

PERSONAL SOFTWARE

ANNO 3 N. 16
MARZO 1984 - L. 3.500

UNA PUBBLICAZIONE DEL GRUPPO EDITORIALE JACKSON



Copia riservata agli abbonati



- **I BOSS DEL PETROLIO: UN GIOCO PER VIC 20**
 - **MUSIC EDITOR CON IL C64**
- **GIOCHI AFRICANI PER ZX SPECTRUM**
 - **SPRITE PER C64**
- **UNA NAVE CARICA DI PROGRAMMI PER ZX81**

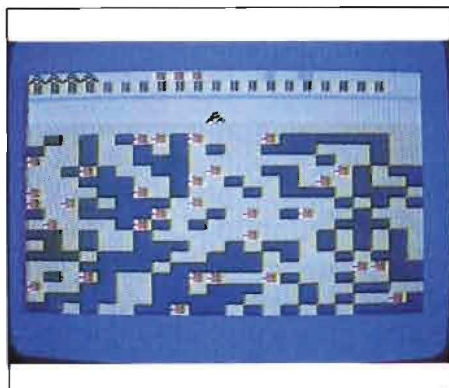
I boss del petrolio

— Parte prima —

Un programma per VIC 20 + joystick

di Enrico Comini

Immaginatevi di essere uno Ewing come J.R. o come Bobby se il primo non vi risultasse simpatico, di essere in definitiva "Boss del petrolio". Estrarre il petrolio non è però semplice, bisogna riuscire a superare molti ostacoli per trarne profitto. Vi sono rocce formate da minerali particolarmente duri che rovinano la trivella ed allora è necessario passare a mezzi più energici, bisogna impiegare qualche candelotto di dinamite, le cui scorte non sono però illimitate. Nelle cavità vuote da cui l'oro nero è già stato estratto, sussiste il pericolo di infiltramento di gas naturali che sono altamente esplosivi, per cui può capitare che mentre la trivella le attraversa si crei una deflagrazione dannosa per l'impianto. Esistono inoltre dei piccoli diavoletti che hanno scelto il petrolio come loro dimora e questa simpatica famiglia di individui non vede di buon occhio le trivelle che succhiano il loro elemento vitale, per cui ogni qualvolta si tenta di pompare il petrolio nel quale vive un diavoleto costui mette fuori uso il pozzo. Comunque il vero magnate del petrolio è un "duro" e non si lascia intimorire; procede sicuro sulla sua strada per fare sempre più grande il suo gruzzolo di sonanti dollari.



Una videata d'esempio di gioco.

Caricamento

Il programma gira sul VIC non espanso; è formato da due programmi separati da lanciarsi consecutivamente. Il primo, che carica in memoria i simboli grafici ed alcune routine in linguaggio macchina per l'utilizzo del joystick, contiene anche le istruzioni per giocare. Una volta che questo primo programma è stato lanciato appaiono sul video le istruzioni e viene automaticamente caricato in memoria il secondo programma che costituisce il gioco vero e proprio.

Alla fine del caricamento del secondo programma appare la scritta "PREMERE STOP SUL TAPE" ed il programma attende fino a che questa operazione non è stata compiuta.

Livelli di difficoltà

Lo schermo che appare dopo che si è premuto lo "STOP" mostra il massimo punteggio per ognuno degli otto livelli di difficoltà. Il programma ritorna a questo schermo

ogni volta che il gioco ha termine. Appena terminato il gioco il punteggio ed il livello di difficoltà appaiono nella parte superiore del video. In fondo a questo quadro appare la scritta "DIFF. LEVEL 12345678" ed il livello di difficoltà va scelto muovendo il joystick a destra e premendo il bottone di fuoco quando il livello al quale si è interessati sta lampeggiando. Il livello 1 è il più semplice, per bambini per così dire. Per i giocatori più anziani (in fatto di computer game s'intende) si consiglia di iniziare a giocare dal livello 2. Più alto è il livello di difficoltà e più difficile diventa il gioco; le variazioni che si hanno per ogni livello sono quelle mostrate dal seguente specchio:

LIVELLO	DINAMITE	ROCCHE DURE	ROCCHE INVISIBILI
1	3	20	NO
2	2	20	NO
3	3	30	NO
4	2	30	NO
5	4	20	SI
6	3	20	SI
7	4	30	SI
8	3	30	SI

Come si gioca

Dopo che si è scelto il livello appare sul video il terreno petrolifero. Questo sarà differente per ogni gioco, è impossibile vedere per due volte lo stesso schermo. Ad ogni gioco si hanno a disposizione 5 torri per l'estrazione del petrolio ognuna delle quali ha a disposizione un condotto con trivella formato da 20 pezzi ed un numero di candelotti di dinamite che dipende dal livello scelto. Nella parte superiore sinistra sono visibili le torri che ancora si hanno a

disposizione ed in quella destra il punteggio.

Nel mezzo vi sono i candelotti di dinamite ancora da utilizzare e sotto questi, i tubi di estensione per la trivella.

Appena si inizia a perforare si vedono queste estensioni diminuire e così le si vedranno aumentare quando la trivella verrà ritirata. La parte bassa dello schermo è il terreno di gioco, i quadrati gialli sono rocce, quelli neri petrolio e quelli irregolari rossi rocce dure. Muovendo il joystick a destra e a sinistra la torre per l'estrazione si sposta sopra il campo fino al punto in cui si decide di perforare. Per perforare è necessario premere il joystick verso il basso. Per ritirare la trivella bisogna portare la leva del joystick verso l'alto.

Mentre la trivella è nel terreno non è possibile spostare il pozzo a destra e a sinistra; non si possono inoltre superare con la trivella le rocce dure e i diavoletti. Cercando di continuare a perforare dove non è possibile, l'unica cosa che si ottiene è quella di spezzare la trivella e perdere così alcuni dei tubi di prolunga che sono disponibili (la diminuzione delle estensioni è visibile nella parte superiore dello schermo).

Questo fatto diventa molto importante se si gioca a livelli di difficoltà superiori al quarto, dove le rocce dure sono invisibili ed allora è molto facile rompere la trivella mentre si cerca di raggiungere un giacimento. Bisogna anche andare molto cauti nel passare con la trivella attraverso gli spazi vuoti dai quali il petrolio è già stato pompato perché, in questi spazi, è possibile che si infiltrino gas naturale che essendo molto esplosivo può anche deflagrare al passaggio della trivella. Il bottone del fuoco si può utilizzare per fare tre cose: se la torre non ha esteso la

1-4	Inizializzazioni.
5-23	Schermo massimi punteggi e selezione difficoltà.
24-25	Nuova torre per l'estrazione del petrolio.
26-29	Trivella.
30-33	Ciclo primario.
34-39	Ciclo secondario.
40-45	Arretramento trivella.
46-60	Aspirazione.
61-68	Parte superiore schermo.
69-77	Inserimento dinamite.
78	Gas naturale.
79-81	Esplosione condotto e pozzo.
82	Suono di campanella.
83-84	Suono esplosione.
85-86	Diavoletti.
87-92	Formazione schermo.
93	Disegno.
99	Suono della perforazione.

Tabella 1. *Lista delle principali routine del programma 2.*

trivella premendo il pulsante si ha a disposizione una nuova torre con tutte le estensioni e tutti i candelotti di dinamite. Se la trivella è dentro ad un giacimento di petrolio o in uno spazio vuoto premendo il pulsante si pompa all'esterno il contenuto della cavità (petrolio o aria), mentre invece se la trivella si trova al di sopra di una roccia (non necessariamente dura) si manda nel terreno un candelotto di dinamite.

Se al momento di sganciare la dinamite ci si trova al di sopra di uno spazio vuoto o di un giacimento di petrolio la dinamite esce dalla trivella e procede attraverso lo spazio vuoto o il petrolio finché non incontra una roccia o un diavoletto.

Estrazione del petrolio

Quando si decide di pompare, tutto il petrolio contenuto negli spazi che stanno ai lati e sopra la trivella,

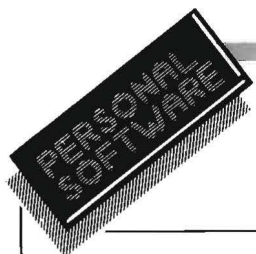
verrà estratto all'esterno. In altre parole tutti i quadrati di petrolio connessi a quello del quale si sta pompando, saranno pompato all'esterno solo se essi giacciono direttamente sopra o ai lati di questo. Ogni quadrato di petrolio che sta al di sotto di quello nel quale si trova la trivella non verrà pompato. Se, mentre il petrolio viene aspirato, si scopre un diavoletto all'interno di questo, allora il pozzo sarà inevitabilmente perso. Se si cerca di pompare in un giacimento che è connesso (come spiegato sopra) con uno spazio contenente un diavoletto già scoperto si causa la distruzione del pozzo. Alla fine di un'estrazione il punteggio viene incrementato. Ogni 100.000 punti raggiunti si ha un pozzo extra. Si ottiene pure un pozzo extra nel caso in cui dal campo è stato estratto tutto il petrolio.

Quando si verifica questa situazione compare un nuovo campo diverso dal precedente.

Come si procede per caricare il programma

La prima cosa da fare è battere il primo programma, salvarlo sul nastro e fare un VERIFY. Quindi bisogna premere STOP sul registratore senza però riavvolgere il nastro, dare il NEW e battere il secondo programma, salvare anche questo sul nastro, tornare dove tale programma inizia e fare un VERIFY. Se tutto è a posto riavvolgere il nastro completamente, dare un NEW e poi un LOAD (attenzione: NON premere STOP dopo che il primo programma è stato caricato).

Ora si può dare il RUN ed il primo programma parte caricando e lanciando automaticamente il secondo programma.



I boss del petrolio

Listato 1. Il programma 1.

```

20 PRINT"☐":PRINT" ISTRUZIONI":P
RINT:PRINT"JOISTICK☐":PRINT" DX & SX=
MUOVE POZZO"
30 PRINT" BASSO=TRIVELLAMENTO":PRINT"
ALTO=ARRETRAMENTO"
40 PRINT:PRINT"BOTTONI FUOCO☐":PRINT
"AL SUOLO=NUOVO POZZO":PRINT"IN PETROL
IO O SPAZIO=POMPA"
60 PRINT:PRINT"ATTENZIONE ☐AL GAS
NEGLI SPAZI VUOTI","SE RILASCI I DIAVOLI N
EL PETROLIO"
70 PRINT:PRINT"ATTENDI LE PROSSIME IS
TRUZIONI "
80 POKE52,28:POKE56,28:POKE45,248:POKE
46,26:CLR
90 FORA=7168TO7375:READB:POKEA,B:NEXT
100 DATA2,138,164,73,74,52,20,0,64,81,
37,146,82,44,40,16,24,24,36,60,90,102,2
31,153
110 DATA20,42,42,20,62,73,20,20,136,34
,136,34,136,34,136,34,148,22,148,34,136
,34,136,34
120 DATA148,22,148,22,148,34,136,34,14
8,22,148,22,148,22,148,22,136,62,188,62
,188,62,188,22
130 DATA170,170,170,170,170,170,170,17
0,150,150,150,170,170,170,170,170,150,1
50,150,150,150
140 DATA170,170,170,150,150,150,150,15
0,150,150,150,150,190,190,190,190,190,1
90,150
150 DATA0,0,0,0,0,0,0,0,20,20,20,0,0,0
,0,0,20,20,20,20,20,0,0,0,20,20,20,20,2
0,20,20,20
160 DATA20,60,60,60,60,60,60,20,0,60,6
0,60,60,60,60,0,0,0,0,255,255,0,0,0
170 DATA18,118,181,153,110,93,197,65,
65,82,150,85,121,181,150,173,2,106,129,
20,64,162,129
180 DATA2,24,60,62,126,124,62,28,0,136
,34,136,34,136,34,136,34
190 FORA=7424TO7640:POKEA,PEEK(25680+A
):NEXT
200 FORA=7464TO7529:READB:POKEA,B:NEXT
210 DATA169,128,141,19,145,169,0,133,1
,133,2,169,127,141,34,145,162,119,236,3
2,145
220 DATA208,4,169,1,133,1,169,255,141,
34,145,162,118,236,17,145,208,4,169,22,
133,1
230 DATA162,110,236,17,145,208,4,169,1
,133,2,162,122,236,17,145,208,4,169,22,
133,2,96
240 LOAD
1500 ,0,0,0,0,0,0,0,20,20,20,0,0,0,00,
20,20,20,20,0,0,0,20,20,20,20,20,20,20

```

Seguito programma 1.

Lista simboli grafici

```

20 : 1 ☐ = SHIFT HOME [CHR$(147)]
1 = CTRL 5 [CHR$(156)]
1 ☐ = CTRL 7 [CHR$(31)]

40 : 1 = CTRL 5 [CHR$(156)]
1 ☐ = CTRL 7 [CHR$(31)]

60 : 1 = CTRL 5 [CHR$(156)]
1 ☐ = CTRL 7 [CHR$(31)]
1 ☐ = CTRL 8 [CHR$(158)]
1 ☐ = CTRL 7 [CHR$(31)]
1 ☐ = CTRL 7 [CHR$(31)]
1 ☐ = CTRL 5 [CHR$(156)]
1 ☐ = CTRL 1 [CHR$(144)]

70 : 1 ☐ = CTRL 7 [CHR$(31)]

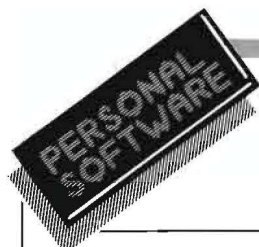
```

Listato 2. Il programma 2.

```

2 PRINT:PRINT" PREMI STOP SUL TAPE"
3 IFPEEK(37137)<100THEN3
4 POKE36879,30:POKE36878,47:DIMAX(21):
M=7768
5 IFZ>BX(T)THENBX(T)=Z
6 POKE36869,240:PRINT"☐"CHR$(28)" "T
,"#Z*100"☐"
7 PRINT:PRINT" LIVELLO","HIGH SCORE"
8 FORA=1TO8:PRINT:PRINT" A,"#BX(A)
*100:NEXT:PRINT:PRINT" DIFF.LIVEL? 1234
5678☐":T=1
9 SYS7464:POKE38852+T,6:T=T+PEEK(1)-PE
EK(2):IFT<10RT>8THENT=1
10 POKE38852+T,7:L=3:IFT/2=INT(T/2)THE
NL=2
11 S=20:IFT=30RT=40RT>6THENS=30
12 N=24:IFT>4THENN=25:L=L+1
13 IFPEEK(37137)>100THEN9
14 POKE36869,255
23 PRINT"☐":M=5:Z=0:K=0:GOSUB87
24 POKEW+X,14:X=10:M=M-1:P=20:Y=L:GOSU
B82:IFM<0THENS
25 GOSUB61:GOTO34
26 A=(R*22)+W+X:C=PEEK(A):P=P-1:GOSU
B67
27 IFC=NORC=30RA>8185THENFORA=1TO3:GOS

```

I boss del petrolio

Seguito programma 2.

```

UB99: NEXT: GOTO30
28 IFRND(1) < .06ANDC=14 THEN78
29 FORB=1T03: POKER,C+B: GOSUB99: NEXT: R=
R+1
30 SYS7464: IFPEEK(1)=22ANDP>0 THEN26
31 IFPEEK(2)=22 THEN40
32 IFPEEK(37137) < 100 THEN46
33 GOTO30
34 SYS7464: A=PEEK(1): B=PEEK(2): IFA=22T
HENR=1: GOTO26
35 IFB=22 THEN34
36 IFPEEK(37137) < 100 THEN24
37 IFAD>0ORB>0 THENPOKEW+X, 14: X=X+A-B: IF
X>21 THENX=21
38 IFX<0 THENX=0
39 POKEW+X, 2: GOTO34
40 IFR=1 THEN34
41 R=R-1: B=(R*22)+W+X: C=PEEK(B): FORA=1
T03: POKEB,C-A: GOSUB99: NEXT
42 P=P+1: GOSUB67: IFR=1 THEN44
43 GOTO30
44 FORA=W+44T08185: IFPEEK(A)=9 THEN34
45 NEXT: M=M+1: FORC=1T03: GOSUB82: NEXT: G
OSUB87: GOSUB61: GOTO34
46 J=2: Q=R-1: FORA=0T021: AX(A)=0: NEXT: B
=(Q*22)+W+X: A=PEEK(B): IFA=7 THEN69
47 AX(X)=1: POKE36877, 252: POKE36878, 36:
V=W+X-22: IFA=12 THENPOKEB, 17: Z=Z+Q: POKEV
,0
48 E=0: F=20: D=1: G=1: I=1: GOSUB52: POKEV,
1: E=21: F=1: D=-1: G=D: I=0: GOSUB52
49 E=0: F=21: D=1: G=-22: I=0: GOSUB52: POKE
V,0: IFC=6 THEN85
50 IFH=0 THENPOKEV, 14: POKE36877, 0: POKE3
6878, 47: GOSUB82: GOSUB61: GOTO30
51 Q=Q-1: GOTO48
52 IFC=6 THENRETURN
53 H=0: FORA=ETOFSTEPD: IFAZ(A)=0 THEN60
54 B=(Q*22)+W+A+G: C=PEEK(B)
55 IFC=9ORC=12 THENPOKEB,C+5: H=1: Z=Z+(Q
+(ABS(I)-1)): AX(A+I)=1: GOTO59
56 IFC=14ORC=17 THENAX(A+I)=1: H=1: GOTO6
0
57 IFC=3 THENC=6: RETURN
58 AX(A+I)=0: GOTO60
59 IFRND(1) < .02ANDC<>12 THENC=6: RETURN
60 NEXT: RETURN
61 PRINT"@"SPC(13)"#Z*100
62 A=INT(Z/1000): IFA=K+1 THENK=K+1: GOSU
B82: GOSUB82: M=M+1
63 IFM<1 THENPOKE7680, 14: GOTO65
64 FORA=7680T07679+M: POKER, 2: NEXT: POKE
A, 14
65 IFY=0 THENPOKE7687, 14: GOTO67
66 FORA=7687T07686+Y: POKER, 19: NEXT: POK
ER, 14
67 IFP<1 THENPOKE7702, 14: RETURN
68 FORB=7702T07701+P: POKEB, 17: NEXT: POK
EB, 14: RETURN
69 A=W+X: B=240: IFY<1 THEN30

```

Seguito programma 2.

```

72 A=A+22: C=PEEK(A): POKER,C+1: IFC=14OR
C=9 THENPOKER,C+4
73 POKE36875, B: FORD=0T0200: NEXT: D=PEEK
(A+22)
74 IFD<>4ANDD<>NANDD<>3ANDAC<8164 THENB=
B-5: POKER,C: GOTO72
75 POKE36875, 0: GOSUB93: POKER+30742, 15
76 IFC<>14ANDC<>9 THENR=R-1
77 Y=Y-1: GOSUB65: GOTO30
78 FORB=1T02: POKER,C+B: GOSUB99: NEXT: GO
SUB93: B=150
79 R=R-1: POKE36878, 36: IFR<1 THENPOKE368
78, 47: GOTO81
80 POKE36877, B: A=(R*22)+W+X: C=PEEK(A):
POKER,C+1: FORD=0T0200: NEXT: POKER,C-3: B=
B+5: GOTO79
81 POKEW+X, 23: GOSUB83: GOTO24
82 FORA=47T032STEP-1: POKE36878, A: POKE3
6876, 237: FORB=1T025: NEXTB, A: POKE36876, 0
: POKE36878, 47: RETURN
83 POKE36877, 220: FORD=47T032STEP-1: POK
E36878, 0: POKE36879, 26: FORE=1T070: NEXT: P
OKE36879, 31
84 NEXT: POKE36877, 0: POKE36878, 47: POKE3
6879, 30: RETURN
85 Z=J: POKEB, 3: POKEB+30720, 2: POKEV, 14:
FORA=0T040: POKE36876, 240: POKE36878, 230:
FORB=1T05: NEXT
86 POKE36878, 0: FORB=1T05: NEXTB, A: POKE3
6876, 0: B=150: GOTO79
87 FORA=7790T08185: POKER, 4: POKER+30720
, 15: NEXT: B=220: C=7812: FORA=1T02: FORD=1T
080
88 E=(RND(0)*B)+C: IFPEEK(E)=9 THEN88
89 POKEE, 9: POKEE+30720, 8: NEXT: B=154: C=
8032: NEXT: FORA=1T05
90 B=(RND(1)*374)+7812: C=PEEK(B): IFC=9
ORC=N THEN90
91 POKEB, N: NEXT: FORA=0T0109: POKE38400+
A, 0: NEXT: FORA=0T021: POKE38444+A, 3: POKE7
724+A, 20: NEXT
92 FORA=0T03: POKE38407+A, 2: NEXT: RETURN
93 POKER, 21: POKER+22, 22: GOSUB83: POKER,
14: POKER+22, 14: RETURN
99 POKE36874, 200: FORE=1T010: NEXT: POKE3
6874, 0: RETURN

```

Lista simboli grafici

- 6 : 1 = SHIFT HOME [CHR\$(147)]
- 1 = CTRL 7 [CHR\$(31)]
- 8 : 1 = CTRL 6 [CHR\$(30)]
- 23 : 1 = SHIFT HOME [CHR\$(147)]
- 61 : 1 = HOME [CHR\$(19)]

Sprite per C 64

— Parte prima —

Figure grafiche animate che danno nuovi sbocchi alla fantasia creativa nella programmazione

di Flavio Stella

Avvicinandosi al C 64, uno dei primi passi è senz'altro l'esplorazione delle possibilità grafiche offerte dai cosiddetti sprite nell'elaborazione grafica ed in particolare nella grafica applicata ai giochi. Questi grafici animati possono essere definiti, colorati e spostati a piacere in sovrapposizione al testo.

Il C 64 è in grado di gestire contemporaneamente fino ad otto sprite e permette quindi un discreto salto di qualità nella programmazione.

La parola Sprite, letteralmente tradotta, significa Folletto, quindi, con la tipica immediatezza e trasparenza di certa terminologia inglese, ci viene suggerito che siamo di fronte a qualcosa di magico, elastico, che compare e svanisce nel nulla nel breve volgere di un secondo. Lo scopo di questo articolo è di illustrare i trucchi che stanno dietro a questa magia e di fornire una serie di strumenti, adattabili alle varie esigenze, che rendano facile e soprattutto immediato il trasferimento dalla teoria alla pratica sotto forma di agili figure colorate.

Il disegno

È sempre possibile, fissata una certa risoluzione, scomporre una fi-

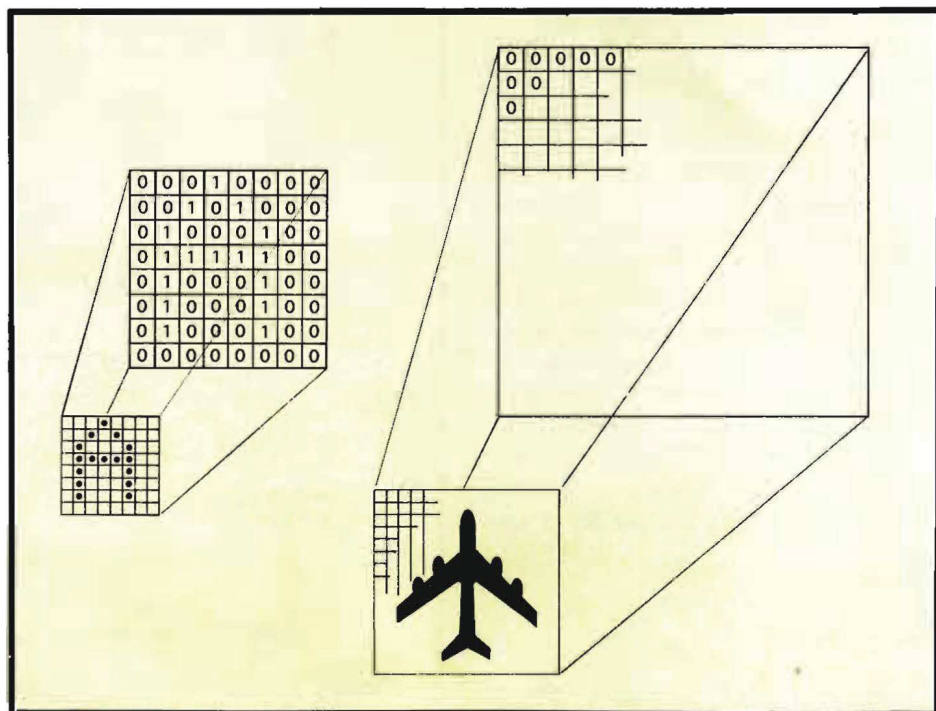


Figura 1. Rappresentazione schematica della trasformazione di un carattere grafico in una matrice binaria.

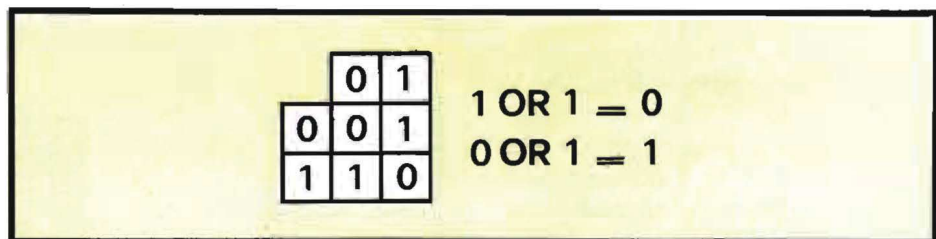


Figura 2. Tabella della verità dell'operazione di or esclusivo con cui si opera sulle locazioni di memoria contenenti lo sprite.

gura in un insieme di punti elementari; a questa rappresentazione puntiforme è facile sovrapporre un reticolo opportunamente proporzionato; segnalando poi con 0, in un casellario corrispondente, ogni spazio non occupato e con 1 ogni punto, avremo ottenuto una matrice binaria i cui elementi saranno memorizzabili bit per bit in qualsiasi computer. Basandosi su questo principio le lettere ed i caratteri grafici disponibili sulla tastiera del C 64 sono codi-

ficati in una matrice di 8 x 8 punti mentre gli sprite occupano uno spazio più grande, precisamente 21 x 24 punti (vedi figura 1).

Per ottenere la codificazione di uno sprite bisognerà quindi procedere come segue:

- tracciare un reticolo con 21 righe di 24 caselle ciascuna e farne una copia,
- comporre il disegno desiderato sul primo schema colorandone le caselle in modo opportuno,

Sprite per C 64

— inserire sulla copia uno 0 in caselle corrispondenti ad un vuoto ed un 1 nelle altre,

— cominciando dall'elemento in alto a sinistra scorrere tutta la matrice binaria attribuendo ad ogni 8 bit binari successivi il loro corrispondente valore decimale (vedi listato 1 - conversione Binario/Decimale),

— concludere la serie dei 63 (21 ★ (24/8)) valori ottenuti con uno 0 che serve da separatore e quadra il numero totale ad una potenza di due ($216 = 64$).

Ora la codifica è completa ed il blocco dei 64 numeri decimali pronto ad essere inserito opportunamente nella memoria con i criteri che verranno esposti nella seconda parte.

Lo strumento proposto per minimizzare questa lunga e noiosa parte del lavoro è il programma del listato 2.

Sprite drawer

Subito dopo l'introduzione questo programma traccia sul video il reticolo 21 x 24 che fa da tavola da disegno ed una lista dei colori disponibili.

Tramite il joystick bisognerà posizionarsi in corrispondenza del colore prescelto e premere FIRE; a questo punto il cursore sarà già pronto, all'interno dello spazio delimitato, a ricevere i comandi di movimento ed a marcare con il colore selezionato la sua posizione, se verrà premuto il tasto del joystick; ripetendo l'operazione su di una casella già piena si otterrà l'effetto contrario, cioè la cancellazione. Il disegno sarà visibile sia sulla tavola dove scorre il cursore sia attraverso lo sprite, aggiornato in tempo reale, che si trova all'interno del riquadro PROVA e che fornisce in termini grafici e di colore l'esatta rappresentazione del risultato ottenuto fino a quel momento (figura 3).

L'operazione creativa non ha limiti di tempo ed è consentito cambiare il colore, una o più volte, pre-

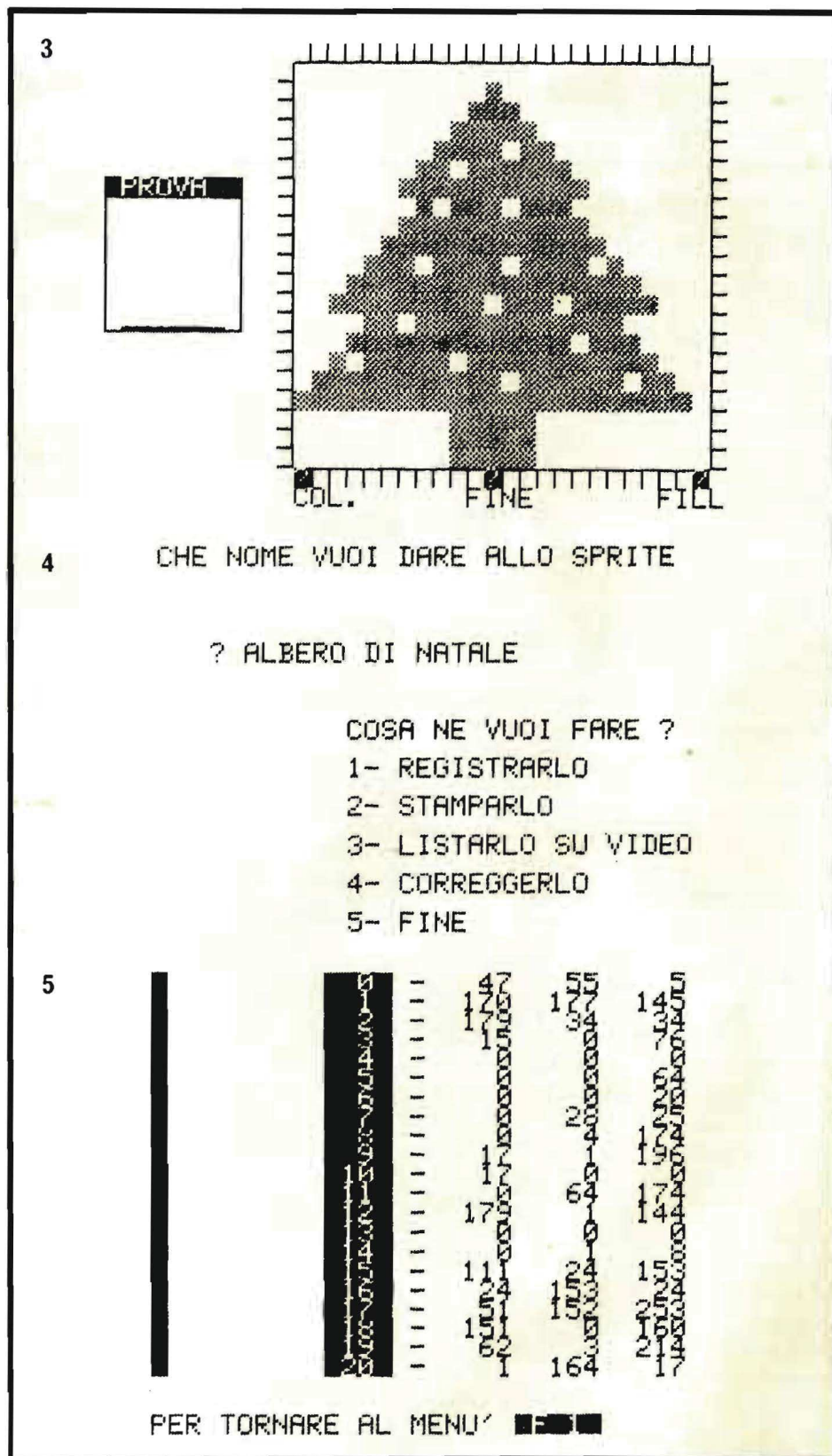



Figure 3/4/5. Tre hard copy del video durante l'esecuzione del programma; lo sprite non viene rilevato dalla stampante ma è presente sullo schermo in tutte e tre le occasioni.


Sprite per C 64

6



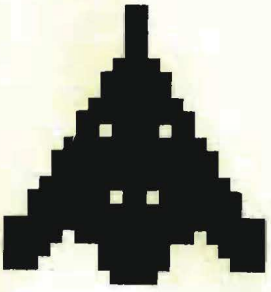
0	112	0	0	112	0	0	48	0	0	0	0
0	252	0	1	186	0	0	89	0	6	104	128
28	112	128	0	121	0	0	120	0	0	124	0
0	204	0	1	140	0	1	135	224	1	128	32
3	0	32	3	0	0	3	7	0	14	7	0
0	7	0	2								

7



0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	240	0	0	252	0	1	255	255
1	254	0	31	255	240	63	255	252	127	255	254
110	247	118	127	255	254	63	255	252	27	221	216
15	255	240	0	0	0	0	0	0	0	0	0
0	0	0	0								

8



0	24	0	0	24	0	0	24	0	0	24	0
0	60	0	0	126	0	0	126	0	0	255	0
1	255	128	1	189	128	3	255	192	7	255	224
7	255	224	15	255	240	31	219	248	31	255	248
127	255	254	125	255	190	120	126	30	112	126	14
0	60	0	0								

mendo il tasto del joystick dopo aver posizionato il cursore sull'apposita casella COL.

La funzione FILL, invece, dà la possibilità di riempire figure di cui siano tracciati solo i contorni risparmiando una notevole quantità di tempo. Questa routine contiene alcune imperfezioni che talvolta costringono a rifinire manualmente l'operazione; il considerare tutte le conformazioni possibili avrebbe richiesto un maggior numero di istruzioni e quindi un ulteriore rallentamento di tutta la procedura senza recare, a mio parere, benefici adeguati. Quando la composizione dello sprite sarà definitivamente ultimata si dovrà uscire premendo il tasto sulla casella FINE, la nuova videata conserverà l'immagine dello sprite e chiederà un nome che individuerà il soggetto ai fini di un successivo utilizzo; immediatamente dopo verrà proposto un menù (figura 4) per la scelta del tipo di archiviazione dei dati ottenuti, sarà possibile:

- registrarli su nastro,
- stamparli su carta (figure 6, 7 e 8),

Figure 6/7/8. Tre sprite stampati con diverso soggetto riportanti la codifica decimale pronta per essere utilizzata nei programmi.



Sprite per C 64

— visualizzarli subito sullo schermo (figura 5).

Sarà sempre offerta, dopo ciascuna di queste routine, la possibilità di rientrare nel menù per operare una nuova scelta, per ripetere il procedimento già utilizzato oppure accedere di nuovo al disegno per correzioni; quest'ultima opportunità è stata prevista per programmare le animazioni; è possibile infatti sovrapporre in rapida successione una serie più o meno lunga di sprite, differenti solo per qualche particolare, per dare l'impressione del movimento.

Caratteri programmabili

Abbiamo già menzionato i caratteri programmabili nella trattazione teorica per far capire che gli sprite altro non sono che grandi caratteri; ma se i due concetti sono così simili perché non utilizzare il programma Sprite drawer anche per i caratteri? Nell'area 21 x 24 destinata ad un sprite trovano posto fino a 6 caratteri 8 x 8.

Composto il disegno, la serie di valori che lo codificano si potrà dedurre con facilità dalla lista su video che presenterà in ogni colonna due gruppi di otto numeri separati da alcuni 0, uno per ciascun carattere disegnato (figura 9).

Il listato 3 vuole fornire uno spunto all'uso di questa possibilità che, affiancata a quanto già visto per gli sprite, arricchisce ancor più le nostre conoscenze. Dopo aver trascritto 64 caratteri dalla ROM alla RAM i primi 6 (@ABCDE) vengono sostituiti con le lettere greche che formano la parola ASTERI, cioè stella, (figura 9) ed i successivi 3 (FGH) con tre invasori stabilizzati (figura 10). Il programma fornisce automaticamente una prova di stampa prima della sua conclusione ma, prolungandolo opportunamente o, più semplicemente, premendo uno dei tasti corrispondenti ai 9 caratteri indicati prima, sarà possibile utilizzare le lettere greche ed i disegni oppure, sostituendo i numeri nelle istru-

BINARIO/DECIMALE

```

10 PRINT"[<1CLR>][<1RVS>][<1R.C>]      CO
NVERSIONE BINARIO/DECIMALE          "
20 PRINT"[<3CRSR D>][<1YEL>]          ";FORI=
0T07:PRINT"[<1CHR$(204)>][<1CHR$(175)>][
<1CHR$(186)>]";NEXT:PRINT"          [<1CHR$(2
04)>][<3CHR$(175)>][<1CHR$(186)>][<2CRSR
D>][<4CRSR U>]"30 PRINT"          ";FORI=0T07
40 GETA$:IFA$=""THEN40
60 IFASC(A$)<48ORASC(A$)>49THEN40
70 PRINT"          "A$;B$=B$+A$:NEXT
80 FORI=0T07:D=D+VAL(MID$(B$,I+1,1))*2↑(
7-I):NEXT:PRINT"          "D
90 B$="" :D=0:PRINT:GOTO20

```

Listato 1. Conversione Binario/Decimale.

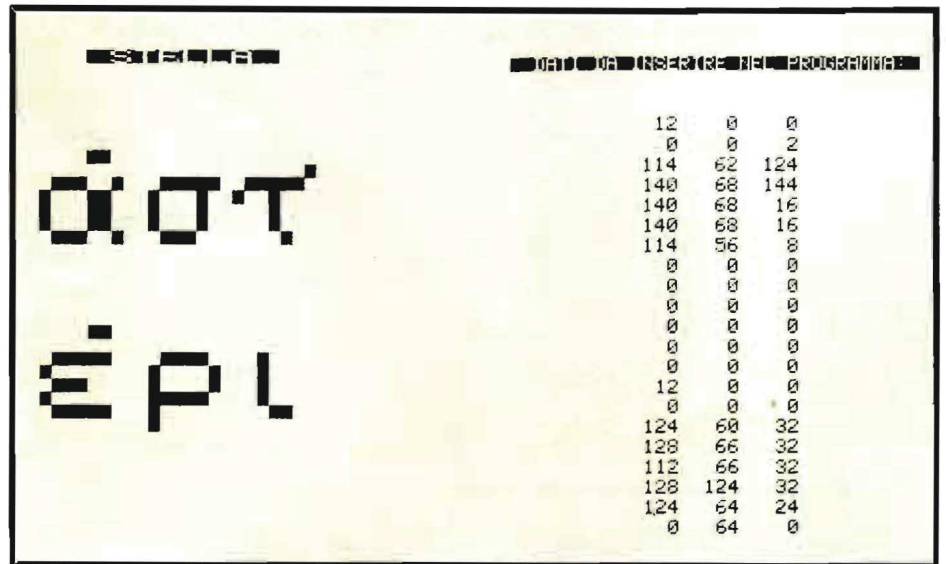


Figura 9. Lettere dell'alfabeto greco che formano la parola ASTERI realizzate con Sprite drawer.

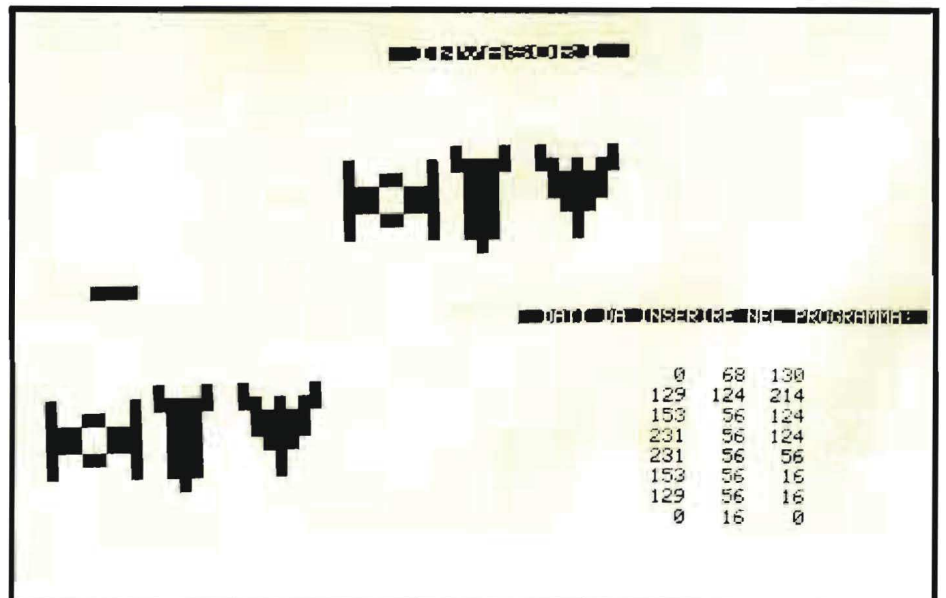
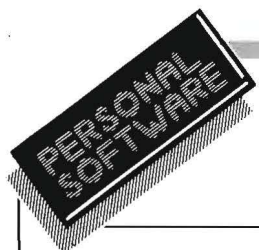


Figura 10. Tre piccole astronavi da utilizzarsi come caratteri programmabili.



Sprite per C 64

zioni DATA e/o aggiungendone di nuovi, creare altri simboli grafici.

Conclusioni

In questa prima parte abbiamo analizzato i meccanismi di formazione della codifica necessaria all'impiego degli sprite e dei caratteri programmabili; mi auguro che con l'aiuto del programma proposto ciascuno possa creare, col tempo e la pazienza necessari, gli effetti che invidiamo ai videogiochi più sofisticati.

Nella seconda parte vedremo come creare le procedure di gestione degli sprite (movimento/colore/posizione/collisioni/ecc.); sarà presentato un programma che legge i dati registrati sul nastro con Sprite drawer e li trasforma immediatamente in figure animate permettendo poi di costruire, intorno al nucleo centrale, i programmi più disparati.

10-30	Inizializza le variabili e abilita lo sprite prova.
40-50	Routine in LM per il posizionamento del cursore.
60-250	Cambia colore allo schermo e prepara la prima videata.
260	Posiziona il cursore e permette la scelta del colore.
270-290	Prepara il riquadro PROVA.
300-390	Gestisce i movimenti, le scelte e la creazione del disegno.
400-520	Scelta del menu archiviazione.
550-590	Registrazione su nastro.
600-760	Stampa su carta.
800-840	Lista su video.
900-930	Gestisce il messaggio di ritorno la menu.
2000-2080	Muove il cursore posizionandolo con la routine in LM.
2100-2200	Lettura joystick.
2500-2510	Calcola, in base alle coordinate del cursore, l'indirizzo del byte (ed il bit al suo interno) interessato all'operazione che la richiama.
3000-3120	Esegue la funzione FILL.
4000-ss.	Introduzione.

Tabella 1. Principali routine del programma Sprite drawer.

SL	Indirizzo del blocco di 64 byte per sprite prova.
V	Indirizzo base dei registri del circuito VICII.
CS	Insieme dei colori disponibili.
X, Y	Coordinate del cursore rispetto allo schermo.
L, YL	Minimo valore x, y ammessi per limitare l'area del disegno.
XH, YH	C.s. ma valore massimo.
CO	Colore prescelto.
P	Indirizzo byte della mappa video sotto il cursore.
PX, PY	Coordinate cursore rispetto al riquadro delimitato.
PM	Indirizzo byte considerato nel blocco sprite.
BI	Valore decimale di un uno nel bit considerato.
NMS	Nome attribuito allo sprite.

Listato 2. Programma Sprite drawer.

I tasti grafici e speciali sono stati racchiusi tra i simboli <>, all'interno di parentesi quadre e preceduti da un numero che specifica quante volte occorre digitare quel determinato carattere. Es. [<8CRSRD>], digitare per 8 volte il tasto di spostamento verso il basso del cursore. Per i caratteri grafici viene indicato il codice ASCII corrispondente; consultare il manuale per la conversione.

Tabella 2. Principali variabili del programma Sprite drawer.

SPRITE DRAWER

```

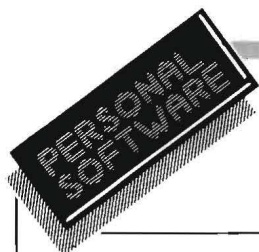
10 GOSUB4000:SL=12288:FORI=SLTOSL+64:POK
EI,0:NEXT:POKE187,0
20 V=53248:POKE2042,192:SL=12288
30 POKEV+23,4:POKEV+29,4:POKEV+4,50:POKE
V+5,130
40 DATA24,166,2,164,182,32,240,255,96
50 FORI=679T0687:READA:POKEI,A:NEXT
60 POKE53280,11:POKE53281,12:C$="["<1BLK>
J["<1WHT>J["<1RED>J["<1CYN>J["<1PUR>J["<1GRN>
J["<1BLU>J["<1YEL>J["<1ARA>J["<1MAR>J["<1R.C>
J["<1G.1>J["<1G.2>J["<1V.C>J["<1AZZ>J["<1G.3>
J":REM SIMBOLI COLORI
80 PRINT["<1CLR>J["<1BLU>J["<1CRSR D>]"TAB
(13):FORI=0T023:PRINT["<1CHR$(186)>]":N
EXT:PRINT
90 FORI=0T020:PRINTTAB(12)["<1CHR$(186)>
]"SPC(24)["<1CHR$(204)>]":NEXT
100 IFHK=1THENPRINT["<1HOME>J["<22CRSR D>
J["<1BLU>]"
110 PRINTTAB(13):FORI=0T023:PRINT["<1CHR
$(207)>]":NEXT:PRINT
120 PRINT["<1CRSR U>]"TAB(13)["<1RVS>J["<
1RED>]?"TAB(24)?"TAB(36)?"

```

```

130 PRINTTAB(13)["<1G.1>]COL:"TAB(23)"FI
NE"TAB(33)" FILL["<1CRSR U>]":IFHK=1THENH
K=0:RETURN
140 PRINT["<1HOME>J["<3CRSR D>] COLORE [
<1CRSR D>J["<8CRSR L>J["<9CHR$(183)>]"
200 PRINT["<1RED>] -["<1RVS>]NERO ["<1BL
K>] ["<1RED>]":PRINT -["<1RVS>]BIANCO ["<1
WHT>] ["<1RED>]":PRINT -["<1RVS>]ROSSO [
<1RED>] ["<1RED>]"
210 PRINT -["<1RVS>]CELESTE["<1CYN>] ["<1R
ED>]":PRINT -["<1RVS>]PORPORA["<1PUR>] ["<
1RED>]":PRINT -["<1RVS>]VERDE ["<1GRN>]
["<1RED>]"
220 PRINT -["<1RVS>]BLU ["<1BLU>] ["<1R
ED>]":PRINT -["<1RVS>]GIALLO ["<1YEL>] ["<
1RED>]":PRINT -["<1RVS>]ARANCIO["<1ARA>]
["<1RED>]"
230 PRINT -["<1RVS>]MARRONE["<1MAR>] ["<1R
ED>]":PRINT -["<1RVS>]ROSSO C["<1R.C>] ["<
1RED>]":PRINT -["<1RVS>]GRIGIO1["<1G.1>]
["<1RED>]"
240 PRINT -["<1RVS>]GRIGIO2["<1G.2>] ["<1R
ED>]":PRINT -["<1RVS>]VERDE C["<1V.C>] ["<
1RED>]":PRINT -["<1RVS>]JAZZURRO["<1AZZ>]
["<1RED>]"
250 PRINT -["<1RVS>]GRIGIO3["<1G.3>] ["<1R

```

Sprite per C 64

Seguito listato Sprite drawer.

```

ED>]:PRINT"[<1CRSR D>] [<1G.1>] [<9CHR$(183)>]"
260 YL=5:YH=20:HH=1:X=10:Y=5:GOSUB2000:H
H=0:CO=Y-5:POKEV+41,CO:POKEV+21,4
270 PRINT"[<1HOME>]":FORI=1TO22:PRINT"
":NEXT
280 PRINT"[<1HOME>] [<7CRSR D>]":PRINT"[<1
BLK>] [<1RVS>] PROVA ":PRINT" [<1CHR
$(207)>] [<6CHR$(183)>] [<1CHR$(208)>]":FO
RI=0TO4
290 PRINT" [<1CHR$(180)>] [<1CHR$(170)>]":NEXT:PRINT" [<1CHR$(204)>] [<6CHR
R$(175)>] [<1CHR$(186)>]":IFPEEK(187)=11T
HENGOSUB3100
300 YL=2:YH=23:XL=13:XH=36:Y=2:X=13:HH=2
2
310 GOSUB2000
320 IFX=13ANDY=23THENHK=1:GOSUB100:POKEV
+21,0:GOTO140:REM CAMBIO COL.
330 IFX=24ANDY=23THENHK=1:GOSUB100:GOTO4
00:REM FINE
340 IFX=36ANDY=23THENHK=1:GOSUB100:GOSUB
3000:GOTO300:REM FILL
350 GOSUB2500:IF(PEEK(PM)ANDBI)=BITHEN38
0
360 POKEP+54272,CO:POKEP,102
370 POKEPM,PEEK(PM)ORBI:GOTO390
380 POKEP,32:POKEPM,PEEK(PM)AND(255-BI):
REM CANCEL
390 GOTO310
400 REM*** FINE ****
405 PRINT"[<1CLR>] [<1CRSR D>] [<1BLU>]
CHE NOME VUOI DARE ALLO SPRITE"
410 INPUT"[<4CRSR D>] ":NM$
420 PRINTTAB(13)" [<3CRSR D>] [<1RED>] COS
A NE VUOI FARE ?"
430 PRINTTAB(13)" [<1CRSR D>] [<1VEL>] 1-
REGISTRARLO"
440 PRINTTAB(13)" [<1CRSR D>] [<1MAR>] 2-
STAMPARLO "
450 PRINTTAB(13)" [<1CRSR D>] [<1VEL>] 3-
LISTARLO SU VIDEO"
460 PRINTTAB(13)" [<1CRSR D>] [<1MAR>] 4-
CORREGGERLO"
465 PRINTTAB(13)" [<1CRSR D>] [<1MAR>] 5-
FINE "
470 GETA$:IFA$=""THEN470
480 IFA$="1"THEN550
490 IFA$="2"THEN600
500 IFA$="3"THEN800
510 IFA$="4"THENPOKE187,11:RUN20
515 IFA$="5"THENEND
520 GOTO470
550 REM *** REGISTRAZIONE SU NASTRO ***
560 PRINT"[<1CLR>] [<4CRSR D>] [<1RED>] A
TTENDERE PREGO !":OPEN1,1,1,NM$
570 FORI=SLTOSL+63:PRINT#1,PEEK(I):NEXTI
:CLOSE1
590 GOTO900
600 REM *** STAMPA SU CARTA ***
610 OPEN5,4
615 FORW=0TO70:PRINT#5,CHR$(175):NEXT:P

```

Seguito listato Sprite drawer.

```

RINT#5,CHR$(10)CHR$(10)CHR$(10)CHR$(10)
620 PRINT#5,CHR$(14)" [<1RVS>] "NM$ "C
HR$(15)CHR$(10)CHR$(10)CHR$(10)CHR$(10):
RI$=""
630 FORY=2TO22:FORX=13TO36:GOSUB2500:IF(
PEEK(PM)ANDBI)=BITHENRI$=RI$+" [<1RVS>] [
<1RVS OFF>]":NEXTX:GOTO640
635 RI$=RI$+" ":NEXTX
640 PRINT#5,RI$CHR$(8):RI$="" :PRINT#5,CH
R$(15):NEXTY
650 PRINT#5,CHR$(10)CHR$(10)CHR$(10)CHR$(
10)
660 PRINT#5,TAB(9)" [<1RVS>] DATI DA INS
ERIRE NEL PROGRAMMA: "CHR$(10)CHR$(10)CH
R$(10)
680 FORY=SLTOSL+63STEP12:ST$="" :FORX=0TO
11
690 ST$=ST$+RIGHT$(" "+STR$(PEEK(Y+X)
)+ " ",5):NEXTX
700 IFY=X>SL+63THENST$=LEFT$(ST$,21)
710 PRINT#5,ST$:NEXTY
720 PRINT#5,CHR$(10)CHR$(10):FORW=0TO70:
PRINT#5,CHR$(175):NEXT:PRINT#5,CHR$(10)
740 CLOSE5
760 GOTO900
800 REM *** LISTA SU VIDEO ***
810 PRINT"[<1CLR>]"
820 FORY=0TO20:N$=RIGHT$(" "+STR$(Y)+"
",4):PRINT" [<1RVS>] "TAB(11)N$ [<1RVS
OFF>] "TAB(7)" - ";
830 F$="" :FORX=0TO2:F$=F$+RIGHT$(" "+
STR$(PEEK(SL+X+Y*3))+ " ",5)
840 NEXTX:PRINTF$:NEXTY
900 PRINT"[<1HOME>] [<24CRSR D>] PER TORN
ARE AL MENU' [<1RVS>] F 1 [<1RVS OFF>] [<1
CRSR U>]"
910 GETA$:IFA$=""THEN910
920 IFA$<>CHR$(133)THEN910
930 PRINT"[<1CLR>] [<5CRSR D>]":GOTO420
2000 REM*** POSIZIONAMENTO CURSORE ***
2010 POKE2,Y:POKE182,X
2020 GOSUB2500:SYS(679):PRINT"[<1RVS>] [
<1RVS OFF>]":FORW=1TO40:NEXT
2030 IFHH<>22THEN2050
2040 IF(PEEK(PM)ANDBI)=BITHENSYS(679):PR
INTMID$(C$,CO+1,1)CHR$(166):FORW=1TO10:N
EXT:GOTO2060
2050 SYS(679):PRINT" ":FORW=1TO10:NEXT
2060 GOSUB2100:IFA=127THEN2020
2070 IFS(4)=0THEN2010
2080 RETURN
2100 REM*** LETTURA JOYSTICK ***
2110 A=PEEK(56320):FORI=0TO4:S(I)=-((AAN
D2^I)=0):NEXT
2120 IFS(0)=1THENY=Y-1:IFYL>YTHENY=Y+1
2130 IFS(1)=1THENY=Y+1:IFYH<YTHENY=Y-1
2135 IFHH=1THENRETURN
2140 IFS(2)=1THENX=X-1:IFXL>XTHENX=X+1
2150 IFS(3)=1THENX=X+1:IFXH<XTHENX=X-1
2200 RETURN
2500 P=1024+X+Y*40:PX=X-13:PY=Y-2:W=INT(
PX/8):PM=SL+PY*3+W:BI=2↑(7-PX+W*8)
2510 RETURN
3000 REM*** FILL ***

```




Sprite per C 64

Seguito listato Sprite drawer.

```

3010 FOR Y=2 TO 22: B=0: NM=1: FOR X=13 TO 36
3020 GOSUB 2500
3030 IF (PEEK(PM) AND BI) = B THEN NM = NM * -1: B =
B + 1: IF B > 2 THEN NM = 1: GOTO 3050
3040 IF NM = -1 THEN POKE PM, PEEK(PM) OR BI: POKE
P + 54272, CO: POKE P, 102
3050 NEXT X, Y: RETURN
3100 FOR Y=2 TO 22: FOR X=13 TO 36: GOSUB 2500
3110 IF (PEEK(PM) AND BI) = B THEN POKE P + 54272
, CO: POKE P, 102
3120 NEXT X, Y: RETURN
4000 PRINT "[<1CLR>][<4CRSR D>]": POKE 5328
0, 2: POKE 53281, 2
4010 PRINT "[<1WHT>] SPRITE DRAWER"
4020 PRINT "[<1WHT>][<1CRSR D>] DI F. STEL
LA (DIC83)"
4030 PRINT "[<8CRSR D>][<1YEL>] INSERISCI
IL JOYSTICK": PRINT "[<1CRSR D>][<1YEL>]
NELLA PORTA #2"
4040 PRINT "[<3CRSR D>] PER COMINCIARE [<
1RVS>] F1 [<1RVS OFF>]"
4050 GETS#: IFS# = "" THEN 4050
4060 RETURN

```

CARATTERI PROGRAM.

```

10 PRINT CHR$(142): POKE 52, 48: POKE 56, 48: CL
R
20 POKE 56334, PEEK(56334) AND 254: POKE I, PEE
K(I) AND 251
30 FOR I=0 TO 511: POKE I + 12288, PEEK(I + 53248)
: NEXT
40 POKE(I), PEEK(I) OR 4: POKE 56334, PEEK(563
34) OR 1
50 POKE 53272, (PEEK(53272) AND 240) + 12
60 FOR I=0 TO 71: READ A: POKE 12288 + I, A: NEXT
70 PRINT "[<1CLR>][<3CRSR D>][<1G.3>]" TAB
(17) "@ABCDEI[<8CRSR D>]": PRINT TAB(9) "[<1W
HT>]F", "G", "H[<6CRSR D>]": END
100 DATA 12, 0, 114, 140, 140, 140, 114, 0: REM A
110 DATA 0, 0, 62, 86, 68, 68, 56, 0: REM S
120 DATA 0, 2, 124, 144, 16, 16, 8, 0: REM T
130 DATA 12, 0, 124, 128, 112, 128, 124, 0: REM E
140 DATA 0, 0, 60, 66, 66, 124, 64, 64: REM R
150 DATA 0, 0, 32, 32, 32, 32, 24, 0: REM I
160 DATA 0, 129, 153, 231, 231, 153, 129, 0
170 DATA 68, 124, 56, 56, 56, 56, 56, 16
180 DATA 130, 214, 124, 124, 56, 16, 16, 0

```

Listato 3. Prova caratteri programmabili.

Scrive, suona, gioca, entusiasma

Gaetano Marano

66 PROGRAMMI PER ZX81

E ZX80 CON NUOVA ROM + HARDWARE

Per le sue qualità e il suo modestissimo prezzo lo ZX 81 della Sinclair è il computer più venduto nel mondo. Oggi, sempre con una modestissima spesa, si può imparare a sfruttare questo eccezionale strumento al limite delle sue capacità. Basta scorrere questo libro per scoprire quante cose lo ZX 81 può fare con l'aggiunta di alcuni semplici ed economici componenti. Ad esempio, tramite un semplice circuito musicale può riprodurre 50 note su 4 ottave e, sempre grazie a una modifica hardware da poche migliaia di lire, lo ZX 81 diventa anche l'unico computer in grado di conferire effetti sonori ai giochi inseriti tra i suoi programmi. Ma non è tutto. Un'altra novità di quest'opera, preziosa anche per chi possiede lo ZX 80 con ROM, è il regalo di alcune tastiere disegnate da sovrapporre a quella sensitiva dell'apparecchio, per ricavarne altre, speciali funzioni.

136 pagine. Lire 12.000 Codice 520 D

Per ordinare il volume
utilizzare l'apposito tagliando
inserito in fondo alla rivista



GRUPPO
EDITORIALE
JACKSON





I SEGRETI DEI PERSONAL

COMMODORE VIC 20 E C 64

La gestione dei dati digitati sulla tastiera del VIC e C 64

di Alessandro Guida

Lo scorso mese si è visto che sia il VIC che il C 64 sfruttano un'interruzione del loro lavoro, provocata dalla linea di interrupt, per controllare se è stato premuto qualche tasto. È stato anche spiegato l'uso del buffer di tastiera per la memorizzazione temporanea del codice ASCII dei tasti interessati nel buffer di tastiera.

Questo buffer, posto nelle locazioni da \$0277 a \$0280, costituisce, quindi, un vero e proprio deposito dove vengono lasciati i caratteri della routine di gestione della tastiera, e dove vengono prelevati in un secondo momento dalle altre routine dell'interprete BASIC.

In pratica, questo buffer è utilizzato diversamente se il computer è in modo diretto o sta eseguendo un programma.

Nel secondo caso i caratteri presenti possono essere letti dall'istruzione GET. Bisogna osservare che in ogni caso il buffer è gestito su base FIFO (First In-First Out), ossia il primo carattere che vi era stato memorizzato sarà il primo a venire letto e tolto dal buffer stesso. Tutti i caratteri seguenti vengono spostati in una posizione e viene aggiornato nella locazione \$C6, il numero di caratteri rimasti.

Quando, invece, il computer si trova in modo diretto (lo stato in cui si possono digitare comandi o linee BASIC direttamente dalla tastiera) ad accedere al buffer è la routine di gestione schermo.

Questa routine preleva i caratteri presenti nel buffer e li deposita sullo schermo, o esegue le operazioni connesse se si tratta dei tasti di movimento cursore o di HOME-CLEAR.

Perciò *lo schermo è il vero deposito* in cui sono conservati i comandi da eseguire o le linee BASIC da memorizzare.

Il computer resta fermo alla routine di gestione schermo, finché non viene premuto il tasto RETURN. A questo punto viene letta la linea dello schermo sulla quale si trova il cursore e ricopiata nel BASIC INPUT BUFFER (\$0200, \$0258).

Qui l'interprete BASIC la analizza e, se la linea comincia con un numero la memorizza come linea di programma, altrimenti la esegue segnalando eventualmente gli errori di sintassi.

Per curiosità, va notato che una linea logica può essere lunga più di una linea sullo schermo. Per que-

sto motivo esiste una tavola in memoria, da \$D9 a \$F0, nella quale è conservato un numero per ogni linea. Un 158 significa che la linea logica termina con la linea dello schermo, altrimenti il valore 30 vuol dire che continua sulla linea seguente.

Tutto ciò spiega come mai se sullo schermo, ad esempio, è già presente un comando, per eseguirlo nuovamente basta spostare il cursore sopra di esso e battere il return.

A questo punto, risulta ovvia una considerazione: è possibile simulare la digitazione di comandi da programma, semplicemente mettendo i caratteri necessari nel buffer di tastiera e memorizzando in \$C6 il numero di caratteri immessi nel buffer. Il programma I illustra quanto appena detto con un esempio.

Quando l'esecuzione del programma si interrompe, il buffer viene svuotato sullo schermo e il return (codice ASCII 13) comunica all'interprete che può leggere la linea dallo schermo. Possiamo anche simulare la digitazione di intere linee BASIC. Poiché, però, queste sono in genere più lunghe di 10 caratteri, verranno stampate sullo schermo e poi nel buffer si introdurranno dei codici di spostamento cursore più il return. In questa maniera terminato il programma i caratteri nel buffer obbligheranno il cursore a spostarsi sulla linea stampata in precedenza, e il return la farà entrare in memoria.

Un'ulteriore accortezza è quella di stampare la linea nello stesso colore dello sfondo.

Questo tipo di simulazione richiede solo un po' di fantasia da parte vostra, poiché apre enormi possibilità.

Il programma 2 è una delle possibili applicazioni. Esso permette di trasformare il contenuto di una zona di memoria in frasi data, ed è utilissimo quando si scrivono delle routine in linguaggio macchina e le si vuole incorporare in un programma BASIC.

Ma torniamo un po' indietro e vediamo cosa accade esattamente quando il computer si trova in modo diretto.

Terminata l'esecuzione di un programma, o dopo l'accensione, il computer passa all'esecuzione di una routine il cui indirizzo di partenza è conservato nel vettore \$0302, \$0303.

Questa è la routine di HANDLE NEW BASIC LINE OR COMMAND (Routine di gestione linee BASIC o comandi), il listato 3 ne contiene le prime istruzioni, che sono le stesse sia per il VIC che per il 64. L'unica differenza tra i due computer è che nel 64 tutti gli indirizzi iniziano con la A mentre nel VIC con la C.

Come si può vedere la prima istruzione è il salto alla subroutine in \$C560 che è proprio quella che, al ricevimento del carattere di RETURN, preleva la linea dallo schermo e la mette nel buffer BASIC,



La gestione dei dati digitali sulla tastiera del VIC e C 64

```

10 REM PROVA SUL FUNZIONAMENTO DEL BUFFER
20 REM DELLA TASTIERA.
30 REM
40 INPUT "TESTO (MAX 9 CAR.)":A$
50 A$=LEFT$(A$,9)
60 REM
70 REM INSERISCE I CARATTERI DEL TESTO NEL
80 REM BUFFER.
90 REM
100 LUNG=LEN(A$)
110 FOR I=1 TO LUNG
120 POKE630+I,ASC(MID$(A$,I,1))
130 NEXT I
140 REM
150 REM AGGIUNGE ANCHE UN RETURN
160 REM
170 POKE630+I,13
180 REM AGGIORNA IL CONTATORE DEI CAR.
190 REM PRESENTI NEL BUFFER.
200 POKE198,LUNG+1
210 REM
220 REM TERMINA IL PROGRAMMA E VIENE
230 REM RIVERSATO IL BUFFER SULLO
240 REM SCHERMO, SUBITO DOPO
250 REM IL MESSAGGIO "READY".

```

Listato 1. Simulazione di un comando dato da tastiera per mezzo di un programma.

come visto prima. Inoltre i registri X e Y conterranno l'indirizzo di partenza del buffer; i due registri vengono salvati in \$7A, \$7B.

Quindi viene chiamata la routine di CHARGET (\$0073) che ha il compito di prelevare il carattere puntato dalle due locazioni \$7A, \$7B. Questa routine fornisce il carattere letto nell'accumulatore A del microprocessore e contemporaneamente setta il flag C (carry) se non si tratta di un numero.

Infatti, viene controllato che tale carattere non sia uno zero (ciò indicherebbe la fine della linea) e, subito dopo, se si tratta di un numero. La presenza di un numero viene interpretata come l'inizio di una linea BASIC, altrimenti come un comando da eseguire in modo diretto. A seconda di un caso o dell'altro vengono eseguite operazioni diverse.

Ma fermiamoci a questa prima parte. Poiché, come abbiamo detto, l'indirizzo di questa routine risiede in un vettore, è possibile modificarlo.

L'idea è quella di interfacciare uno o più comandi dati in modo diretto. Possiamo infatti sostituire l'inizio della routine di Handle con una nostra che controlli se la linea digitata contiene uno dei nuovi comandi. In questo caso verranno eseguite le operazioni da noi previste, se no si ritornerà alla routine originale. Tutto ciò non comporta troppe difficoltà, a patto di lasciare inalterate le prime tre istruzioni che costituiscono il nucleo vitale della routine originale.

Il comando che vogliamo aggiungere è KEY, per completare il programma dei tasti di funzione. Vogliamo fare in modo che:

KEY1, LIST

associi al tasto F1 la stringa LIST e così via cambiando il numero che segue la parola KEY.

Vogliamo anche che la parola KEY da sola faccia apparire la lista dei tasti associati a tutti i tasti funzione.

Il listato 4 contiene il disassemblato parziale della routine che realizza ciò, ed è abbondantemente commentato. In esso è possibile vedere applicato quanto appena detto. Il programma 5, invece, è il modulo BASIC per caricarlo.

Ricordate di abilitare i tasti di funzione prima di far girare il programma 5 per attivare il comando KEY, utilizzando il programma visto il mese scorso.

Naturalmente, i più volenterosi che abbiano seguito fin qui questo articolo, potranno fondere insieme le due routine.

Sperando di avervi dato delle idee utili vi auguro buon lavoro.

```

2 INPUT "INIZIO LM":S
4 HBX=S/256:POKE829,HBX:POKE828,S-HBX*256
10 INPUT "FINE LM":E
15 HBX=E/256:POKE831,HBX:POKE830,E-HBX*256
16 PRINT "OK";
17 POKE833,234:POKE832,96
20 NL=PEEK(833)*256+PEEK(832):N$="000"
22 S=PEEK(828)+PEEK(829)*256:E=PEEK(830)+PEEK(831)*256
30 PRINT "OK"NL"DATA";
40 NL=NL+10:HBX=NL/256
45 POKE833,HBX:POKE832,NL-HBX*256
60 FOR NB=0 TO 9:BY=PEEK(S+NB)
70 BY$=RIGHT$(N$+RIGHT$(STR$(BY),LEN(STR$(BY))-1),3)
75 IF NB<>9 THEN BY$=BY$+" "
80 PRINT BY$;
90 NEXT NB
100 IF S+10>E THEN 200:REM FINE
110 PRINT:PRINT "RUN20"
120 S=S+10:HBX=S/256
130 POKE829,HBX:POKE828,S-HBX*256
140 POKE831,19:POKE832,13:POKE833,13:POKE198,3
150 END
200 POKE831,19:POKE832,13:POKE198,2
210 END

```

Listato 2. Programma che fa largo uso delle caratteristiche del buffer di tastiera. Riporta il contenuto di una zona di memoria in linee BASIC sotto forma di frasi DATA.

```

;ROUT. HANDLE NEW BASIC LINE ORIGINALE
C483 20 60 C5 JSR $C560 ;Prende la linea dallo schermo e la
;memorizza nel buffer Basic
C486 86 7A STX $7A ;Aggiorna i puntatori all'inizio
C488 84 7B STY $7B ;del buffer basic
C48A 20 73 00 JSR $0073 ;Routine CHARGET, legge il primo car.
C48D AA TAX ;
C48E F0 F0 BEQ $C48D ;Se = 0 ritorna ad attendere un'altra
;sullo schermo.
C490 A2 FF LDX $FF ;
C492 86 3A STX $3A ;
C494 90 06 BCC $C49C ;Se il carattere e' un numero (C=0)
;salta a $C49C, altrimenti continua.

```

Listato 3. Disassemblato delle prime istruzioni della routine di Handle New BASIC Line originale, che ci apprestiamo a modificare.