

# software

C-128



## La casa stregata

di Moreno Andreini  
Capalbio Scalo

Quando mi è venuto in mente l'idea di fare un adventure non credevo che questo avrebbe raggiunto le dimensioni che poi ha raggiunto nella stesura finale.

Nei paesi di lingua anglosassone questi tipi di giochi sono molto comuni e molto apprezzati, il più delle volte più apprezzati dei videogame.

### Per chi vuole il listato

Il listato di questo programma è lungo alcuni metri. In conseguenza di ciò, si è ritenuto opportuno non pubblicarlo, sia perché avrebbe occupato troppo spazio sulla rivista sottraendone ad altri argomenti, sia perché una digitazione senza errori di un listato così lungo appare poco probabile.

Chi è interessato al programma può ordinare, secondo il solito sistema, il disco o la cassetta in redazione. È anche possibile «pescare» direttamente (e gratuitamente) il programma per via telematica, dal nostro servizio MC-Link; questo ovviamente vale per chi è attrezzato in tal senso. Ricordiamo che per ottenere una casella su MC-Link è sufficiente telefonare (con un modem e un programma di comunicazione) al numero 06/4510211.

Questo programma è disponibile su disco presso la redazione. Vedere l'elenco dei programmi disponibili e le istruzioni per l'acquisto a pag. 186.

Qui in Italia invece non hanno molto successo, forse dipenderà dal fatto che le più belle adventure sono scritte in lingua inglese, e che per poterci giocare bisogna avere una buona conoscenza della lingua.

La prima cosa che dobbiamo chiederci quando decidiamo di scrivere un programma di questo tipo è, quale sarà lo scenario, o meglio in quale epoca ambienteremo la nostra avventura.

Dobbiamo vagliare molto attentamente questo fatto, infatti da ciò può dipendere il successo o l'insuccesso della nostra fatica.

Rischiamo molte volte di cadere nel visto e rivisto, ma indubbiamente è difficile riuscire a trovare una trama che sia bella, avvincente e nel medesimo tempo originale.

Io ho ambientato questa mia nel genere horror per vari motivi, non ultimo perché è un genere che a me piace in particolar modo.

Non mi sono posto inoltre il problema se la mia avventura fosse piaciuta o meno; infatti chi si siede davanti al computer e si accinge ad una fatica simile deve prescindere dal pensiero che il suo programma possa piacere o meno, infatti devono essere presi in considerazione diversi aspetti dell'avventura:

1) Non preoccuparsi eccessivamente della fantasia del giocatore, infatti una persona senza fantasia non si siederà mai davanti ad un computer per giocare con un adventure, si stuferà prima.

2) Mettersi sempre nei panni del giocatore e considerare tutte le possibili risposte che potrebbero essere date da chi siede davanti al video.

3) Arricchire il più possibile il programma con effetti sonori, sprite e grafica (il 128 si presta molto bene a ciò).

Il gioco che mi appresto a presentare non è una vera e propria avventura nel senso classico della parola, infatti non ci troveremo mai di fronte a situazioni imbarazzanti dove non sapremo cosa fare o quali comandi dare, ma piuttosto il gioco è una specie di labirinto disseminato di pericoli e di oggetti utili che servono per sfuggire a questi pericoli; non è detto però che tutti gli oggetti siano utili in egual misura.

Gli oggetti ed i pericoli sono disseminati per le stanze della casa, i pericoli sono rappresentati dai fantasmi che popolano la casa, mentre gli aiuti sono gli oggetti più comuni per allontanare i fantasmi.

Per spiegare che cosa è un adventure farò una breve descrizione della logica che porta alla costruzione del gioco.

In modo generale una stanza od una locazione possono essere qualunque cosa, per esempio una foresta può essere una stanza, la riva del mare può essere ugualmente una stanza, così come un deserto, la cima di un albero oppure la cucina di una casa.

L'importante, dopo aver buttato su carta la trama di massima, è tracciare una mappa del percorso. Questa mappa deve essere trascritta su di una griglia quadrata, per esempio:

Ogni casella può rappresentare una locazione o stanza, ma non tutte necessariamente devono essere stanze.

Per esempio se voi decidete che la partenza del gioco debba avvenire dalla casella n. 1, voi da qui potrete anda-

re o alla locazione n. 2, oppure alla locazione n. 6.

Se voi decidete che la locazione n. 1 rappresenti un sentiero, e la n. 2 un fiume, mentre la n. 6 non volete che entri in gioco, dovete far in modo che il giocatore sia impossibilitato a muovere sulla casella n. 6.

Dovete altresì fare in modo che il giocatore che si trovi sulla casella n. 2, possa tornare, se vuole, sulla casella n. 1.

Naturalmente nelle stanze metterete dei pericoli e degli oggetti. Il programma deve essere in grado di far sì che se il giocatore raccoglie un oggetto in una determinata locazione e lo depone in un'altra, una volta che si ritrovi a passare in quella determinata locazione dove ha deposto l'oggetto, questo si debba trovare effettivamente là: si avrà così l'impressione di muoversi veramente tra le varie stanze. Il tutto è realizzato semplicemente con una formuletta matematica ed esattamente: se associamo ad una variabile P il punto di partenza (P=1 in questo caso), il

giocatore, come visto sopra, potrà andare solo ad ovest (2) oppure a sud (6).

Se decidiamo per l'ovest, allora la formula sarà  $p=p+1$ , in questo momento ci troveremo sulla casella n. 2, se decidiamo di tornare indietro (alla n. 1) allora la formula diverrà  $p=p-1$ .

In ugual maniera se decidessimo dalla locazione n. 1 di andare nella n. 6 la formula cambierà così:  $p=p+5$  (5 perché la griglia è composta da un quadrato di 5 caselle per lato) infatti p in questo caso diverrà 6. Analogamente, per tornare indietro la formula sarà  $p=p-5$ .

Quindi quando tracciate una mappa dovete avere l'accortezza di tracciarla quadrata, poi le caselle che non vi servono rimarranno inutilizzate.

Un altro consiglio che posso darvi è quello di prendere un foglio di carta molto grande perché potrete comodamente aggiungere locazioni.

La trama del mio gioco è molto semplice. La vostra fidanzata o la vo-

stra ragazza è partita per una vacanza in un paese un po' isolato e da quando è partita non ha dato più sue notizie. Voi, preoccupato, vi mettete in viaggio per cercarla ed arrivate a questo paese dove vi indirizzano verso una casa nei boschi.

La casa è stregata e abitata da innumerevoli fantasmi, vampiri, licantropi, zombie e chi più ne ha più ne metta.

Vostro scopo è quello di evitare i pericoli e riuscire a liberare la ragazza.

In questa casa però c'è anche un fantasma buono che io ho chiamato Beppe.

Questo fantasma, se entrerete nella sua stanza, vi darà dei consigli preziosi e in più vi farà vedere la mappa della casa.

Beppe è dislocato in 3 stanze, ma solo se prima entrerete nella prima stanza con questo fantasma sarete abilitati per vedere la mappa: mi spiego meglio. Voi potete entrare in una stanza con Beppe, ma in questo caso riceverete solo consigli e non la videata della mappa.

Ci sono vari oggetti nelle stanze, ogni volta che ne prendete uno un inventario permanente (sulla parte alta dello schermo) vi terrà al corrente di ciò di cui siete in possesso. In modo analogo, ogni volta che lascerete un oggetto, questo sparirà dall'inventario; il numero limite degli oggetti che potete portare è 3.

Questo inventario è realizzato mediante sprite; io non li ho definiti in variabili stringa o con linee data poiché ho usato il metodo di definirli mediante la funzione del 128 attraverso la quale si possono definire gli sprite (SPRDEF) e li ho quindi salvati come file binari (vedere manuale).

La griglia della casa è composta da 12 caselle per lato, quindi ne avrei in teoria 144 utili.

In realtà io ne ho usate 42, ma vi assicuro che sono più che sufficienti per perderci la testa.

Inoltre ho usato della musica e degli effetti sonori; la musica di introduzione l'ho presa sul manuale mentre alcuni effetti sonori li ho ottenuti dopo una paziente ricerca.

In più, all'inizio del gioco, compare una videata in alta risoluzione (copiata anche quella): comunque ognuno di voi con un po' di fantasia può sbizzarrirsi come vuole.

### Spiegazione delle linee

10-120: azzeramento variabili, impostazione colori, richiesta di non togliere il disco.

1300-1780: musica.

1800-2000: in queste linee di data ci sono le definizioni degli oggetti. In

### Titolatore

```

10 GRAPHIC0,1:COLOR0,1:COLOR4,1:COLOR5,5:SCNCLR
20 CHAR,14,3," TAPE LABEL ",1:COLOR5,8:CHAR,6,6,"ETICHETTE PER CASSETTE AUDIO"
30 COLOR5,11:CHAR,6,9,"PROGRAMMA PER COMMODORE 128"
40 COLOR5,14:CHAR,10,11,"E STAMPANTE MPS 1000"
60 COLOR5,6:CHAR,19,17,"BY":COLOR5,7:CHAR,11,19,"MARIA TERESA INTINI":SLEEP2
70 SCNCLR:DIMAS(18),BS(18),CS(18),DS(18):BS="I"
80 COLOR5,8:CHAR,13,0," DATI LATO 1 ",1:COLOR5,4
90 FORS=1TO18:IFS<10THENYS=" ":ELSEYS=""
100 CHAR,12,2,"-----":COLOR5,6
110 CHAR,0,2+S:PRINT"TILOLO"YS:S:INPUTA$(S):IFLEN(A$(S))>18THEN110
120 NEXT:PRINTCHR$(7):COLOR5,3
130 CHAR,11,23,"DATI CORRETTI S/N",1:GETKEYA$:IFA$="N"THEN80:ELSESCNCLR
140 COLOR5,8:CHAR,13,0," DATI LATO 2 ",1:COLOR5,4
150 CHAR,12,2,"-----":COLOR5,6
160 FORS=1TO18:IFS<10THENYS=" ":ELSEYS=""
170 CHAR,0,2+S:PRINT"TILOLO"YS:S:INPUTC$(S):IFLEN(C$(S))>18THEN170
180 NEXT:PRINTCHR$(7):COLOR5,3
190 CHAR,11,23,"DATI CORRETTI S/N",1:GETKEYA$:IFA$="N"THEN140:ELSESCNCLR
200 COLOR5,8:CHAR,9,0," DATI BORDO CASSETTA ",1:COLOR5,4
201 CHAR,18,2,"-----":COLOR5,6
210 CHAR,0,3:INPUT"SIGLA CASSETTA ",NS:IFLEN(NS)>5THEN210
220 CHAR,0,4:INPUT"TILOLO LATO A ",OS:IFLEN(OS)>18THEN220
230 CHAR,0,5:INPUT"TILOLO LATO B ",SS:IFLEN(SS)>18THEN230:ELSECOLOR5,3
240 PRINTCHR$(7):CHAR,11,7,"DATI CORRETTI S/N",1:GETKEYA$:IFA$="N"THEN210
250 PRINTCHR$(7):SCNCLR:CHAR,4,2,"VUOI IL NEAR LETTER QUALITY S/N",1:GETKEYA$
260 IFA$="S"THENOPEN4,4:PRINT#4,CHR$(31):CLOSE4:GOTO280
270 OPEN4,4:PRINT#4,CHR$(159):CLOSE4
280 COLOR5,8:CHAR,14,5," ATTENDI... ",1:PRINTCHR$(7):PRINTCHR$(7)
290 FORS=1TO18:FORT=1TO19-LEN(C$(S)):C$(S)=C$(S)+" ":NEXT
300 FORT=1TO19-LEN(A$(S)):A$(S)=A$(S)+" ":NEXT:NEXT
310 FORS=1TO18:BS(S)=BS+A$(S)+BS:DS(S)=C$(S)+BS:NEXT
320 FORT=1TO19-LEN(OS):OS=OS+" ":NEXT:FORT=1TO19-LEN(SS):SS=SS+" ":NEXT
330 FORT=1TO6-LEN(NS):NS=NS+" ":NEXT
340 OPEN4,4:OPEN6,4,6:PRINT#6,CHR$(27):CLOSE6
350 PRINT#4,CHR$(14)CHR$(18)"I SIDE A SIDE B "CHR$(146):CHR$(15)
360 PRINT#4," "
370 FORS=1TO18:PRINT#4,BS(S):DS(S):NEXT
380 PRINT#4," "
390 PRINT#4,"| "CHR$(18)"LATO A"CHR$(146)" "OS"| "
400 PRINT#4,"| "CHR$(14)N$CHR$(15)"| "
410 PRINT#4,"| "CHR$(18)"LATO B"CHR$(146)" "S$"| "
420 PRINT#4,"| "
430 PRINT#4,"| "
440 PRINT#4,"| "
450 PRINT#4,"| "
460 PRINT#4,"| "
470 PRINT#4,"| "
480 PRINT#4,"| "
490 PRINT#4,"| "
500 CLOSE4:FORS=1TO6:PRINTCHR$(7):NEXT:COLOR5,3:SCNCLR
510 CHAR,8,12,"ANCORA UN'ETICHETTA S/N",1:GETKEYA$:IFA$="S"THENRUN70:ELSEEND

```

**Frog 2000**

```

10 GRAPHICO.1:COLOR0.1:COLOR4.1:COLOR5.6
20 CHAR.5.2."| | | |"
30 CHAR.5.3."| | | |"
40 CHAR.5.4."| | | |"
50 CHAR.5.5."| | | |":COLOR5.3:CHAR.9.7."THE FROG 2000"
60 COLOR5.12:CHAR.6.12."PROGRAMMA PER COMMODORE 128"
70 COLOR5.6:CHAR.19.16."BY":COLOR5.7:CHAR.11.18."MARIA TERESA INTINI":COLOR5.8
80 BLOAD"SPRITE":BLOAD"LM":SYS2816
90 CHAR.6.20."PREMI UN TASTO PER INIZIARE".1:GETKEYAS
100 SCNCLR:CHAR.4.4."SCEGLI IL LIVELLO DI DIFFICOLTA'"
110 COLOR5.11:CHAR.14.6."[1 - 2 - 3]".1
120 GETKEYAS:IFAS<"1"ORAS>"3"THEN120
130 V=VAL(AS)*4:GRAPHIC1.1:COLOR1.12:FORD=OTO24:CHAR.31.D." ".1:NEXT
140 FORS=OTO6:DRAW1.0,S*30TO255,S*30:NEXT:COLOR1.11
150 FORS=1TO8:SPRITES.1.S+2.1:NEXT:FORS=1TO6:MOVSPRS.0.S*32+20:NEXT
160 CHAR.33.10."VITE".1:COLOR1.4:CHAR.33.2."GIRI".1:COLOR1.8:CHAR.33.6."TEMPO".1
170 MOVSPR8.126.237:MOVSPR7.295.220:SPRITE7.0
180 VI=6:G=0:J=250:POKEJ.0:TI$="000000":GOSUB290
190 COLLISION1.240:IFPEEK(J)=1THENGOSUB280
200 IFC=1THENC=0:GOTO250
210 COLOR1.4:GS=STR$(G):CHAR.33.3.GS
220 COLOR1.11:VS=STR$(VI):CHAR.33.11.VS
230 COLOR1.8:CHAR.33.7.RIGHT$(TI$,4):IFTI$="000059"THENQ=2:GOTO310:ELSEGOTO190
240 C=1:FORS=1TO6:MOVSPRS.90#0:NEXT:COLLISION1:COLLISION2:COLLISION1.240:RETURN
250 MOVSPR6.0.212:VI=VI-1:SOUND1.2000.150.1.200.10.2.60
260 FORT=1TO9:FORS=1TO16:MOVSPR8.126.237:SPRITE7.1.S.0:NEXT:NEXT:SPRITE7.0
270 IFVI=0THENQ=1:GOTO310:ELSEGOSUB290:GOTO190
280 G=G+1:MOVSPR8.126.237:SOUND1.5000.20.2.4900.100.0:POKEJ.0:RETURN
290 MOVSPR1.90#V:MOVSPR2.270#V+2:MOVSPR3.90#V+2
300 MOVSPR4.90#V+3:MOVSPR5.270#V+1:MOVSPR6.270#V+3:RETURN
310 FORS=1TO8:SPRITES.0:MOVSPRS.0#0:NEXT:GRAPHICO.1
320 IFQ=2THENMS="HAI FINITO IL TEMPO A TUA DISPOSIZIONE":Q=0
330 IFQ=1THENMS=" SEI STATO INVESTITO TROPPE VOLTE ":Q=0
340 CHAR.1.3.M$.1
350 COLOR5.6:CHAR.11.6."GIOCHI ANCORA S/N":GETKEYAS:IFAS="S"THEN100
360 COLOR5.8:CHAR.15.10."FINE GIOCO":SLEEP1:END

```

particolare modo il primo numero indica in quale stanza si trova l'oggetto, la prima stringa indica l'oggetto vero e proprio mentre la seconda è la descrizione dell'oggetto.

2000-2057: chiede se vogliamo le istruzioni se si va alla subroutine 50000.

Viene poi chiesto se vogliamo riprendere un gioco interrotto precedentemente e salvato (esiste la possibilità in qualunque momento di interrompere il gioco e di salvarlo su disco per riprenderlo successivamente, basta digitare la parola FINE alla richiesta di istruzioni), se si va alla subroutine 55000.

2060-2140: carica gli oggetti immagazzinati nel file binario e presenta il gioco.

2141: va alla subroutine 60030 per effetto sonoro pioggia.

2145: questa linea si occupa di settare a 138 (la mia locazione di partenza) la variabile p se la variabile d=0, cioè se il gioco non è stato caricato da disco, infatti un gioco caricato da disco avrà una locazione di partenza diversa.

2146: si occupa dell'inventario e di azzerare (con la subroutine 15000) tutti gli sprite degli oggetti.

2147: questa linea è un po' difficile da spiegare. Qui la variabile p viene posta uguale alla variabile p2; ogni volta che ci muoveremo per le stanze la variabile p2 conterrà il valore della locazione in cui siamo.

2150-2158: in queste righe sta il nocciolo del movimento. Infatti da qui, secondo che valore assume la variabile p, ci possiamo muovere per le varie stanze.

2160-2166: queste linee pongono gli oggetti nelle giuste stanze.

2175-2179: viene indicato in quale direzione siamo abilitati ad andare.

2180: input per colloquiare con il programma.

2183-2185: varie possibilità di risposta nella locazione di partenza.

2190-2205: testa le direzioni che contengono una sola lettera.

2210-2250: controlla se sono state usate due parole.

2260-2310: converte la stringa immessa nell'input (i\$) in verbo (ve\$) e nome (no\$).

2320-2360: subroutine per scandire comandi ve\$.

2400-2430: per acquisire i vari oggetti.

2440-2460: tutte le possibilità per ingannare il computer sono vagliate, in più viene conteggiato il numero degli oggetti presi e se si tenta di prenderne di più la macchina lo fa notare garbatamente.

2641-2470: abilita gli sprite degli oggetti presi.

2500-2560: disabilita gli sprite quando possiamo gli oggetti.

2600-2670: se ci troviamo in una stanza con un pericolo e chiediamo di uccidere il mostro ci viene chiesto come.

2700-2800: in queste linee viene chiesto di specificare l'oggetto con cui vogliamo uccidere i mostri.

3000-3040: routine del movimento.

3100-7130: descrizione delle stanze.

10000-10030: viene richiesto se siete incappato in qualche incidente se volete giocare ancora.

20000: viene conteggiato il tempo da cui avete iniziato a giocare, se questo tempo è superiore a 10 minuti il vampiro si sveglia dalla sua bara e voi siete morto.

**Variabili**

s1: quando si entra nella prima stanza con Beppe viene posta ad 1, quindi quando entreremo nelle altre stanze sempre con Beppe ci verrà mostrata la mappa, se s1=0 la mappa non viene mostrata.

s2: viene posta a 1 quando i ragni sono stati uccisi.

s3: viene posta a 1 quando il serpente è stato ucciso.

s4: viene posta a 1 quando il licantropo è stato ucciso.

s5: viene posta a 1 quando il vampiro è stato ucciso.

s6: viene posta a 1 quando l'assassino è stato ucciso.

s7: viene posta a 1 quando lo zombie è stato ucciso.

iv: inventario.

ti\$: tempo da quando iniziato il gioco se > 10 min. e s5=0 allora si sveglia il vampiro.

d: se viene caricato un vecchio gioco d è messa ad 1.

p: valore della stanza sulla griglia.

p2: nuovo valore che assume la posizione dopo lo spostamento.

n: nord.

e: est.

s: sud.

w: ovest.

Nella stesura di questo gioco devo molto al libro di Mike Grace edito dalla casa editrice Jackson.

**Titolatore**

di M. Teresa Iutini  
Putignano (BA)

Titolatore è un programma che, in unione con la stampante MPS 1000, consente di realizzare etichette per audiocassette. Prima di descriverne l'uso e il funzionamento occorre sapere che l'etichetta è così formata;

**PARTE FRONTALE:** vengono scritti i titoli dei brani (o dei programmi) incisi sui due lati della cassetta, 18 per lato per un totale di 36 titoli avente ognuno la lunghezza massima di 18 caratteri alfanumerici;

**BORDO CASSETTA:** viene scritto un codice di riconoscimento della cassetta di massimo 5 caratteri alfanumerici, oltre ai 2 titoli (lato A e lato B) della cassetta.

**RETRO CASSETTA:** vengono scritte, a discrezione dell'utente, fino a 6 righe di testo contenenti ad esempio nome e indirizzo del proprietario della cassetta o eventuali note.

Dato il run sul nostro monitor appare una prima schermata contenente il titolo del programma. Dopo una brevissima pausa, vengono chiesti i 18 titoli per il lato A, viene fatta una domanda di conferma sull'esattezza dei dati introdotti e se si risponde positivamente, si ripete il discorso inserimento dati anche sul lato B. In caso

contrario è possibile, prima di proseguire, correggere eventuali errori. Successivamente vengono richiesti i dati per il bordo della cassetta: la sigla, il titolo del lato A e quello del lato B. Introdotti pazientemente i dati dobbiamo rispondere alla domanda inerente la qualità di stampa, rispondendo con un «S» si attiva il modo NLQ che consente una riproduzione su carta eccellente, rispondendo invece con un «N» avremo una stampa molto più veloce ma ci dovremo accontentare dei caratteri draft.

A questo punto termina il lavoro di introduzione dati e il nostro 128, dopo aver eseguito alcune manipolazioni sulle stringhe, dà l'OK alla stampante. Il listato appare subito molto breve e

semplice ma a mio parere molto funzionale: analizziamolo.

**LINEE 10-60** settaggio dei colori e stampa su video della prima schermata.

**LINEA 70** vengono dimensionate le matrici e attribuito alla variabile B\$ il carattere corrispondente al codice chr\$(98) barretta verticale che utilizzeremo per stampare su carta i contorni dell'etichetta.

**LINEE 80-130** vengono richiesti i 18 titoli per il lato A e effettuata la routine di scelta (proseguire o correggere). **LINEE 140-190** come le linee 80-130 però per il lato B.

**LINEE 200-250** inserimento dati per il bordo della cassetta.

**LINEE 250-280** scelta modo di stampa PICA o NLQ e comunicazione della scelta fatta alla stampante.

**LINEE 290-330** vengono aggiunti degli spazi in coda alle stringhe che ne abbiano bisogno per poter effettuare il corretto incolonnamento dei dati.

**LINEE 354-500** sono le linee di scambio dati con la stampante, a lavoro finito una serie di beep ci avviserà di recuperare la carta e tagliare l'etichetta.

**LINEA 510** consente di far ripartire il programma o di terminare il lavoro.

Per concludere, aggiungiamo che, a parte la selezione dei modi di scrittura, il programma funziona anche con le altre stampanti Commodore.

Frog 2000

Dump della routine in LM.

```
>00B00 78 A9 0D BD 14 03 A9 0B 8D 13 03 58 60 AD 00 DC:
>00B10 C9 7D D0 03 4C 48 0B C9 7E D0 03 4C 56 0B C9 7B:
>00B20 D0 03 4C 6F 0B C9 77 D0 03 4C 62 0B C9 7A D0 03:
>00B30 4C 8F 0B C9 76 D0 03 4C 7A 0B C9 79 D0 03 4C A4:
>00B40 0B C9 75 D0 03 4C 89 0B 4C 65 FA AD E5 11 69 01:
>00B50 8D E5 11 4C CE 0B 38 AD E5 11 E9 02 8D E5 11 4C:
>00B60 CE 0B 18 AD E4 11 69 02 8D E4 11 4C CE 0B 38 AD:
>00B70 E4 11 E9 02 8D E4 11 4C CE 0B 38 AD E5 11 E9 02:
>00B80 8D E5 11 18 AD E4 11 69 02 8D E4 11 4C CE 0B 38:
>00B90 AD E5 11 E9 02 8D E5 11 38 AD E4 11 E9 02 8D E4:
>00BA0 11 4C CE 0B 38 AD E4 11 E9 02 8D E4 11 18 AD E5:
>00BB0 11 69 02 8D E5 11 4C CE 0B 18 AD E5 11 69 02 8D:
>00BC0 E5 11 18 AD E4 11 69 02 8D E4 11 4C CE 0B AD E5:
>00BD0 11 C9 29 D0 04 A9 01 85 FA AD E5 11 C9 EF F0 03:
>00BE0 4C E8 0B A9 ED 3D E5 11 AD E4 11 C9 FA F0 03 4C:
>00BF0 F7 0B 49 FB 8D E4 11 AD E4 11 C9 0C F0 03 4C 65:
>00C00 FA A9 0E BD E4 11 4C 65 FA FF 00 FF 00 FF 00 FF:
```

Dump dell'area immagazzinamento Sprite.

```
>00E00 1F FF 00 FF FF 80 E1 01 80 E1 01 80 E1 01 80 E1:
>00E10 09 88 61 6D FC 3F 0B FC 3F 81 FC 1F C1 FC 1F FF:
>00E20 FF 22 01 B3 49 01 11 1C 00 38 0B 00 10 00 00 00:
>00E30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00:
>00E40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00:
>00E50 3F FB 00 61 04 00 C1 04 1F FF FF 3F FF 3F FF 3F FF:
>00E60 FF 1C 7F C7 09 7F 92 03 BF 88 01 00 10 00 00 00:
>00E70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00:
>00E80 00 00 00 FF FF FC 00 FF FC 00 9C 80 DD 6C 9B DB:
>00E90 9C 8C D7 6C 86 D1 9C 86 FF FF FE FF FE E3 FF FF:
>00EA0 1F C9 FE 4F 1C 00 E0 08 00 40 00 00 00 00 00 00:
>00EB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00:
>00EC0 1F FF 00 FF FF 80 E1 01 80 E1 01 80 E1 01 80 E1:
>00ED0 09 88 61 0D FC 3F 0B FC 3F 81 FC 1F C1 FC 1F FF:
>00EE0 FF 22 01 B3 49 01 11 1C 00 38 0B 00 10 00 00 00:
>00EF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00:
>00F00 00 FF F0 01 FF FF 01 B1 87 01 81 87 11 91 87 3F:
>00F10 B1 86 3F D1 FC 3F 83 FC 3F 87 FC FF FF FC C1 80:
>00F20 62 88 80 49 1C 00 1C 0B 09 08 00 00 00 00 00 00:
>00F30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00:
>00F40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00:
>00F50 3F FB 00 61 04 00 C1 04 1F FF FF 3F FF 3F FF 3F FF:
>00F60 FF 1C 7F C7 09 7F 92 03 BF 88 01 00 10 00 00 00:
>00F70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00:
>00F80 07 FF EC 0F FF F0 1F FF FB 1F FF FB 30 FF BC 38:
>00F90 FF 1C 30 7E 0C 38 FF 1C 3D FF 8C 3F E7 FC 1F E7:
>00FA0 FB 0F FF F0 06 99 60 01 99 00 E0 99 07 3E 7E 7C:
>00FB0 FF B1 FF 00 FF 00 FF B1 FF 1F 00 FB E0 00 07 00:
>00FC0 00 20 C1 0D 11 E2 00 0B F4 00 07 38 00 07 38 00:
>00FD0 07 38 00 0F 3C 00 17 3A 00 23 F1 00 07 FB 00 09:
>00FE0 E4 00 10 02 00 00 00 00 00 00 00 00 00 00 00 00:
>00FF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00:
```

Frog 2000

di M. Teresa Iutini Putignano (BA)

The frog 2000 è un giochino suddiviso in due parti: una in Basic e l'altra in L.M. Dato il run, viene visualizzata la solita schermata introduttiva e vengono caricati con l'istruzione blood due file, il programma in LM e i dati relativi agli sprite che ho preferito introdurre in memoria tramite i versatili comandi del monitor piuttosto che da Basic con le istruzioni READ e DATA. Effettuato il caricamento viene chiesto al giocatore il livello di difficoltà desiderato e una volta fatta la scelta viene disegnata una strada a 6 corsie su cui scorrono velocemente altrettante auto. Il nostro compito è quello di aiutare uno scarabeo ad attraversare la strada senza subire incidenti.

Il gioco ha una durata di 60 secondi e in questo tempo occorre far compiere al nostro animaletto il più alto numero di attraversamenti tenendo conto che se esso viene investito per più di 6 volte il gioco termina anche se non è

## Routine in LM Lettura Joystick.

```

MONITOR
PC SR AC XR YR SP
; 000 00 00 00 00 F8
00B00 78 SEI
00B01 A9 0D LDA ##0D
00B03 8D 14 03 STA #0314
00B06 A9 0B LDA ##0B
00B08 8D 15 03 STA #0315
00B0B 58 CLI
00B0C 60 RTS
00B0D AD 00 DC LDA #0C00
00B10 C9 7D CMP ##7D
00B12 D0 03 BNE #0B17
00B14 4C 4B 0B JMP #0B4B
00B17 C9 7E CMP ##7E
00B19 D0 03 BNE #0B1E
00B1B 4C 56 0B JMP #0B56
00B1E C9 7B CMP ##7B
00B20 D0 03 BNE #0B25
00B22 4C 6F 0B JMP #0B6F
00B25 C9 77 CMP ##77
00B27 D0 03 BNE #0B2C
00B29 4C 62 0B JMP #0B62
00B2C C9 7A CMP ##7A
00B2E D0 03 BNE #0B33
00B30 4C 8F 0B JMP #0B8F
00B33 C9 76 CMP ##76
00B35 D0 03 BNE #0B3A
00B37 4C 7A 0B JMP #0B7A
00B3A C9 79 CMP ##79
00B3C D0 03 BNE #0B41
00B3E 4C A4 0B JMP #0BA4
00B41 C9 75 CMP ##75
00B43 D0 03 BNE #0B48
00B45 4C B9 0B JMP #0BB9
00B48 4C 65 FA JMP #FA65
00B4B AD E5 11 LDA #11E5
00B4E 69 01 ADC ##01
00B50 8D E5 11 STA #11E5
00B53 4C CE 0B JMP #0BCE
00B56 38 SEC
00B57 AD E5 11 LDA #11E5
00B5A E9 02 SBC ##02
00B5C 8D E5 11 STA #11E5
00B5F 4C CE 0B JMP #0BCE
00B62 1B CLC
00B63 AD E4 11 LDA #11E4
00B66 69 02 ADC ##02
00B68 8D E4 11 STA #11E4
00B6B 4C CE 0B JMP #0BCE
00B6E 38 SEC
00B6F AD E4 11 LDA #11E4
00B72 E9 02 SBC ##02
00B74 8D E4 11 STA #11E4
00B77 4C CE 0B JMP #0BCE
00B7A 38 SEC
00B7B AD E5 11 LDA #11E5
00B7E E9 02 SBC ##02
00B80 8D E5 11 STA #11E5
00B83 1B CLC
00B84 AD E4 11 LDA #11E4
00B87 69 02 ADC ##02
00B89 8D E4 11 STA #11E4
00B8C 4C CE 0B JMP #0BCE
00B8F 38 SEC
00B90 AD E5 11 LDA #11E5
00B93 E9 02 SBC ##02
00B95 8D E5 11 STA #11E5
00B98 38 SEC
00B99 AD E4 11 LDA #11E4
00B9C E9 02 SBC ##02
00B9E 8D E4 11 STA #11E4
00BA1 4C CE 0B JMP #0BCE
00BA4 38 SEC
00BA5 AD E4 11 LDA #11E4
00BAB E9 02 SBC ##02
00BAA 8D E4 11 STA #11E4
00BAD 1B CLC
00BAE AD E5 11 LDA #11E5
00BB1 69 02 ADC ##02
00BB3 8D E5 11 STA #11E5
00BB6 4C CE 0B JMP #0BCE
00BB9 1B CLC
00BBA AD E5 11 LDA #11E5
00BBD 69 02 ADC ##02
00BEF 8D E5 11 STA #11E5
00BC2 1B CLC
00BC3 AD E4 11 LDA #11E4
00BC6 69 02 ADC ##02
00BC8 8D E4 11 STA #11E4
00BCB 4C CE 0B JMP #0BCE
00BCE AD E5 11 LDA #11E5
00BD1 C9 29 CMP ##29
00BD3 D0 04 BNE #0BD9
00BD5 A9 01 LDA ##01
00BD7 85 FA STA #FA
00BD9 AD E5 11 LDA #11E5
00BDC C9 EF CMP ##EF
00BDE F0 03 BEQ #0BE3
00BE0 4C E8 0B JMP #0BE8
00BE3 A9 ED LDA #ED
00BE5 8D E5 11 STA #11E5
00BE8 AD E4 11 LDA #11E4
00BEB C9 FA CMP ##FA
00BED F0 03 BEQ #0BF2
00BEF 4C F7 0B JMP #0BF7
00BF2 A9 F8 LDA ##F8
00BF4 8D E4 11 STA #11E4
00BF7 AD E4 11 LDA #11E4
00BFA C9 0C CMP ##0C
00BFC F0 03 BEQ #0C01
00BFE 4C 65 FA JMP #FA65
00C01 A9 0E LDA ##0E
00C03 8D E4 11 STA #11E4
00C06 4C 65 FA JMP #FA65

```

scaduto il tempo. Tutto ciò è semplice ma è utile per comprendere il funzionamento dei comandi del Basic 7.0 inerenti gli sprite e le loro collisioni.

Vediamo di analizzare il programma. Innanzitutto la routine in LM locata da B00 a C08 consente di muovere lo scarabeo con il joystick entro i limiti dello schermo e qualora detto animale (impersonato dallo sprite n. 8) si porti nell'estrema posizione superiore del video, viene posto nella locazione 250 il valore numerico 1 che il programma Basic utilizzerà per incrementare la variabile che controlla il numero di attraversamenti e per altre cose ancora...

Andando con ordine nell'analisi del listato, dapprima incontriamo nel gruppo di linee 10-70 la routine introduttiva.

La linea 80 contiene le istruzioni di caricamento routine in LM e dati per gli sprite oltre che la SYS per il movimento dello scarabeo tramite il joystick. Dalla linea 90 a 120 viene chiesto il livello di difficoltà.

Le linee 130-170 contengono le istruzioni di creazione del campo di gioco e l'attivazione degli sprite mentre la variabile V, presente in linea 130, contiene il valore del livello di difficoltà.

In 180 sono settate le variabili di gioco, il numero di vite iniziali, il numero di giri, la variabile ti\$ e viene posta nella locazione di memoria 250 il valore 0.

Da 190 a 230 si svolge il ciclo principale del gioco; tramite queste linee vengono aggiornati e visualizzati i parametri del gioco (vite, tempo, giri).

In 240 è posta la subroutine di gestione collisioni a cui si perviene tramite il comando collision di linea 190. Viene fra l'altro settata la variabile C che, se assume valore 1, fa sì che dalla linea 200 il flusso del programma devii dal ciclo principale per saltare al gruppo di linee 250-270, incaricate di segnalare l'avvenuto incidente decrementando la variabile VI (vite) ed emettendo un segnale acustico.

In 280 vi è invece la subroutine a cui si salta dopo aver compiuto un attraversamento ed ha il compito di incrementare la variabile G (giri).

Segue in 290-300 una routine che viene chiamata per mettere in movimento gli sprite raffiguranti le varie auto che circolano per strada.

Infine da 310 a 360 vi è la routine di fine gioco.