

CREATE GIOCHI ARCADE COL VOSTRO SPECTRUM

con ampie descrizioni di oltre 30 routines e 18 fantastici programmi.

di DANIEL HAYWOOD



edizioni **Jce**

CREATE
GIOCHI ARCADE
COL VOSTRO
SPECTRUM

con ampie descrizioni di oltre 30
routines e 18 fantastici programmi.

di Daniel Haiwood

traduzione di Angelo Cattaneo



Via dei Lavoratori, 124
CINISELLO BALSAMO (MI)

Tutti i diritti sono riservati, nessuna parte di questo libro e della cassetta software allegata puo' essere riprodotta, posta in sistemi di archiviazione, trasmessa in qualsiasi forma o mezzo elettronico, meccanico, di fotocopiatura, ecc., senza l'autorizzazione scritta dell'Editore.

Nel testo sono stati introdotti programmi di valore didattico. L'Editore non risponde dei possibili errori che si verificano nei listati e nei relativi risultati.

Prima edizione: INTERFACE PUBLICATIONS 1983

Publicato in Gran Bretagna da:
Interface Publications,
44-46 Earls Court Road,
London, W8 6EJ.

Copyright © Daniel Haywood 1983

Copyright © per l'edizione italiana: Edizioni JCE, 1984

Prima edizione: Giugno 1984

Stampato in Italia da:
Gemm Grafica S.r.l.
Via Magretti-Paderno Dugnano (MI)

INDICE

Prefazione

-Tim Hartnell

Capitolo 1

-Lettura della Tastiera

"Sketchpad"

Capitolo 2

-Grafica definita dall' utente

"Generatore di caratteri", "Olocausto"

Capitolo 3

-Suoni

Capitolo 4

-Uso degli Operatori Logici

"Bomber"

Capitolo 5

-Movimento

"Zap", "Invaders", "Snakes", "Squash!"

Capitolo 6

-Scrolling

"Meteors", "Gran Prix", "Scromble", "Slalom"

Capitolo 7

-Uso degli array

"Flea race", "Snakey", "ICBM"

Capitolo 8

-PEEK e POKE

"Circuit" parte prima e seconda, "Chopper", "Raindrops"

Capitolo 9

-Come rendere piu' interessanti i Games.

PREFAZIONE - Tim Hartnell

D' ora in poi non dovrete piu' sacrificare le cento lire in sala giochi; infatti, con l' aiuto dei programmi presentati in questo libro (sempre accompagnati dai chiarimenti tecnici del caso) potrete ottenere gli stessi identici effetti.

L' autore Daniel, nello stendere l' opera, presume che voi conosciate gia' le basi di programmazione dello Spectrum per cui procede spedito presentando i programmi piu' interessanti senza tralasciare pero' la descrizione del perche' venga seguita una certa strada anziche' un' altra.

Il contenuto didattico trova posto alla fine di ogni programma dove e' riportata la spiegazione dei vari blocchi che lo compongono. Ciononostante, siamo certi che il volume verra' apprezzato principalmente per il suo donare all' utente diverse ore di divertimento, lo stesso che abbiamo provato noi nel verificare i listati per la pubblicazione.

Tim Hartnell

Tim Hartnell, oltre ad essere il fondatore del Gruppo Nazionale Inglese degli ZX Users, ha anche scritto una trentina di libri sugli ZX e su altri computers.

LETTURA DELLA TASTIERA

Il riuscire a "leggere" la tastiera, e' indispensabile per poter scrivere un programma di giochi. Questo primo capitolo spiega appunto le varie funzioni legate alla tastiera e in quale modo esse agiscano.

INKEY\$ rende il carattere del tasto premuto sottoforma di stringa esplorando la tastiera in una frazione di secondo. Così' come in altri computers, tale funzione ha una azione "continuativa", cioè' continua a leggere il carattere relativo al tasto per tutto il tempo in cui questo rimane premuto.

INKEY\$ agisce anche in combinazione con i tasti di Shift rendendo i relativi caratteri maiuscoli o i relativi simboli. CODE INKEY\$, come e' intuibile, porta il codice del carattere memorizzato da INKEY\$ il quale, nel caso non sia azionato alcun tasto, contiene una stringa vuota (ad es. " "). In questo caso CODE INKEY\$ ritorna il valore zero.

Lo Spectrum possiede anche un secondo comando molto usato : IN. Il vantaggio di "IN" rispetto ad "INKEY\$" e' che puo' essere usato per leggere piu' di un tasto contemporaneamente; cosa che INKEY\$ non puo' fare.

Provate a battere :

```
10 POKE 23692,-1:PRINT INKEY$:GO TO 10
```

e rilevate cosa succede premendo due tasti contemporaneamente come potrebbe capitare durante un game di invaders. La risposta e' semplice : non succede nulla in base a quanto sopra spiegato.

La funzione di INKEY\$ viene prevalentemente usata per variare le coordinate delle colonne a cui sono riferiti gli oggetti (ad esempio una base missilistica). Un sistema per effettuare dette variazioni e' quello di adottare gli statements IF...THEN, molto piu' comodi che non l'uso di operatori logici. Nella routine e' combinato anche il controllo per evitare che la base missilistica non provochi errori portandosi all' esterno dei margini dello schermo.

'p' contiene le coordinate delle colonne della base missilistica.

Versione 1: IF...THEN

```
100 IF INKEY$="8" AND p<31 THEN LET p=p+1  
110 IF INKEY$="5" AND p>0 THEN LET p=p-1
```

Versione 2: Operatori logici

```
100 LET p=p+(INKEY$="8" AND
p<31)-(INKEY$="5" AND p>0)
```

INKEY\$ puo' anche venir impiegato per stabilire limiti di tempo. In questo caso e' sottoposto alla funzione PAUSE la quale puo' essere interrotta con la pressione di un tasto qualsiasi per poter far riprendere al programma il suo regolare svolgimento. Una forma del genere e' usata assai spesso. Eccovi un esempio

```
10
20
.
.
.
.
.
5499 REM Altra partita ?
5500 PRINT AT 5,2;"Vuoi giocare ancora ?";
      AT 7,0;"Hai 5 secondi
per rispondere s/n"
5510 PAUSE 250
5520 IF INKEY$<>"s" THEN PRINT AT 10,13;"O.K.":
      STOP
5530 RUN
```

In unione a GOTO, la funzione INKEY\$ sostituisce efficacemente gli statements IF...THEN come potete verificare dalla parte finale del programma che segue

```
10000000 GOTO 50000
11000000 REM TO programma presente
12000000 PRINT "Programma presente"
13000000 REM TO 1 AND *60:30
14000000 REM Ultimo uomo perso
15000000 GOTO 50000
16000000 REM Istruzioni
17000000 REM Istruzioni presenti
```


errata, anche ad eventuali domande che seguano. Per evitare questo contrattempo e' sufficiente inserire la linea che segue
5590 IF INKEY\$<>" THEN GOTO 5590.

Esaminiamo ora dettagliatamente la funzione IN.

Innanzitutto IN non rende una stringa bepsi' un numero e poi non esplora in un sol colpo tutta la tastiera ma prende in considerazione blocchi di cinque tasti alla volta. Ogni singolo tasto sottrae un numero funzione della sua posizione nella fila dei cinque, da un secondo numero fisso caratteristico della tastiera quando non risulta azionato alcun tasto.

Prendiamo come esempio la fila coi numeri dall'1 al 5. La locazione di IN e' 63486, mentre il numero caratteristico e', di solito, 255.

Il tasto "1" sottrae 2 elevato a 0=1; per cui, nel caso sia questo l' unico tasto azionato, il numero reso sara' 255-1=254.

Il "2" sottrae 2 elevato a 1=2; per cui da solo rende 255-2=253. Similmente il "3" sottrae 2 elevato a 2=4; il "4" sottrae 2 elevato alla 3=8 e "5" sottrae 2 elevato alla 4=16.

Nel caso in cui ne venga premuto piu' di uno, si avranno le varie combinazioni. In presenza di una azione simultanea su tutti e cinque i tasti, IN 63486 rendera' :

255-16-8-4-2-1=244.

Eccovi la serie completa delle locazioni cui si riferisce la funzione IN :

IN 63486	: tasti 1,2,3,4,5
IN 64510	: tasti Q,W,E,R,T
IN 65022	: tasti A,S,D,F,G
IN 65278	: tasti Caps Shift,Z,X,C,V
IN 61438	: tasti 6,7,8,9,0
IN 57342	: tasti Y,U,I,O,P
IN 49150	: tasti H,J,K,L,ENTER
IN 32766	: tasti B,N,M,Symbol Shift,Break/Space

I primi quattro blocchi, si trovano nella meta' sinistra della tastiera ed i relativi valori vanno considerati in ordine di importanza man mano che si va verso l' interno, per cui i tasti verso il bordo hanno valore 1 e quelli posti al centro 16. Per i rimanenti quattro blocchi, il discorso, ovviamente, si inverte.

Per essere piu' chiari, la cifra 1 viene sottratta alla pressione dei tasti "1","Q","p" e "0" come invece la 16 riguarda "G","5","6" e "B".

Un handicap di IN sta' nel fatto che il numero caratteristico non sempre e' 255, per cui possono venir resi anche numeri errati.

Un metodo attendibile per riportare ogni volta il valore caratteristico a 255, e' quello di inserire un BEEP prima della lettura della tastiera.

In linea di massima, e' piu' semplice usare INKEY\$ anche se in alcuni casi particolari, come appunto l' azionamento simultaneo di due tasti, la funzione IN si dimostra indispensabile.

Per verificare la versatilita' di IN, provate a far girare il programma che segue riguardante un semplice "Sketchpad" che vi

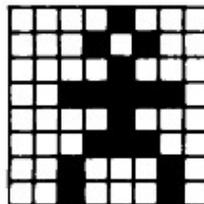
DIS. MUQU. CANC.

J O E .

GRAFICA DEFINITA DALL' UTENTE

Una delle peculiarita' dello Spectrum e' la possibilita' di creare facilmente nuovi caratteri. Ogni carattere, disposto in una griglia 8*8 pixel, e' formato dalla combinazione di 64 punti ciascuno dei quali puo' essere "pieno" o "vuoto" (settato o resettato) a seconda che debba essere visibile o meno.

Considerate l' omino che segue :



Come si puo' notare, e' composto in un array (matrice) di 8*8 punti. Il primo problema da risolvere e' come inserirlo nel computer e visto che questo accetta solo numeri, vediamo di eseguire la trasformazione necessaria.

Il carattere viene definito da 8 numeri : uno riferito alla prima riga, un secondo riferito alla seconda riga e cosi' via. Il comando interessato e' BIN, forma tronca che sta per "binary", accompagnato da un numero binario formato da tanti zeri quanti sono gli spazi e da tanti 1 quanti sono i punti. Gli otto numeri dell' omino di cui sopra sono quindi :

```
I   riga - BIN 00001000
II  riga - BIN 00010100
III riga - BIN 00001000
IV  riga - BIN 00111110
V   riga - BIN 00001000
VI  riga - BIN 00011100
VII riga - BIN 00100010
VIII riga - BIN 00100010
```

Il manuale riporta che BIN 00000000 puo' venir sostituito da 0 il quale e', evidentemente il suo equivalente decimale, ma non dice che anche qualsiasi altro numero binario ha un suo sostituto in decimale, per cui, se preferite operare in decimale, eseguite ogni volta la conversione b/d. Se andate di fretta, non avrete da far altro che battere il comando diretto PRINT BIN seguito dal numero binario per ricevere all'istante dal computer l'equivalente decimale. Se invece avete un attimo di tempo, soffermatevi a leggere quanto segue sul come ricavarli a mano.

DATA 129,90,60,90,60,102,153,195

DATA 129,66,60,219,165,24,36,195

DATA 195,165,24,60,90,36,24,231

DATA 129,102,60,255,90,255,129,231

DATA 60,126,219,219,255,126,66,102

Per gli invaders in movimento scrivere prima una versione e quindi l' altra presentandole alternativamente :

DATA 60,126,235,235,255,126,36,231

DATA 60,126,215,215,255,126,66,102

DATA 66,36,189,173,255,60,36,231

DATA 36,36,60,247,189,189,36,36

DATA 129,189,189,173,239,60,36,231

DATA 231,165,189,181,52,255,129,231

DATA 0,24,60,90,126,24,36,90

DATA 60,90,126,60,24,36,90,0

DATA 198,34,20,42,28,34,17,230

DATA 99,68,40,84,56,68,136,103

Ora gli "alieni", ben conosciuti dagli appassionati di giochi spaziali. Con questo sistema l' effetto si crea presentando un carattere alternato ad altri due :

DATA 60,126,255,170,170,255,126,60

A sinistra :

DATA 3,7,15,10,10,15,7,3

A destra :

DATA 192,224,240,160,160,240,224,192

Altro metodo, e' quello di stampare gli alieni (tutti lunghi non piu' di due caratteri) di seguito nell' ordine. Si ottiene un effetto assai realistico specialmente a velocita' sostenute.

Primo "alieno"

Sinistra : DATA 63,127,255,73,73,255,127,63
Destra : DATA 252,254,255,36,36,255,254,252

Secondo "alieno":

sinistra : DATA 63,127,255,36,36,255,127,63
Destra : DATA 252,254,255,146,146,255,254,252

Terzo "alieno"

Sinistra DATA 63,127,255,146,146,255,127,63
Destra DATA 252,254,255,73,73,255,254,252

Basi missilistiche

DATA 24,24,24,60,126,255,219,219

DATA 0,24,24,24,255,255,255,255

DATA 24,24,24,60,255,255,255,0

Sinistra DATA 1,1,7,9,31,127,255,255
Destra DATA 0,0,192,32,240,252,254,254

Vediamo qualche astronave

Sinistra : DATA 0,252,32,33,18,127,15,1
Destra : DATA 0,0,240,8,4,228,255,248

Sinistra : DATA 31,32,79,255,255,79,32,31
Destra : DATA 0,0,60,235,235,60,0,0

Sinistra : DATA 0,248,32,16,15,63,15,3
Destra : DATA 0,0,0,224,16,8,207,248

Alcuni asteroidi assomigliano ad astronavi. Sono formati da due disegni uno per la sagoma orizzontale-verticale, l'altro per quella diagonale.

Rivolto verso nord-ovest :
DATA 192,176,76,67,44,40,16,16

Rivolto verso ovest :
DATA 0,7,25,98,132,98,25,7,

Rivolto verso sud-ovest :
DATA 16,16,40,44,67,76,176,192

Rivolto verso sud :
DATA 198,170,146,68,68,40,40,16

Rivolto verso sud-est :
DATA 8,8,12,44,194,50,13,3

Rivolto a est :
DATA 0,224,152,70,33,70,152,224
Rivolto a nord-est :
DATA 3,13,50,194,44,12,8,8
Rivolto a nord :
DATA 16,40,40,68,68,146,170,198

I prossimi grafici sono aerei.

Un tipo di Jet
Sinistra : DATA 0,0,224,112,120,255,60,15
Destra : DATA 0,0,0,0,224,248,30,248

Un modello normale :
Sinistra : DATA 192,192,227,158,127,0,0,0
Destra : DATA 0,228,244,60,244,36,80,32

Un secondo tipo di Jet
Sinistra : DATA 14,199,224,255,127,1,0,0
Destra : DATA 0,0,96,248,252,224,240,112

Un modello particolare quale usato nel programma "Olocausto".

Sinistra : DATA 192,224,240,248,143,255,15,1
Centrale : DATA 0,7,9,17,255,255,224,255
Destra : DATA 0,192,32,16,254,255,30,248

Le prossime otto file di dati si riferiscono ad un aereo inclinato in ognuna dello otto direzioni della rosa dei venti.

Verso nord :
DATA 8,8,28,62,127,8,8,28

Verso nord-est :
DATA 1,2,124,60,28,172,68,32

Verso est :
DATA 16,24,156,255,156,24,16,0

Verso sud-est :
DATA 32,68,172,28,60,124,2,1

Verso sud :
DATA 28,8,8,127,62,28,8,8

Verso sud-ovest :
DATA 4,34,53,56,60,62,64,128

Verso nord-ovest :
DATA 128,64,62,60,56,53,34,4

Se preferite stare con i piedi per terra, eccovi una sequenza di carri armati e di navi.

Sinistra DATA 0,16,31,31,63,127,63,0
Destra DATA 0,0,252,0,240,248,240,0

Sinistra DATA 0,0,63,0,15,31,15,0
Destra DATA 0,8,248,248,252,254,252,0

DATA 0,0,64,63,112,126,255,126
DATA 0,0,20,252,14,126,255,126

Nave da guerra rivolta a sinistra

Sinistra DATA 0,0,2,115,23,255,126,63
Destra DATA 192,192,224,239,236,255,255,254

ora rivolta a destra

Sinistra DATA 3,3,7,247,55,255,255,127
Destra DATA 0,0,64,204,232,255,254,252

Ecco ora i dati per ottenere un carro armato direzionato in vari modi.

Verso nord :
DATA 8,73,73,93,127,93,65,65

Verso nord-est :
DATA 33,66,188,60,60,29,2,4

Verso est :
DATA 254,16,56,63,56,16,254,0

Verso sud-est :
DATA 4,2,29,60,60,188,66,33

Verso sud :
DATA 65,65,93,127,93,73,73,8

Verso sud-ovest :
DATA 32,64,184,60,60,61,66,132

Verso ovest :
DATA 127,8,28,252,28,8,127,0

Verso nord-ovest :
DATA 132,66,61,60,60,184,64,32

Concludendo la selezione dei semoventi terrestri ecco qualche auto :

In alto a sinistra : DATA 31,31,4,5,55,59,55,6
In alto a destra : DATA 248,248,32,160,236,220,236,96

In basso a sinistra: DATA 6,6,7,19,31,19,1,7
In basso a destra : DATA 96,96,224,200,248,200,128,224

Sinistra : DATA 224,159,119,239,239,119,159,224
Destra : DATA 28,242,190,191,191,190,242,28

DATA 0,66,189,239,239,189,66,0

DATA 102,153,239,223,223,239,153,102

Lanciamo le bombe

DATA 32,16,160,92,30,31,15,7

DATA 54,28,8,28,62,62,28,8

Qui' di seguito trovate alcuni "pacman" disegnati in modo che ognuno dei primi quattro si alterni col quinto in funzione della propria direzione.

Pacman rivolto a destra
DATA 60,127,252,240,240,252,127,60

Pacman rivolto in su'
DATA 66,66,231,231,255,255,126,60

Pacman rivolto a sinistra
DATA 60,254,63,15,15,63,254,60

Pacman rivolto in giu'
DATA 60,126,255,255,231,231,66,60

Pacman con la bocca chiusa
DATA 60,126,255,255,255,255,126,60

Spettro
DATA 56,124,214,214,254,254,170,170

Ora un po' di frutta

Fragola : DATA 24,82,247,255,255,126,60,24
Lampone : DATA 4,8,8,86,171,213,106,60
Ciliegia : DATA 8,8,20,20,34,99,243,96
Pesca : DATA 44,110,231,247,247,247,102,44
Banana : DATA 2,3,7,14,30,124,248,0
Prugna : DATA 8,16,24,60,124,62,60,24
Mela : DATA 24,82,255,255,255,255,126,36

Per concludere la parte della grafica definita dall'utente, presentiamo una serie alternativa di numeri riferiti a forme spesso usate in fasi particolari dei giochi TV.

Un esempio e' il punteggio, inserito in una variabile di stringa con STR\$ e quindi scelto per mezzo di un loop tra i vari caratteri, l'ultimo dei quali determinato da LEN. Il codice (CODE) di ciascun carattere viene determinato aggiungendo il numero corrispondente alla differenza tra il CODE del digit interessato e il CODE della grafica stessa definita dall'utente.

Ma poiche' gli esempi valgono da sempre piu' delle parole battute la routine che segue. La variabile "S" contiene il punteggio mentre gli user-graphic da 0 a 9 derivano dalle lettere da A e J impostate in "graphic mode". La variabile S# contiene il punteggio da ricavare dalla serie di numeri.

```

40000 LEFT $$=STR$ a
40010 FOR i=1 TO LEN S$
40020 LEFT S$(i)=CHR$(CODE S$(i)+
40030 NEXT i
40040 RETURN

```

Eccovi i grafici :

```

0 DATA 0,60,66,66,98,98,98,60
1 DATA 0,8,8,8,12,12,12,12,
2 DATA 0,124,2,2,60,96,96,62
3 DATA 0,124,2,2,60,6,6,126
4 DATA 0,64,96,100,100,126,4,4
5 DATA 0,62,64,64,60,6,6,126
6 DATA 0,62,64,64,124,70,70,126
7 DATA 0,120,8,8,12,12,12,12
8 DATA 0,60,66,66,60,70,70,60
9 DATA 0,60,66,66,60,6,6,126

```

Dopo aver esaminato alcuni possibili caratteri, vediamo il programma sotto riportato che vi permettera' di effettuare la definizione per conto vostro :

```

9500 PRINT TAB 7;"Generatore di
generat tteri"
9510 PRINT ("per creare grafici
definiti da ll'utente su una matr
ice di 0x0")

```

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 480. 481. 482. 483. 484. 485. 486. 487. 488. 489. 490. 491. 492. 493. 494. 495. 496. 497. 498. 499. 500. 501. 502. 503. 504. 505. 506. 507. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 633. 634. 635. 636. 637. 638. 639. 640. 641. 642. 643. 644. 645. 646. 647. 648. 649. 650. 651. 652. 653. 654. 655. 656. 657. 658. 659. 660. 661. 662. 663. 664. 665. 666. 667. 668. 669. 670. 671. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 838. 839. 840. 841. 842. 843. 844. 845. 846. 847. 848. 849. 850. 851. 852. 853. 854. 855. 856. 857. 858. 859. 860. 861. 862. 863. 864. 865. 866. 867. 868. 869. 870. 871. 872. 873. 874. 875. 876. 877. 878. 879. 880. 881. 882. 883. 884. 885. 886. 887. 888. 889. 890. 891. 892. 893. 894. 895. 896. 897. 898. 899. 900. 901. 902. 903. 904. 905. 906. 907. 908. 909. 910. 911. 912. 913. 914. 915. 916. 917. 918. 919. 920. 921. 922. 923. 924. 925. 926. 927. 928. 929. 930. 931. 932. 933. 934. 935. 936. 937. 938. 939. 940. 941. 942. 943. 944. 945. 946. 947. 948. 949. 950. 951. 952. 953. 954. 955. 956. 957. 958. 959. 960. 961. 962. 963. 964. 965. 966. 967. 968. 969. 970. 971. 972. 973. 974. 975. 976. 977. 978. 979. 980. 981. 982. 983. 984. 985. 986. 987. 988. 989. 990. 991. 992. 993. 994. 995. 996. 997. 998. 999. 1000.

| | |
|-----------|--|
| Note | |
| 10 | Inizio |
| 20 | Loop per aereo |
| 30 | Predisporre l' aereo successivo |
| 40-70 | Loop per muovere l' aereo |
| 80-100 | Aereo successivo |
| 170 | Fine della gara |
| 3000-3190 | Predisporre la corsa dell'aereo successivo |
| 4000-4020 | Disegna il suolo |
| 4030 | Posizione della casa |
| 4100-4190 | Disegna lo schermo |
| 4500-4510 | Sgancia la bomba |
| 4520 | Esplosione |
| 4530-4540 | Setta le variabili di "yes" e "bomb" |
| 5000-5490 | Inizializzazione |
| 9000-9030 | Grafica definita dall' utente |
| 9100-9260 | Dati per la grafica. |

Volendo, potete anche dotare il vostro Spectrum di un box sonoro (ottimo quello della Tenkolek) per amplificare il segnale presente sulla presa di EAR. Una buona amplificazione si ottiene anche usando da buffer il registratore stesso. Per far cio', collegare un cavetto tra l' uscita EAR dello Spectrum e l' ingresso MIC del registratore dopodiche' dare il "play".

USO DEGLI OPERATORI LOGICI

Gli operatori logici sono :

AND;OR;NOT;=><(diverso da);<>;<=>

Tali funzioni vengono completate dalle espressioni matematiche +, -, * (moltiplicato) e / (diviso). Gli operatori logici possono sostituire IF...THEN come nell'espressione IF A=D THEN LET C=1 equivalente a "A=B". Se l'espressione e' vera, cioe' se 'A' e' realmente uguale a 'B', l'azione prosegue attribuendo, nel nostro caso, a 'C' il valore 1. Analogamente agisce LET C=(A=B), il cui nucleo e' "(A=B)", le parentesi non sono necessarie all'atto pratico ma chiariscono maggiormente il concetto. Il computer valuta anche qui l'uguaglianza e se questa risulta vera, rende a C il valore 1, viceversa, cioe' nel caso in cui A sia diverso da B, rende 0. Facciamo un esempio, supponete che tanto A quanto B valgano 5 :

```
C=(A=B)           (oppure C=A=B)
C=(5=5)
E' 5 uguale a 5 ? Si, per cui
C= VERO
C= 1
```

Si nota chiaramente come, l'espressione logica, operi alla stregua degli statement IF...THEN con la sola differenza che, questi mantengono il precedente valore della variabile e non la pongono a 0 come invece succede usando gli operatori logici se questa risulta falsa.

Il segno "=" non funge solamente da comparatore

```
IF A<>B THEN LET C=1
```

```
LET C=(A<>B)
```

```
IF A<B THEN LET C=1
```

```
LET C=(A<B)
```

```
IF A>B THEN LET C=1
```

```
LET C=(A>B)
```


qui notiamo che la seconda e la terza linea riguardano la stessa funzione come pure la quarta e la quinta. Il loro impiego permette di inserire espressioni in statement relativi a "operazioni logiche". Nell'esempio, LET C=(A=3 AND B=2), il valore di C e' 1 solamente se A e' uguale a 3 e B e' uguale a 2. La funzione OR raggiunge lo stesso risultato con l' aiuto di IF...THEN. NOT invece, inverte il valore dell' espressione permettendo l' esecuzione sia nel caso di risultato vero (1), sia nel caso di risultato falso (0). In altre parole, trasforma gli "1" in "0" e gli "0" in "1".

Visto cio' e' ora possibile comprendere l' uso dei segni +, * e / come operatori logici. Se consideriamo le espressioni facenti parte di IF (espressione) THEN... , se ne deduce che :

"A<B" equivale a "A<>B"

"A+B" equivale a "A OR B". Notate pero' che se A vale 5 e B vale -5, il risultato della somma e' 0 anche se sono entrambi veri.

"A*B" equivale a "A AND B"

"A/B" equivale ad A stesso, a meno che A sia talmente piccolo rispetto a B che il risultato si approssimi a 0, rivelandosi in tal caso falso.

Gli esempi portati finora per A e B valgono naturalmente anche per le stringhe A# e B#.

Come esempio, supponiamo ora di voler "simulare" l' espressione IF A=B THEN LET C=3 con operatori logici. La forma equivalente risultera' :

```
LET C=(X AND espressione)
```

dove "X" e' il numero (o la stringa) e "C" la grandezza a cui viene attribuito il valore "X" se l' espressione e' vera. Nel caso in cui risulti falsa, "C" varra' 0.

Impiegando l' espressione di cui sopra si otterra'

```
LET C=(3 AND A=B)
```

Anche la funzione OR puo' assumere una tale configurazione, naturalmente gli argomenti saranno diversi :

```
LET C=(X OR espressione)
```

"X" e' sempre solo un numero in quanto con OR non si possono usare stringhe. Il valore di C in questo caso risulta 1 con espressione vera e "X" con espressione falsa.

A dimostrazione di cio', fate girare il programma che segue mettendo in pratica quanto avete imparato dal capitolo e analizzando i risultati.

CRASH,
LAND,
SCREEN,
UPDATE,
LOOP

"Tokens" per le subroutines

Le lettere maiuscole alle linee 60,140,160,5000,8540 e 8560 vanno battute in "graphic mode" e servono a generare i caratteri richiesti.

MOVIMENTO

Gran parte dei games, per non dire tutti, posseggono oggetti in movimento che rappresentano figure spesso bizzarre come i "pacmen" e le navi spaziali. Le difficoltà piu' evidenti nello scrivere programmi nascono dal fatto che :

- a) il BASIC non e' sufficientemente veloce per muovere contemporaneamente diverse immagini,
- b) l' animazione e' a scatti; piu' a lungo una immagine si ferma sullo schermo e peggio e',
- c) gli oggetti non devono travalicare i confini imposti dallo schermo,
- d) le varie parti non possono "sapere" in che direzione spostarsi; ad esempio gli invasori spaziali si muovono solo a destra e a sinistra,
- e) il computer deve esplorare lo schermo per rilevare le eventuali collisioni di due oggetti (ad esempio, la racchetta con la palla).

Tutto cio' si verifica su uno scenario di sottofondo mantenuto fisso. L' oggetto viene stampato (PRINT) o disegnato (PLOT) in relazione ad una serie di coordinate x e y, quindi cancellati e ridisegnati con una o con tutte e due le coordinate cambiate di valore il che dona l' illusione del movimento.

L' effetto dello spostamento a scatti e' appunto dovuto al fenomeno di cancellazione-ristampa e la sua intensita' dipende dal tempo intercorrente tra le due azioni. Va da se' che piu' rapidamente la figura viene ripresentata dopo essere stata cancellata, piu' si ha la sensazione di un movimento costante. Il programma che segue, stampa un quadratino nero al centro e lo sposta fino alla fine dello schermo :

```

10 GOTO 30
20 GOTO 30
30 GOTO 30
40 GOTO 30
50 GOTO 30
60 GOTO 30
70 GOTO 30
80 GOTO 30
90 GOTO 30
100 GOTO 30
110 GOTO 30
120 GOTO 30
130 GOTO 30
140 GOTO 30
150 GOTO 30
160 GOTO 30
170 GOTO 30
180 GOTO 30
190 GOTO 30
200 GOTO 30
210 GOTO 30
220 GOTO 30
230 GOTO 30
240 GOTO 30
250 GOTO 30
260 GOTO 30
270 GOTO 30
280 GOTO 30
290 GOTO 30
300 GOTO 30
310 GOTO 30
320 GOTO 30
330 GOTO 30
340 GOTO 30
350 GOTO 30
360 GOTO 30
370 GOTO 30
380 GOTO 30
390 GOTO 30
400 GOTO 30
410 GOTO 30
420 GOTO 30
430 GOTO 30
440 GOTO 30
450 GOTO 30
460 GOTO 30
470 GOTO 30
480 GOTO 30
490 GOTO 30
500 GOTO 30
510 GOTO 30
520 GOTO 30
530 GOTO 30
540 GOTO 30
550 GOTO 30
560 GOTO 30
570 GOTO 30
580 GOTO 30
590 GOTO 30
600 GOTO 30
610 GOTO 30
620 GOTO 30
630 GOTO 30
640 GOTO 30
650 GOTO 30
660 GOTO 30
670 GOTO 30
680 GOTO 30
690 GOTO 30
700 GOTO 30
710 GOTO 30
720 GOTO 30
730 GOTO 30
740 GOTO 30
750 GOTO 30
760 GOTO 30
770 GOTO 30
780 GOTO 30
790 GOTO 30
800 GOTO 30
810 GOTO 30
820 GOTO 30
830 GOTO 30
840 GOTO 30
850 GOTO 30
860 GOTO 30
870 GOTO 30
880 GOTO 30
890 GOTO 30
900 GOTO 30
910 GOTO 30
920 GOTO 30
930 GOTO 30
940 GOTO 30
950 GOTO 30
960 GOTO 30
970 GOTO 30
980 GOTO 30
990 GOTO 30

```

Provate lo e vedrete il quadratino lampeggiare scorrendo verso destra. Come si puo' migliorare un tale effetto ? Un rimedio e' quello di inserire il calcolo di avanzamento, di cui alla linea 40, tra il disegno e la cancellazione dell' oggetto il quale, pertanto, si ferma sullo schermo il tempo necessario al calcolo delle sue nuove coordinate.

Il risultato si ha col programma che segue in cui e' stata prevista una temporanea memorizzazione (linea 20) del vecchio valore di "y" per non far stampare erroneamente la linea 30 :

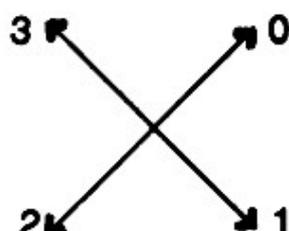
Queste proprietà vengono usate in molti giochi tra cui lo "squash" e' quello che le evidenzia maggiormente. Il nome della variabile usata per la direzione e' di solito "d" oppure "dir" e, nell'esempio dell'invader a "V", assume due valori ben precisi :

- a) 1, nel qual caso l' incremento vale 1 e l' oggetto si sposta verso destra,
- b) -1, in cui si ha un decremento con l' oggetto che si sposta verso sinistra.

Sempre nello stesso esempio, il valore del "marker" di direzione viene inserito direttamente nel calcolo :

```
LET Y=Y+direzione
```

e una tale forma puo' essere usata in qualsiasi momento per indicare dove l' oggetto deve prendere posto ma non per calcolare il movimento. Il valore del "marker" e' compreso tra 0 e 3 ed i numeri singoli si riferiscono alle direzioni :



E' molto difficile manipolare detti numeri per inserirli direttamente nei calcoli per il movimento, pero' e' possibile ricorrere all' aiuto degli operatori logici visti un capitolo addietro. Guardate :

'x' e' la coordinata x (\downarrow)

'y' e' la coordinata y (\rightarrow)

'd' e' il puntatore di direzione con i valori corrispondenti al diagramma.

Cosa succede a 'x' e 'y' per diversi valori.

```
1000 LET x=x+(d=1 OR d=2)-(d=0 OR d=3)
```

```
1010 LET y=y+(d<2)-(d>1)
```

Se un oggetto, nel suo spostamento urta una linea orizzontale (vedremo come piu' avanti analizzando SCREEN\$ e ATTR), puo', sempre usando gli operatori logici, rimbalzare cambiando direzione :

```

10000 REM Urto della linea orizzo
10001 LET d=d+1-(2 AND (d#3 OR d#
1))

```

Se non siete del tutto convinti che una sola linea possa cambiare correttamente i valori di "d", battete quanto segue e confrontatelo col diagramma :

```

? " 1 INPUT "IL valore di 'd' e'
10001 INPUT " " ; d
10002 PRINT " " ; d
10003 INPUT " " ; d
10004 PRINT " " ; d
10005 INPUT " " ; d
10006 PRINT " " ; d
10007 INPUT " " ; d
10008 PRINT " " ; d
10009 INPUT " " ; d
10010 PRINT " " ; d

```

Un altro metodo, si presenta con l'adozione delle stringhe di variabili e di VAL. Per vedere l'effetto di dette funzioni, battete questo programma dimostrativo :

```

10001 INPUT " " ; N
10002 PRINT " " ; N
10003 INPUT " " ; N
10004 PRINT " " ; N
10005 INPUT " " ; N
10006 PRINT " " ; N
10007 INPUT " " ; N
10008 PRINT " " ; N
10009 INPUT " " ; N
10010 PRINT " " ; N
10011 INPUT " " ; N
10012 PRINT " " ; N
10013 INPUT " " ; N
10014 PRINT " " ; N
10015 INPUT " " ; N
10016 PRINT " " ; N
10017 INPUT " " ; N
10018 PRINT " " ; N
10019 INPUT " " ; N
10020 PRINT " " ; N
10021 INPUT " " ; N
10022 PRINT " " ; N
10023 INPUT " " ; N
10024 PRINT " " ; N
10025 INPUT " " ; N
10026 PRINT " " ; N
10027 INPUT " " ; N
10028 PRINT " " ; N
10029 INPUT " " ; N
10030 PRINT " " ; N
10031 INPUT " " ; N
10032 PRINT " " ; N
10033 INPUT " " ; N
10034 PRINT " " ; N
10035 INPUT " " ; N
10036 PRINT " " ; N
10037 INPUT " " ; N
10038 PRINT " " ; N
10039 INPUT " " ; N
10040 PRINT " " ; N
10041 INPUT " " ; N
10042 PRINT " " ; N
10043 INPUT " " ; N
10044 PRINT " " ; N
10045 INPUT " " ; N
10046 PRINT " " ; N
10047 INPUT " " ; N
10048 PRINT " " ; N
10049 INPUT " " ; N
10050 PRINT " " ; N
10051 INPUT " " ; N
10052 PRINT " " ; N
10053 INPUT " " ; N
10054 PRINT " " ; N
10055 INPUT " " ; N
10056 PRINT " " ; N
10057 INPUT " " ; N
10058 PRINT " " ; N
10059 INPUT " " ; N
10060 PRINT " " ; N
10061 INPUT " " ; N
10062 PRINT " " ; N
10063 INPUT " " ; N
10064 PRINT " " ; N
10065 INPUT " " ; N
10066 PRINT " " ; N
10067 INPUT " " ; N
10068 PRINT " " ; N
10069 INPUT " " ; N
10070 PRINT " " ; N
10071 INPUT " " ; N
10072 PRINT " " ; N
10073 INPUT " " ; N
10074 PRINT " " ; N
10075 INPUT " " ; N
10076 PRINT " " ; N
10077 INPUT " " ; N
10078 PRINT " " ; N
10079 INPUT " " ; N
10080 PRINT " " ; N
10081 INPUT " " ; N
10082 PRINT " " ; N
10083 INPUT " " ; N
10084 PRINT " " ; N
10085 INPUT " " ; N
10086 PRINT " " ; N
10087 INPUT " " ; N
10088 PRINT " " ; N
10089 INPUT " " ; N
10090 PRINT " " ; N
10091 INPUT " " ; N
10092 PRINT " " ; N
10093 INPUT " " ; N
10094 PRINT " " ; N
10095 INPUT " " ; N
10096 PRINT " " ; N
10097 INPUT " " ; N
10098 PRINT " " ; N
10099 INPUT " " ; N
10100 PRINT " " ; N

```

La collisione di due oggetti, come ad esempio un missile ed un invasore spaziale, viene rilevata facilmente testando la variabile dei due oggetti stessi : IF X missile=X invader AND Y missile=Y invader THEN...

Ciononostante, la via piu' facile, resta quella di esplorare direttamente lo schermo per mezzo di uno dei tre sistemi sottoelencati (anche se il terzo lo sconsigliamo) :

1) SCREEN#. Questa funzione rende una stringa e va inserita con la forma "SCREEN# (x,y)"dove "x" e "y" rappresentano le coordinate cosi'come per lo statement PRINT AT.Non lavora con gli UDG,ma solo con i caratteri che ritornano normali anche nel caso in cui siano inversi.

2) ATTR. Rende un numero che dipende dagli "Attributi" della cella definita dalle coordinate. Gli attributi sono il colore, il flashing e il brightness. Il relativo statement e' "ATTR (x,y)" della medesima fisionomia del precedente.

ATTR e' la somma di :

128 (10000000 binario) se il carattere lampeggia con FLASH 1,
64 (01000000 binario) se il carattere ha luminosita' piu'
accentuata con BRIGHT 1.

0 il colore di PAPER che occupa il terzo, il quarto e il
quinto posto dell' equivalente binario.

ATTR e' molto usato, in quanto non distingue se il carattere
che sta esaminando e' un UDG o meno.

3) POINT. Rende 1 oppure 0. La forma del suo statement e'
classica "POINT (x,y) in cui "x" e "y" vengono impiegati come
in PLOT x,y. Il risultato della funzione e' 1 se le coordinate
'pescano' un punto, mentre e' 0 se riscontrano un vuoto.

Vediamo ora la velocita'. Deve essere tale da mettere in
condizione il giocatore di poter sparare agli invaders che
scendono spostandosi casualmente a destra o a sinistra. Sia la
base che gli alieni, possono uscire dai margini laterali dello
schermo per riapparire dal lato opposto nella stessa posizione.
Gli invaders, quattro in tutto, vengono presentati dal computer
uno per volta rendendo la gara assai interessante anche perche'
a loro volta possono sparare alla base.

Le coordinate di ogni invader, sono memorizzate in due arrays,
l' array "a" per le coordinate verticali, il "b" per le
coordinate orizzontali. I suoi movimenti sono funzione di
quelli della base e vengono determinati dalla variabile "i"
presente nella linea 20. Fatta questa doverosa presentazione
eccovi il programma con relativa documentazione.

Elenco delle variabili

| | |
|---------------|---|
| h | Record |
| a\$(.) | Array di caratteri contenente la forma degli
alieni. |
| e\$(.) | Array contenente l' immagine dell' esplosione. |
| n | Punti. |
| men | Numero delle basi rimaste. |
| a(.),
b(.) | Coordinate x e y degli invasori. |
| p | Numero della colonna della base. |
| i | Usato nel loop principale per il movimento degli
alieni. |
| il | Usato per calcolare "i". |
| t1 | Numero di pixel della base nello sparo. |
| dr | Sviluppa la traiettoria in salita dello sparo
della base. |
| | Usato per calcolare se l' invader e' stato
colpito. |
| t2,
t1 | Numero di pixel delle coordinate x e y quando
spara l' alieno. |
| dr | Sviluppa la traiettoria in discesa dello sparo
degli alieni. |
| x,
i, | |

n Variabili per il controllo dei loop.
 m,
 c# Usate per la fine della gara.

Le lettere maiuscole nelle linee 140,1010,1020,1040,1510,1520, 7710,7720 riferiti agli statement di PRINT sono tutti caratteri definiti dall'utente(UDG)

Note

10 Inizializza le variabili e scrive le istruzioni.
 20-100 Parte del loop per muovere gli alieni e per sparare.
 105-160 Parte del loop per muovere la base e per sparare
 1000-1060 Calcolo del fuoco
 1400-1420 Fuoco
 1500-1540 Calcolo del fuoco
 1900-1920 Fuoco
 2000 Incrementa il punteggio
 3000-3020 Esplosione della base
 4000-4070 Fine di una parte
 4080-4180 Fine della gara
 7000-7100 Scrive le istruzioni
 7500-7530 Inizializza le variabili permanenti come ad esempio il punteggio.
 7600-7800 Inizializza le altre variabili e disegna lo schermo
 9000-9030 Definisce i caratteri UDG
 9050-9150 Dati per gli UDG
 9500 Dati per la fine
 9990-9999 Fine.

```

10 GO TO 7000: REM istr/iniz
20 LET i1=i: LET i=i+1-(4 AND
i1) GO +3
30 IF a(i) >= 2 THEN PRINT AT a(
i) (i) (i) : a(i) = a(i) + 1: LET b(i) = b
(i) + 1 + INT (RND*3)
40 IF a(i) = 30 THEN LET b(i) = 0
50 IF a(i) = 17 THEN LET b(i) = 31
60 IF a(i) = 17 OR a(i) = 18 THEN
PRINT AT a(i), 0; " "
70 IF a(i) = 0 THEN GO SUB 3000
80 IF a(i) >= 2 THEN PRINT AT a(
i) (i) (i) : a(i) = a(i)
90 IF a(i) >= 2 AND RND > .9 THEN
GO SUB 1000: REM SPARO
100 IF a(i) = 0 THEN GO TO 4000: R
EM FIN
110 PRINT "INIZIO" - (INKEY# = "1")
120 LET P=P+
(INKEY# = "N") - (INKEY# = "1")

```


v,
i,
l Variabili di controllo per i loops ecc.

I caratteri grafici da definire battendo le maiuscole in "graphic mode", trovano posto nelle linee :
300,"B",310,"U","D","L","R",330,"0",510,"A",1200,"C",5000,"0".

Note

10 Inizializzazione e istruzioni
20-50 Tastiera
60-70 Testa per vedere se qualcosa e' stato colpito
90 Tratta il limite di tempo
100-110 Tratta la coda
120 Inserisce i valori nell' array
300-310 Subroutine per il disegno della testa
500-550 Disegna una nuova mela, il punteggio e suona il motivetto
1000-1010 Predispone i record
1020-1050 Istruzioni
1060-1080 Istruzioni circa i limiti di tempo

Nel caso possediate uno Spectrum a 16K, dovete apportare le modifiche che seguono :

Linee 20,30,110,520,5020,1170,1180. Cambiare ogni numero 500 in 180 e ogni 499 in 179. Se non fate cio' il programma impegna troppa memoria e si blocca con l' errore 4 "Out of memory".

1090-1150 Selezione dei livelli di abilita'
1160 Colori, che potete anche variare
1170-1190 Inizializzazione
1200-1220 Disegna lo schermo
4000-4020 Definisce gli UDG
4030-4100 Dati per gli UDG
5000-5050 Non disegna la coda alla fine della gara
5100-5110 Punti
5120 Presenta il record precedente se non e' stato battuto
5130-5240 Presenta il nuovo record nel caso sia stato ottenuto
5250-5320 Presenta la tavola dei record e il "bit per la fine"
9000 Dati per il motivetto

Serpenti

```
10 GO SUB 4000: GO TO 1000
20 LET h=h-1: IF h=0 THEN LET
30 h=3000
40 LET j=j-1: IF j=0 THEN LET
50 j=3000
60 LET a$=INKEY$: IF a$>"4" AN
70 d$a<"0" THEN LET d$a=a$
```



```

0000 110 DATA 1,2005,20,17,2005,00,00,00,00,00
0004 0004,0007,0010,0014,0017,0020,0024,0027,0030
0000 0004,0007,0010,0014,0017,0020,0024,0027,0030

```

Uno dei pochi svantaggi dello scrolling, e' che transla tutto quanto riportato sullo schermo ivi compreso il punteggio, che, come si sa, permane costantemente nella parte alta.

Pertanto, usando un tale effetto, e' necessario memorizzare il risultato senza presentarlo se non alla fine del gioco.

Ed ora eccovi alcuni interessanti programmi basati sullo scrolling. Iniziamo da "Grand Prix". Siete alla guida di un bolide blu attraverso un tracciato di cui dovete evitare i bordi e nello stesso tempo le auto piu' lente che vi si presentano davanti. Lo spostamento coi soliti tasti 5 e 8.

Elenco delle variabili

| | |
|------|---|
| p | Coordinate della colonna dell' auto del giocatore |
| s | Punti |
| x(.) | Usato nella cancellazione del fumo |
| car | Posizione prossima dell' auto da corsa |
| ink | Colore dell' auto da corsa |
| rnd | Numero random per lo snodarsi della pista |
| life | Auto a disposizione (inizialmente 5) |
| l | Coordinata della colonna della pista |
| i, | |
| n | Variabili di controllo dei loop |
| s, | |
| END, | |
| LOOP | Memorizzano i numeri delle subroutines |

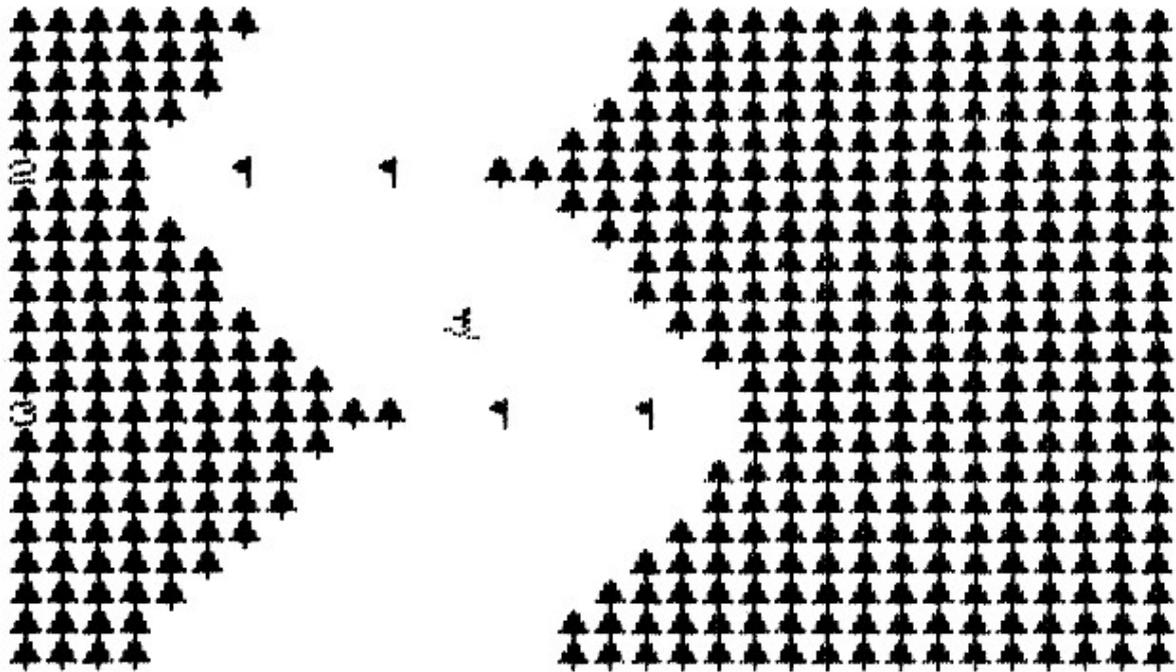
Gli UDG presenti in questo programma, corrispondono alle lettere maiuscole di cui alle linee :

30,60,110,130,2000 e 5120.

```

1000 GO SUB 5000: REM INIZ
1010 REM ***ROUTINE PRINCIPALE**
1020 REM OKE 20002,-1: PRINT : REM
1030 PRINT INK 0; AT 9,P; "FG": PR
1040 AT 10,P; " "
1050 LET P=P+(INKEY#="8")-(INKEY
1060 IF ATTR (11,P) <> 110 OR ATTR
1070 (11,P+1) <> 110 THEN GO TO END
1080 INK 1; AT 10,P; "AB"; AT
1090 P
1100 REM CHZ$ = S + 1
1110 REM INZ$ = AT 7,X(1); " "
1120 REM X(1) = X(2): LET X(2) = X(3)
1130 REM X(3) = P
1140 REM C = INT (RND*9+L+2): LE
1150 T = INT (RND*4+2)

```

- z# Stringa contenente una fila intera di alberi presentata con dei PRINT
- s(,) Array relativo al motivetto
- m Si alterna tra 0 e 8 per suonare la prima o la seconda meta' della melodia
- e Marker per rilevare se lo sciatore si e' schiantato o no
- s Loop che predispone il conteggio delle porte
- g# Stringa contenente la porta. Contiene i caratteri di controllo del colore. Lo sciatore deve passarvi in mezzo ma non attorno
- n, Usate come variabili di controllo dei loops
- i



Caratteri UDG sono presenti alle linee 310-graphic B; 320 graphic E; e 5120-graphic D.

USO DEGLI ARRAY

Gli array (matrici) sono delle variabili e come tali vanno usati anche se e' possibile affidargli diversi altri compiti. Esistono, come per le variabili, due tipi di array : quelli numerici che contengono unicamente numeri e quelli di caratteri che memorizzano lettere o stringhe. Per comprenderne bene la struttura, immaginateli disposti come una tabellina divisa in settori ciascuno dei quali ospita un numero o un carattere. Prima di poterli usare, vanno dimensionati per mezzo dello statement DIM :

```
10 DIM A (500)
```

In tal modo si prenota lo spazio in memoria per la "tabella" di nome A, la quale inizialmente ha tutti i suoi elementi a 0. Il nome e' seguito da un numero racchiuso tra parentesi chiamato "subscript" della variabile. Analogamente avviene per gli array di caratteri.

```
10 DIM A# (500)
```

significa che l'array A e' formato da 500 elementi ognuno dei quali contiene un carattere che inizialmente e' uno spazio. Provate ad aggiungere alla linea di cui sopra :

```
20 LET A# (1)="Ciao"
```

e quindi date il RUN. Fatto cio' inserite il comando diretto, PRINT A# (1) e vedrete che la risposta sullo schermo non sara' "Ciao" bensì solo "C" in quanto ogni singolo elemento non puo' memorizzare piu' di un carattere. Volendo pertanto sistemare "Ciao" in un array dovrete ricorrere a piu' di una dimensione come dimostrato facilmente da :

```
10 DIM A#(100,5)
```

In questo modo avrete dimensionato una matrice da 100*5=500 elementi in totale. Battete ora :

```
20 LET A#(1)= "Ciao"
```

fatelo girare e quindi date PRINT A#(1). A questo punto la risposta sara' "Ciao" in virtu' del fatto che A#(1) viene considerato come un array a se' stante lungo 5 caratteri. L'analisi di ogni sua parte e' facilmente ottenibile studiando lo statement :

```
30 PRINT A#(1),A#(1,3)
```



```

00000 NEXT n: NEXT i: RETURN
00100 DATA 102,102,24,129,126,0,3
00200 1
00110 DATA 51,51,204,204,51,51,20
4,204

```

Gli UDG sono insiti nel listato alle linee 50-graphic A, 90-graphic B, 5320-graphic A.

Ogni giocatore puo' scommettere su quale pulce vincera'; se entrambi sbagliano, perdono la cifra puntata mentre se vincono incrementano il bottino.

Le matrici possono essere impiegate anche come "marker" dello stato di qualsiasi cosa presente sullo schermo. Lo vedremo tra poco nell'impegnativo programma "ICBM".

Gli array di stringhe trovano particolare uso nella presentazione del display. Battete il programmino che segue il quale, anche senza l'uso di array, da' l'impressione del movimento continuo :

```

10 LET a$="████████████████████"
20
30 LET i=1
40 FOR R=0: PAPER 0: CLS
50 PRINT AT 10,1;a$(i TO 30);a
#(1 TO i)
60 LET i=i+1-(30 AND i=30)
70 GO TO 50

```

Sullo stesso principio, il prossimo che di linee in movimento ne presenta tre :

```

10 LET a$="████████████████████"
20
30 LET i=1
40 FOR R=0: PAPER 0: CLS
50 FOR R=7: INK 1
60 PRINT AT 0+R*2,1;a$(i TO 30)
);a$(1 TO i)
70 NEXT i
80 LET i=i+1-(30 AND i=30)
90 GO TO 50

```

Piu' difficile si presenta il discorso quando le tre linee devono scorrere in direzioni differenti. Il programmino che segue impiega due variabili denominate "i" per sfilare la linea di centro verso destra e le due laterali verso sinistra.

```

10 LET a$="████████████████████"
20 LET i l=1: LET i r=1

```


| | |
|-----------|-------------------------------------|
| 70 | Disegna il serpente e i blocchi |
| 80 | Musica |
| 90 | Punti |
| 200-210 | Fine della partita |
| 220-240 | Bit di fine gara |
| 250-4990 | Fine del programma |
| 5000-5010 | Inserisce i dati nell' array |
| 5020-5030 | Definisce le altre variabili |
| 8000-8030 | Dati per il motivo durante la gara |
| 8100 | Dati per il motivo di fine gara |
| 8200-8240 | Dati per messaggio a fine programma |

Il secondo programma, "ICBM", e' una versione del piu' noto "Missile command".

Scopo del game, e' il salvataggio di sei citta', mostrate sottoforma di blocchi, da uno stormo di missili che si presentano ad ondate successive di venti elementi, ognuna piu' veloce della precedente. Potete distruggere i missili in caduta posizionandovi al disotto e sparando con lo 0.

La vostra posizione varia in senso antiorario col tasto "6" e in senso orario col "7". Se avete qualche perplessita', consultate la freccia nella parte superiore dello schermo. Il game ha termine con la distruzione delle sei citta'. Ad ogni 10.000 punti vi viene regalata una citta', il numero dei punti aumenta ad ogni ondata e varia a seconda dei missili colpiti e delle munizioni risparmiate...il resto lo scoprirete da voi.

Elenco delle variabili

| | |
|-------|--|
| mis | Missile in discesa dal cielo |
| f(.) | Array relativo ai missili |
| m(.) | Punto di origine dei missili situato sulla parte alta dello schermo |
| x(.) | Coordinata orizzontale della testa del missile |
| y(.) | Coordinata verticale della testa del missile |
| d(.) | Numero dei pixel necessario al missile per raggiungere la citta' da distruggere |
| b(.) | Citta' selezionata dal missile. Se e' la 3 il missile centra la base |
| c(.) | Array che riporta quale delle citta' sia stata distrutta |
| a(.) | Citta' selezionata (in b(.)) sottoforma del numero di pixel orizzontale |
| c | Numero delle citta' distrutte |
| x | Coordinata orizzontale della base |
| y | Coordinata verticale della base |
| b\$ | Usata per l' input |
| dir | Direzione della base 0=su, 1=su a destra, 2 a destra ecc... |
| move | Numero dei pixel della base mossi ogni volta (diminuisce man mano che la gara si fa piu' difficoltosa) |
| ammo | Munizioni residue |
| count | Missili per ogni ondata |

ink Colore di INK
z Usato come controllo dei loop per vedere se ogni missile colpisce (linee 420-460)
n1 Valore per predisporre un nuovo missile
sc Punti
dd Grado di difficoltà'
paper Colore di PAPER
te Valore temporaneo di b (.) per calcolare la scrittura della colonna per l' esplosione della città'
city Quale città' e' esplosa
mark Usato per calcolare se tutte le città' sono state colpite
wave Conto delle ondate
abm Punti delle munizioni rimaste
city Punti delle città' rimaste
down Numero dei pixel relativi ai missili caduti dopo ogni ondata
ink 1 Colore delle munizioni a terra
sc 1 Punteggio in 10.000, usato per calcolare quando dare una città' gratis
xc Memorizzazione delle città' elargite. Se tutte le città' sono ancora in piedi dopo aver raggiunto i 10.000 punti, la città' viene conservata per ulteriori bisogni
nc Posizionamento random della nuova città'
a1,
a2,
a3 Usati per la casualità' dei colori di INK e PAPER

```

10 GO TO 3000: REM INIZ
20 GO TO 500
30 REM INIZ
40 LET B=0
50 IF B=0 THEN B=1+(4 AND MIS=4)
60 INVERSE B THEN GO TO 200
70 PLOT B(MIS),153: DRAW X(MIS)
80 Y(MIS),Y(MIS)-153
90 LET Y(MIS)=Y(MIS)-down
100 LET X(MIS)=X(MIS)+(d(MIS)*d
110)
120 INVERSE 0
130 IF Y(MIS)<16 THEN GO SUB 60
140 GO TO 190
150 PLOT B(MIS),153: DRAW X(MIS)
160 Y(MIS),Y(MIS)-153
170 IF C=0 THEN GO TO 4000: REM
INIZ
180 INVERSE 1: PLOT X-2,Y: DRAW
190 X,Y: PLOT X,Y: DRAW X-4,Y-4
200 LET DIR=DIR+1: IF DIR=8: TH
210 LET DIR=0
220 IF DIR=1 THEN LET DIR=DIR+
230 AND DIR=7)

```


| | |
|-----------|--|
| 3100-3140 | Inizializzazione delle variabili (anche 3220-3230) |
| 3200 | Array per i missili |
| 3240 | Predispone i missili |
| 3300 | Inizializza l'attraversamento del cursore |
| 3400 | Disegna lo schermo |
| 4000 | Fine della gara |
| 9000-9020 | Definisce gli UDG |
| 9100-9130 | Dati per gli UDG |

PEEK e POKE

Sono senza dubbio due tra i comandi piu' usati dello Spectrum. Per capire bene le funzioni che essi svolgono, e' necessario conoscere l' hardware della macchina e il suo funzionamento. Come gia' saprete, ogni computer possiede una memoria la quale si divide in due tipi ben distinti. La ROM (abbreviazione che sta per Read Only Memory = memoria a sola lettura) contiene le informazioni necessarie al funzionamento della CPU (Central Processing Unit = Unita' centrale di processo o piu' semplicemente microprocessore) le quali rimangono memorizzate anche quando l' apparecchio viene spento. Il contenuto della ROM non puo' venire alterato in alcun modo.

Il secondo tipo di memoria si chiama RAM (Random Access Memory = Memoria ad accesso casuale) e memorizza i dati provenienti dalla CPU. Le informazioni e i dati in gioco, sono tutto quanto presente in un programma e visualizzato sullo schermo a partire dalla grafica definita dall' utente (UDG) per arrivare alle variabili del sistema. Al contrario di quanto accade per la ROM, il contenuto della RAM va perso quando si toglie corrente e puo' venir alterato in qualsiasi momento.

Vediamo ora in quale modo viene memorizzata una informazione. Immaginatevi la memoria suddivisa in compartimenti ognuno dei quali in grado di contenere un numero; sia la quantita' dei primi che quella dei secondi dipende dal tipo di computer. Lo Spectrum dispone di 65536 (da 0 a 65535) compartimenti, chiamati bytes capaci ognuno di memorizzare un numero compreso entro la gamma 0-255. Per comodita' i bytes vengono raggruppati a mille per volta e ognuno di tali gruppi e' conosciuto col nome di "Kilobyte", abbreviato con "K". Per essere piu' precisi, un K equivale a 1024 bytes.

Il comando PEEK vi permette di accedere a questi bytes con la forma "PEEK n" dove "n" e' appunto il byte da esaminare.

Il comando POKE, invece, vi abilita addirittura a cambiare il contenuto per mezzo dello statement "POKE n,m" in cui "n" e' il byte, o meglio l' indirizzo, presso il quale effettuare la variazione e "m" il valore del nuovo dato compreso tra 0 e 255. POKE viene quindi usato per memorizzare nuovi dati, in una area di memoria riservata e PEEK per andarli a ripescare.

Vi sono diversi modi per riservare la memoria onde proteggere i dati da alterazioni accidentali. Uno di questi sistemi prevede l' uso dello statement di REM.

Se la prima linea e' :

```
1 REM AAA
```

La memoria viene prenotata sotto il segno delle A. Provate a battere questa linea facendola seguire dal comando diretto POKE 23760,127 e vedrete che, listando nuovamente la linea, al posto della prima A troverete il simbolo di copyright corrispondente

appunto al carattere 127. Provate con altri valori. L'indirizzo 23760 si riferisce appunto al primo carattere dopo il REM presente alla linea 1. Per pochi dati potete quindi utilizzare lo statement di REM dotandolo di tanti caratteri quanti sono i dati per poi POKare questi ultimi nei vari indirizzi messi a disposizione dallo statement stesso.

Una seconda via per proteggere i dati la mette a disposizione CLEAR il quale li preserva anche dal comando NEW che ha il potere di azzerare tutto quanto.

Di solito i caratteri a disposizione dell'utente, che sono in tutto 21, occupano gli ultimi 168 bytes della memoria in quanto ogni carattere ne impiega 8. Nello Spectrum da 16 K, essi occupano gli indirizzi da 32600 a 32767, in quello da 48 K gli indirizzi impegnati partono da 65368 e arrivano fino a 65535. Lo statement "CLEAR n" predispone una variabile di sistema (che vedremo piu' avanti) all'indirizzo "n".

Se "n" e' piu' alto di 32600 nello Spectrum da 16 K si perde un carattere-user, predisponendolo piu' basso, si prenota una zona di memoria da n+1 a 32599. Analogamente accade nella versione da 48 K con i relativi indirizzi.

Con POKE si predispongono le variabili di sistema le quali vengono considerate dalla CPU per valutare quanto sta succedendo e prendere determinate decisioni. Usando correttamente questa funzione e' possibile creare un gran numero di effetti speciali. POKando, ad esempio, 23609 col numero 100 potete generare, alla pressione di qualsiasi tasto, un beep piu' consistente del normale che puo' risultare utile nelle operazioni di INPUT dei games. POKando un numero maggiore di 1 all'indirizzo 23692, il computer non vi chiederà piu' lo "scroll?" ogni 22 linee di programma. Per cambiare il colore delle scritte relative ai messaggi sulla parte bassa dello schermo, non dovrete far altro che eseguire un POKE 23624 seguito da un numero a caso. POKE 23624 altera quindi gli ATTRibuti e i colori delle scritte o degli oggetti presenti nella parte alta dello schermo. Dando POKE 23659,1:CLS riempirete lo schermo di strisce la cui ampiezza dipende dal valore POKato in 23693.

Gli indirizzi 23677 e 23678 riguardano la funzione DRAW la quale puo' tracciare linee in tutte le direzioni partendo dall'ultimo punto stabilito da PLOT. Nell'esempio che segue, la posizione di plot orizzontale non e' 20 bensì 120 (100+20), così come quella verticale non e' 60 ma 140 (80+60).

```
10 PLOT 100,80
20 DRAW 20,60
```

Perche' una eventuale linea 30 possa iniziare a tracciare dai valori 0,0 e' necessario l'intervento di alcuni calcoli. Per mezzo delle variabili di sistema 23677 e 23678 potete simulare un DRAW assoluto. La linea 20 puo' essere riscritta nel modo che segue :

```
20 DRAW 120-PEEK 23677,140-PEEK 23678
```

e se prima avreste dovuto battere

```
30 DRAW -120,-140
```

ora dovete inserire

```
30 DRAW 0-PEEK 23677,0-PEEK 23678
```

Lo Spectrum puo' disporre del tempo in modo assai ben definito grazie alle tre variabili di sistema 23674,23673 e 23672. La piu' bassa delle tre viene incrementata ogni 50esimo di secondo fino a che non raggiunge 255. dopodiche' torna a 0 facendo partire il conteggio nell' indirizzo 23673 il quale, una volta raggiunto 255 abilita il terzo che, a sua volta raggiunge 255, resettandosi e facendo ripartire tutto da capo.

Il calcolo del tempo (espresso in secondi) e' assai semplice :

```
LETsecs=(65536*PEEK 23674+256*PEEK 23673+PEEK 23672)/150
```

Questa funzione e' di grande aiuto in games tipo auto da corsa e in qualsiasi caso in cui la gara debba essere portata a termine nel piu' breve tempo possibile. Alcune volte puo' succedere che il calcolo dei secondi dia un risultato sbagliato. Nel caso in cui, ad esempio, la locazione 23673 contenga 0 in conseguenza del fatto che lo 23672 abbia raggiunto 255 e poi si sia a sua volta azzerata, il computer puo' valutare contemporaneamente i valori 0 delle due variabili e di conseguenza considerare un tempo doppio di quello reale falsando i risultati. Per ovviare a tale fenomeno, e' necessario POKare a 0 ognuno degli indirizzi.

Oltre ai 22 caratteri, lo Spectrum ne puo' definire altri 96. Il segreto sta' nel bit del POKE il quale, se impiegato negli indirizzi 23606 e 23607, vi permette di cambiare il punto di partenza del set dei caratteri presente da CHR# 32 (spazio) a CHR# 127 (marchio di copyright).

Ecco come dovete fare :

- 1) Disegnate i nuovi caratteri da mettere al posto dei vecchi e calcolatene i dati.
- 2) Decidete i caratteri del set originale da ridefinire e prendete nota di quelli che dovranno ridefinirli. Di solito i caratteri da ridefinire (UDG=User-defined graphic) vengono scelti tra i primi del set originale. Se volete definire lettere e numeri tipo "era spaziale" e' bene segnarsi sempre l' accoppiamento tra i vecchi numeri con i nuovi e tra le vecchie lettere con le nuove. Questo perche', mentre state scrivendo lo statement di PRINT vi ricordate le equivalenze, ma, una volta fatto girare il programma, i caratteri ordinari lasciano il posto a quelli ridefiniti scomparendo dal contesto delle linee.

dr Direzione dell' auto da 0 a 7. (0-su,2-destra ecc...)

g Marcia. Da 1 (piu' lenta) a 4 (piu' veloce)

al Legge la tastiera con IN per vedere se e' stato premuto il tasto 0 per l' accelerazione

dl Come il precedente ma stavolta riferito al tasto A per la decelerazione

mv Verifica se sia stato battuto uno dei tasti per spostare l' auto (O oppure P)

a,
b Coordinate che la subroutine usa per disegnare qualsiasi cosa. Tale subroutine e' richiamata da due diversi posti, "a" e "b" vengono assegnati prima di chiamare la subroutine in modo che vengano PRINTate le coordinate esatte

t Tempo (in secondi) impiegato per completare il circuito. Viene comparato con "h"

z# Usata come stringa d' ingresso

LOOP,
ALTER,
NORM,
TIME,
START,
NEXT CAR,
END,
CRASH,
FINISH,
BYE

Sono tutti "tokens" per la partenza delle subroutines: ALTER e' la subroutine usata per inserire il set alternativo di caratteri; NORM (normale) e' per cambiare di nuovo; TIME presenta tempo; START inizializza la partenza di una nuova gara; NEXT CAR inizializza una nuova auto; END e' il punto in cui finisce il programma quando l' auto urta contro qualcosa; CRASH e' per quando l' auto urta contro il circuito e FINISH e' per quando l' auto raggiunge la griglia di partenza.

Nel listing sono presenti degli user-graphic alle linee 5300 (quattro I) e 5310 (lettera C).

Circuito (Parte 2)

```

100 GO TO 5000: REM init/instr
200 REM COORDINATE SPECTRUM ISSUE 2
300 REM variare 10000 con 101,
400 REM 254 con 100 e 253 con 1
500 LET a1=IN 64510-255: LET dl
=IN 65022-255: LET mv=IN 57342

```



```

      GO TO 1000
      PRINT "CRASH"
      PRINT "*****CRASH*****"
      PRINT "*****FINISH*****"
      PRINT "*****NEW GAME*****"
      PRINT "Another game? (Y/N)"
      PRINT "Press ENTER to play"
      PRINT "PAPER 1: INK 2: C"
      PRINT "FLASH 1: IN"

```

Note

| | |
|-----------|--|
| 10 | Inizializzazione |
| 100 | Legge la tastiera |
| 120 | Cambia la direzione |
| 140 | Cambia le coordinate |
| 160 | Disegna l' auto |
| 180 | Cambia marcia |
| 200 | Suono |
| 220 | Cancella l' auto |
| 240 | Verifica se l' auto ha urtato qualcosa |
| 1000 | Set di caratteri alternativo |
| 1500 | Set di caratteri normale |
| 2010 | Il tempo viene inserito in una stringa |
| 2030-2040 | Scriva il tempo |
| 5000 | Miglior tempo inizializzato |
| 5010-5070 | "Tokens" per le subroutines inizializzate |
| 5100 | Disegna le strisce |
| 5130 | Colori |
| 5150-5170 | Disegna lo schermo |
| 5300 | Scriva il tempo |
| 5310 | Auto rimaste |
| 5320 | Scriva sulla 23esima riga |
| 5430 | Azzeri i contatori del tempo |
| 6500-6590 | Dati usati alla linea 5150 per scrivere sullo schermo |
| 7000 | Cancella il marker delle marce |
| 7010-7020 | Calcola il tempo e lo presenta |
| 7030 | Salta a "CRASH" o a "FINISH" |
| 7500 | Ruota l' auto |
| 7510 | Riposiziona il circuito |
| 7520-7540 | Diminuisce il numero delle auto e salta di conseguenza |
| 8000 | Disegna l' auto |
| 8030 | Congratulazioni |
| 8050 | Inserisce il nuovo tempo nella variabile "best time" |
| 8080 | Riposiziona la griglia |
| 9000 | Verifica per un' altra gara |
| 9020 | Tutto pronto |
| 9980-9999 | Allegorie |

Seguono due ottimi programmi sviluppati originalmente per tutt' altro. L' idea originale era di usare una racchetta o una superficie scorrevole per catturare i blocchi che piovono dal cielo. Ogni blocco puo' valere da uno o due punti. I blocchi non catturati piombano al suolo accatastandosi fino a formare un muro che, se raggiunge l' altezza della racchetta, fa finire la gara.

Il primo dei due games basati su tale principio e' stato chiamato "Chomper".

Dovete mangiare il cibo che cade dal cielo composto da pere, salsicce, birra, polli arrosto e hamburger. Cadono anche dei

fucili che dovrete recuperare per poter sparare ai mostri che crescono mangiando quanto non siete riusciti ad acchiappare. Le istruzioni sono contenute nel programma stesso. Le coordinate dei tre oggetti che cadono contemporaneamente (siano questi cibo o armi) sono memorizzate in array per poterli muovere tramite loop.

```

Graphic A      : linee 1000 e 3000 (doppio)
Graphic B      : linea 4000
Graphic J e K  : linee 2010,4040 e 5410 (primo set della linea)
Graphic L e M  : linea 2000 (primo set)
Graphic N e O  : Linea 2000 (secondo set) e 5410 (secondo set)
Graphic P      : linea 7010 (primo)
Graphic Q      : linea 7010 (secondo)
Graphic R      : linee 7020 e 7050
Graphic S      : linea 7030

```

Elenco delle variabili

```

i          Variabile di controllo dei loop per far cadere gli
          oggetti dal cielo
a(.)      Array per le coordinate verticali degli oggetti
b(.)      Array per le coordinate orizzontali degli oggetti
t2        Altezza dell' ultimo oggetto mangiato dal mostro.
          Usato per vedere se il mostro ha raggiunto l'
          altezza del "Chomper"
c(.)      Memorizza il colore di PAPER di ogni oggetto in
          caduta
d(.)      Memorizza il colore di INK di ogni oggetto in
          caduta
c#(.)     Memorizza la forma dell' oggetto (ad esempio una
          pera)
m         Usato per alternare il movimento della bocca de
          "Chomper"
a$        Per leggere la tastiera
p         Coordinata orizzontale del giocatore
gun       Numero dei fucili recuperati
rnd       Usata per selezionare la specie di cibo da far
          cadere
n         Controllo per il loop di lettura dei colori del
          cibo
t1        Usata per ingrassare il mostro quando questo ha
          mangiato
z         Variabile di controllo per la cancellazione dei
          mostri quando siano stati colpiti
s         Punti
i,
n,
z         Variabili di controllo per gli altri loop

```


| | |
|-----------|---|
| 100 | Cambia la posizione |
| 110 | Ristampa l' uomo |
| 1000 | Stampa lo splash, aggiorna il punteggio e
inizializza una nuova goccia |
| 2000 | Disegna il nuovo livello |
| 2010 | Incrementa il livello dell' acqua e verifica se e'
ad altezza d' uomo |
| 5005-5030 | Istruzioni |
| 5040-5050 | Tutto pronto |
| 5060 | UDG |
| 5070 | Inizializza le variabili |
| 5080 | Inizializza le gocce negli array |
| 5090-5110 | Disegna lo schermo |
| 8000 | Disegna l' omino carico e suona il motivetto |
| 8010-8090 | Bit finale (punti, nuova gara ecc...) |
| 9000 | Inizializza le variabili stringa per l' omino |
| 9010-9040 | Definisce gli user-graphic e li pone nelle
variabili stringa |
| 9050-9130 | Dati per gli UDG |
| 9990-9999 | Fine del game |

COME RENDERE PIU' INTERESSANTI I GAMES

L' attrattiva di un game molto dipende dal modo in cui viene presentato. Non cadete nella prolissita', eliminate le frange inutili senza fare piu' di quanto il listato non richieda. Badate anche a non introdurre troppe presentazioni o infioriture che, se da un lato migliorano l' aspetto del display, dall' altro rendono piu' lento il programma.

Eccovi alcuni preziosi suggerimenti :

-Stendete le istruzioni con una nota di humor rendendole piu' divertenti e fuori dalla norma apportando qua' e la' piccoli ritocchi tanto da far capire che i games servono a divertire e non vanno presi troppo seriamente.

-In casi particolari, il computer deve dare l' impressione di commentare l' evento. Ad esempio, in un game tipo "Mastermind", potete inserire prima e dopo ogni mossa, frasi del genere : "Siamo duri eh?..." seguito da una pausa e poi "finalmente ci sei arrivato!". Anche qui pero', occhio a non esagerare perche' cadreste nel noioso.

-Per dare forma agli oggetti usate gli UDG e non simboli come l' asterisco, il punto esclamativo o simili che non rendono assolutamente alcuna idea. La gente non ricorda se la bomba e' grossa la meta' dell' aereo che la sgancia, bensì nota il realismo delle scene e del succedersi delle azioni.

-Sperimentate diversi colori usando le piu' svariate combinazioni, e magari randomizzando i colori di PAPER e di INK in funzione di certi eventi cosi' come avviene nel programma ICBM.

-Generate spesso le esplosioni, sono cariche di effetto. State pero' attenti che all' atto dell' esplosione, il resto del programma rimane inattivo per cui un abuso in questo senso porterebbe a interruzioni troppo frequenti che ostacolerebbero l' effetto prefissato.

-Dovendo inserire effetti musicali, non affidatevi a frequenze di note random, ma ricorrete a filastrocche piacevoli come spiegato nel contesto del volume.

Fate attenzione alla velocita' con la quale il programma gira. E' vero che piu' una azione e' veloce e meglio e', ma anche qui non bisogna esagerare sconfinando nell' impossibile. Se ritenete troppo rapido lo svolgimento di determinate fasi, rallentatelo frapponendo piacevoli effetti musicali. Ovviamente la velocita' di un game dipende da piu' fattori quali ad esempio : le tecniche usate, ivi compresa la compattezza del programma in codice, l' organizzazione dei componenti e quella delle variabili. Un programma con "x" linee comprendenti ognuna uno statement gira piu' lentamente di "x" statement raggruppati nella stessa linea.

Il componente piu' importante di un game e' il loop principale, seguito dalle subroutines, prima quelle piu' usate poi quelle meno (come le istruzioni). Quando il programma incontra GO TO o

GO SUB, esplora in successione i numeri delle linee fino a che non raggiunge quella richiesta. Poiche' il computer parte dall'inizio del listato, piu' vicina si trova la linea alla quale va eseguito il salto e piu' rapidamente questo avviene. Stesso discorso vale quando il computer consulta una variabile. Quelle insite nel primo blocco considerato vengono inizializzate per prime e quindi rintracciate piu' velocemente.

Il BASIC viene spesso criticato per essere un linguaggio "non strutturato" a causa appunto del GO TO che, se usato impropriamente, dirotta fuori destinazione il flusso del programma. E' possibile a tale scopo, strutturare i programmi in modo da far loro assumere le caratteristiche del PASCAL il quale impiega dei loop formati da GO SUB. Questo procedimento si basa su di un "sottoprogramma", richiamato dal loop principale, in grado di fare le operazioni piu' svariate come causare un' esplosione, far sparare un laser, oppure muovere gli invaders.

Provate a dare un' occhiata a quanto segue, e tenetene presente come intelaiatura di games :

```

10 GO SUB 5000: REM inizializz
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
290
300
310
320
330
340
350
360
370
380
390
400
410
420
430
440
450
460
470
480
490
500
510
520
530
540
550
560
570
580
590
600
610
620
630
640
650
660
670
680
690
700
710
720
730
740
750
760
770
780
790
800
810
820
830
840
850
860
870
880
890
900
910
920
930
940
950
960
970
980
990
1000
1010
1020
1030
1040
1050
1060
1070
1080
1090
1100
1110
1120
1130
1140
1150
1160
1170
1180
1190
1200
1210
1220
1230
1240
1250
1260
1270
1280
1290
1300
1310
1320
1330
1340
1350
1360
1370
1380
1390
1400
1410
1420
1430
1440
1450
1460
1470
1480
1490
1500
1510
1520
1530
1540
1550
1560
1570
1580
1590
1600
1610
1620
1630
1640
1650
1660
1670
1680
1690
1700
1710
1720
1730
1740
1750
1760
1770
1780
1790
1800
1810
1820
1830
1840
1850
1860
1870
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
2000
2010
2020
2030
2040
2050
2060
2070
2080
2090
2100
2110
2120
2130
2140
2150
2160
2170
2180
2190
2200
2210
2220
2230
2240
2250
2260
2270
2280
2290
2300
2310
2320
2330
2340
2350
2360
2370
2380
2390
2400
2410
2420
2430
2440
2450
2460
2470
2480
2490
2500
2510
2520
2530
2540
2550
2560
2570
2580
2590
2600
2610
2620
2630
2640
2650
2660
2670
2680
2690
2700
2710
2720
2730
2740
2750
2760
2770
2780
2790
2800
2810
2820
2830
2840
2850
2860
2870
2880
2890
2900
2910
2920
2930
2940
2950
2960
2970
2980
2990
3000
3010
3020
3030
3040
3050
3060
3070
3080
3090
3100
3110
3120
3130
3140
3150
3160
3170
3180
3190
3200
3210
3220
3230
3240
3250
3260
3270
3280
3290
3300
3310
3320
3330
3340
3350
3360
3370
3380
3390
3400
3410
3420
3430
3440
3450
3460
3470
3480
3490
3500
3510
3520
3530
3540
3550
3560
3570
3580
3590
3600
3610
3620
3630
3640
3650
3660
3670
3680
3690
3700
3710
3720
3730
3740
3750
3760
3770
3780
3790
3800
3810
3820
3830
3840
3850
3860
3870
3880
3890
3900
3910
3920
3930
3940
3950
3960
3970
3980
3990
4000
4010
4020
4030
4040
4050
4060
4070
4080
4090
4100
4110
4120
4130
4140
4150
4160
4170
4180
4190
4200
4210
4220
4230
4240
4250
4260
4270
4280
4290
4300
4310
4320
4330
4340
4350
4360
4370
4380
4390
4400
4410
4420
4430
4440
4450
4460
4470
4480
4490
4500
4510
4520
4530
4540
4550
4560
4570
4580
4590
4600
4610
4620
4630
4640
4650
4660
4670
4680
4690
4700
4710
4720
4730
4740
4750
4760
4770
4780
4790
4800
4810
4820
4830
4840
4850
4860
4870
4880
4890
4900
4910
4920
4930
4940
4950
4960
4970
4980
4990
5000
5010
5020
5030
5040
5050
5060
5070
5080
5090
5100
5110
5120
5130
5140
5150
5160
5170
5180
5190
5200
5210
5220
5230
5240
5250
5260
5270
5280
5290
5300
5310
5320
5330
5340
5350
5360
5370
5380
5390
5400
5410
5420
5430
5440
5450
5460
5470
5480
5490
5500
5510
5520
5530
5540
5550
5560
5570
5580
5590
5600
5610
5620
5630
5640
5650
5660
5670
5680
5690
5700
5710
5720
5730
5740
5750
5760
5770
5780
5790
5800
5810
5820
5830
5840
5850
5860
5870
5880
5890
5900
5910
5920
5930
5940
5950
5960
5970
5980
5990
6000
6010
6020
6030
6040
6050
6060
6070
6080
6090
6100
6110
6120
6130
6140
6150
6160
6170
6180
6190
6200
6210
6220
6230
6240
6250
6260
6270
6280
6290
6300
6310
6320
6330
6340
6350
6360
6370
6380
6390
6400
6410
6420
6430
6440
6450
6460
6470
6480
6490
6500
6510
6520
6530
6540
6550
6560
6570
6580
6590
6600
6610
6620
6630
6640
6650
6660
6670
6680
6690
6700
6710
6720
6730
6740
6750
6760
6770
6780
6790
6800
6810
6820
6830
6840
6850
6860
6870
6880
6890
6900
6910
6920
6930
6940
6950
6960
6970
6980
6990
7000
7010
7020
7030
7040
7050
7060
7070
7080
7090
7100
7110
7120
7130
7140
7150
7160
7170
7180
7190
7200
7210
7220
7230
7240
7250
7260
7270
7280
7290
7300
7310
7320
7330
7340
7350
7360
7370
7380
7390
7400
7410
7420
7430
7440
7450
7460
7470
7480
7490
7500
7510
7520
7530
7540
7550
7560
7570
7580
7590
7600
7610
7620
7630
7640
7650
7660
7670
7680
7690
7700
7710
7720
7730
7740
7750
7760
7770
7780
7790
7800
7810
7820
7830
7840
7850
7860
7870
7880
7890
7900
7910
7920
7930
7940
7950
7960
7970
7980
7990
8000
8010
8020
8030
8040
8050
8060
8070
8080
8090
8100
8110
8120
8130
8140
8150
8160
8170
8180
8190
8200
8210
8220
8230
8240
8250
8260
8270
8280
8290
8300
8310
8320
8330
8340
8350
8360
8370
8380
8390
8400
8410
8420
8430
8440
8450
8460
8470
8480
8490
8500
8510
8520
8530
8540
8550
8560
8570
8580
8590
8600
8610
8620
8630
8640
8650
8660
8670
8680
8690
8700
8710
8720
8730
8740
8750
8760
8770
8780
8790
8800
8810
8820
8830
8840
8850
8860
8870
8880
8890
8900
8910
8920
8930
8940
8950
8960
8970
8980
8990
9000
9010
9020
9030
9040
9050
9060
9070
9080
9090
9100
9110
9120
9130
9140
9150
9160
9170
9180
9190
9200
9210
9220
9230
9240
9250
9260
9270
9280
9290
9300
9310
9320
9330
9340
9350
9360
9370
9380
9390
9400
9410
9420
9430
9440
9450
9460
9470
9480
9490
9500
9510
9520
9530
9540
9550
9560
9570
9580
9590
9600
9610
9620
9630
9640
9650
9660
9670
9680
9690
9700
9710
9720
9730
9740
9750
9760
9770
9780
9790
9800
9810
9820
9830
9840
9850
9860
9870
9880
9890
9900
9910
9920
9930
9940
9950
9960
9970
9980
9990
10000
10010
10020
10030
10040
10050
10060
10070
10080
10090
10100
10110
10120
10130
10140
10150
10160
10170
10180
10190
10200
10210
10220
10230
10240
10250
10260
10270
10280
10290
10300
10310
10320
10330
10340
10350
10360
10370
10380
10390
10400
10410
10420
10430
10440
10450
10460
10470
10480
10490
10500
10510
10520
10530
10540
10550
10560
10570
10580
10590
10600
10610
10620
10630
10640
10650
10660
10670
10680
10690
10700
10710
10720
10730
10740
10750
10760
10770
10780
10790
10800
10810
10820
10830
10840
10850
10860
10870
10880
10890
10900
10910
10920
10930
10940
10950
10960
10970
10980
10990
11000
11010
11020
11030
11040
11050
11060
11070
11080
11090
11100
11110
11120
11130
11140
11150
11160
11170
11180
11190
11200
11210
11220
11230
11240
11250
11260
11270
11280
11290
11300
11310
11320
11330
11340
11350
11360
11370
11380
11390
11400
11410
11420
11430
11440
11450
11460
11470
11480
11490
11500
11510
11520
11530
11540
11550
11560
11570
11580
11590
11600
11610
11620
11630
11640
11650
11660
11670
11680
11690
11700
11710
11720
11730
11740
11750
11760
11770
11780
11790
11800
11810
11820
11830
11840
11850
11860
11870
11880
11890
11900
11910
11920
11930
11940
11950
11960
11970
11980
11990
12000
12010
12020
12030
12040
12050
12060
12070
12080
12090
12100
12110
12120
12130
12140
12150
12160
12170
12180
12190
12200
12210
12220
12230
12240
12250
12260
12270
12280
12290
12300
12310
12320
12330
12340
12350
12360
12370
12380
12390
12400
12410
12420
12430
12440
12450
12460
12470
12480
12490
12500
12510
12520
12530
12540
12550
12560
12570
12580
12590
12600
12610
12620
12630
12640
12650
12660
12670
12680
12690
12700
12710
12720
12730
12740
12750
12760
12770
12780
12790
12800
12810
12820
12830
12840
12850
12860
12870
12880
12890
12900
12910
12920
12930
12940
12950
12960
12970
12980
12990
13000
13010
13020
13030
13040
13050
13060
13070
13080
13090
13100
13110
13120
13130
13140
13150
13160
13170
13180
13190
13200
13210
13220
13230
13240
13250
13260
13270
13280
13290
13300
13310
13320
13330
13340
13350
13360
13370
13380
13390
13400
13410
13420
13430
13440
13450
13460
13470
13480
13490
13500
13510
13520
13530
13540
13550
13560
13570
13580
13590
13600
13610
13620
13630
13640
13650
13660
13670
13680
13690
13700
13710
13720
13730
13740
13750
13760
13770
13780
13790
13800
13810
13820
13830
13840
13850
13860
13870
13880
13890
13900
13910
13920
13930
13940
13950
13960
13970
13980
13990
14000
14010
14020
14030
14040
14050
14060
14070
14080
14090
14100
14110
14120
14130
14140
14150
14160
14170
14180
14190
14200
14210
14220
14230
14240
14250
14260
14270
14280
14290
14300
14310
14320
14330
14340
14350
14360
14370
14380
14390
14400
14410
14420
14430
14440
14450
14460
14470
14480
14490
14500
14510
14520
14530
14540
14550
14560
14570
14580
14590
14600
14610
14620
14630
14640
14650
14660
14670
14680
14690
14700
14710
14720
14730
14740
14750
14760
14770
14780
14790
14800
14810
14820
14830
14840
14850
14860
14870
14880
14890
14900
14910
14920
14930
14940
14950
14960
14970
14980
14990
15000
15010
15020
15030
15040
15050
15060
15070
15080
15090
15100
15110
15120
15130
15140
15150
15160
15170
15180
15190
15200
15210
15220
15230
15240
15250
15260
15270
15280
15290
15300
15310
15320
15330
15340
15350
15360
15370
15380
15390
15400
15410
15420
15430
15440
15450
15460
15470
15480
15490
15500
15510
15520
15530
15540
15550
15560
15570
15580
15590
15600
15610
15620
15630
15640
15650
15660
15670
15680
15690
15700
15710
15720
15730
15740
15750
15760
15770
15780
15790
15800
15810
15820
15830
15840
15850
15860
15870
15880
15890
15900
15910
15920
15930
15940
15950
15960
15970
15980
15990
16000
16010
16020
16030
16040
16050
16060
16070
16080
16090
16100
16110
16120
16130
16140
16150
16160
16170
16180
16190
16200
16210
16220
16230
16240
16250
16260
16270
16280
16290
16300
16310
16320
16330
16340
16350
16360
16370
16380
16390
16400
16410
16420
16430
16440
16450
16460
16470
16480
16490
16500
16510
16520
16530
16540
16550
16560
16570
16580
16590
16600
16610
16620
16630
16640
16650
16660
16670
16680
16690
16700
16710
16720
16730
16740
16750
16760
16770
16780
16790
16800
16810
16820
16830
16840
16850
16860
16870
16880
16890
16900
16910
16920
16930
16940
16950
16960
16970
16980
16990
17000
17010
17020
17030
17040
17050
17060
17070
17080
17090
17100
17110
17120
17130
17140
17150
17160
17170
17180
17190
17200
17210
17220
17230
17240
17250
17260
17270
17280
17290
17300
17310
17320
17330
17340
17350
17360
17370
17380
17390
17400
17410
17420
17430
17440
17450
17460
17470
17480
17490
17500
17510
17520
17530
17540
17550
17560
17570
17580
17590
17600
17610
17620
17630
17640
17650
17660
17670
17680
17690
17700
17710
17720
17730
17740
17750
17760
17770
17780
17790
17800
17810
17820
17830
17840
17850
17860
17870
17880
17890
17900
17910
17920
17930
17940
17950
17960
17970
17980
17990
18000
18010
18020
18030
18040
18050
18060
18070
18080
18090
18100
18110
18120
18130
18140
18150
18160
18170
18180
18190
18200
18210
18220
18230
18240
18250
18260
18270
18280
18290
18300
18310
18320
18330
18340
18350
18360
18370
18380
18390
18400
18410
18420
18430
18440
18450
18460
18470
18480
18490
18500
18510
18520
18530
18540
18550
18560
18570
18580
18590
18600
18610
18620
18630
18640
18650
18660
18670
18680
18690
18700
18710
18720
18730
18740
18750
18760
18770
18780
18790
18800
18810
18820
18830
18840
18850
18860
18870
18880
18890
18900
18910
18920
18930
18940
18950
18960
18970
18980
18990
19000
19010
19020
19030
19040
19050
19060
19070
19080
19090
19100
19110
19120
19130
19140
19150
19160
19170
19180
19190
19200
19210
19220
19230
19240
19250
19260
19270
19280
19290
19300
19310
19320
19330
19340
19350
19360
19370
19380
19390
19400
19410
19420
19430
19440
19450
19460
19470
19480
19490
19500
19510
19520
19530
19540
19550
19560
19570
19580
19590
19600
19610
19620
19630
19640
19650
19660
19670
19680
19690
19700
19710
19720
19730
19740
19750
19760
19770
19780
19790
19800
19810
19820
19830
19840
19850
19860
19870
19880
19890
19900
19910
19920
19930
19940
19950
19960
19970
19980
19990
20000
20010
20020
20030
20040
20050
20060
20070
20080
20090
20100
20110
20120
20130
20140
20150
20160
20170
20180
20190
20200
20210
20220
20230
20240
20250
20260
20270
20280
20290
20300
20310
20320
20330
20340
20350
20360
20370
20380
20390
20400
20410
20420
20430
20440
20450
20460
20470
20480
20490
20500
20510
20520
20530
20540
20550
20560
20570
20580
20590
20600
20610
20620
20630
20640
20650
20660
20670
20680
20690
20700
20710
20720
20730
20740
20750
20760
20770
20780
20790
20800
20810
20820
20830
20840
20850
20860
20870
20880
20890
20900
20910
20920
20930
20940
20950
20960
20970
20980
20990
21000
21010
21020
21030
21040
21050
21060
21070
21080
21090
21100
21110
21120
21130
21140
21150
21160
21170
21180
21190
21200
21210
21220
21230
21240
21250
21260
21270
21280
21290
21300
21310
21320
21330
21340
21350
21360
21370
21380
21390
21400
21410
21420
21430
21440
21450
21460
21470
21480
21490
21500
21510
21520
21530
21540
21550
21560
21570
21580
21590
21600
21610
21620
21630
21640
21650
21660
21670
21680
21690
21700
21710
21720
21730
21740
21750
21760
21770
21780
21790
21800
21810
21820
21830
21840

```




Il volume descrive dettagliatamente tutte le tecniche di stesura di giochi ARCADE, partendo dalla lettura della tastiera e toccando la definizione grafica, l'impiego del suono e l'uso degli operatori logici, per migliorare la qualità dei programmi. Altri argomenti esaminati in dettaglio sono l'animazione degli oggetti, lo scrolling dello schermo e l'impiego dei comandi PEEK e POKE per il loro uso più corretto.

Il tutto è accompagnato da 18 programmi la maggior parte dei quali o sono inediti, oppure riguardano versioni migliorate di games di grande successo come "INVADERS" e "BOMBER". I più interessanti sono stati registrati sulla **cassetta allegata al volume** al fine di farvi risparmiare ore di digitazione.

Cod. 9003

L. 25.000
prezzo comprensivo di cassetta

