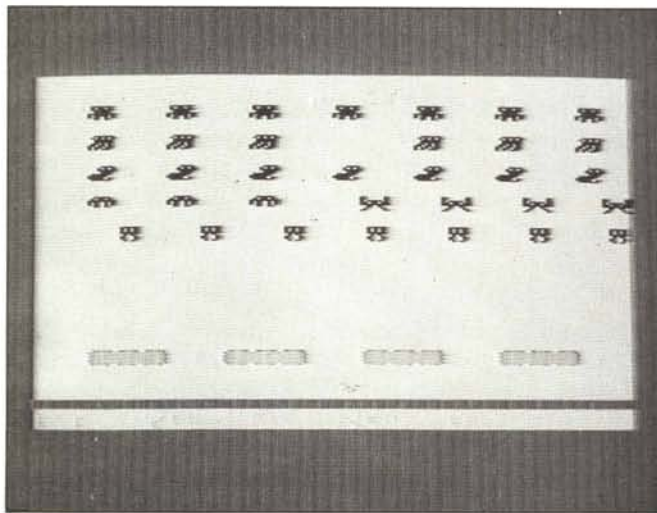


Space invaders

di Mauro Da Lio - Spinea (VE)

Se già avevamo pubblicato giochi allo stesso livello di quelli commerciali (valga per tutti l'ormai storico VIC-MAZE), questo INVADERS è senz'altro superiore a molti di quelli. Lo dichiariamo apertamente: si tratta di uno dei migliori giochi che ci sia capitato di avere per il VIC, superiore a quelli di moltissime case, anche famose, indipendentemente dal supporto (cassetta, disco o cartuccia) e dal prezzo. Diremo di più: manca un pizzico di velocità in più per renderlo pari a quelli delle sale giochi (ma allora doveva esser tutto in LM, mentre è scritto largamente in BASIC), rispetto al quale ha però il vantaggio di presentare nientepopodimeno che 18 varianti di gioco — non so se mi spiego: due opzioni per nove livelli l'una — che lo rendono forse superiore. La parola all'artefice, autore anche di un buon articolo.



Il programma che presentiamo consente di giocare anche in notturna ...

Elenco cassette per il VIC 20

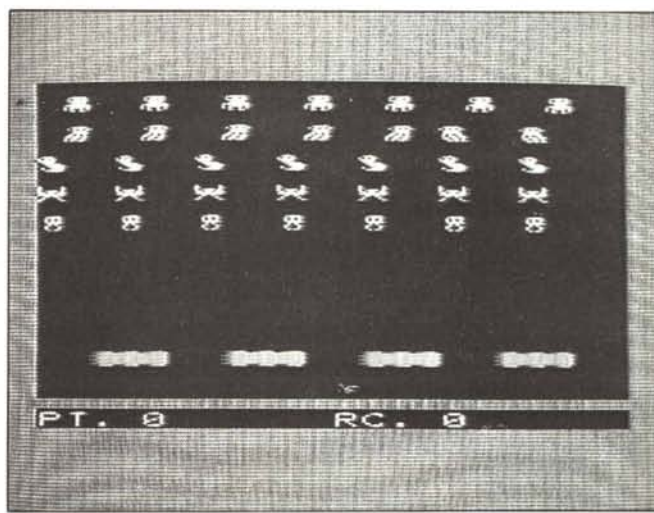
Presso la redazione sono disponibili le cassette relative ad alcuni dei programmi pubblicati nella rubrica di software per il VIC 20. Per l'ordinazione inviare l'importo (a mezzo assegno, c/c o vaglia postale) a Technimedia srl, Via Valsolda 135, 00141 Roma.

codice	programma	MC n.	config.	lire
CVC/01	VIC-Maze	19	base	17000
CVC/02	Pic-man	23	base	17000
CVC/03	Briscola	25	base	17000
CVC/04	Grand Prix	28	base	17000
CVC/05	Frogger	26	6K	17000
CVC/06	Invaders	29	16K	23000
DVC/01	EXMA	27/28	16K	15000

zione del programma principale e delle subroutine.

Vediamo di giustificare questi due fatti: se qualcuno di voi ha mai provato a scrive-

re un programma di gioco si sarà quasi immediatamente reso conto che il problema più grosso, almeno restando in ambiente Basic, è senza ombra di dubbio quel-



Descrizione del programma

Il programma Space invaders è una versione per VIC 20 del celeberrimo gioco. Il programma è lungo circa 10K ma richiede per funzionare la presenza di uno spazio di 2.5K in testa al programma stesso. Per questo motivo gira solo con l'espansione da 16K anche se è senz'altro possibile ottenerne una versione adatta agli 8K. Il modo per ricavare il suddetto spazio di 2.5K in testa al Basic (subito dopo la video Ram) è spiegato nelle istruzioni per il gioco ed in questo momento non ci torneremo sopra.

Un'occhiata al listato fa subito capire che si tratta di un programma anomalo almeno per un paio di motivi: innanzi tutto si osserva che forse il 70% del programma è fatto da istruzioni DATA, ed in secondo luogo si osserva una non naturale dispo-

```

1 GOTO900
2 IFUSR(A(A)+K6)=1THENPRINT""
4 RETURN
6 IFUSR(A(A)+K6)=1THENPRINT"_"
8 RETURN
10 IFUSR(A(A)+K6)=1THENPRINT"|"
12 RETURN
14 IFUSR(A(A)+K6)=1THENPRINT"#"
16 RETURN
18 IFBM=0THENBM=A(A):RETURN
20 RETURN
30 IFJTHEN42
31 IFUSR(BM)=8THENPRINT" ";
32 BM=BM+K6:ONUSR(BM)GOTO34,4,36,4,4,34,40,4,35
34 PRINT"■ ";:RETURN
35 BM=0:RETURN
36 FG=1:RETURN
40 J=14:SYS$PRINT"■P"::RETURN
42 J=J-1:U=USR(BM):PRINTE$(J)::IFJTHENRETURN
44 BM=0:RETURN
48 IFC=0THEN60
50 IFUSR(C)=4THENPRINT" ";
51 C=C-K6:IFC<0THENC=0:SYSIR:RETURN
52 ONUSR(C)GOTO53,56,4,4,4,4,54,55,4
53 PRINTC#:RETURN
    
```

Istruzioni per il caricamento del gioco:

- 1) Spegner il computer
- 2) Inserire l'espansione da 16K e collegare un Joystick
- 3) Riaccendere il computer ed attendere il messaggio:
.... CBM BASIC V2
19967 Bytes Free
Ready.
- 4) Digitare POKE 642,28:SYS58232 (return)

A questo punto deve comparire sul video la scritta:

```
.... CBM BASIC V2 ....
17407 Bytes Free
Ready.
```

Il presente comando serve a spostare l'inizio della memoria RAM destinata al Basic alla pagina 28. (Se si trovano delle difficoltà al presente punto 4 si può eseguire in alternativa il punto 4' come indicato alla fine di queste note). Si riservano in questo modo circa 2500 byte in testa (e non in coda) al programma Basic. Questi 2500 byte sono utilizzati dal programma per inserirvi la mappa dei caratteri speciali ridefiniti e tutte le routine in L.M. utilizzate dal programma.

5) Caricare il programma con LOAD. Data la lunghezza del programma il tempo richiesto per questa operazione è di circa 3 minuti.

6) Dare il comando RUN. A questo punto dovrebbe comparire a caratteri cubitali la scritta:

```
SPACE
INVADERS
```

7) Dopo un certo tempo necessario al programma per definire tutti i caratteri grafici e le routine in L.M. compare la scritta:

Difficoltà (1-9)?

Bisogna selezionare il grado di difficoltà con cui si intende iniziare il gioco.

Fatto questo compare la domanda:

Alieni Visibili?

Se si risponde di Sì (S) ci si ritroverà

nella prima variante del gioco a combattere contro un battaglione di alieni verdi in un cielo nero e quindi perfettamente visibili. Se si risponde di No (tasto N) si dovrà combattere di notte contro degli alieni neri in un cielo nero. Gli alieni risulteranno visibili solo quando i bagliori delle esplosioni squarceranno il cielo. (I loro colpi invece continueranno ad essere visibili).

Premuto quindi il tasto S od N a seconda della variante di gioco desiderata, vedremo comparire una tabella dei punti che possiamo guadagnare annientando i vari tipi di alieni. (Ovviamente se abbiamo selezionato il gioco notturno saranno visibili solo i punti e non gli alieni!).

Si faccia attenzione che nel caso in cui si colpisca e si distrugga una delle proprie barriere difensive si perdono 55 punti. Se invece si riesce a colpire al volo la grossa bomba gialla che di tanto in tanto gli invasori lasciano cadere si guadagna la bellezza di 256 punti.

Quest'ultima manovra però è alquanto pericolosa perché non è detto che la bomba anche se colpita esploda. Vi è quindi un serio pericolo di vedere il nostro cannoncino andare in fumo.

Quando saremo stanchi di ammirare la tabella dei punti e al tempo stesso pronti al combattimento premiamo il tasto rosso del FUOCO per incominciare.

Non mi dilungo a questo punto sulla strategia del gioco e sui piccoli trucchi che giocatori incalliti come me possono scoprire da soli.

Il mio record personale in qualità del "Creativo" del gioco si aggira sui 17000 punti.

Faccio invece ancora notare che si dispone solo di un cannoncino laser e non di 3 come nelle macchinette da bar (non mi pareva che avesse senso mettere 3 cannoncini in un gioco in cui non si devono inserire le 100 lire per giocare!).

Sotto lo schermo, in basso a sinistra, si vede indicato con PT, il punteggio che stiamo realizzando, mentre a destra, indicato con R.C., il record del giorno.

Il gioco termina quando il nostro cannoncino laser viene colpito dagli alieni o quando anche uno solo di essi riesce ad invaderci.

Alla fine vedremo comparire dei commenti sul video e la domanda:

VUOI GIOCARE ANCORA?

Se si risponde Sì (tasto S) il gioco riprende ed il record non viene cancellato.

Un'ultima osservazione riguarda il modo non standard di tenere il Joystick.

Tenete il Joystick in modo che la posizione marcata con TOP risulti alla vostra destra. In questo modo il Joystick stesso sarà più stabile e il pulsante di fuoco sarà più facilmente azionabile con la mano destra.

Con il Joystick così orientato per spostare il cannoncino a destra sarà sufficiente tirare a destra la leva (in realtà top), per andare a sinistra bisogna invece spostare la leva a sinistra (in realtà down). Per sparare si usa il tasto rosso (fire). E possibile sparare solo quando il laser è carico (cannoncino azzurro). Buon divertimento!

Nota al punto 4) delle istruzioni per il gioco

L'operazione indicata serve a spostare l'inizio del Basic (che normalmente con l'espansione da 16K si trova subito dopo la video RAM alla pagina 18 cioè all'indirizzo $18 \times 256 = 4608$) in modo da farlo cadere in pagina 28, cioè all'indirizzo $28 \times 256 = 7168$.

In questo modo è possibile ricavare 2.5 K nei soli 4K indirizzabili dal chip controllore del video.

Un modo alternativo a quello indicato nel punto 4) per ottenere sempre lo stesso risultato di spostare l'inizio del Basic a 7168 è il seguente:

4') Digitare: POKE 642,28 : POKE 44,28 (return)

POKE 7168,0 : NEW (return)

In ogni caso il programma segnala un errore se l'inizio del Basic non è nella posizione esatta.

```
54 PT=PT-55:GOTO57
55 PT=PT+K6:C=0:SYSIR:J=14:SYSES:PRINT"R#":RETURN
56 PT=PT+IT(PEEK(K5)-04)
57 SYSIR:IFITHENU=USR(E1):PRINT" ";
58 E1=C:C=0:I=14:SYSES:RETURN
60 IFPEEK(K1)THEN66
62 POKEV.68:SYSIO:C#CC*(PEEK(K2)):C=K0+PEEK(K3):RETURN
66 U=USR(PU):PRINT"R#PT":RETURN
70 IFUSR(A(A))<>2THENF(M)=F(A):A=M:VA=VA-GD:NA=NA-1:RETURN
72 PRINT"R#A(A)=A(A)+D:IFUSR(A(A))=6THENIFE1=A(A)THENI=0
74 PRINTA#(A,B):ONRND(0)*VAGOTO18,6,10,14,2
76 RETURN
100 B=1-B:SYSTX:IFPEEK(K4)THENF=3ANDF+1:D=D*(F)
110 GOSUB70:IFNA=0THEN192
120 SYSPL:PRINTL#(PEEK(K2)):GOSUB48:SYSMC:IFPEEK(K5)THEN800
138 IFBMTHEGOSUB30:IFFGTHEN800
150 SYSEX:SYSIV:IFPEEK(K5)THEN700
160 SYSPL:PRINTL#(PEEK(K2)):GOSUB48:SYSMC:IFPEEK(K5)THEN800
170 IFITHENI=I-1:U=USR(E1):PRINTE#(I);
190 M=A:A=F(A):IFA#MTHEN110
191 GOTO100
192 GG=GG+1:IFGG>9THENG=9
194 GD=GD(GG):VA=VA(GG):GOSUB200
196 M=34:NA=35:F=3:A=0:B=0:GOTO100
200 FORB=0TO4:FORA=0TO5:K=A+B*7:R(K)=B*02+A*3+K6*GG:U=USR(R(K))
```

(continua a pag. 114)

lo della velocità di esecuzione. Un gioco lento non vale niente o quasi. Per accelerarlo sono possibili diverse strade: si può ad esempio compilare il programma oppure non usare il Basic ma un linguaggio molto più veloce come ad esempio il FORTH, o ancora scrivere il programma completamente in linguaggio macchina, oppure infine adoperare una soluzione mista costituita da un programma Basic che chiama un adeguato numero di subroutine in linguaggio macchina.

La prima soluzione non è certamente praticabile col VIC, visto che non esiste in commercio il compilatore Basic. La seconda soluzione è secondo me la migliore.

Il presente programma tuttavia l'ho scritto quando ancora non conoscevo il FORTH. Volendo evitare la fatica di scri-

vere un programma completamente in linguaggio macchina non mi restava che l'ultima soluzione: realizzare un programma Basic che nella fase di gioco vera e propria riuscisse a raggiungere la velocità desiderata grazie all'aiuto di velocissime subroutine in L.M.

Sempre con lo stesso motivo si spiega la disposizione delle subroutine in Basic: quelle relative alla fase di gioco sono state tutte poste nella parte iniziale del programma perché li sono trovate più rapidamente in seguito ad un comando del tipo GO-SUB. Per lo stesso motivo si è preferito usare le variabili anche per contenere dei valori costanti perché in Basic viene eseguita più rapidamente una istruzione del tipo $C = A + B$ piuttosto che $C = 102 + 34$. Vediamo quindi a grandi linee che cosa fa il programma: subito dopo il RUN il programma salta alla linea 900 dove controlla che ci sia il famoso spazio di 2.5K in testa al Basic necessario per inserire i codici L.M. e la mappa dei caratteri ridefiniti. In seguito, dalla linea 1000 in poi, il programma esegue le seguenti operazioni: salto alla subroutine 7000 che visualizza l'intestazione, quindi lettura dei DATA che vengono messi al loro posto a formare le suddette routine. Alcuni DATA servono invece a costruire la mappa dei caratteri che nel presente programma è posta all'indirizzo 6144. Ci si può divertire a fermare il programma mentre si sta giocando (tasto R/S senza premere RESTORE) e quindi a stampare i vari caratteri sul video. Si potrà così osservare che i caratteri alfabetici non sono stati alterati mentre quelli grafici si (una cinquantina), per formare le figurine degli alieni e di tutto quello che compare sul video durante il gioco. Una modifica che si può fare agevolmente consiste, se lo si desidera, nell'alterare questi dati in modo da ottenere le figure di alieni o altro che più ci aggradano. Ogni blocco di DATA è accompagnato dal suo bravo ciclo FOR ... NEXT per la lettura ed è preceduto da una istruzione REM che suggerisce qualche cosa sul significato stesso dei dati. Nel caso i dati rappresentino delle routine in L.M., viene anche definita una variabile che provvede a contenere l'indirizzo di partenza di quella subroutine.

Vediamo in breve che cosa fanno le principali subroutine in L.M.:

SYSPL — chiama l'esecuzione di una routine che provvede a leggere il joystick. La subroutine provvede anche a muovere il cursore nella posizione in cui deve essere disegnata la forma del cannoncino. In questo modo è sufficiente eseguire di seguito il comando PRINT LS (qualche cosa) per avere immediatamente stampato il cannoncino al posto giusto.

SYSMC — provvede a muovere verso il basso tutti i colpi alieni visibili nello schermo. La subroutine ha ovviamente la necessità di comunicare al Basic se il cannone sia stato colpito. Questo risultato si ottiene scrivendo un flag in una opportuna locazione di memoria che viene successivamente letta dal Basic con l'istruzione PEEK.

(segue da pag. 113)

```

210 PRINT#(K,0):NEXT:FORK=0T033:F(K)=K+1:NEXT:F(34)=0:RETURN
700 SYSIR:U=USR(6*K6):PRINT"GLI ALIENI HANNO VINTO"
710 FORK=0T0200:NEXT
800 FG=PEEK(K3)+5120:SYSIR:U=USR(FU):PRINT"PT"
810 SYSES:FORI=0T014
820 U=USR(FG):PRINTEX(I):SYSEX:FORJ=0T010
830 POKE664,12+RND(0)*(14-I)/8:POKE3685,38+RND(0)*(14-I)/4:NEXT:NEXT:IFFT<=HI
THEN850
840 HI=PT:U=USR(8*K6):PRINT"MI HAI FATTO IL RECORD!"
850 FT=0:U=USR(10*K6):PRINT"MI VUOI GIOCARE ANCORA?"
860 GETG:IFG#="S"THENGOSUB7000:GOTO5000
865 IFG#="N"THENEND
870 GOTO860
900 IFPEEK(44)=28THEN1000
905 PRINT"MBATTERE!"
910 PRINT"POKE642,28:SYS58232"
920 PRINT"RILEGGERE IL PROGRAMMA":END
1000 A=0:U=0:GOSUB7000
1010 DIMF(34),R(34),P(34,1),L(3),CC(3),E(14),EX(14),D(3),IT(9),GD(9),VA(9)
1200 REM---DIFFICOLTA---
1210 FORK=1T09:READGD(K),VA(K):NEXT
1230 DATA7,252,6,214,5,178,4,144,3,110,2,74,1,39,.5,22,.25,13,5
1400 REM---ALIENI-----
1410 FORB=0T04:FORA=0T06:K=A+B*7
1420 A#(K,0)=CHR(30)+CHR(192+2*B)
1430 A#(K,1)=CHR(30)+CHR(193+2*B)
1440 NEXT:NEXT
1500 REM---PUNTI-----
1510 FORK=0T09:READDT(K):NEXT
1520 DATA100,100,70,70,60,60,40,40,25,25
1530 REM---LASER-----
1540 FORK=0T03:READA,B,C:L#(K)=CHR(A)+CHR(B)+CHR(C):NEXT
1560 DATA106,107,32,108,109,32,32,112,113,32,110,111
1570 REM---COLPI LASER---
1580 FORK=0T03:READC:CC#(K)=CHR(C):NEXT
1590 DATA231,232,233,234
1600 REM---ESPLOSIONE---
1610 FORK=0T014:READA,B:E#(K)=CHR(A)+CHR(B):NEXT
1620 DATA5,32,158,248,5,247,28,246,156,245
1630 DATA5,244,158,243,156,242,5,241,158,240
1640 DATA5,239,156,238,5,237,158,236,156,235
1650 REM---DIREZIONI---
1660 FORK=0T03:READD#(K):NEXT
1670 DATA1,256,-1,256
2000 REM---USR-----
2005 FORK=5120T05145:READC:POKEK,C:NEXT:POKE0,76:POKE1,0:POKE2,20
2010 DATA32,170,209,166,100,164,101,24,32,240,255
2040 DATA177,209,168,133,255,185,0,18,168,169,0,32,145,211,96
2044 REM---SYSPL-----
2046 FORK=5186T05264:READC:POKEK,C:NEXT:PL=5186
2050 DATA166,252,164,253,173,31,145,205,31,145,208,248,133,87
2060 DATA41,50,74,74,74,74,74,133,251,169,4,36,87,208,16,232
2070 DATA224,4,144,11,162,0,200,192,20,144,4,160,19,162,3,169
2080 DATA0,36,87,208,12,202,16,9,162,3,136,16,4,160,0,162
2090 DATA0,134,252,132,253,162,20,24,32,240,255,169,13,141,134,2,96
2094 REM---SYSTX-----
2096 FORK=5265T05329:READC:POKEK,C:NEXT:TX=5265
2100 DATA169,0,170,189,0,16,201,64,144,4,201,74,144,46
2105 DATA189,21,16,201,64,144,4,201,74,144,35
2110 DATA189,220,16,201,64,144,4,201,74,144,24
2115 DATA189,241,16,201,64,144,4,201,74,144,13
2118 DATA138,24,105,22,201,220
2120 DATA144,203,169,0,133,254,96,169,1,133,254,96
2125 REM---CODICI USR---
2128 FORK=4608T04735:READC:POKEK,C:NEXT
2130 DATA9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9
2140 DATA9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9
2150 DATA1,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9
2160 DATA9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9
2170 DATA2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3
2180 DATA3,3,7,7,7,7,7,7,7,7,7,7,7,7,7,7
2190 DATA1,8,1,5,5,5,5,4,4,4,4,6,6,6,6,6,6
2200 DATA6,6,6,6,6,6,6,6,6,6,9,9,9,9,9,9
2400 REM---MOV. COLPI-
2410 FORK=5330T05453:READC:POKEK,C:NEXT:MC=5330
2420 DATA169,0,133,255,169,17,133,88,169,206,133,87
2430 DATA198,87,165,87,201,255,144,9,198,88,169,15,197,88,144,1,96
2440 DATA160,0,177,87,168,170,185,0,18,201,5,208,226
2450 DATA160,0,169,32,145,87,160,22,177,87,168,185,0,18
2460 DATA201,9,208,3,76,222,20,201,2,208,3,76,222,20
2470 DATA201,7,208,12,160,22,169,1,24,113,87,145,87,76,222,20
2480 DATA201,3,208,11,152,185,182,18,61,165,18,5,255,133,255
2490 DATA160,22,138,145,87,165,87,133,89,165,88,24,105,132,133,90
2500 DATA169,7,145,89,76,222,20
2600 REM---COD. COLPI-
2610 FORK=4864T04875:READC:POKEK,C:NEXT
2620 DATA15,252,3,255,63,240,255,192,192,48,12,3
2700 REM---INVASION?---
2710 FORK=5454T05479:READC:POKEK,C:NEXT:IV=5454
2720 DATA169,0,133,255,162,21,189,184,17,168,185,0,18,201,2
2730 DATA240,4,202,16,242,96,169,1,133,255,96
2800 REM---ESPLOSIONE---
2810 FORK=0T014:READA,B,C:EX#(K)=E#(A)+E#(B)+E#(C):NEXT
2820 DATA0,14,0,14,12,13,13,11,12,11,10,12,10,0,9,9,9,8,7,8,7
2830 DATA5,7,6,4,6,4,3,4,3,2,3,4,2,2,1,1,2,1,0,1,1,0,0,0
2850 REM---SYSES-----
2855 FORK=5480T05490:READC:POKEK,C:NEXT:ES=5480
2860 DATA169,200,141,13,144,169,15,141,31,19,96
2900 REM---ESPLOSIONE---
2910 FORK=4895T04909:READC:POKEK,C:NEXT

```

```

2920 DATA0,4,5,5,4,5,5,6,7,8,7,9,11,13,15
2925 REM---SET IRQ---
2930 FORK=5520T05532:READC:POKEK,C:NEXT:I0=5520
2940 DATA120,169,157,141,20,3,169,21,141,21,3,88,96
3000 REM---CARATTERI---
3002 FORK=6144T06655:POKEK,PEEK(K+26624):NEXT
3008 FORK=6656T07119:READC:POKEK,C:NEXT
3010 DATA126,90,126,189,255,60,102,135
3020 DATA126,90,126,60,255,189,165,36
3030 DATA124,170,124,124,86,91,109,53
3040 DATA62,85,62,62,106,218,182,172
3050 DATA112,168,248,112,62,31,79,62
3060 DATA14,21,31,14,124,248,242,124
3070 DATA129,129,189,219,255,36,66,129
3080 DATA60,90,255,165,165,36,0,0
3090 DATA62,42,42,62,20,34,34,20
3100 DATA124,84,84,124,40,68,130,130
3110 DATA0,0,2,3,11,2,3,3
3120 DATA192,200,224,240,248,224,240,48
3130 DATA0,0,0,0,0,0,0,0
3140 DATA48,50,184,252,254,184,252,204
3150 DATA3,35,11,15,47,11,15,12
3160 DATA0,32,128,192,224,128,192,192
3170 DATA12,140,46,63,191,46,63,51
3180 DATA0,128,0,0,128,0,0,0
3190 DATA126,255,255,255,255,255,126
3200 DATA66,231,255,255,255,255,126
3210 DATA2,195,255,255,255,255,126
3220 DATA0,131,220,255,255,255,126
3230 DATA0,131,211,255,255,255,126
3240 DATA0,1,81,247,255,255,126
3250 DATA0,0,17,215,247,255,126
3260 DATA0,0,17,215,247,255,126
3270 DATA0,0,0,25,211,255,255,126
3280 DATA0,0,0,1,82,255,255,126
3290 DATA0,0,0,1,82,87,255,126
3300 DATA0,0,0,0,64,86,255,126
3310 DATA0,0,0,0,86,88,126
3320 DATA0,0,0,0,0,24,106
3330 DATA0,0,0,0,0,0,0
3340 DATA102,60,60,126,126,60,24
3350 DATA0,0,0,0,0,0,0
3360 DATA128,64,128,64,128,64,128,64
3370 DATA32,16,32,16,32,16,32,16
3380 DATA8,4,8,4,8,4,8,4
3390 DATA2,1,2,1,2,1,2,1
3400 DATA192,128,192,128,192,128,192,128
3410 DATA48,32,48,32,48,32,48,32
3420 DATA12,8,12,8,12,8,12,8
3430 DATA3,2,3,2,3,2,3,2
3440 DATA0,32,22,24,44,8,0,0
3450 DATA64,41,22,24,44,74,16,0
3460 DATA196,41,54,28,44,75,146,40
3470 DATA206,41,183,252,62,75,147,186
3480 DATA131,92,155,230,90,75,147,76
3490 DATA81,70,217,38,208,198,81,12
3500 DATA8,178,76,147,104,131,193,147
3510 DATA8,144,8,129,42,32,69,72
3520 DATA136,20,0,130,65,64,42,68
3530 DATA8,4,8,1,146,64,164,2
3540 DATA64,20,8,66,12,32,66,0
3550 DATA0,16,72,0,18,0,128,0
3560 DATA64,2,0,32,0,2,128,0
3570 DATA0,16,0,0,0,128,0,2
3580 DATA0,255,255,255,255,255,0,0
3900 REM---SUONO LASER---
3910 FORK=5533T05552:READC:POKEK,C:NEXT
3920 DATA174,47,19,202
3940 DATA208,2,162,10,142,47,19,189,47,19,141,12,144,76,191,234
3950 FORK=4911T04921:READC:POKEK,C:NEXT
3960 DATA1,227,224,221,218,215,212,209,206,203,200
3965 REM---RESET IRQ---
3970 FORK=5553T05575:READC:POKEK,C:NEXT:IR=5553
3980 DATA120,169,191,141,20,3,169,234,141,21,3,169,0,141,12,144,169,52,141,14,14
4,88,96
4870 REM---SYSEX-----
4875 FORK=5576T05617:READC:POKEK,C:NEXT:EX=5576
4880 DATA174,31,19,224,0,240,9,202,142,31,19,208,4,142,13,144
4890 DATA96,169,240,45,14,144,29,31,19,141,14,144
4900 DATA173,73,19,201,0,240,6,189,57,19,141,15,144,96
4910 REM---BAGLIORI---
4920 FORK=4922T04936:READC:POKEK,C:NEXT:FL=4937
4930 DATA14,14,14,14,14,110,62,238,206,238,222,190,238,14
4970 V=36878:V1=V-1:V2=V-2:MX=5373:K0=5121:O2=512:O4=64:PU=5635
4980 K1=251:K2=252:K3=253:K4=254:K5=255:K6=256
5000 GOSUB8000:POKEV+1,14:POKEV,52:POKEV-9,206:GOSUB9000
5020 PRINT"O":U=USR(5122)
5025 PRINT"O"
5030 PRINT"O"
5110 GOSUB200:POKEK2,0:POKEK3,10:POKEK4,0
5120 A=0:B=0:C=0:F0=0:I=0:J=0:BM=0:M=34:NA=35
5130 F=3:GOTO100
7000 POKE36879,110:POKE36869,134:PRINT"O"
7010 PRINT"O"
7020 PRINT"O"
7030 PRINT"O"
7040 PRINT"O"
7050 PRINT"O"
7060 PRINT"O"
7070 PRINT"O"
7080 PRINT"O"

```

(continua a pag. 116)

Infatti tutte le volte che viene eseguito SYSMC segue subito dopo IF PEEK...

SYSTX — guarda se gli alieni hanno raggiunto uno dei due bordi dello schermo nel qual caso bisogna che cambino direzione. Anche questa subroutine provvede a comunicare col Basic nel modo precedentemente visto.

SYSIV — guarda se per caso gli alieni hanno raggiunto l'ultima riga nel qual caso si è completata l'invasione e la partita è persa.

Una particolare subroutine è la:

USR (posizione) — questa subroutine assolve essenzialmente a due compiti: 1) muove il cursore alla posizione indicata nello schermo. La variabile "posizione" è così collegata alle coordinate di schermo: posizione = colonna + riga * 256/2) restituisce un codice al Basic. Il codice è legato al codice di schermo del carattere presente nella posizione indicata (si vedano nel listato i DATA preceduti dalla REM — CODICI USR —) ed è ad esempio 2 per gli alieni, 1 per lo spazio, 9 per un carattere alfabetico e così via.

In questo modo con la costruzione ON USR(P) GOSUB... si ottiene il risultato di posizionare il cursore nel posto voluto sullo schermo (in attesa della successiva PRINT) e, cosa ancora più importante, di indirizzare il programma verso la subroutine più opportuna a seconda di cosa fosse contenuto in quella posizione. Si evitano in questo modo interminabili confronti IF...THEN.

Faccio anche notare che disegnare i caratteri sullo schermo attraverso delle PRINT risulta notevolmente più veloce che non con delle POKE per tutta una serie di motivi, non ultimo quello che le POKE devono essere almeno 2: una alla mappa caratteri ed una alla mappa colori (ma ce ne sono degli altri come la necessità di calcolarsi gli indirizzi...).

Ultima curiosità che intendo soddisfare è quella riguardante il modo in cui è stato ottenuto il suono del laser: si tratta di una subroutine in linguaggio macchina che viene agganciata alla normale routine di interrupt (IRQ) che ha la frequenza di 60 Hz.

L'"agganciamento" viene ottenuto alterando il vettore di IRQ presente alle locazioni 788-789. Lo "sganciamento" (fine del suono) si ottiene ripristinando il vettore iniziale. Sono due subroutine pure in L.M. che provvedono a ciò (SYSIQ, SYSIR).

Per ritornare al nostro programma Basic vediamo che da 5000 in poi sta l'inizializzazione del gioco mentre a 8000 e 9000 troviamo altre due subroutine di presentazione. Il LOOP principale di gioco va da 100 a 199 ed è lui che chiama tutte le subroutine sopra viste (ed altre ancora). Il loop stesso è preceduto da tutte le subroutine Basic necessarie al gioco stesso: movimento della bomba, del colpo laser del cannone, degli alieni eccetera.

Tra 200 ed 899 troviamo invece le subroutine che producono l'esplosione finale del cannoncino, l'aggiornamento del record ecc..

(segue da pag. 115)

```

7090 PRINT"II 00 100 200 300 400 500 600 700 800 900"
7100 RETURN
8000 POKE198,0:PRINT"NUM >VARIANTI DI GIOCO:"
8010 PRINT"NUM 1=DIFFICOLTA'?(1-9) 0=0":
8020 GETG$:G0=VAL(G$):IFG0<10R0G0>9THEN8020
8030 G0=G0(GG):VA=VA(G0):PRINTG$
8040 PRINT"NUM ALIENI VISIBILI? 0=0":
8050 GETG$:IFG$="S"THENPRINT"1":CN$="S":POKEFL,0:GOTO8080
8060 IFG$="N"THENPRINT"0":CN$="N":POKEFL,1:GOTO8080
8070 GOTO8050
8080 FORK=0TO34:FORU=0TO1:A$(K,U)=CN$+RIGHT$(A$(K,U),1):NEXT:
8090 POKE162,0:WAIT162,64:RETURN
9000 PRINT"NUM PUNTI"
9010 PRINT"NUM FORA=0TO4:PRINTTAB(4)A$(A*7,0)TAB(14)"DT(A*2):PRINT:NEXT
9020 PRINT"NUM PUNTI", "NUM 55":PRINT
9030 PRINT"NUM PUNTI", "NUM 256":PRINT
9035 POKE162,0:WAIT162,64
9040 PRINT"NUM PER COMINCIARE PREMI"TAB(51)"[FUOC0]";
9050 WAIT37137,32,32:RETURN

```

Scroll per Loader

di Marco Bodon - Padova

Rovistando nella memoria del VIC ho trovato alcune POKE interessanti: tra queste vi segnalo la 36865, il cui contenuto (usualmente 38) indirizza l'altezza dell'immagine rispetto allo schermo. Per capir meglio provare il seguente programmino:

```

10 REM * SCROLL
20 FOR A=0TO150
30 POKE 36865,A
40 FOR T=0TO99:NEXT
50 NEXT A
60 FOR A=150TO0STEP-1
70 POKE 36865,A
80 FOR T=0TO99:NEXT
90 NEXT A
98 POKE 36865,38
99 END

```

```

1000 REM=====
1001 REM= VIC LOADER=
1002 REM= VERS. 2.0=
1003 REM=====
1010 :
1020 PRINT"7" POKE 36865,255
1030 PRINTTAB(22)"MC MICROCOMPUTER PRESENTA"
1040 PRINT"PRINT"PRINT
1050 PRINT"NUM PROGRAMMA PER IL MICROCOM"
2000 B=38400-7680
2010 READ T:IF T=999 THEN 2100
2020 S=T+7878:POKE S,81:POKE B+S,7
2030 GOTO 2010
2040 :
2100 REM * SCROLL
2110 FORA=150TO0STEP-1
2120 POKE36865,A
2130 FORT=0TO99:NEXT
2140 NEXT A
2150 GOTO 3000
2200 DATA 0,4,6,9,10,12,13,14,17,18
2208 DATA 22,26,28,30,36,38,41
2209 DATA 44,48,50,52,58,60,63
2210 DATA 66,70,72,74,80,82,85,88
2211 DATA 92,94,96,101,104,107,111
2212 DATA 113,116,118,122,126,129
2213 DATA 134,138,141,143,144
2214 DATA 145,146,149,150,999
2300 :
3000 PRINTCHR$(15)
3010 POKE888,129:POKE 631,131:POKE 632,13
3011 POKE198,2:END

```

La lettera proseguiva con altre curiosità (che non è detto non vengano usate, prima o poi); questa invece la usiamo per miglio-

rare il programmino VIC LOADER che vi abbiamo proposto su MC 26. Se quella vecchia semplicemente scriveva in cubitale VIC-20 per poi caricare il programma successivo in auto-run (disabilitando il LIST), questo fa salire la scritta dal basso ed inoltre disabilita anche il RUN-STOP.

Va detto che questi programmi che manipolano le variabili di sistema possono anche creare dei problemi (tipo blocchi totali) proprio perché non è detto che il computer usi il contenuto di una cella solo per quello che noi crediamo, cosa questa molto importante per il 64 ma un pò meno per il VIC.

Istruzioni per l'uso del LOADER

Un breve riepilogo non guasta, anche a beneficio di chi ha incontrato difficoltà con il n. 26.

Si tratta di caricare due programmi l'uno dopo l'altro e in modo che le operazioni del primo rimangano inalterate dopo il caricamento del secondo. Bisognerà quindi:

- 1) digitare il LOADER;
- 2) registrarlo sul nastro, senza dare il RUN;
- 3) digitare il programma successivo, uno qualsiasi che vi interessi;
- 4) registrarlo subito dopo il LOADER, sempre senza dare il RUN;
- 5) riavvolgere il nastro e caricare il primo dei due. Non serve il RUN.

Succederà che il caricatore (LOADER) scriverà alcune cose sullo schermo, poi disabiliterà alcuni tasti o funzioni — tipo il LIST o il RUN STOP — ed infine si caricherà il programma successivo, che non sarà quindi accessibile ma si potrà usare normalmente. **MC**

Frogger per tutti (da 6.5K in poi)

di Giovanni Beani
Marina di Pietrasanta (LU)

Sono un giovane di 22 anni, studente di informatica all'Università di Pisa, e anche se seguo MCmicrocomputer da un paio d'anni non vi ho mai scritto a causa della mia estrema pigrizia.

Riferendomi al n° 26 di MC, nell'articolo del gioco VIC FROGGER voi affermate testualmente: "C'è però un grosso problema: il programma, assolutamente non parametrico, gira solo ed esclusivamente nella versione da 6.5K RAM; la riconversione richiederebbe un lavoro pedante... pubblicheremo con solerzia le modifiche per la versione da 8K in su". Così voglio sollevarvi dalla fatica di riscrivere completamente il programma: infatti basta modificare le linee 10 e 20 come mostrato nel listato 1, e poi battere le seguenti istruzioni:

```
POKE 642,32: POKE 648,30: SYS 64818
```

dopodiché non rimane che caricare, da cassetta o da disco (beato chi può!) il programma, e dare il RUN. Tutto girerà alla perfezione.

Miracolo? No, niente di speciale o di estremamente complicato. Vediamo le istruzioni ad una ad una.

POKE 642,32:
sapendo che in 641-642 (\$0281-\$0282) si ha l'indirizzo di partenza della memoria, con la parte più significativa in 642, basta cambiare il valore contenuto in questa locazione (che è 16 per il VIC in configurazione di base, 4 per l'espansione da 3K, 18 con espansione da 8 o 16K) per spostare in avanti l'inizio della memoria utente, e più precisamente all'inizio dell'espansione dei primi 8K (8192; \$2000).

POKE 648,30:
Nel byte 648 (\$0288) si trova il numero di pagina della mappa video (che è a 30-\$1E - per il VIC di base e con i 3K in più, a 16-\$10 - se si dispone di 8K o più), quindi mettendoci 30 si porta automaticamente l'inizio della mappa video a 7680

(\$1E00) con l'espansione da 3K.

SYS 64818:
ovviamente dovremo anche ripristinare tutti i vettori di I/O in RAM perché tutto funzioni, e per ottenere lo scopo basta chiamare parte della routine che parte all'accensione della macchina (che parte da 64802 - \$FD22), dall'indirizzo 64818. Da lì, infatti, si ha il salto alla subroutine che inizializza i vettori di salto (da \$0314 a \$0333), dopo di che si passa a quella che si prende cura dei due circuiti integrati 6522 del VIC, quindi si rivolge alle variabili di sistema, per tornare infine al BASIC.

Spero di aver dato il mio contributo al mondo sempre più vasto dei microfanatici.

* * *

Le modifiche proposte dal lettore rendono possibile l'inserimento del Frogger nel catalogo di software proposto da MC: ricordiamo ancora che è richiesto un minimo di 6.5K RAM.

TELCOM DUE CARTE IN PIU'

LA NUOVA STAMPANTE JUKI A MARGHERITA
LA NUOVA STAMPANTE MITSUI AD AGHI

JUKI 6100

finalmente una stampante A MARGHERITA accessibile a tutti per il suo BASSO COSTO. La caratteristica fondamentale di una stampante a margherita è la stampa a carattere pieno che garantisce una qualità di scrittura indispensabile per la corrispondenza automatica e il trattamento delle parole. L'alto prezzo di queste stampanti ne ha sempre impedito l'utilizzo in sistemi a basso costo quali personals, desk-top computers e micro in genere.

JUKI 6100 è la prima stampante a margherita che garantisce:

- prestazioni elevate
- affidabilità
- qualità di stampa
- prezzo coerente con i piccoli sistemi

JUKI 6100: un passo avanti in tecnologia
un passo indietro nel costo

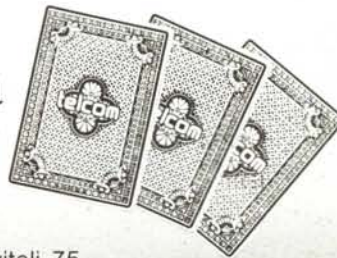
MITSUI Serie MC

Le stampanti della MITSUI rappresentano la continuità Telcom nella politica del "LOW-COST" con caratteristiche e qualità competitive:

- Velocità di 120/160 caratteri al secondo
- Testina a 9 aghi; 80/132 colonne
- Trascinamento per fogli singoli, per rotoli o per modulo continuo
- Interfaccia parallela e seriale
- Scrittura normale; espansa e NLQ
- Set di 95 caratteri
- Affidabilità elevata
- Ridotto costo di esercizio

MITSUI Serie MC: le stampanti "giuste" per micro e personals

gioca la carta
telcom



Telcom s.r.l. - 20148 Milano - Via M. Civitali, 75
Tel. 4047648 (3 linee ric. aut.) - Telex 335654 TELCOM I

