+" +CHR$ 144
+
260 LET j$="" +p$(1 TO 11)+" "
270 PRINT AT 0,0: PAPER 1: j$
280 REM 10 spazi
290 FOR l=1 TO 10 STEP 2
300 PRINT AT l,0: PAPER 1: INK
310 IF l<9 THEN PRINT AT l+1,0:
320 PRINT AT l,0: INK 7: j$
330 NEXT l
340 PRINT AT 10,0: PAPER 1: INK

```

Figura 55

PROGRAMMI VARI

FISICA DEI REATTORI NUCLEARI

Il programma (Figura 57) simula il processo di 'Fissione Nucleare' che è alla base del funzionamento appunto delle centrali nucleari che producono energia. La Fissione Nucleare è una reazione provocata da neutroni veloci che colpiscono degli atomi di Uranio 235 i quali si spaccano producendo due atomi più piccoli di Bario e Kripto ed altri due neutroni che colpendo a loro volta degli atomi di Uranio provocano la cosiddetta 'reazione a catena'. La reazione produce anche energia che viene sfruttata per produrre elettricità e calore. La Figura 58 mostra una fase intermedia della reazione di fissione. In questo programma è molto importante inserire bene gli spazi nelle varie stringhe.

```
10 REM FISICA DEI REATTORI
20 REM NUCLEARI
30 REM
40 REM -----
L1 50 ORDER 0: PAPER 0: INK 7: C
L3
110 DATA 0,0,0,04,04,0,0,0,04
0500 0500 0500 0500 0500 0500 0500 0500 0500 0500
0510 LET U=PEEK 23675+256*PEEK 2
0570
130 FOR a=0 TO U+23
140 READ C: POKE a,C: NEXT a
150 PRINT " SIMULATORE DI FISSI
ONE NUCLEARE"
160 PRINT " AT 0,00; INK 6; CHR# 1
44: INK 7: " Neutro"
170 PRINT " AT 4,000; INK 4; CHR# 1
46: INK 7: " Uranio 235"
180 PRINT " AT 6,00; INK 5; CHR# 1
48: INK 7: " Bario 139"
190 PRINT " AT 8,00; INK 2; CHR# 1
49: INK 7: " Kripto"
200 INK 8: PRINT " AT 2,15; CHR# 1
44: INK 4: PRINT " AT 6,15; CHR# 14
0: AT 14,7; CHR# 146; AT 14,23; CHR#
146; AT 18; CHR# 146; AT 18,11; C
146; AT 18; CHR# 146; AT 18,
07; CHR# 146
```

Figura 57 (continua)

```

14010 LET a#=CHR$ 16+CHR$ 5+CHR$
14020 CHR$ 100+CHR$ 10+CHR$ 0+CHR$
14030 LET b#=CHR$ 16+CHR$ 5++CHR$
14040 LET b#=""
14050 PRINT AT 3,0; INK 0; "Dremer
14060 INPUT LINE a#"
14070 PRINT AT 3,0;"
14080 FOR L=3 TO 8
14090 PDUS L
14100 PRINT AT (-1,15;" "); AT (L,15
14110 NEXT L
14120 PRINT AT 6,14;"a#"
14130 PDUS L
14140 PRINT AT (L+0,15-1); AT (L+0
14150 PDUS L+0,15+L); AT (L+0,
14160 PRINT AT 14,6;"a"; AT 14,22;"a
14170 PDUS L
14180 PRINT AT 14,6;"a"; AT 14,22;"a
14190 PDUS L
14200 PDUS L
14210 PDUS L
14220 PDUS L
14230 PDUS L
14240 PDUS L
14250 PDUS L
14260 PDUS L
14270 PDUS L
14280 PDUS L
14290 PDUS L
14300 PDUS L
14310 PDUS L
14320 PDUS L
14330 PDUS L
14340 PDUS L
14350 PDUS L
14360 PDUS L
14370 PDUS L
14380 PDUS L
14390 PDUS L
14400 PDUS L
14410 PDUS L
14420 PDUS L
14430 PDUS L
14440 PDUS L
14450 PDUS L
14460 PDUS L
14470 PDUS L
14480 PDUS L
14490 PDUS L
14500 PDUS L
14510 PDUS L
14520 PDUS L
14530 PDUS L
14540 PDUS L
14550 PDUS L
14560 PDUS L
14570 PDUS L
14580 PDUS L
14590 PDUS L
14600 PDUS L
14610 PDUS L
14620 PDUS L
14630 PDUS L
14640 PDUS L
14650 PDUS L
14660 PDUS L
14670 PDUS L
14680 PDUS L
14690 PDUS L
14700 PDUS L
14710 PDUS L
14720 PDUS L
14730 PDUS L
14740 PDUS L
14750 PDUS L
14760 PDUS L
14770 PDUS L
14780 PDUS L
14790 PDUS L
14800 PDUS L
14810 PDUS L
14820 PDUS L
14830 PDUS L
14840 PDUS L
14850 PDUS L
14860 PDUS L
14870 PDUS L
14880 PDUS L
14890 PDUS L
14900 PDUS L
14910 PDUS L
14920 PDUS L
14930 PDUS L
14940 PDUS L
14950 PDUS L
14960 PDUS L
14970 PDUS L
14980 PDUS L
14990 PDUS L
15000 PDUS L

```

Figura 57

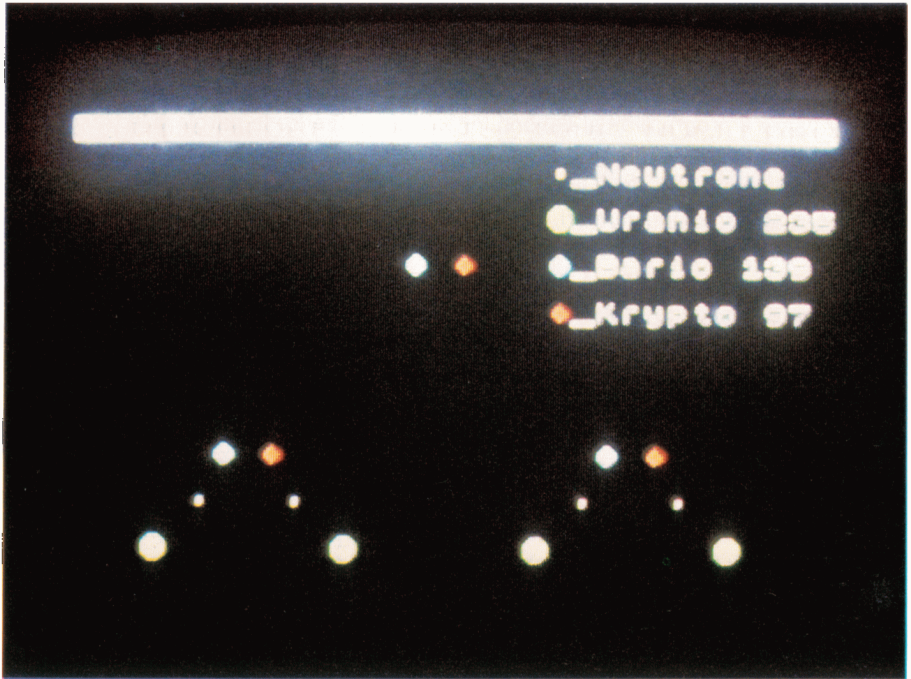


Figura 58

LETTURA VELOCE

È possibile esercitarsi a riconoscere più velocemente parole e frasi, in modo da riuscire a leggere più rapidamente quando occorre, usando il programma di Figura 59. Per usare il programma si devono innanzitutto inserire le parole o le frasi (dalla cui lunghezza dipende anche il grado di difficoltà) in una serie di istruzioni DATA a partire dalla linea 1001. Ogni linea deve contenere una sola frase di massimo 32 caratteri e/o spazi, segni, numeri, ecc.

Le frasi non devono contenere delle keyword ma solo i caratteri con codice da 32 a 126 (vedere appendice A del manuale Spectrum); inoltre le linee devono essere numerate di seguito (1001, 1002, 1003, ecc.) senza saltarne nessuna e devono concludersi con la linea 9999 DATA "#" che serve al programma per contare le frasi. Le parole o le frasi possono essere scritte in qualsiasi lingua ed il loro numero può essere anche di alcune centinaia ed è limitato solo dalla capacità di memoria dello Spectrum (lo spectrum 16K può contenere oltre 500 parole e/o frasi di 10/15 caratteri ciascuna).

Naturalmente è bene che la persona che sceglie ed introduce le frasi sia diversa da quella che deve usare il programma, e questo per evitare che il test di lettura veloce si trasformi in realtà in un test di memoria (nel senso di ricordare le frasi che si sono introdotte).

Inserite le frasi si può far girare il programma. La prima frase viene sempre visualizzata per un tempo di 0.1 secondi che viene indicato in anticipo (Figura 60), quindi esce la frase scelta casualmente (Figura 61). Sparita la frase il computer chiede di scriverla (Figura 62), a questo punto si deve inserire la frase esattamente (rispettando anche gli spazi e i caratteri maiuscoli o minuscoli).

Se la frase inserita è uguale a quella scelta dal Computer esce la scritta di Figura 63, se invece è sbagliata esce la scritta di Figura 64.

Inoltre ogni volta che si riconosce esattamente una frase il tempo di visualizzazione della successiva viene dimezzato (fino ad un minimo di 0.02 sec.), mentre ogni volta che si sbaglia viene raddoppiato (fino ad un massimo di 1 sec.).

```
10 REM LETTURA VELOCE
20 REM -----
100 BORDER 0: PAPER 0: INK 6: C
LS
110 LET b$="" : REM 32 spazi
120 LET k$=b#+b#+b$
130 PRINT ; PAPER 1; k$
140 FOR a=1 TO 10000
150 RESTORE 1000+a
160 READ c$: IF c$="#" THEN GO
TO 180
170 NEXT a
180 LET nf=a-1
```

Figura 59 (continua)

GREEN

Durata Frase 0.2 Sec.

Figura 61

Scrivete la frase che avete
Visto e premete **ENTER**

Figura 62

Frase: **ESATTA**

Figura 63

Frase: **SBAGLIATA !!**

Figura 64

SCRITTE ROTANTI

Una delle applicazioni del programma 'Scritte Rotanti' (Figura 67) può essere nella pubblicità e nell'informazione in genere.

Dopo avere premuto RUN e ENTER si inserisce prima la frase (anche di molti caratteri) e poi la velocità di rotazione, dopodichè i caratteri, i numeri, ecc. della frase inserita inizieranno a ruotare da destra a sinistra.

Le Figure 68 e 69, 70 e 71, mostrano due esempi di scritte rotanti ciascuno dei quali visto in due differenti momenti; comunque il modo migliore per rendersi conto del funzionamento è quello di provare il programma in pratica.

Se si vuole rendere stabile la frase in modo da registrarla su nastro insieme al programma si può sostituire la linea 110 INPUT a\$ con la linea 110 LET a\$="frase" inoltre se nella frase si vogliono inserire anche dei caratteri speciali (Figura 72) si può inserire la relativa routine per caricarli in memoria tra la linea 1115 e la linea 2221.

Infine inserendo nel programma le linee di Figura 73 si può fermare la rotazione della frase ogni volta che si preme il tasto 'p' e fino a che non si preme ENTER.

```
10 REM SCRITTE ROTANTI
20 REM -----
99 GO SUB 1111
100 BORDER 0: PAPER 0: CLS
110 INPUT "Inserire frase e pre
mere ENTER ", LINE a$
120 FOR a=1 TO LEN a$
130 IF CODE a$(a)<32 OR CODE a$
(a)>164 THEN LET a$(a)=" "
140 NEXT a
150 INPUT "Inserire velocità (
1--10) e premere ENTER ":t:
IF t<1 OR t>10 THEN GO TO 150
160 FOR a=3 TO 7
170 PRINT AT a,0: PAPER 1:
REM 30 spazi
180 NEXT a
190 IF LEN a$<32 THEN DIM h$(32
-LEN a$+10)
195 IF LEN a$>31 THEN DIM h$(10
)
200 LET b#=a#+h$
210 LET c#=b$
220 PRINT AT 5,0: INK 5:c$(1 TO
30)
230 LET c#=c$(2 TO )
240 IF LEN c$<32 THEN LET c#=c$
+b$
250 PAUSE t*5
```

Figura 67 (continua)

```
260 GO TO 220
1111 REM -----
1112 REM routine per eventuali
1113 REM caratteri speciali
1114 REM -----
2222 RETURN
```

Figura 67



QUESTA E' UNA SCRITTA ROTANTE...

Figura 68



TE.... QUESTA E' UNA SC

Figura 69



US. 8_1495 SCAN._1230 2_2

Figura 70



Figura 71



Figura 72

```
255 IF INKEY$="P" THEN GO TO 27  
260 GO TO 220  
270 INPUT LINE  E $  
280 GO TO 220
```

Figura 73

CONTAPEZZI

Un uso insolito dello Spectrum può essere quello di Contapezzi (Figura 74) capace di contare da 0 a 99.999.999 pezzi.

Facendo girare il programma, ogni volta che si preme ENTER il numero viene incrementato di uno (Figura 75). Oltre a ciò, si può inserire anche un numero qualsiasi da aggiungere (per esempio 334925) che premendo ENTER viene sommato al numero presente sullo schermo oppure sottratto, se il numero inserito è preceduto dal segno '-'.
Se si cerca di sommare o di sottrarre un numero che farebbe scendere il totale sotto lo 0 o lo farebbe salire oltre il massimo, il programma fa uscire sul video la frase 'NUMERO ERRATO O ECCESSIVO'.

```
100 REM          CONTAPPEZZI CH 0 H
200 REM          99.999.999 PEZZI
300 REM          -----
1000 BORDER 0: PAPER 0: INK 6: C
L5
110 PRINT INK 4;"Premere ENTER
per 1 solo pezzo"
120 PRINT
130 PRINT INK 4;"o introdurre N
pezzi in piu' in"
140 PRINT
150 PRINT INK 4;"meno (con -) e
premere ENTER"
160 PRINT AT 10,0; INK 5;"L I I I
170 PRINT AT 11,0; INK 5;"I I I
180 PRINT AT 12,0; INK 5;"I I I
190 LET n=0
200 LET c=0
210 INPUT LINE n$
220 IF n$="" THEN LET n$="1"
230 IF CODE n$<>45 THEN FOR a=1
TO LEN n$
240 IF CODE n$=45 THEN FOR a=2
TO LEN n$
250 IF CODE n$(a)<48 OR CODE n$(
a)>57 THEN GO TO 600
260 NEXT a
270 IF (n+VAL n$)>(128-1) OR (n
+VAL n$)<0 THEN GO TO 600
280 LET n=n+VAL n$
290 DIM b$(1,0-LEN STR$ n)
300 LET b$=STR$ n+b$(1)
310 FOR p=1 TO LEN b$-1
320 LET c=p*2+14
```

Figura 74 (continua)

Premere **ENTER** per 1 solo pezzo
o introdurre N° pezzi in più o in
meno (con -) e premere **ENTER**

N°PEZZI 53774923

Figura 75

SIMULATORE DI PORTE LOGICHE

Il programma (Figura 76) simula il funzionamento delle sei porte logiche che sono alla base di qualsiasi circuito digitale incluso i Computer. Inoltre lo stesso microprocessore Z80 usato dallo Spectrum ha nel suo set di istruzioni, tre istruzioni logiche (AND, OR, XOR) il cui funzionamento è simile a quello delle corrispondenti porte digitali.

La scelta delle porte e la selezione di '1' e '0' ai due ingressi 'A' e 'B' avviene premendo i seguenti tasti:

Tasto	Funzione
1	ingresso 'A' a '1'
2	ingresso 'A' a '0'
3	ingresso 'B' a '1'
4	ingresso 'B' a '0'
5	porta AND
6	porta NAND
7	porta OR
8	porta NOR (Figura 77)
9	porta EX-OR
0	porta EX-NOR


```

445000 PRINT AT 1,13; "NOR"
445000 PLOT 112,110: DRAW 38,20,PI/3
445000 DRAW 0,40,PI/3
445000 PLOT 104,112: DRAW 0,40,PI/3
445000 GO TO 1000
445000 LAMP P=5
445000 FOR L=1 TO 7: PRINT AT L,11
445000 : NEXT L
445000 PRINT AT 1,13; "EX-OR"
445000 PLOT 112,110: DRAW 38,20,PI/3
445000 DRAW 0,40,PI/3
445000 PLOT 104,112: DRAW 0,40,PI/3
445000 GO TO 1000
445000 PLOT 96,143: DRAW 19,0
445000 PLOT 96,119: DRAW 19,0
445000 PLOT 151,132: DRAW 16,0
445000 GO TO 1000

```

Figura 76

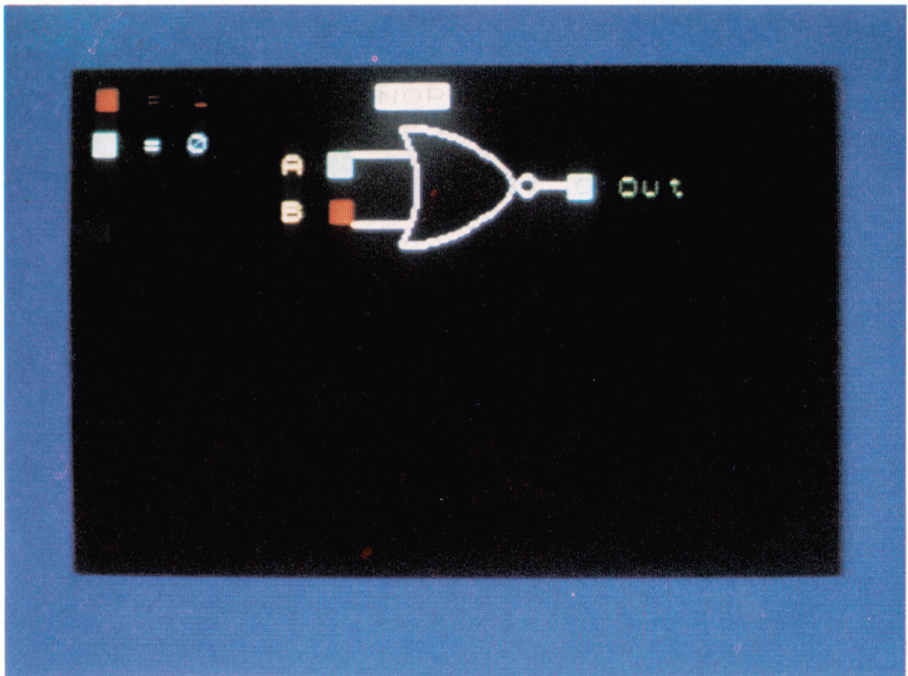


Figura 77

PROGETTAZIONE DI MULTIVIBRATORE MONOSTABILE CON 555

Una delle applicazioni più interessanti del Computer è nella progettazione di circuiti elettronici. Un esempio pratico è il programma di Figura 78 che permette di calcolare i valori di un Multivibratore Monostabile con il timer 555.

Durante il funzionamento il programma chiede prima il calcolo che si desidera effettuare (Figura 79), quindi dopo aver inserito i dati, fornisce il progetto completo (Figura 80); infine premendo il tasto 'C' si può avere anche una copia su carta del circuito.

Per quanto riguarda i valori di Ra, C, t inseriti, questi possono avere la seguente escursione minima e massima:

Ra	da 1kΩ a 10000kΩ
C	da 0.00001μF a 1000μF
t	da 0.01ms a 10000ms

```

100 REM
200 REM
300 REM
400 REM
500 REM
600 REM
700 REM
800 REM
900 REM
1000 REM
1100 REM
1200 REM
1300 REM
1400 REM
1500 REM
1600 REM
1700 REM
1800 REM
1900 REM
2000 REM
2100 REM
2200 REM
2300 REM
2400 REM
2500 REM
2600 REM
2700 REM
2800 REM
2900 REM
3000 REM
3100 REM
3200 REM
3300 REM
3400 REM
3500 REM
3600 REM
3700 REM
3800 REM
3900 REM
4000 REM
4100 REM
4200 REM
4300 REM
4400 REM
4500 REM
4600 REM
4700 REM
4800 REM
4900 REM
5000 REM
5100 REM
5200 REM
5300 REM
5400 REM
5500 REM
5600 REM
5700 REM
5800 REM
5900 REM
6000 REM
6100 REM
6200 REM
6300 REM
6400 REM
6500 REM
6600 REM
6700 REM
6800 REM
6900 REM
7000 REM
7100 REM
7200 REM
7300 REM
7400 REM
7500 REM
7600 REM
7700 REM
7800 REM
7900 REM
8000 REM
8100 REM
8200 REM
8300 REM
8400 REM
8500 REM
8600 REM
8700 REM
8800 REM
8900 REM
9000 REM
9100 REM
9200 REM
9300 REM
9400 REM
9500 REM
9600 REM
9700 REM
9800 REM
9900 REM
10000 REM

```

Figura 78 (continua)

```

T 000000000000
D 0004000000
C 0000000000
0100000000
0200000000
0300000000
0400000000
0500000000
0600000000
0700000000
0800000000
0900000000
1000000000
1100000000
1200000000
1300000000
1400000000
1500000000
1600000000
1700000000
1800000000
1900000000
2000000000
2100000000
2200000000
2300000000
2400000000
2500000000
2600000000
2700000000
2800000000
2900000000
3000000000
3100000000
3200000000
3300000000
3400000000
3500000000
3600000000
3700000000
3800000000
3900000000
4000000000
4100000000
4200000000
4300000000
4400000000
4500000000
4600000000
4700000000
4800000000
4900000000
5000000000
5100000000
5200000000
5300000000
5400000000
5500000000
5600000000
5700000000
5800000000
5900000000
6000000000
6100000000
6200000000
6300000000
6400000000
6500000000
6600000000
6700000000
6800000000
6900000000
7000000000
7100000000
7200000000
7300000000
7400000000
7500000000
7600000000
7700000000
7800000000
7900000000
8000000000
8100000000
8200000000
8300000000
8400000000
8500000000
8600000000
8700000000
8800000000
8900000000
9000000000
9100000000
9200000000
9300000000
9400000000
9500000000
9600000000
9700000000
9800000000
9900000000

A J: 0004000000 INK 5; "Pre
0000000000 PRINT AT 1,10; "Monostabile
0000000000 PRINT SUB 4000
0004000000 PRINT AT 10,0; INK 5; "
0007000000 PRINT AT 10,0; INK 6; " il t
0010000000 PRINT AT 10,0; INK 6; " il t
0013000000 PRINT AT 10,0; INK 6; " il t
0016000000 PRINT AT 10,0; INK 3; "test
0019000000 LET K=CODE INKEY$
0022000000 IF K=67 OR K=99 THEN GO TO
0025000000
0028000000 IF K>40 AND K<50 THEN GO TO
0031000000 *100
0034000000 GO TO 340
0037000000 PRINT AT 10,0; "Took care Pa
0040000000 PRINT AT 10,0; "
0043000000 PRINT AT 10,0; "
0046000000 PRINT AT 10,0; "
0049000000 PRINT AT 10,0; "
0052000000 PRINT AT 10,0; "
0055000000 PRINT AT 10,0; "
0058000000 PRINT AT 10,0; "
0061000000 PRINT AT 10,0; "
0064000000 PRINT AT 10,0; "
0067000000 PRINT AT 10,0; "
0070000000 PRINT AT 10,0; "
0073000000 PRINT AT 10,0; "
0076000000 PRINT AT 10,0; "
0079000000 PRINT AT 10,0; "
0082000000 PRINT AT 10,0; "
0085000000 PRINT AT 10,0; "
0088000000 PRINT AT 10,0; "
0091000000 PRINT AT 10,0; "
0094000000 PRINT AT 10,0; "
0097000000 PRINT AT 10,0; "
0100000000 PRINT AT 10,0; "

```

Figura 78 (continua)


```

7701 IF X=1 THEN RETURN
7700 LET X#=" K"+CHR$ 144: LET U
#=" " +CHR$ 145+"T": LET Z#=" m#%
c.
7800 IF Ra>=1000 THEN LET X#=" M
"+CHR$ 144
7800 IF Ra>=1000 THEN LET Ra=Ra/
10000
7800 LET r#=STR$ Ra: IF LEN r#>4
THEN LET r#=r$(1 TO 4)
7810 IF C<1 THEN LET U#=" OF"
7810 IF C<1 THEN LET C=C*1000
7810 LET c#=STR$ c: IF LEN c#>4
THEN LET c#=c$(1 TO 4)
7820 IF t>=1000 THEN LET z#=" se
c.
7822 IF t>=1000 THEN LET t=t/100
0
7827 LET t#=STR$ t: IF LEN t#>4
THEN LET t#=t$(1 TO 4)
7830 PRINT AT 5,1;" ";AT 1
2,1;" ";AT 11,22;"
7830 PRINT AT 5,5-LEN r#;r#;X#
7840 PRINT AT 12,5-LEN c#;c#;U#
7850 PRINT AT 11,22;t#;Z#
8800 RETURN

```

Figura 78

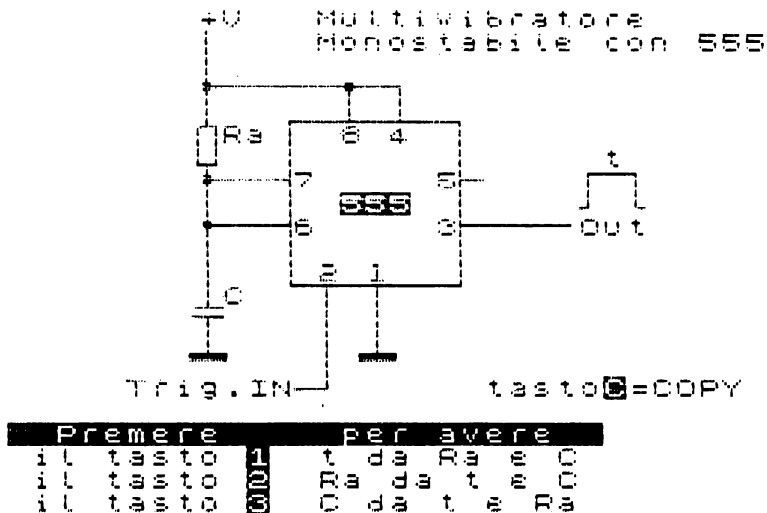


Figura 79

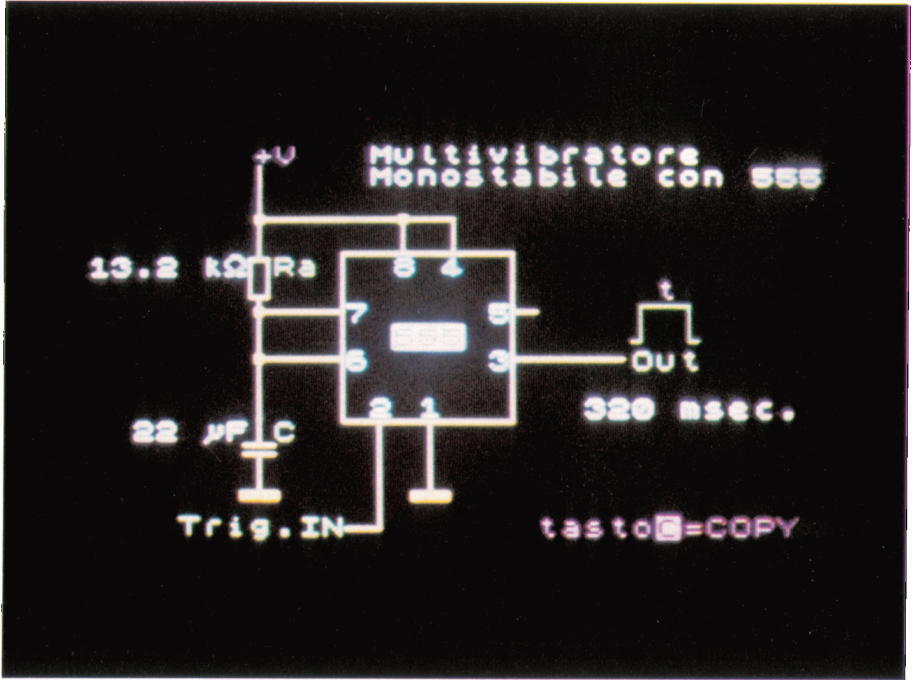


Figura 80

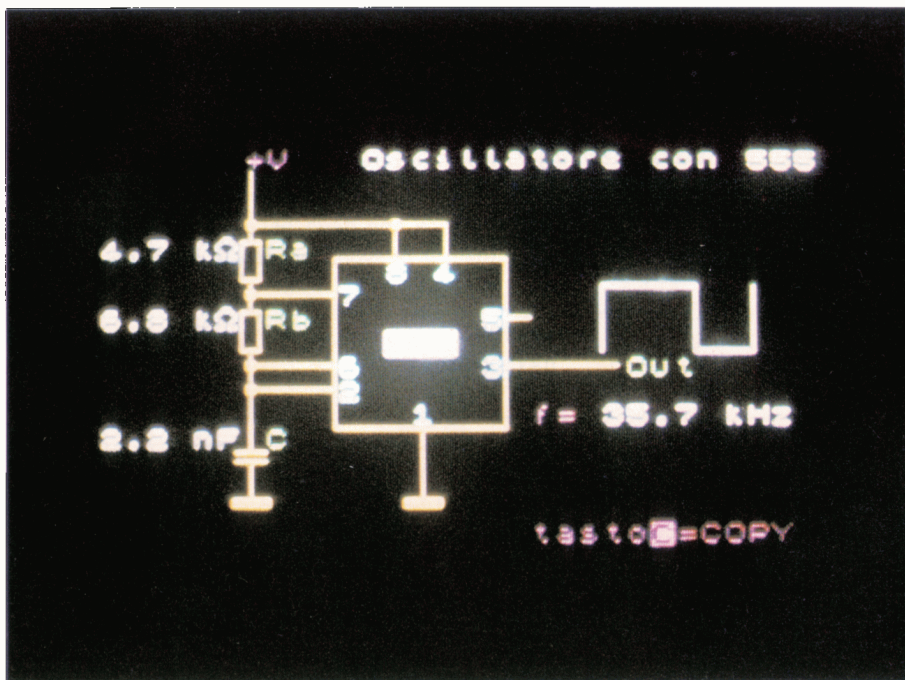


Figura 83

LEGGE DI OHM

Il terzo programma di progettazione elettronica, ed ultimo di questo Capitolo, è soprattutto un programma didattico sulla legge di OHM (Figura 84). Inserendo infatti i dati come vengono richiesti dal computer si ottiene un risultato come quello di Figura 85.

Da notare che il computer chiede sempre tutti e tre i dati e che bisogna perciò inserire 0 quando il computer chiede il valore che deve essere calcolato.

Le escursioni dei valori di V, R, I inseriti possono essere le seguenti:

V	da 0.000001V a 1000000V
R	da 0.001 Ω a 1000 M Ω
I	da 0.000001A a 1000A

```

100 REM      LEGGE DI OHM
200 REM      -----
300 BORDER 0: PAPER 0: INK 0: 0
L5
100 LET U=PEEK 23675+256*PEEK 2
300
110 DATA 50,60,100,100,100,40,20
0,0,10,254,1000,100,100,100,100,1
0,100,100,100,100,100,100,254,1
000
120 FOR a=U TO U+20
130 READ d: DOKE a: NEXT a
140 PLOT 0,100: DRAW 0,16: DRAW
150 0,10: DRAW 0,140: DRAW 100,0:
UC
1500 PRINT AT 5,17:CHR# 145:AT 0
1510 PRINT AT 5,17:CHR# 145
1520 PLOT 1,1: DRAW 1,10
1530 PLOT 1,1: DRAW 1,10
1540 PLOT 1,1: DRAW 1,10
1550 PLOT 1,1: DRAW 1,10
1560 PLOT 1,1: DRAW 1,10
1570 PLOT 1,1: DRAW 1,10
1580 PLOT 1,1: DRAW 1,10
1590 PLOT 1,1: DRAW 1,10
1600 PLOT 1,1: DRAW 1,10
1610 PLOT 1,1: DRAW 1,10
1620 PLOT 1,1: DRAW 1,10
1630 PLOT 1,1: DRAW 1,10
1640 PLOT 1,1: DRAW 1,10
1650 PLOT 1,1: DRAW 1,10
1660 PLOT 1,1: DRAW 1,10
1670 PLOT 1,1: DRAW 1,10
1680 PLOT 1,1: DRAW 1,10
1690 PLOT 1,1: DRAW 1,10
1700 PLOT 1,1: DRAW 1,10
1710 PLOT 1,1: DRAW 1,10
1720 PLOT 1,1: DRAW 1,10
1730 PLOT 1,1: DRAW 1,10
1740 PLOT 1,1: DRAW 1,10
1750 PLOT 1,1: DRAW 1,10
1760 PLOT 1,1: DRAW 1,10
1770 PLOT 1,1: DRAW 1,10
1780 PLOT 1,1: DRAW 1,10
1790 PLOT 1,1: DRAW 1,10
1800 PLOT 1,1: DRAW 1,10
1810 PLOT 1,1: DRAW 1,10
1820 PLOT 1,1: DRAW 1,10
1830 PLOT 1,1: DRAW 1,10
1840 PLOT 1,1: DRAW 1,10
1850 PLOT 1,1: DRAW 1,10
1860 PLOT 1,1: DRAW 1,10
1870 PLOT 1,1: DRAW 1,10
1880 PLOT 1,1: DRAW 1,10
1890 PLOT 1,1: DRAW 1,10
1900 PLOT 1,1: DRAW 1,10
1910 PLOT 1,1: DRAW 1,10
1920 PLOT 1,1: DRAW 1,10
1930 PLOT 1,1: DRAW 1,10
1940 PLOT 1,1: DRAW 1,10
1950 PLOT 1,1: DRAW 1,10
1960 PLOT 1,1: DRAW 1,10
1970 PLOT 1,1: DRAW 1,10
1980 PLOT 1,1: DRAW 1,10
1990 PLOT 1,1: DRAW 1,10
2000 PRINT AT 5,18: "U:"
2010 PRINT AT 1,19: "I:"
2020 PRINT AT 1,20: "R:"
2030 PRINT AT 0,21: "U_:"
2040 INK 7: BRIGHT 0
2050 LET c=0

```

Figura 84 (continua)

```

0310 INPUT "Inserire U (in Volt)
oppure 0 se preferisci ENTER ";v
0311 IF v=0 THEN GO TO 0320
0315 IF v<1E-6 OR v>1E6 THEN GO
TO 0310
0320 INPUT "Inserire I (in Amper)
oppure 0 se preferisci ENTER ";i
0321 IF i=0 THEN GO TO 0330
0325 IF i<1E-6 OR i>1E3 THEN GO
TO 0320
0330 INPUT "Inserire R (in Ohm)
oppure 0 se preferisci ENTER ";r
0331 IF r=0 THEN GO TO 0340
0335 IF r<1E-3 OR r>1E9 THEN GO
TO 0330
0340 IF v<>0 THEN LET c=c+1
0341 IF i<>0 THEN LET c=c+1
0342 IF r<>0 THEN LET c=c+1
0343 IF v<>0 THEN GO TO 0777
0344 IF i<>0 THEN LET v=i*r
0345 IF r<>0 THEN LET r=v/i
0346 IF v<1E-6 OR v>1E6 THEN GO
TO 0330
0347 IF i<1E-6 OR i>1E3 THEN GO
TO 0330
0348 IF r<1E-3 OR r>1E9 THEN GO
TO 0330
0420 LET x$="U": LET y$=CHR$ 144
:
0430 LET z$="R":
0440 IF v<1 THEN LET x$="BU"
0450 IF v<10000 THEN LET x$="KU"
0460 IF v>10000 THEN LET x$=v/100
0
0470 IF r>=1E6 THEN LET y$="M"+C
IR# 144
0480 IF r>=1E6 THEN LET r=r/1E6
0490 IF r>=1000 THEN LET y$="K"+
OI# 144
0500 IF r>=1000 THEN LET r=r/100
0
0510 IF i<1 THEN LET z$="BR"
0520 IF i<1 THEN LET i=i*1000
TO 0530 LET v$=STR$(v): IF LEN v$>5
TO 0540 LET r$=STR$(r): IF LEN r$>5
TO 0550 LET i$=STR$(i): IF LEN i$>5
TO 0560 LET z$=z$(1 TO 5)
0570 PRINT AT 0,00;v$; INK 4;x$
:
0580 PRINT AT 4,00;r$; INK 4;y$

```

Figura 84 (continua)

```

5990 PRINT AT 6,23;i$; INK 4;z$
6000 GO TO 300
7777 PRINT AT 11,7;"VALORI ERRAT
I"
8888 PAUSE 100: PRINT AT 11,7;"
9999 GO TO 300

```

Figura 84

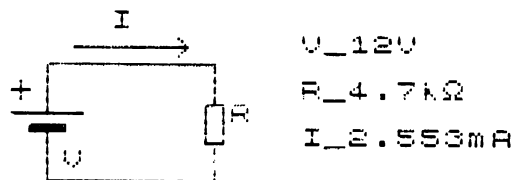


Figura 85

UTILITY

ROUTINE 'PAINT' IN BASIC

Una istruzione di cui si sente la mancanza quando si usa l'alta risoluzione grafica dello Spectrum per disegnare, è l'istruzione 'PAINT' (o FILL su altri computer) che serve per colorare l'interno di figure chiuse senza dover inventare ogni volta una routine apposita. La routine PAINT (Figura 86) può essere usata a tale scopo; basta infatti inserire nelle variabili 'xp' ed 'yp' le coordinate di un punto all'interno della figura da colorare (il più possibile al centro) e nella variabile 'ip' il colore desiderato, e quindi chiamare la routine con l'istruzione 'GOSUB 7000'.

Naturalmente se la figura da colorare è molto complessa oppure rimangono degli spazi non colorati, è possibile indirizzare più punti all'interno della figura e chiamare altrettante volte la routine.

È importante che all'interno della figura non vi siano già delle scritte o altro che possa fermare il pennello della routine, inoltre nel programma non devono essere utilizzate le variabili 'xp', 'yp', 'ip', 'o' e 'q' che vanno riservate alla routine stessa.

In Figura 86 insieme alla routine è compreso anche un breve programma esempio che disegna due figure (Figura 87) e le colora (Figura 88) mostrando in pratica il funzionamento di 'PAINT'.

```
10 REM PROGRAMMA ESEMPIO
20 BORDER 0: PAPER 0: CLS
30 INK 7: PLOT 0,0: DRAW 255,0
: DRAW 0,175: DRAW -255,0: DRAW
0,-175
40 REM CERCHIO azzurro
50 INK 5: CIRCLE 50,50,30
60 LET ip=5: LET xp=50: LET yp
=50: GO SUB 7000
70 REM QUADRATO giallo
80 INK 6: PLOT 150,100: DRAW 4
5,0: DRAW 0,45: DRAW -45,0: DRAW
0,-45
90 LET ip=6: LET xp=170: LET yp
=120: GO SUB 7000
1000 STOP
9999 REM -----
7000 REM routine PAINT in Basic
```

Figura 86 (continua)

```

7000 INK IP
70010 FOR q=xp TO 255
70020 FOR o=yf TO 175
70030 PLOT q,o
70040 IF POINT (q,o+1)=0 THEN NEX
T
70050 IF POINT (q+1,yf)=0 THEN NE
XT q
70060 FOR q=xp TO 255
70070 FOR o=yf-1 TO -175 STEP -1
70080 PLOT q,o
70090 IF POINT (q,o-1)=0 THEN NEX
T
70100 IF POINT (q+1,yf-1)=0 THEN
NEXT q
70110 FOR q=xp-1 TO -255 STEP -1
70120 FOR o=yf TO 175
70130 PLOT q,o
70140 IF POINT (q,o+1)=0 THEN NEX
T
70150 IF POINT (q-1,yf)=0 THEN NE
XT q
70160 FOR q=xp-1 TO -255 STEP -1
70170 FOR o=yf-1 TO -175 STEP -1
70180 PLOT q,o
70190 IF POINT (q,o-1)=0 THEN NEX
T
70200 IF POINT (q-1,yf-1)=0 THEN
NEXT q
70210 RETURN

```

Figura 86

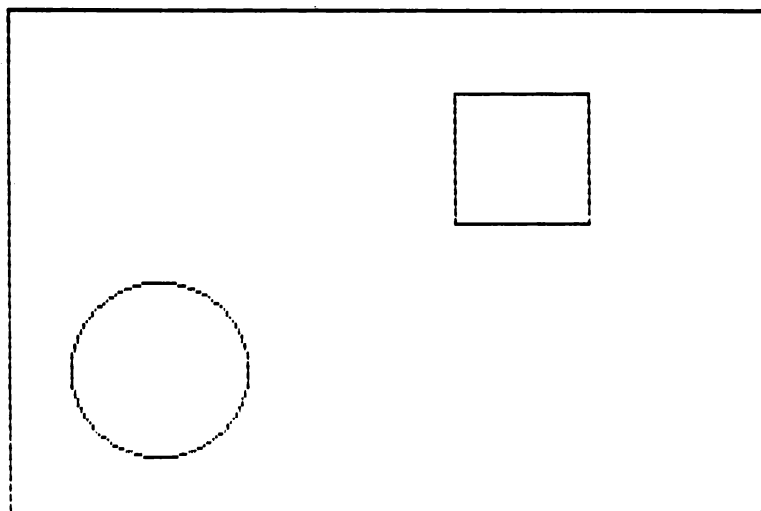


Figura 87

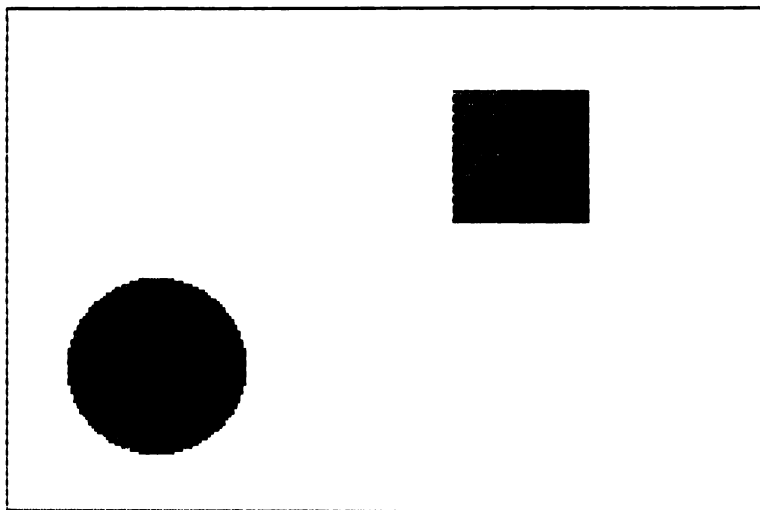


Figura 88

NOTE SULLE ROUTINES 'SCROLL'

Le prossime quattro utilities sono le routines 'SCROLL'.

Si tratta di quattro routines in linguaggio macchina, le quali, ogni volta che vengono chiamate da una istruzione 'USR' spostano di un carattere tutto il contenuto dello schermo rispettivamente verso sinistra, verso destra, verso l'alto o verso il basso.

Tali routines sono molto utili, inserite in appositi programmi, per estendere le dimensioni virtuali dello schermo, che diventa una finestra su uno schermo più ampio non visibile il quale può avere anche per esempio 50 righe per 80 colonne.

Altri usi sono per spostare in un dato senso una serie di dati o delle immagini (una versione modificata degli Scroll a sinistra ed a destra è usata per esempio nel gioco 'Tiro alla fune', in questo libro, per spostare rapidamente tutta la parte superiore dello schermo a destra o a sinistra secondo le necessità del programma).

Da notare che lo Spectrum ha già lo Scroll verso l'alto che però, a differenza della routine in linguaggio macchina, si blocca ogni 22 linee visualizzando la scritta 'scroll?' che chiede se lo scroll deve continuare.

Importante: in tutte e quattro le routines (che comprendono anche un esempio di funzionamento) le istruzioni in linguaggio macchina sono caricate nella zona di memoria riservata ai caratteri speciali, naturalmente se tale zona serve al programma utilizzatore, la o le routines possono essere memorizzate tra la fine dello stack e l'inizio della zona degli UDG usando il comando CLEAR nel modo indicato al capitolo 26 del manuale dello Spectrum.

SCROLL A SINISTRA

La prima routine (Figura 89) che è lunga 24 Bytes, scrolla tutto lo schermo verso sinistra di un carattere ogni volta che il programma incontra un'istruzione tipo: LET s=USR n dove 'n' è l'indirizzo della RAM dove comincia la routine e che nel programma esempio è (u+144).

```
66 REM ROUT. SCROLL A SINISTRA
77 REM -----
99 BORDER 0: PAPER 0: INK 6: C
Ls
100 DATA 6,192,33,1,64,229,209,
27,197,1,31,0,237,176,43,54,0,19
,35,35,193,16,241,201
104 LET U=PEEK 23675+256*PEEK 2
3676
110 FOR a=U+144 TO U+167
120 READ d: POKE a,d: NEXT a
999 REM -----

1000 REM      programma esempio
1010 REM      -----
1015 FOR x=1 TO 100
1020 FOR a=0 TO 359
1030 PRINT AT 10+9*SIN (PI/9*a),
31; ". :
1040 LET s=USR (U+144)
1044 PAUSE S
1050 NEXT a
1060 NEXT x
```

Figura 89

SCROLL A DESTRA

Anche questa routine (Figura 90) è lunga 24 bytes e scrolla tutto lo schermo verso destra di un carattere ad ogni istruzione tipo LET s=USR n dove 'n' è di nuovo l'indirizzo di partenza della routine nella RAM, nell'esempio quindi (u+144).

```

      66 REM      ROUT. SCROLL A DESTRA
      77 REM      -----
      99 BORDER 0: PAPER 0: INK 7: C
L 6
100 DATA 6,190,33,254,67,229,20
101 197,1,31,0,207,104,36,64,0,
102 43,43,103,10,241,201
103 LET U=PEEK 23675+256*PEEK 2
104 FOR a=U+144 TO U+167
105 READ d: POKE a,d: NEXT a
106 REM      -----

1000 REM      programma esempio
1010 REM      -----
1011 LET t=1
1015 FOR x=1 TO 100
1020 FOR a=0 TO 21
1030 IF t=1 THEN PRINT AT a,0;".
1033 IF t=-1 THEN PRINT AT 21-a,
0;".
1040 LET s=USR (U+144)
1044 REUS E s
1050 NEXT a
1055 LET t=t*-1
1060 NEXT x
```

Figura 90

RENUMBER

Un programma di Renumber è molto utile nella programmazione per riordinare automaticamente tutti i numeri di linea appena cominciano ad essere un po' confusi.

La versione di Figura 93 realizza tale riordino escluso i numeri dopo GOTO e GO-SUB che vanno modificati singolarmente.

Il programma va fatto girare con un RUN 9020 o con un GOTO 9020. Dopo di ciò appariranno le scritte in Input che chiedono l'inserimento prima del numero da cui iniziare la rinumerazione (per esempio 3), poi del nuovo numero di linea iniziale (per esempio 100) ed infine della distanza desiderata tra linea e linea (per esempio 10). Naturalmente è possibile effettuare anche più di una rinumerazione per esempio per avere le prime 16 linee rinumerate con 100, 110, 120, ecc... e le altre 27 linee rinumerate con 1000, 1050, 1100, ecc.

La Figura 94 mostra come esempio un programma che ha bisogno di essere rinumerato, mentre la Figura 95 mostra una delle possibili rinumerazioni dello stesso ottenuta con il programma 'Renumber' appena descritto.

Per comodità il Renumber può essere memorizzato su una cassetta con il comando SAVE "renumber" e caricato in memoria quando occorre tramite il comando MERGE "renumber" che consente di inserire il Renumber nella RAM senza cancellare il programma che state scrivendo. Naturalmente per usare il Renumber il programma da rinumerare non deve usare le linee da 9000 a 9140, inoltre vanno riservate al funzionamento del Renumber le variabili 'im', 'nir', 'nli', 'dtl', 'nx'.

```
00000 REM          RENUMBER
00010 STOP
00020 INPUT "Inserire N° linea ini
nio renumber",,nir
00030 INPUT "Inserire nuovo N° lin
ea iniziale",,nli
00040 INPUT "Inserire distanza tr
a linee",,dtl
00050 LET im=PEEK 23635+256*PEEK
00060 -1
00060 LET nx=PEEK (im+1)*256+PEEK
(im+2)
00070 IF nx<nir THEN GO TO 9120
00080 IF nx=9000 THEN GO TO 9140
00090 POKE (im+1),INT (nli/256)
01000 POKE (im+2),nli-256*INT (nli
/256)
01100 LET nli=nli+dtl
01200 LET im=im+4+PEEK (im+3)+256
*PEEK (im+4)
01300 GO TO 9060
01400 LIST
```

Figura 93

LETTORE MEMORIA RAM/ROM

Un'altra utility interessante è il lettore della memoria del Computer (Figura 96) con il quale si può leggere il contenuto di una parte qualsiasi della ROM o della RAM a partire da un indirizzo inserito all'inizio.

La Figura 97 mostra un esempio di lettura della RAM ed in particolare proprio di una zona nella quale è scritto il programma di Figura 96; come si vede il lettore memoria visualizza l'indirizzo, il contenuto decimale ed il corrispondente carattere di ciascun Byte esaminato. Se il carattere ha un codice inferiore a 32 esce la scritta '—', mentre, se il carattere è uno di quelli grafici definibili, viene visualizzata la lettera corrispondente preceduta da una 'G' inversa.

Il programma ha un funzionamento continuo premendo ENTER ogni volta che esce la scritta 'scroll?' e può essere fermato con un Break. Un'altra possibilità è quella di numerare il programma per esempio con i numeri da 8000 a 8140 in modo da caricarlo da una cassetta nella RAM con un comando 'MERGE' insieme ad un altro programma che si vuole esaminare; in questo caso il lettore memoria va fatto girare con un RUN 8000 o con un GOTO 8000.

Le variabili riservate al funzionamento di questo programma sono: 'ip', 'w\$', 'w'.

```
10 REM LETTORE MEMORIA RAM/ROM
20 REM -----
99 BORDER 0: PAPER 0: INK 7: C
LS
100 INPUT "Inserire indirizzo d
i partenza ( 0--65535 ) e preme
re ENTER",ip
110 IF ip<0 OR ip>65535 THEN GO
TO 100
120 FOR w=ip TO 65535-ip
130 IF w=ip OR w-INT (w/10)*10=
0 THEN PRINT " Indirizzo Conten
uto Carattere"
140 LET w$=" " + STR$ w + "
"
150 PRINT w$(1 TO 15);
160 LET w$=STR$ PEEK w + "
"
170 PRINT w$(1 TO 8);
180 IF PEEK w>31 AND PEEK w<144
OR PEEK w>164 THEN PRINT CHR$ P
EEK w
188 IF PEEK w>143 AND PEEK w<16
5 THEN PRINT "G ";CHR$ (PEEK w-7
9)
190 IF PEEK w<32 THEN PRINT "--
"
200 NEXT w
```

Figura 96

CONVERSIONE DECIMALE/ESADECIMALE ED INVERSO

Spesso le routines in linguaggio macchina usate in alcuni programmi vengono caricate in memoria a partire da stringhe o istruzioni DATA che contengono i loro codici scritti in esadecimale.

In questo ed in molti altri casi è comodo disporre di un programma come quello di Figura 98 che converte numeri decimali in esadecimali e viceversa. Il programma può convertire numeri decimali da 0 a 16777215 e numeri esadecimali da 0 a FFFFFFF con una capacità quindi di 24 Bit sufficiente perciò per essere usato anche nella programmazione dei più recenti microprocessori a 16 Bit.

```

10 REM ██████████ CONVERSIONE DECIMALE /
200 REM ██████████ ESADECIMALE E INVERSO ██████████
300 REM ██████████
99 BORDER 0: PAPER 0: INK 7: C
Ls
100 PRINT AT 0,0;"tasto [D] per i
nserire N° Decimale"
110 PRINT AT 0,0;"tasto [H] per i
ns. N° Esadecimale"
120 LET t=CODE INKEY$
130 IF t=80 OR t=100 THEN GO TO
150
140 IF t=72 OR t=104 THEN GO TO
270
150 GO TO 120
160 INPUT "Inserire N° Decim. (0.
/16777215.) e premere [ENTER] ";
d
170 LET f=d
180 IF d<0 OR d>(2↑24-1) THEN G
O TO 160
190 LET c$="0123456789ABCDEF"
200 LET h$=""
210 FOR a=0 TO 0 STEP -1
220 LET h#=h#+c$(1+INT (f/16↑a))
230 LET f=f-(INT (f/16↑a)*16↑a)
240 NEXT a
241 LET s=1
242 FOR a=1 TO LEN h$
244 IF h$(a)="0" THEN LET s=s+1
246 IF CODE h$(a)>48 THEN GO TO
290
248 NEXT a
250 LET h#=h$(s TO ): GO TO 370
270 INPUT "Inserire N° Esadecim.
(0./FFFFFF.) e premere [ENTER] ";
LINE h$

```

Figura 98 (continua)

```

000000 IF LEN Y# > 6 THEN GO TO 270
000000 LET X# = Y#
000000 IF HTOR X# = 1 THEN GO TO 270
000000 FOR a = 1 TO LEN X#
000000 IF (a) < 103 THENZ LET X#(a) = CHR# ((C
000000 X#(a)) - 32)
000000 IF (a) < 71 THENZ LET X#(a) = CHR# ((C
000000 X#(a)) - 7)
000000 IF CODE X#(a) < 48 OR CODE X#
(a) > 63 THENZ GO TO 270
000000 NEXT a
000000 LET d = 0
000000 FOR a = 1 TO LEN X#
000000 LET d = d + (CODE X#(a) - 48) * 16↑
(a)
000000 NEXT a
000000 PRINT AT 6, 0; INK 6; "N" Dec
i 6, d
000000 PRINT AT 6, 0; INK 6; "N" Ess
d 6, d
000000 GO TO 100

```

Figura 98

ANIMAZIONI

VIAGGIO SPAZIALE

La più bella delle animazioni è senzaltro il Viaggio Spaziale (Figura 99) che simula appunto il viaggio di una astronave nello spazio con le stelle che dal centro dell'immagine fuggono verso l'esterno. Prima di ottenere l'effetto occorre aspettare circa venti minuti durante i quali il programma esegue una serie di operazioni che servono al funzionamento poi dell'animazione. Importante: Per settare correttamente gli attributi di colore il programma funziona con INK 0 e PAPER 0 durante l'esecuzione delle linee da 500 a 1000 per cui se fermate il programma in questa fase può sembrare che si sia cancellato, in realtà il programma c'è e basta eseguire in modo diretto l'istruzione 'INK 7' per rivederlo.

```

10 REM
20 REM
30 BOARD
40 INK 7:
50 CLEAR
60 DATA 33,36,117,17,10,88,1,10
70 17,19,20,21,23,27,33,217,10,200
80 17,19,20,21,23,27,33,217,10,200
90 FOR a=300 TO 3000 STEP 4
100 READ d: POKE a,d: NEXT a
110 LET attr=255
120 GO TO 500
130 INK 7:
140 FOR a=0 TO 360 STEP 4
150 LET k=PI/180*a
160 LET l=10+RND*10
170 FOR s=1 TO 5
180 LET x=100+l*COS k
190 LET y=80+l*SIN k
200 IF x<0 OR x>255 THEN GO TO
210 IF y<0 OR y>175 THEN GO TO
220 IF RND<.2 THEN PLOT INK 6:
230 LET l=l+30
240 NEXT s
250 NEXT a
260 LET k=USR 30001

```

Figura 99 (continua)

```

0010  PAUSE 7
0020  PRINT @K=USR 00013
0030  PRINT @K=7
0040  PRINT @K=USR 00025
0050  PRINT @K=7
0060  GO TO 300
0070  HNZK @: 300
0080  FOR a=0 TO 360 STEP 4
0090  LET k=PI/180*a
0100  LET l=10.5+r*SIN k
0110  LET c=15.5+r*COS k
0120  IF l<0 OR l>21 THEN GO TO 5
0130
0140  IF c<0 OR c>31 THEN GO TO 5
0150
0160  PRINT AT l,c; INK 7; "■"
0170  NEXT a: NEXT r
0180  LET mem=300740
0190  FOR a=0 TO 700
0200  POKE (mem+a),PEEK (attr+a)
0210  NEXT a
0220  CLS
0230  FOR a=0 TO 360 STEP 4
0240  LET k=PI/180*a
0250  LET l=10.5+r*SIN k
0260  LET c=15.5+r*COS k
0270  IF l<0 OR l>21 THEN GO TO 7
0280
0290  IF c<0 OR c>31 THEN GO TO 7
0300
0310  PRINT AT l,c; INK 7; "■"
0320  NEXT a: NEXT r
0330  LET mem=300740
0340  FOR a=0 TO 700
0350  POKE (mem+a),PEEK (attr+a)
0360  NEXT a
0370  CLS
0380  FOR a=0 TO 360 STEP 4
0390  LET k=PI/180*a
0400  LET l=10.5+r*SIN k
0410  LET c=15.5+r*COS k
0420  IF l<0 OR l>21 THEN GO TO 9
0430
0440  IF c<0 OR c>31 THEN GO TO 9
0450
0460  PRINT AT l,c; INK 7; "■"
0470  NEXT a: NEXT r
0480  LET mem=31440
0490  FOR a=0 TO 700
0500  POKE (mem+a),PEEK (attr+a)
0510  NEXT a
0520  CLS
0530  GO TO 100

```

Figura 99


```

350 FOR l=17 TO 0 STEP -1
360 LET g=l+1: LET h=l+2: LET s
=l+3
370 PRINT AT l,18;a$;AT g,18;b$
;AT h,18;c$;AT s,18;" "
380 NEXT l
390 PRINT AT 0,18;b$;AT 1,18;c$
;AT 2,18;" "
400 PRINT AT 0,18;c$;AT 1,18;" "
410 PRINT AT 0,18;" "

```

Figura 100

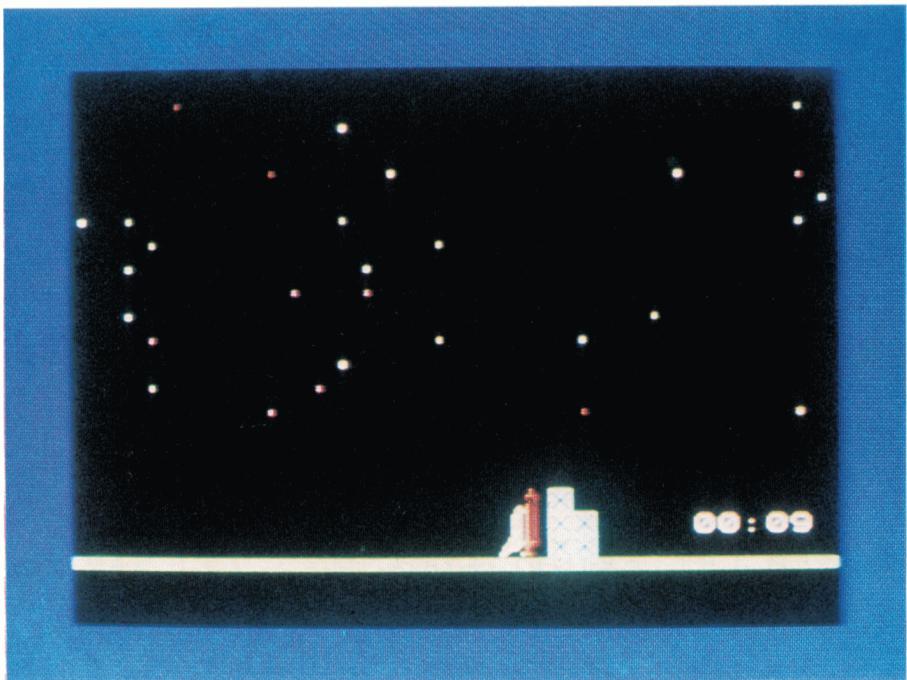


Figura 101

ANIMAZIONE ESPLOSIONE

In Figura 102 il programma che disegna un asteroide (Figura 103) il quale viene colpito e ridotto in tanti pezzi da un missile.

```
10 REM ANIMAZIONE ESPLOSIONE
20 REM -----
100 BORDER 0: PAPER 0: CLS
110 DATA 120,66,49,47,20,95,
50 34,4,34,81,4,41,16,133,136,4,
00 130,20,8,102
7 120,10,4,80,137,18,165,0,1
000,132,10,164,70,30,100,10
1300 LET U=PEEK 03075+0500*PEEK 0
00700
140 FOR a=U TO U+30
150 READ d: POKE a,d: NEXT a
160 LET a#=CHR$ 16+CHR$ 0+CHR$
144
170 LET b#=CHR$ 16+CHR$ 2+CHR$
145+CHR$ 140
180 LET c#=CHR$ 16+CHR$ 2+CHR$
147+CHR$ 140
190 FOR a=0 TO 6 STEP 0.5
200 HX=a: CHRCLE 50,100,a+1
210 CHRCLE 57,100,a
220 NEXT a
230 PRINT AT 21,22);a#
240 FOR a=100
250 PRINT AT a+1,a+2); " ";AT a,a
+1);a#
260 FOR a=1
270 NEXT a
280 PRINT AT 5,6);b#;AT 6,6);c#
290 REPEAT 40
300 PRINT AT 5,6); " ";AT 6,6); "
```

Figura 102



Figura 103

CALEIDOSCOPIO

Nel programma 'Caleidoscopio' (Figura 104) dodici diversi oggetti di colori casuali appaiono e scompaiono continuamente e velocemente sul video, simmetricamente nelle quattro zone in cui è diviso (Figura 105).

```

10 REM          CALEIDOSCOPIO
20 REM
100 BORDER    : PAPER 0 : CLS
110 GOTO 100
120 GOTO 100
130 GOTO 100
140 GOTO 100
150 GOTO 100
160 GOTO 100
170 GOTO 100
180 GOTO 100
190 GOTO 100

```

Figura 104 (continua)

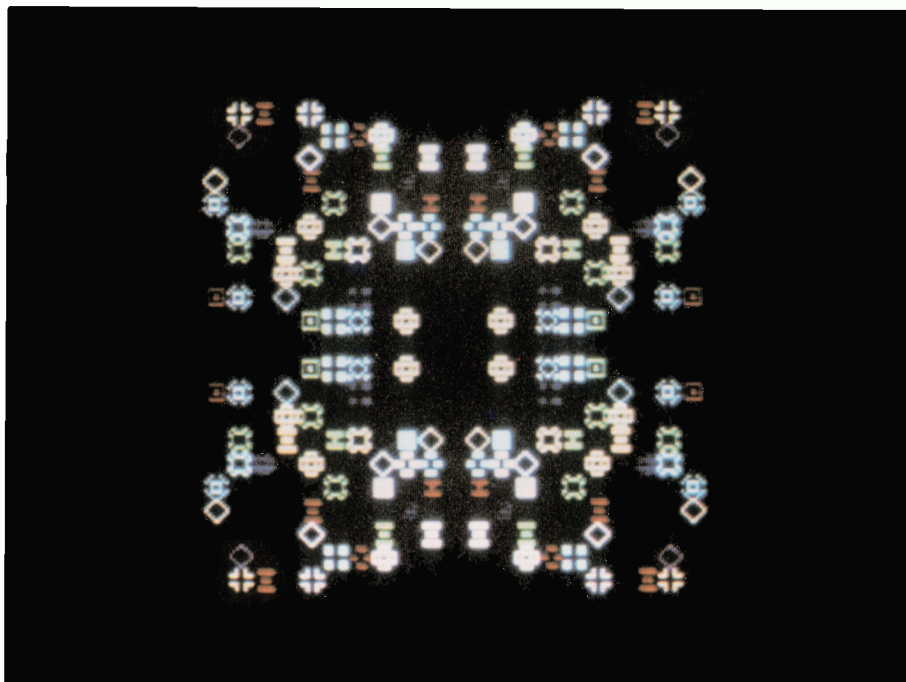


Figura 105

```

010000 DATA 00,00,00,00,1000,1000,1000,
1000,00,1000,1000,1000,00,000,1000,1000,
1000
020000 DATA 00,00,1000,00,000,1000,00
030000 LET U=PEEK 0000700+00000*PEEK 0
040000 FOR a=0 TO U+95
050000 READ d: POKe a,d: NEXT a
060000 RANDOMIZE
070000 LET x=144+INT (RND*30): IF
x>105 THEN LET x=120
080000 LET l=1+INT (RND*10)
090000 LET c=0+INT (RND*10)
100000 INK 1+INT (RND*7)
110000 POK INT AT l,c)CHR$ x:AT l,30
120000 CHR$ x:AT 00-l,00-c:CHR$ x:AT
00-l,c)CHR$ x
130000 GO TO 010

```

Figura 104

ROMBOSPIRALE

Altra animazione è 'Rombospirale' (Figura 106) che disegna e cancella alternativamente in continuazione, una spirale a forma di rombo formata da caratteri e colori casuali (Figura 107).

```

000100 100000 ROMBOSPIRALE
000200 ROMBOSPIRALE
000300 ROMBOSPIRALE
000400 ROMBOSPIRALE
000500 ROMBOSPIRALE
000600 ROMBOSPIRALE
000700 ROMBOSPIRALE
000800 ROMBOSPIRALE
000900 ROMBOSPIRALE
001000 ROMBOSPIRALE
001100 ROMBOSPIRALE
001200 ROMBOSPIRALE
001300 ROMBOSPIRALE
001400 ROMBOSPIRALE
001500 ROMBOSPIRALE
001600 ROMBOSPIRALE
001700 ROMBOSPIRALE
001800 ROMBOSPIRALE
001900 ROMBOSPIRALE
002000 ROMBOSPIRALE
002100 ROMBOSPIRALE
002200 ROMBOSPIRALE
002300 ROMBOSPIRALE
002400 ROMBOSPIRALE
002500 ROMBOSPIRALE
002600 ROMBOSPIRALE
002700 ROMBOSPIRALE
002800 ROMBOSPIRALE
002900 ROMBOSPIRALE
003000 ROMBOSPIRALE
003100 ROMBOSPIRALE
003200 ROMBOSPIRALE
003300 ROMBOSPIRALE
003400 ROMBOSPIRALE
003500 ROMBOSPIRALE
003600 ROMBOSPIRALE
003700 ROMBOSPIRALE
003800 ROMBOSPIRALE
003900 ROMBOSPIRALE
004000 ROMBOSPIRALE
004100 ROMBOSPIRALE
004200 ROMBOSPIRALE
004300 ROMBOSPIRALE
004400 ROMBOSPIRALE
004500 ROMBOSPIRALE
004600 ROMBOSPIRALE
004700 ROMBOSPIRALE
004800 ROMBOSPIRALE
004900 ROMBOSPIRALE
005000 ROMBOSPIRALE
005100 ROMBOSPIRALE
005200 ROMBOSPIRALE
005300 ROMBOSPIRALE
005400 ROMBOSPIRALE
005500 ROMBOSPIRALE
005600 ROMBOSPIRALE
005700 ROMBOSPIRALE
005800 ROMBOSPIRALE
005900 ROMBOSPIRALE
006000 ROMBOSPIRALE
006100 ROMBOSPIRALE
006200 ROMBOSPIRALE
006300 ROMBOSPIRALE
006400 ROMBOSPIRALE
006500 ROMBOSPIRALE
006600 ROMBOSPIRALE
006700 ROMBOSPIRALE
006800 ROMBOSPIRALE
006900 ROMBOSPIRALE
007000 ROMBOSPIRALE
007100 ROMBOSPIRALE
007200 ROMBOSPIRALE
007300 ROMBOSPIRALE
007400 ROMBOSPIRALE
007500 ROMBOSPIRALE
007600 ROMBOSPIRALE
007700 ROMBOSPIRALE
007800 ROMBOSPIRALE
007900 ROMBOSPIRALE
008000 ROMBOSPIRALE
008100 ROMBOSPIRALE
008200 ROMBOSPIRALE
008300 ROMBOSPIRALE
008400 ROMBOSPIRALE
008500 ROMBOSPIRALE
008600 ROMBOSPIRALE
008700 ROMBOSPIRALE
008800 ROMBOSPIRALE
008900 ROMBOSPIRALE
009000 ROMBOSPIRALE
009100 ROMBOSPIRALE
009200 ROMBOSPIRALE
009300 ROMBOSPIRALE
009400 ROMBOSPIRALE
009500 ROMBOSPIRALE
009600 ROMBOSPIRALE
009700 ROMBOSPIRALE
009800 ROMBOSPIRALE
009900 ROMBOSPIRALE
010000 ROMBOSPIRALE

```

Figura 106 (continua)

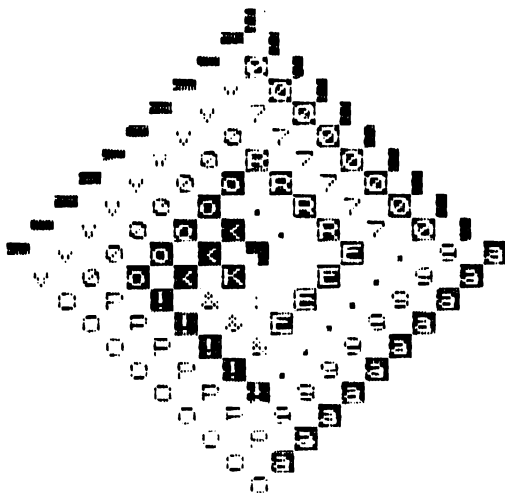


Figura 107

```

00000000 FOR P=1 TO 11-D
00000070 PRINT AT 23-P-D,17-P;CHR$ X
00000000 NEXT P
00000000 GO SUB 370
00000010 FOR P=1 TO 11-D
00000000 PRINT AT 13-P,5+P+D;CHR$ X
00000000 NEXT P
00000040 LET C=C*(-1)
00000000 GO TO 100*
00000070 IF C=1 THEN LET X=33+INT (R
ND*11): INVERSE INT (RND*2): IN
X INT (RND*7)+1
00000000 RETURN

```

Figura 106

PROGRAMMA UNIVERSALE ANIMAZIONE CON PRINT AT

Una complessa animazione può richiedere decine di linee con PRINT AT. Il programma di Figura 108 permette di condensare tutte le istruzioni necessarie all'animazione in sei stringhe.

Per ogni singolo movimento le sei stringhe contengono in codice i seguenti dati:

l\$	codice linea
c\$	codice colonna
i\$	codice INK
p\$	codice PAPER
x\$	carattere (anche grafico o speciale)
t\$	codice tempo tra un PRINT AT e il successivo

I codici ed i rispettivi valori per, linee, colonne, tempi e colori INK e PAPER, sono nella Tabella 2; per quanto riguarda i caratteri, nella stringa x\$ è presente direttamente il carattere e non un suo codice.

Per un corretto funzionamento è essenziale che le sei stringhe abbiano la stessa lunghezza.

Nel programma è anche prevista una zona (a partire dalla linea 1000), dove inserire eventuali routines (come nell'esempio) per caricare in memoria caratteri speciali o per disegnare all'inizio figure o altre cose necessarie all'animazione.

I dati contenuti nelle stringhe del programma di Figura 108, mostrano, come esempio di animazione, il movimento di un insetto in una tana con una entrata ed un'uscita, in Figura 109 è visibile l'inizio di tale animazione.

```

10 REM Programma Universale
20 REM Animazione con PRINT AT
30 REM con dati esempio
40 REM -----
99 BORDER 4: PAPER 0: CLS
100 GO SUB 1000
110 REM Codici Linee in l$
120 LET l$="55555555555554444333
333444455555444433333333"
130 REM Codici Colonne in c$
140 LET c$="778899::;<<<<====>
>????@#@@ARBBBBCCCCDDEEFFG"
150 REM Colori in i$
160 LET i$="7060500616360646062
60616065636061606460650607"
165 REM Fondi in p$

```

Figura 108 (continua)

MUSICA ED EFFETTI SONORI

MINIORGANO

Nonostante alcune limitazioni, dovute al fatto che lo Spectrum non possiede un integrato indipendente per la generazione dei suoni ma è lo Z80 che li produce, è possibile realizzare con il programma di figura 110 un Miniorgano a 5 ottave (61 note) che usa 18 dei 40 tasti dello Spectrum.

Nel programma è utilizzata una routine in linguaggio macchina lunga 89 Bytes che serve a ridurre al minimo il tempo di lettura della tastiera ed il calcolo del codice nota dell'istruzione BEEP. La funzione dei tasti usati dal Miniorgano è la seguente;

Tasto	Funzione
1	prima ottava
2	seconda ottava
3	terza ottava
4	quarta ottava
5	quinta ottava
Z	DO
S	DO#
X	RE
D	RE#
C	MI
V	FA
G	FA#
B	SOL
H	SOL#
N	LA
J	LA#
M	SI
'SYMBOL SHIFT'	DO (ottava superiore)

Per usare il Miniorgano occorre premere contemporaneamente uno dei cinque tasti di ottava ('1'...'5') ed uno dei tredici tasti di nota (per es. 'G').

Importante: Non inserire nel programma l'istruzione BORDER altrimenti non fun-

ziona; se si vuole un dato colore del bordo lo si può inserire con un comando diretto prima di scrivere il programma.

```

1000 REM          GENERATORE DI MUSICA CASUALE
1010 REM          CASUALE - 130 NOTE
1020 REM          -----
1030 INPUT "Inserire durata nota";d: IF d<.1 OR d>1 TH
1040 EN DO
1050 LET f=0
1060 FOR n=1 TO 1000
1070 LET r=INT (RND*20) -10
1080 LET f=f+r
1090 IF f>60 THEN LET f=f-20
1100 IF f<-60 THEN LET f=f+20
1110 BEEP d,f
1120 NEXT n

```

Figura 110

GENERATORE DI MUSICA CASUALE – 130 NOTE

Il programma di Figura 111 produce musica utilizzando in modo casuale le centotrenta note diverse dello Spectrum. All'inizio occorre inserire la durata richiesta per ogni nota.

```

1000 REM          GENERATORE DI MUSICA
1010 REM          CASUALE - 130 NOTE
1020 REM          -----
1030 INPUT "Inserire durata nota";d: IF d<.1 OR d>1 TH
1040 EN DO
1050 LET f=0
1060 FOR n=1 TO 1000
1070 LET r=INT (RND*20) -10
1080 LET f=f+r
1090 IF f>60 THEN LET f=f-20
1100 IF f<-60 THEN LET f=f+20
1110 BEEP d,f
1120 NEXT n

```

Figura 111

METRONOMO

Il Metronomo (Figura 112) può produrre da 60 a 300 battute al minuto a seconda del valore inserito all'inizio nella richiesta di Input fatta dal programma.

```
10 REM METRONOMO
200 REM -----
100 INPUT "Inserire N° battute
al minuto (60--300) e premere
ENTER "; b
110 IF b<60 OR b>300 THEN GO TO
100
120 LET t=3000/b-1
130 BEEP .02,-15
140 PAUSE t
150 GO TO 130
```

Figura 112

SCALE MUSICALI

Il programma di Figura 113 produce, continuamente le scale musicali alternativa-mente a salire ed a scendere. All'inizio può essere stabilita la durata delle note delle scale.

```
10 REM SCALE MUSICALI
200 REM -----
100 INPUT "Inserire durata note
(0.1--2) "; d: IF d<.1 OR d>2 TH
EN GO TO 100
110 DIM a$(12,4)
120 LET a$(1)="DO "; LET a$(2)
="DO# "; LET a$(3)="RE "; LET a
$(4)="RE# "; LET a$(5)="MI "; L
ET a$(6)="FA "; LET a$(7)="FA#
"; LET a$(8)="SOL "; LET a$(9)="
SOL# "; LET a$(10)="LA "; LET a$
(11)="LA# "; LET a$(12)="SI "
130 FOR o=-5 TO 4
140 FOR n=0 TO 11
150 INK 6: PRINT AT 10,10;a$(n+
1); " Ottava ";o+5; " "
160 BEEP d,o*12+n
170 NEXT n: NEXT o
180 FOR o=4 TO -5 STEP -1
190 FOR n=11 TO 0 STEP -1
200 INK 5: PRINT AT 10,10;a$(n+
1); " Ottava ";o+5; " "
210 BEEP d,o*12+n
220 NEXT n: NEXT o
230 GO TO 130
```

Figura 113

PROGRAMMA UNIVERSALE PER MUSICA

Suonare con il Computer il vostro pezzo musicale preferito può voler dire realizzare un programma con, nel caso dello Spectrum, centinaia di linee con istruzioni BEEP.

Tali istruzioni possono essere compattate usando il programma di Figura 114 nel quale le note sono inserite in codice in una stringa (a\$). Ad ogni nota tra la '-36' e la '+56' corrisponde un carattere ASCII da inserire nella stringa a\$ secondo la Tabella 3.

In a\$ le Pause sono inserite con '!'.
 !

La durata delle note è stabilita all'inizio del programma dal valore (in secondi) inserito nella variabile 'd' (linea 99); se nel pezzo musicale occorrono a volte note di durata maggiore basta inserire nella stringa più codici della stessa nota di seguito.

```

10 REM          PROGRAMMA UNIVERSALE
20 REM          PER MUSICA
30 REM          -----
99 LET d=0.5: REM durata note
in secondi
100 REM        NOTE NELLA STRINGA a$
110 LET a$="##$%&'()*+,-./012345
6789:;<=>!!!!?@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~"
120 FOR n=1 TO LEN a$
130 IF CODE a$(n TO ) < 33 OR CODE
E a$(n TO ) > 127 OR CODE a$(n TO
)=34 THEN PRINT "NOTA ";n;" ERRA
TA"
140 NEXT n
150 FOR n=1 TO LEN a$
160 IF CODE a$(n TO )=33 THEN G
O TO 200
170 BEEP d, CODE a$(n TO )-71
180 NEXT n
190 STOP
200 PAUSE 50*d
210 NEXT n
    
```

Figura 114

CHARACTER	NOTE	DURATION	BEEP
ASCII			

!	PAUSA		
#	DO	2	-36
\$	DO#	2	-36

Tabella 3 (continua)

/rN<X&ECC-Iw&wvovZZfzCHlG7mO00D0&v u 8... 00\00t40mP0\• 1~+*~0\0\0%
gWp77z3p&00fGfW077z3p&00fGfW077z3p&00fGfW077z3p&00fGfW077z3p&
D00IDHm000HID00IDHm000HID00IDHm000HID00IDHm000HID00IDHm000HID00
fGF# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
#

0000000000000000000000000000000000000000000000000000000000

00000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000

Tabella 3 (continua)

OROLOGIO (Figura 116)

```
1000 REM OROLOGIO
1010 REM -----
1020 FOR n = 1 TO 60
1030 PRINT n = 1 TO 60
1040 PRINT n .001, -5
1050 PRINT n .001, -12
1060 PRINT n .004
1070 NEXT n
```

Figura 116

MITRA (Figura 117)

```
1000 REM MITRA
1010 REM -----
1020 FOR n = 1 TO 10
1030 FOR c = 1 TO 3 + INT (RND * 10)
1040 PRINT c .02, -24
1050 FOR t = 1 TO 5: NEXT t
1060 NEXT c
1070 PRINT 20 + INT (RND * 30)
1080 NEXT n
```

Figura 117

SIRENA AMERICANA (Figura 118)

```
1000 REM SIRENA AMERICANA
1010 REM -----
1020 FOR a = 1 TO 100
1030 PRINT a .001, 12
1040 PRINT a .001, 13
1050 PRINT a .001, 14
1060 PRINT a .001, 15
1070 PRINT a .001, 16
1080 PRINT a .001, 17
1090 PRINT a .001, 18
1100 PRINT a .001, 19
1110 PRINT a .001, 20
1120 PRINT a .001, 19
1130 PRINT a .001, 18
1140 PRINT a .001, 17
1150 PRINT a .001, 16
1160 PRINT a .001, 15
1170 PRINT a .001, 14
1180 PRINT a .001, 13
1190 NEXT a
```

Figura 118

SIRENA A 2 TONI (Figura 119)

```

10000 P SIRENA A 2 TONI
10100 P -----
10200 P a = 1 TO 30
10300 P , 70 : PRRM tono 1
10400 P , 70 : PRRM tono 2
10500 P NEXT a

```

Figura 119

SIRENA MULTITONO (Figura 120)

```

10000 P SIRENA MULTITONO
10100 P -----
10200 P a = 1 TO 30
10300 P , 1, -7 : PRRM tono 1
10400 P , 1, -4 : PRRM tono 2
10500 P , 1, -10 : PRRM tono 3
10600 P , 1, -10 : PRRM tono 4
10700 P , 1, -4 : PRRM tono 5
10800 P , 1, -4 : PRRM tono 6
10900 P NEXT a
11000 P (anche piu' di 4 toni)

```

Figura 120

SQUILLO DEL TELEFONO (Figura 121)

```

10000 P SQUILLO DEL TELEFONO
10100 P -----
10200 P f = 1 TO 8
10300 P , 1 TO 40
10400 P , 2, 25
10500 P t = 1 TO 1 : NEXT t
10600 P NEXT f
10700 P PRRM 80
10800 P NEXT n

```

Figura 121

SEGNALE TELEFONICO DI LINEA LIBERA (Figura 122)

```
1000 REM          SEGNALE TELEFONICO
1010 REM          DI LINEA LIBERA
1020 REM          -----
1030 FOR n=1 TO 10
1040 BEEP 1.4,16
1050 PAUSE 70
1060 NEXT n
```

Figura 122

SEGNALE TELEFONICO DI LINEA OCCUPATA (Figura 123)

```
1000 REM          SEGNALE TELEFONICO
1010 REM          DI LINEA OCCUPATA
1020 REM          -----
1030 FOR n=1 TO 30
1040 BEEP .2,14
1050 PAUSE 10
1060 NEXT n
```

Figura 123

GENERATORE DI SUONI CASUALI (Figura 124)

```
1000 REM          GENERATORE DI SUONI
1010 REM          CASUALI
1020 REM          -----
1030 FOR n=1 TO 1000
1040 LET t=10: REM o t=5 o t=20
1050 BEEP RND/t,RND*60-20
1060 NEXT n
```

Figura 124

DIN-DON (Figura 125)

```
1000 REM          DIN-DON
1010 REM          -----
1020 FOR n=1 TO 10
1030 BEEP .002,23
1040 PAUSE .005
1050 BEEP .005,19
1060 PAUSE .005
1070 NEXT n
```

Figura 125

GRILLI (Figura 126).

```
10000  RR      [5]
10010  RR
10020  RR  -----
10030  RR  k=1 TO 50
10040  RR  k=1 TO 3
10050  RR  P .02,36
10060  RR  t=1 TO 1: NEXT t
10070  RR  k
10080  RR  6
10090  RR  NEXT n
```

Figura 126

GIOCHI

CACCIA AL SOMMERSIBILE

Il gioco (Figura 127) consiste nel trovare un sommergibile muovendo tramite i tasti 5, 6, 7, 8 un simbolo che rappresenta la nave o l'aereo antisommergibile.

Naturalmente finchè la nave non gli passa sopra il sommergibile viaggia nascosto sottacqua spostandosi casualmente in tutte le direzioni, il cacciatore ha però la possibilità di scovarlo seguendo il suono di una sorta di 'SONAR' prodotto dal computer.

Infatti tale suono AUMENTA di frequenza se ci si sta avvicinando e diminuisce se ci si sta allontanando o se il sommergibile si sposta in senso opposto.

Appena la nave o aereo trova il sommergibile, questo appare (Figura 128) insieme ad una scritta che indica in quante mosse ci si è riusciti.

Ovviamente ogni volta che si inizia il gioco la nave parte dallo stesso angolo ed il sommergibile viene nascosto nello stesso punto.

Il gioco può essere reso più difficile facendo spostare più spesso il sommergibile sostituendo la linea 310 con:

```
310 GO SUB 350
```

Un'ulteriore difficoltà può essere introdotta facendo spostare il sommergibile di spazi più ampi, sostituendo le linee 350 e 360 con:

```
350 LET k=INT (RND*5) -2
```

```
360 LET h=INT (RND*5) -2
```

```
10 REM CHOCCH HL SOMMERSIBILE
20 REM -----
99 BORDER 0: PAPER 1: INK 7: C
Ls
100 DATA 1,130,75,48,4,28,50,19
3,0,48,16,48,96,200,255,0
110 LET u=PEEK 23875+255*PEEK 2
3678
111 FOR w=u TO u+15
112 READ g: POKE w,g: NEXT w
120 FOR s=1 TO 176
130 PRINT CHR$ 144;CHR$ 144;CHR$ 144;CHR
$ 144;CHR$ 144;
140 NEXT s
150 LET l=21: LET c=0
160 LET x=20: LET y=8
```

Figura 127 (continua)

```

170 LET M=0: LET f=-60
180 INK 4: PRINT AT L,C;"#"
185 BEEP 0.3,f
190 INK 7: PRINT AT L,C;CHR$ 14
4
200 IF C>0 AND INKEY$="5" THEN
LET C=C-1
210 IF L<21 AND INKEY$="6" THEN
LET L=L+1
220 IF L>0 AND INKEY$="7" THEN
LET L=L-1
230 IF C<31 AND INKEY$="8" THEN
LET C=C+1
240 IF CODE INKEY$>52 AND CODE
INKEY$<57 THEN GO TO 300
250 IF L=4 AND C=X THEN GO TO 2
260 GO TO 180
270 INK 6: PRINT AT 4,X;CHR$ 14
5
277 BEEP 2,-30

```

Figura 127 (continua)

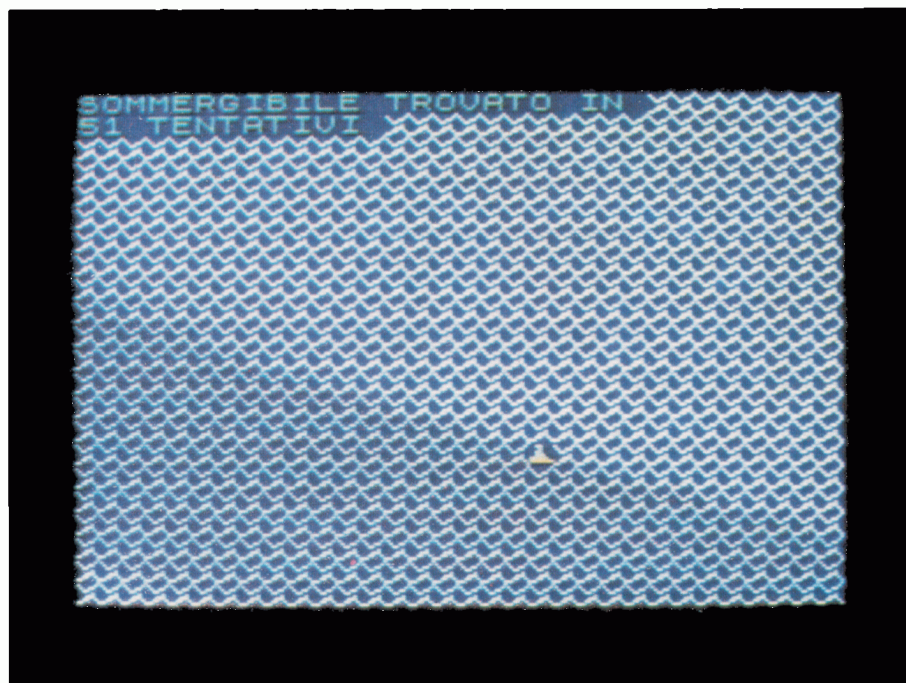


Figura 128

```

280 INK 4: PRINT AT 0,0;"SOMMER
GIBILE TROVATO IN ";AT 1,0;m;" T
ENTATIVI "
290 PAUSE 50: STOP
300 LET m=m+1
310 IF RND<0.5 THEN GO SUB 350
320 LET j=53-ABS (y-l)-ABS (c-x
)
330 LET f=-55+j*2
340 GO TO 250
350 LET k=INT (RND*3)-1
360 LET h=INT (RND*3)-1
370 IF x+k<=31 AND x+k>=0 THEN
LET x=x+k
380 IF y+h<=21 AND y+h>=0 THEN
LET y=y+h
390 RETURN

```

Figura 127

TIRO ALLA FUNE

Altro gioco è il Tiro alla Fune (Figura 129) con il quale devono giocare due persone.

All'inizio appaiono sullo schermo due omini (Figura 130) con una fune.

L'omino di sinistra è Giallo e viene comandato dal tasto 'A' mentre l'omino di destra è Bianco e viene comandato dal tasto 'L'. Ogni volta che appare il rettangolo rosso al centro dello schermo il giocatore che preme per primo il proprio tasto ha la possibilità di tirare i due uomini dalla propria parte, se però uno dei giocatori preme il proprio tasto prima che appaia il rettangolo, i due omini si sposteranno nel senso opposto favorendo l'avversario.

Il gioco continua finchè uno dei due giocatori non riesce a trascinare l'avversario completamente nella propria zona (Figura 131).

È importante durante il gioco che i giocatori non premano tasti diversi da 'A' e 'L'.

Il tempo di apparizione del rettangolo rosso ed il ritardo tra una apparizione e l'altra sono casuali e vengono stabiliti rispettivamente dalle linee 290 e 280 che possono essere eventualmente modificate per rendere il gioco più difficile.

Con le linee attuali il tempo di apparizione ed il ritardo sono rispettivamente di 200-600 msec. e di 1-3 sec.

Importante: Non inserire in questo programma l'istruzione BORDER altrimenti non funziona più. Se si vuole un colore per il bordo lo si può stabilire con un comando diretto prima di inserire il programma.


```

0000 POKE 00400,INT (RND*100)+50
0000 POKE 00401,INT (RND*20)+10
0000 GO SUB USR (U+40)*500
0010 POKE 22831,0: POKE 22832,0
0000 GO TO 0000
0000 PRINT AT 2,1: INK 6: "Vince
il GIALLO"
0010 PRINT AT 12,2: INK 5: "Preme
re ENTER per continuare"
0020 INPUT e$
0030 RUN
0040 RETURN
0050 PRINT AT 2,16: INK 7: "Vince
il BIANCO"
0060 PRINT AT 12,2: INK 5: "Preme
re ENTER per continuare"
0070 INPUT e$
0080 RUN

```

Figura 129



Figura 130

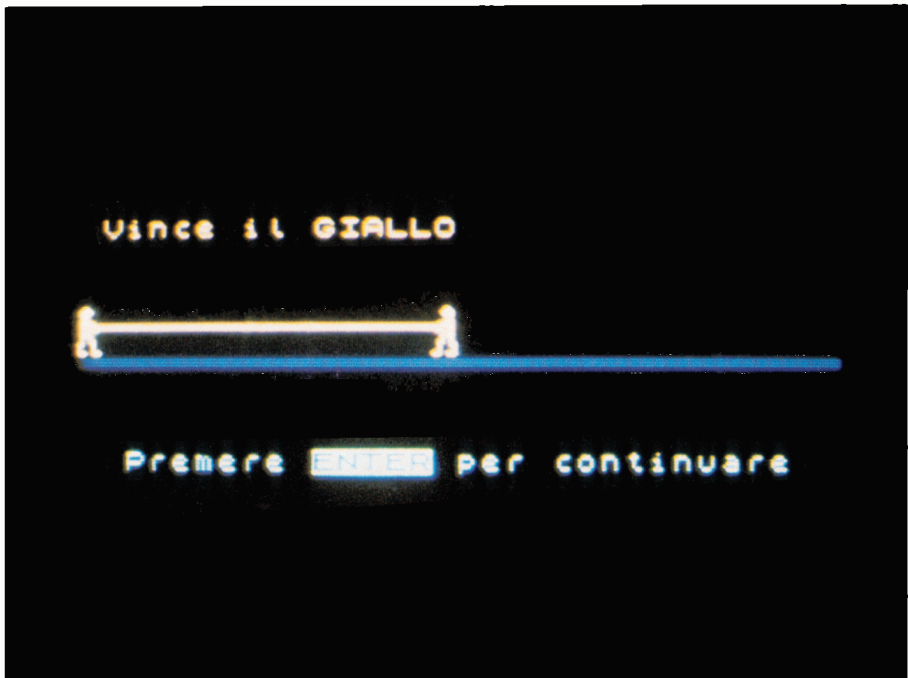


Figura 131


```

0350 LET K$=SCREEN$(L,C): LET J
#0360 SCREEN$(L,C+1): LET K$=K$+J$
0370 PRINT AT L,C: PAPER 0: INK
0377 PAUSE 3
0380 PRINT AT L,C: PAPER P: INK
0390 NEXT X
0400 PRINT AT L,C: FLASH 1: PAPE
R P: INK I: K$
0410 INK 6: PRINT AT 12,15: "Prem
ere" ENTER"; AT 13,15: "per continu
are"
0420 INPUT E$
0430 FLASH 0: GO TO 100

```

Figura 132

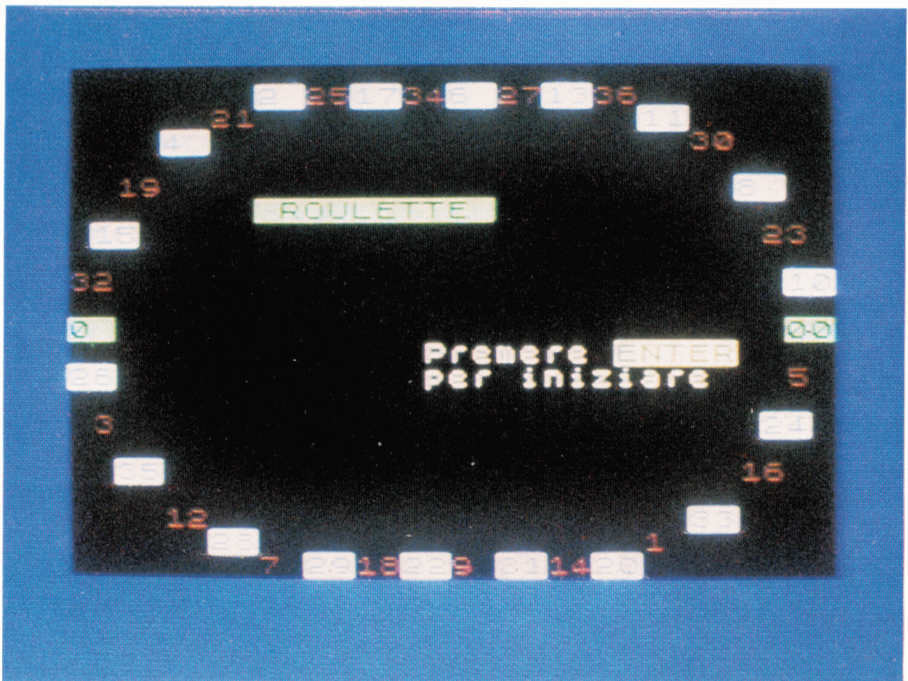


Figura 133

CAMPO MINATO

Questo programma (Figura 134) simula un campo in cui vi sono 20 mine nascoste casualmente. Il giocatore deve spostare, usando i tasti 5, 6, 7, 8, l'omino dalla posizione di partenza nell'angolo in basso a sinistra alla posizione di arrivo nell'angolo in alto a destra senza incappare in una delle mine nascoste. Per attraversare il campo minato senza danni, il giocatore può contare sul proprio intuito e sulla propria fortuna, se riesce appare sul video la scritta 'SALVO !!!'. Aggiungendo la linea 245 (Figura 135) e facendo girare il programma, si può avere un'idea di come vengono disposte le mine sul campo (Figura 136). Naturalmente durante il gioco le mine devono essere nascoste per cui si deve prima togliere dal programma la linea 245.

Ora con le mine nascoste può capitare di terminare il gioco senza danno (Figura 137) oppure può succedere prima dell'arrivo di passare su una delle mine (Figura 138).

Il numero delle mine può essere diminuito o aumentato per rendere più facile o più difficile il gioco. Se per esempio si vogliono nascondere 12 mine invece di 20 basta modificare il contenuto delle linee 190, 210 e 400, sostituendo, all'interno delle linee stesse, tutti i 20 con dei 12.

```

100 REM
110 REM
120 LET U=256*PEEK 23676+PEEK 2
130 FOR a=U TO U+31: READ w: PO
140 a,w: DATA NEXT a
150 50,50,50,170,85,170,85,170,85,1
160 70,50,50,170,85,170,85,170,85,17
170 0,50,50,10,254,106,40,40,200,0,1
180 0,0,10,140,50,104,204
190 FOR DER=0: PAPER 0: CLS
200 FOR a=1 TO 20
210 HNZK 4: PRINT CHR$(144);CHR$
220 CHR$(144);CHR$(144);CHR$(144);C
230 HNZK CHR$(144);CHR$(144);
240 NEXT a
250 PRINT
260 LET l=01
270 LET c=0
280 HNZK 6: PRINT AT 0,31;"@ "
290 HNZK 80: PRINT AT l,c;"@ "
300 PAPER 30
310 DIM M(20,2)
320 RANDOMIZE
```

Figura 134 (continua)

```

0010 FOR a=1 TO 20
0020 LET g=INT (32*RND)
0030 LET f=INT (22*RND)
0040 IF g<6 AND f>15 THEN GO TO
0050
0060 LET m(a,1)=f
0070 LET m(a,2)=g
0080 NEXT a
0090 INK 6: PRINT AT l,c;CHR$ 14
0100
0110 BEEP .1,30
0120 INK 0: PRINT AT l,c;"■"
0130 LET x=l
0140 LET y=c
0150 IF c>0 AND INKEY$="5" THEN
0160 LET c=c-1
0170 IF l<21 AND INKEY$="6" THEN
0180 LET l=l+1
0190 IF l>0 AND INKEY$="7" THEN
0200 LET l=l-1
0210 IF c<31 AND INKEY$="8" THEN
0220 LET c=c+1
0230 IF x<>l OR y<>c THEN GO TO
0240
0250 GO TO 280
0260 FOR a=1 TO 20
0270 IF m(a,1)=l AND m(a,2)=c TH
0280 GO TO 450
0290 NEXT a
0300 IF l=0 AND c=31 THEN GO TO
0310
0320 GO TO 280
0330 INK 7: PRINT AT l,c;CHR$ 14
0340
0350 BEEP 2,-40: STOP
0360 PRINT AT 0,31;CHR$ 146
0370 INK 6: PRINT AT 0,0;"
0380 SALVO !!!"

```

Figura 134

```

0245 INK 5: PRINT AT f,g;CHR$ 14
0250

```

Figura 135

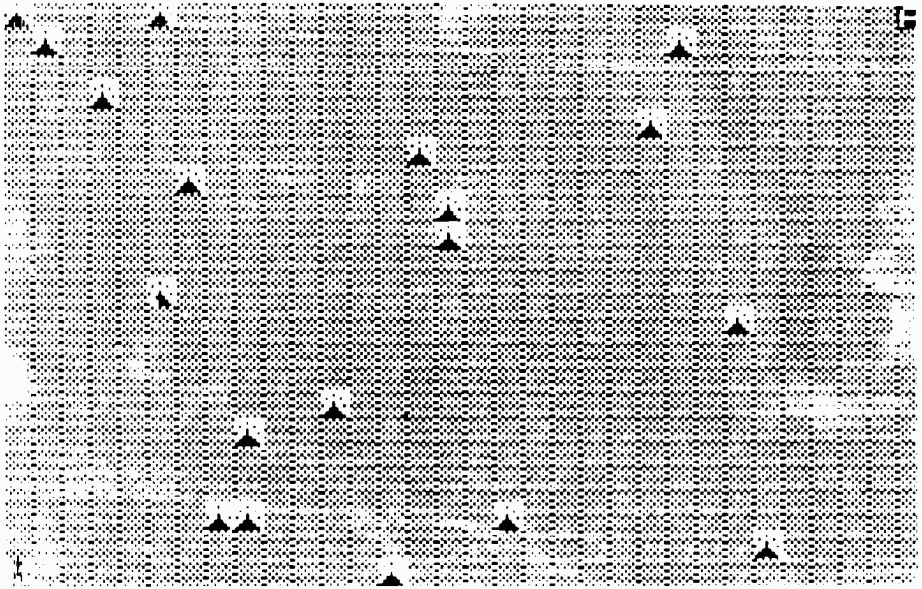


Figura 136

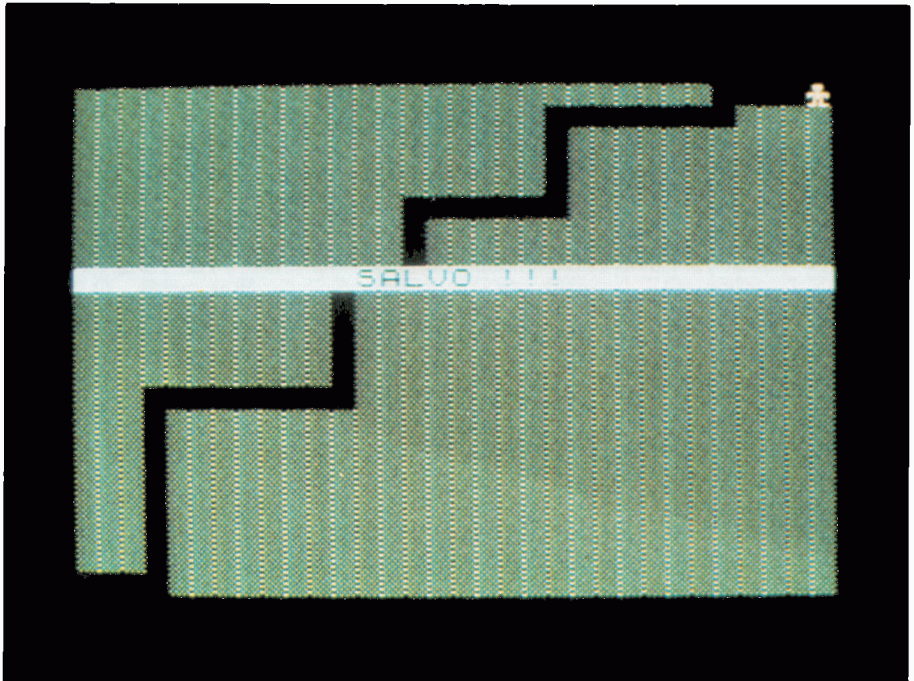


Figura 137

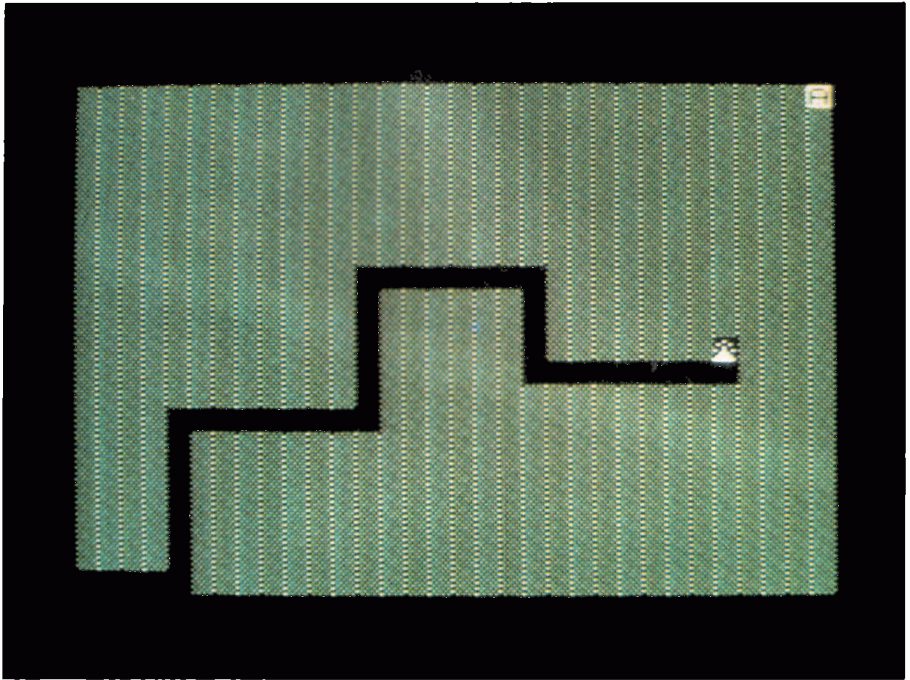


Figura 138

FLIPPER GIAPPONESE

In questo 'Flipper Giapponese' (Figura 139), trenta palline cadono dall'alto attraverso una serie di chiodini, finendo in modo casuale su delle basi che indicano dei punteggi diversi, punteggi che vengono automaticamente sommati al totale (Figura 140).

Il numero delle palline può essere modificato cambiando la cifra '30' nella linea 290.

```
100 REM FLIPPER GIAPPONESE
1000 REM -----
1100 BORDER 4: PAPER 1: CLS
1110 DATA 24,24,0,0,0,0,0,0,0,0,0,0
1120 LET U=PEEK 23675+256*PEEK 2
1200 FOR W=U TO U+15
1400 READ N: POKE W,N: NEXT W
1500 INK 2: PRINT AT 2,10;"
1600 LET A#=CHR$ 144+CHR$ 128
1700 FOR A=1 TO 4: LET A#=A#+A#:
NEXT A
1800 LET B#="" +A#
1900 FOR B=0 TO 12
2000 INK 7: PRINT AT 4+A,13-A;A#
(1 TO (A+3)*2+1)
2100 NEXT B
2200 INK 8: INVERSE 1: PRINT AT
2300 PRINT AT 10,3;"05021
2400 PRINT AT 10,3;"00526
2500 INVERSE 0
2600 INK 7: PRINT AT 21,0;"Preme
2700 INPUT LINE E$
2800 PRINT AT 21,16;"PUNTEGGIO
2900 PRINT AT 21,0;"
3000 LET S=0
3100 FOR A=1 TO 30
3200 LET C=3: LET C=16
3300 INK 6: PRINT AT 3,3;A;"
3400 PRINT AT 2,16;CHR$ 145: PAU
3500 PRINT AT 2,16;"": AT 3,16
3600 FOR Z=1 TO 12
3700 LET Z=AND
```

Figura 139 (continua)


```

7 34 IF X<0.5 THEN LET Y=-1: IF
Y>0.5 THEN LET Y=1
IF Y=0 THEN LET Y=1
PRINT AT 0,0: " " LET Y=1
LET I=I+1
PRINT AT 0,0+5:CHR$ 145
LET C=C+7
GOTO 0.05,-20
NEXT I
IF C=15 OR C=17 THEN LET S=
S+10
IF C=13 OR C=19 THEN LET S=
S+20
IF C=11 OR C=21 THEN LET S=
S+40
IF C=9 OR C=23 THEN LET S=S
+80
IF C=7 OR C=25 THEN LET S=S
+140
IF C=5 OR C=27 THEN LET S=S
+200
IF C=3 OR C=29 THEN LET S=S
+260
PRINT 7: PRINT AT 21,26: S
PAUSE 10: PRINT AT 0,0: " "
NEXT S
GO TO 250

```

Figura 139

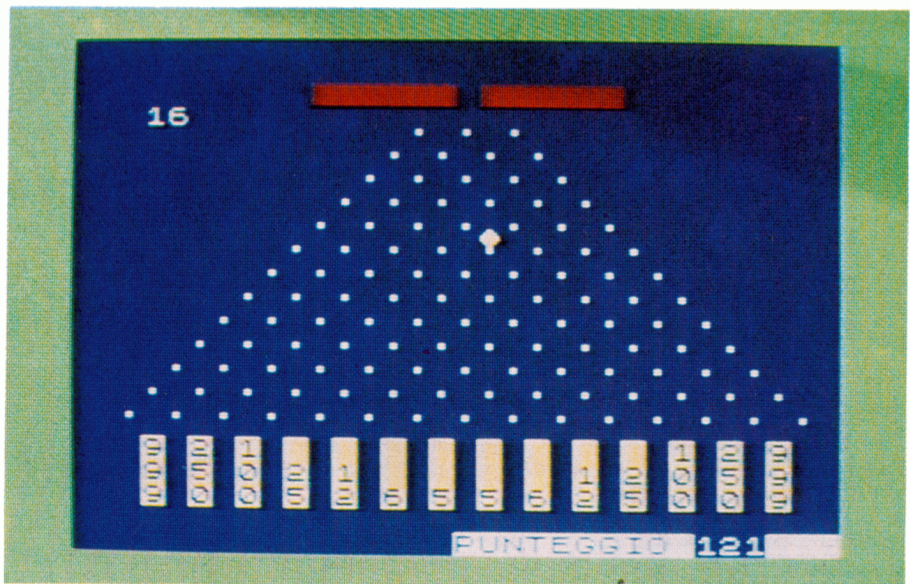


Figura 140

TAVOLA DI NUMERI E LETTERE

La prima parte del programma di Figura 141 disegna sul video una tavola di sedici caselle di cui quindici occupate dai numeri da 0 a 9 più le lettere da A a E. Subito dopo, le quindici caselle vengono mescolate per apparire alla fine sullo schermo nel modo indicato in Figura 142.

A questo punto, usando i tasti 5, 6, 7, 8 si possono spostare le varie caselle con i numeri e le lettere per cercare di ritornare alla disposizione iniziale (Figura 143) nel minor numero possibile di mosse. Durante il gioco o alla fine, ogni volta che si preme il tasto 'p', si può avere il numero aggiornato delle mosse fatte.

```

0100 REM          TAVOLA DI NUMERI
0200 REM          E LETTERE
0300 REM          -----
0400 BORDO DEB 1: PAPER 0: INK 2: C
L 0
1110 DIM W##(4, 4)
1200 FOR I=0 TO 4
1300 FOR J=0 TO 3
1400 LET W##(I, J) = INT(4*RNDF)
1500 NEXT J
1600 NEXT I
1700 PRINT W##
1800 FOR I=0 TO 4
1900 FOR J=0 TO 3
2000 PRINT W##(I, J);
2100 IF J<3 THEN PRINT " ";
2200 IF J=3 THEN PRINT " : ";
2300 NEXT J
2400 PRINT " "
2500 NEXT I
2600 LET P = 0
2700 LET C = 4
2800 HZK B0 = 1 TO 500
2900 LET X = 1 + INT(4*RNDF)
3000 IF C < 4 AND X = 1 THEN GO SUB
3100 IF C > 1 AND X = 2 THEN GO SUB
3200 IF C < 4 AND X = 3 THEN GO SUB
3300 IF C > 1 AND X = 4 THEN GO SUB
3400 NEXT P
3500 LET P = P + 1
3600 IF P = 10 THEN PRINT AT
3700 INK 4; "TOTALE "; P; " MOSSE"
3800 IF C < 4 AND INKEY$ = "5" THEN
3900 SUB 3800
4000 IF C > 1 AND INKEY$ = "6" THEN
4100 SUB 4400
4200 IF C < 4 AND INKEY$ = "7" THEN
4300 SUB 5000

```

Figura 141 (continua)


```

330 IF X=6 THEN PRINT AT (L+3,C)
CHR$(133);CHR$(141);CHR$(141);CHR$(
141
340 LET K=K+X
350 IF Y=0 THEN GO TO 380
3600 NEXT C
3700 NEXT L
3800 PRINT
3900 PRINT "          "); INK 6: PRINT
4000 PRINT "dadi"; "          punteggio = ";K
4100 REM

```

Figura 144

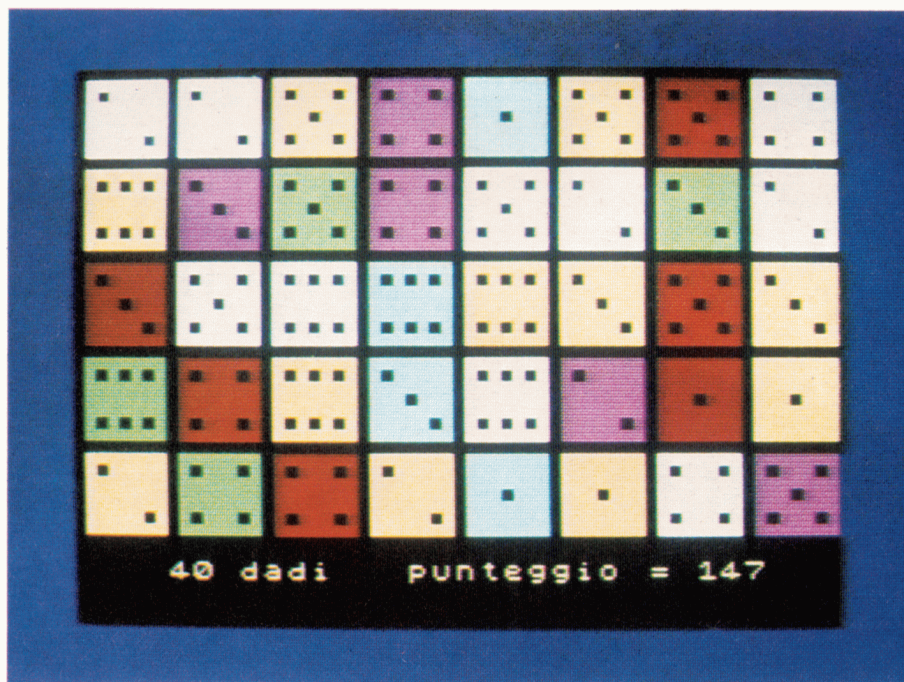


Figura 145

La particolarità dei 77 programmi per lo Spectrum contenuti in questo libro' è' oltre alla qualità, soprattutto la varietà delle applicazioni e degli argomenti per la realizzazione dei quali si è cercato di utilizzare al massimo le nuove interessanti prestazioni dello Spectrum quali le nuove istruzioni BASIC, la velocità, il Suono, e soprattutto il Colore e la Grafica tra cui specialmente l'altra risoluzione e la possibilità di creare dei Caratteri Speciali. Le applicazioni vanno dalla Grafica (30) a, Musica (5), Utility (8), Animazioni (6), Giochi (7), Effetti Sonori (12), Elettronica (4) ed altri programmi (5).

Un cenno particolare merita senz'altro la 'Business Grafica', presente con 9 programmi, e realizzata con una efficacia visiva e una utilizzabilità pratica precedentemente possibile solo su personal e business computers molto più costosi.

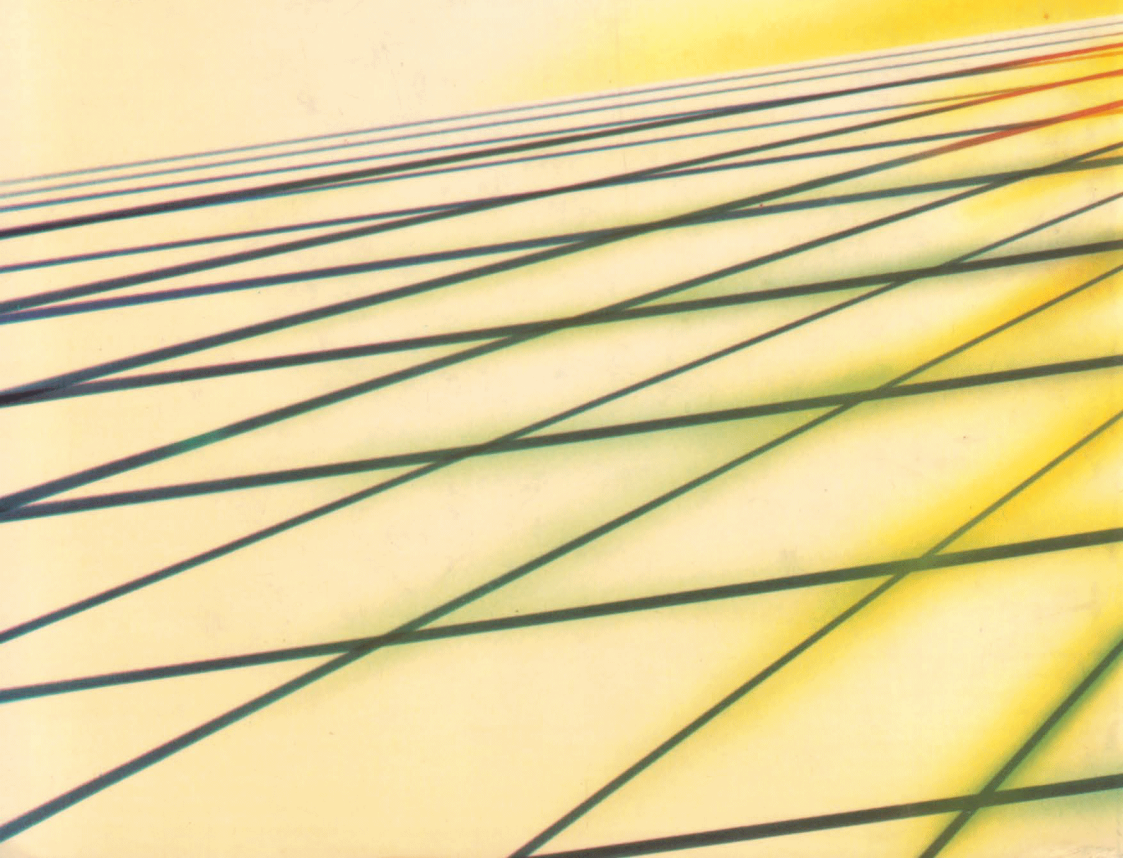
Dello stesso Autore:

- 66 PROGRAMMI PER ZX81 E ZX80 Con Nuova ROM + Hardware.
- GLI AMPLIFICATORI DI NORTON QUADRUPLI LM3900 & LM359 con Esperimenti.

L. 16.000

Cod. 555A

ISBN 88-7056-143-7





77 PROGRAMMI PER

SPECTRUM

di Gaetano Marano

GRUPPO
EDITORIALE
JACKSON

