

Italia 128

di Marco Gastaldi-Beinette

Il programma, del quale non pubblichiamo il listato data l'eccessiva lunghezza, non sarà particolarmente interessante per quanto riguarda la tecnica di programmazione, ma sicuramente lo è dal punto di vista dell'originalità. È un gioco educativo che consiste nel cercare su di una cartina geografica rappresentante l'Italia, una città capoluogo di provincia che di volta in volta il computer segnalerà. Sullo schermo vengono rappresentate 23 città fra cui compare naturalmente quella da cercare. Il giocatore può, tramite un mirino mosso da joystick, posizionarsi sulla città voluta e premere in seguito il tasto di fuoco. In caso di successo si passa a cercare la prossima città mentre, se il capoluogo puntato non era quello richiesto, si hanno a disposizione altri due tentativi. Dopo il terzo errore il computer segnala l'esatta posizione della città da cercare. Al termine di dieci prove viene segnalato un giudizio sulla prestazione del giocatore ed il programma termina.

Questo programma può essere un valido pretesto per insegnare ai bambini la geografia. È stato di grande aiuto anche per me e devo confessare che, grazie al programma, ho finalmente imparato dove si trovano queste benedette città italiane!

Descrizione del programma

«Italia 128» è scritto in Basic e utilizza due routine in LM. La prima, locata nel bank 0 all'indirizzo decimale 32768, ha la funzione di immettere la schermata

È disponibile, presso la redazione, il disco con il programma pubblicato in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 251

Si prega gentilmente il signor Marco Gastaldi di contattare la redazione.

rappresentante l'Italia nella memoria video. Questo si è reso necessario perché un costante accesso al disco avrebbe rallentato eccessivamente l'esecuzione del programma. Ho pensato quindi di caricare una volta sola i dati del disegno e di depositarli in un'area di memoria libera; precisamente all'indirizzo 40960. La routine si occuperà ogni volta di effettuare il trasferimento, rendendo tale operazione trasparente al giocatore.

La seconda routine gestisce il movimento dello sprite-mirino ed è locata nel bank 15 all'indirizzo 5632.

Una volta lanciato, il programma carica la schermata grafica, le routine sopracitate e i dati per gli sprite che vengono memorizzati nei due vettori stringa ad otto elementi S0 ed S1.

Quindi, dopo la presentazione in cui si ascolta ad una voce un pezzo dell'inno nazionale, il programma richiede l'inserimento del nome del giocatore. A questo punto inizia il gioco sul quale non ho niente da aggiungere.

Descrizione del listato

10-250

caricamento della routine LM, schermata grafica, dati sprite. Presentazione;

260

disabilitazione dei tasti run/stop e restore;

290-370

programma principale. Da qui vengono chiamate tutte le subroutine del programma;

400-480

viene suonato l'inno nazionale

510-610

posizionamento degli sprite sullo schermo;

630-720

dimensionamento delle variabili;

740-850

input nome del giocatore;

890-1240

questa è la subroutine principale del gioco;

900-950

viene chiamata la routine per lo schermo grafico, vengono visualizzati i punti, la manche e vengono disegnati i capoluoghi;

1000-1020

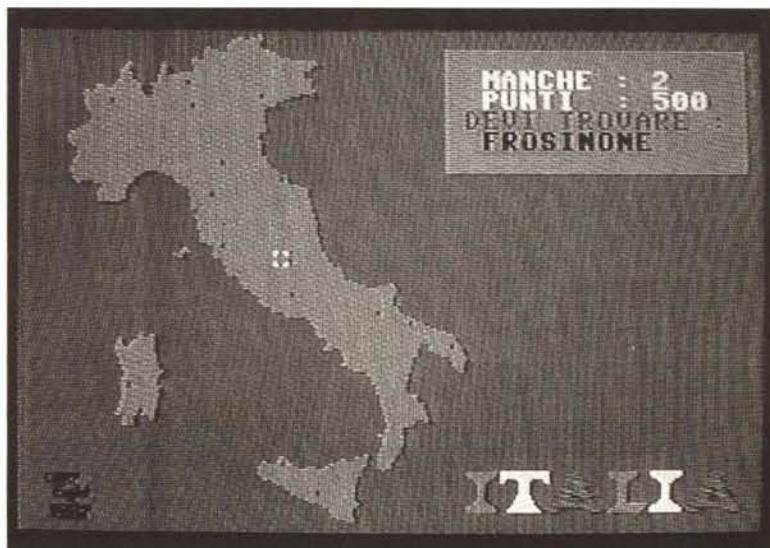
viene chiamata la routine di gestione joystick e si controlla se le coordinate dello sprite-mirino coincidono con quelle del capoluogo da cercare?

1070-1090

indovinato! si incrementa il punteggio e si torna alla linea 890;

1110-1240

sbagliato! uno sprite a forma di mano indica l'esatta posizione del capoluogo;



- 1260-1440** fine del gioco e possibilità di ricominciare;
1470-1570 linee data contenenti i capoluoghi con le coordinate di visualizzazione.

Variabili principali

- X()** vettore coordinate di visualizzazione sull'asse x
Y() vettore coordinate di visualizzazione sull'asse y
CS() vettore contenente i nomi dei capoluoghi
NS nome del giocatore
PX utilizzata nella subroutine di input nome
TE numero dei tentativi
MA numero delle marche
R numero casuale indicante il capoluogo
R1 utilizzata per la funzione RND
SY
SX coordinate x, y dello sprite-mirino
PU punti del giocatore
AS,J,I variabili di uso generale e per cicli for-next

Nota. I capoluoghi usati nel programma sono 89 anziché i reali 95. Vogliate scusare la mia dimenticanza... Naturalmente anche così il programma è ugualmente interessante.

Pointer per C-128

di Tommaso Masi - Abano Terme (PD)

Pointer è una breve utility in LM che farà comodo a tutti coloro che amano utilizzare nei loro programmi tecniche di interfacciamento verso l'utente dell'ultima generazione permettendo di selezionare funzioni e opzioni nel modo più intuitivo possibile. Pointer, infatti, consente di gestire un puntatore mediante

Pointer per C-128 - Programma caricatore.

```

10 BANK 15:CK=0
20 FOR J=0 TO 255
30 READ J$
40 POKE 3072+J,DEC(J$)
50 CK=CK+DEC(J$)
60 NEXT
70 IF CK<>24765 THEN LIST:PRINT CHR$(17):"ERRORE NE
I DATA!":END
80 BSAVE "POINTER".B15.P3072 TO P3328
90 END
100 DATA 00,00,07,00,00,00,58,01,FA,02
110 DATA 00,07,07,FA,CE,02,0C,D0,CB,AD
120 DATA 00,DC,AE,09,0C,CA,D0,03,AD,01
130 DATA DC,A0,00,A2,00,4A,B0,01,88,4A
140 DATA B0,01,C8,4A,B0,01,CA,4A,B0,01
150 DATA E8,4A,8E,00,0C,8C,01,0C,A9,00
160 DATA 2A,8D,0D,0C,8A,D0,0B,98,D0,08
170 DATA AD,0B,0C,8D,0C,0C,D0,1D,CE,0C
180 DATA 0C,D0,03,EE,0C,0C,A0,00,A9,01
190 DATA AA,2D,0A,0C,F0,03,20,71,0C,C8
200 DATA C8,0A,D0,F2,AD,0C,0C,8D,02,0C
210 DATA 4C,65,FA,48,AD,01,0C,F0,1B,0A
220 DATA B9,D7,11,D0,0D,CD,08,0C,F0,10
230 DATA 18,69,01,90,08,CD,05,0C,F0,06
240 DATA 38,E9,01,99,D7,11,AD,00,0C,F0
250 DATA 51,0A,B9,D6,11,B0,23,CD,06,0C
260 DATA D0,10,8A,2D,E6,11,F0,02,A9,01
270 DATA CD,07,0C,F0,39,B9,D6,11,18,69
280 DATA 01,99,D6,11,D0,2E,8A,0D,E6,11
290 DATA D0,25,CD,03,0C,D0,10,8A,2D,E6
300 DATA 11,F0,02,A9,01,CD,04,0C,F0,16
310 DATA B9,B6,11,38,E9,01,99,D6,11,C9
320 DATA FF,D0,09,8A,49,FF,2D,E6,11,8D
330 DATA E6,11,68,60,A9,0C,A2,0E,D0,04
340 DATA A9,FA,A2,65,78,8E,14,03,8D,15
350 DATA 03,8D,0D,0C,58,60

```

un joystick e, poiché la routine agisce a livello di interrupt, il movimento è indipendente da quanto sta eseguendo il programma principale: per esempio, è possibile giocherellare con il puntatore mentre sullo schermo scorre l'interminabile elenco di una directory...

Poiché sinora abbiamo parlato di puntatori, bisogna precisare che Pointer consente di governare con il joystick il movimento di un numero di sprite compreso tra 1 e 8: questi possono essere

sovrapposti per dar luogo a figure complesse (ad esempio per imitare la freccetta di Amiga come nel demo che trovate in queste pagine) o essere sparpagliati sullo schermo, ma in questo caso l'utilità di Pointer non sarebbe chiara (ad ogni modo i programmatori più «demenziali» sappiano che è possibile). Trattandosi di sprite, il puntatore può assumere qualunque forma e perciò potete usare Pointer per governare frecce, cursori, mirini, matitine, ico-

Programma dimostrativo.

```

10 POKE 3075.24:POKE 3076.0:REM X MIN
20 POKE 3077.50: REM Y MIN
30 POKE 3078.87:POKE 3079.1:REM X MAX
40 POKE 3080.249:REM Y MAX
50 POKE 3081.2:REM NUM. JOYSTICK
60 POKE 3082.7:REM SPRITE USATI 1-2-3
70 POKE 3083.8:REM RITARDO
80 FOR J=0 TO 191: POKE 3584+J.0:NEXT
90 FOR J=0 TO 31: READ J%: POKE 3584+J.J%:NEXT
100 FOR J=3 TO 25:READ J%:POKE 3640+J.J%:NEXT
110 FOR J=3 TO 28:READ J%:POKE 3712+J.J%:NEXT
120 COLOR 0.7:COLOR 1.2:COLOR 4.7
130 MOVSPR 1.183.149:MOVSPR 2.183.149:MOVSPR 3.183.
149
140 GRAPHIC 1.1
150 SPRITE 1.1.1:SPRITE 2.1.8:SPRITE 3.1.3
160 SYS 3306
170 IF PEEK(3085) THEN 170
180 X1%=RSPPOS(1.0)-24:Y1%=RSPPOS(1.1)+50
190 IF PEEK(3085)=0 THEN 190
200 X2%=RSPPOS(1.0)-24:Y2%=RSPPOS(1.1)-50:BOX 1.X1%
.Y1%.X2%.Y2%:BOX 0.X1%.Y1%.X2%.Y2%:IF PEEK(3085) TH
EN 200
210 BOX 1.X1%.Y1%.Y2%.Y2%
220 X1%=X1%+24:X2%=X2%+24:Y1%=Y1%+50:Y2%=Y2%+50
230 IF X1%>X2% THEN X%=X1%:X1%=X2%:X2%=X%
240 IF Y1%>Y2% THEN Y%=Y1%:Y1%=Y2%:Y2%=Y%
250 POKE 3075.X1%-INT(X1%/256)*256:POKE 3076.INT(X1
%/256):POKE 3077.Y1%
260 POKE 3078.X2%-INT(X2%/256)*256:POKE 3079.INT(X2
%/256):POKE 3080.Y2%
270 MOVSPR 1.X1%.Y1%:MOVSPR 2.X1%.Y1%:MOVSPR 3.X1%.
Y1%
280 IF PEEK(3085)=0 THEN 280
290 IF PEEK(3085) THEN 290
300 SYS 3312
310 GRAPHIC CLR
320 END
330 DATA 252.0.0.130.0.0.130.0
340 DATA 0.132.0.0.130.0.0.145
350 DATA 0.0.104.128.0.4.64.0
360 DATA 2.32.0.1.64.0.0.128
370 DATA 124.0.0.4.0.0.8.0
380 DATA 0.4.0.0.2.0.0.1
390 DATA 0.0.0.128.0.0.64
400 DATA 254.0.0.254.0.0.252.0
410 DATA 0.254.0.0.255.0.0.111
420 DATA 128.0.7.192.0.3.192
430 DATA 0.1.192

```

ne: insomma tutto quello che le necessità dei vostri programmi permettono di rappresentare con degli sprite.

Il movimento che si ottiene con Pointer procede a velocità ridotta quando inizia, accelerando progressivamente per raggiungere, in un tempo programmabile, la velocità di regime di 60 pixel al secondo.

Questo al fine di ottenere, nelle applicazioni che lo richiedono, la massima

precisione di puntamento. Al riguardo è importante sapere che il movimento avviene solo nelle otto direzioni permesse dai comuni joystick.

Per ottenere l'interazione tra il puntatore, governato autonomamente da Pointer, e il vostro programma ci sono diversi modi:

- leggere le coordinate del puntatore con la funzione RSPPOS;
- rilevare le sovrapposizioni del punta-

tore con oggetti sullo schermo con la funzione BUMP;

— leggere lo stato del pulsante di fuoco mediante la funzione PEEK (3085) che restituisce il valore 1 se non è premuto e 0 se è premuto.

Rimane infine da illustrare un'ultima caratteristica di Pointer. Se desiderate che il movimento sia limitato da una determinata area dello schermo (per esempio, la superficie di una finestra), potete vincolare il puntatore indicando le coordinate dell'angolo superiore sinistro e di quello inferiore destro, proprio come fate con l'istruzione WINDOW, ma attenzione: dovete usare coordinate assolute e non valori di riga e di colonna; in pratica, quelle che si usano di solito nell'istruzione MOVSPR. Si noti che se si usa lo stesso valore per l'ascissa minima e massima, si renderanno possibili solo movimenti in verticale, mentre lo stesso valore per le ordinate lascerà libero il puntatore di muoversi solo lungo l'asse orizzontale.

Passiamo adesso alle modalità tecniche di utilizzo dell'utility:

— anzitutto bisogna fissare l'area di movimento indicando il valore minimo della X nelle locazioni 3075-3076, rispettivamente per il low byte e l'high byte, e il valore minimo della Y nella locazione 3077, e i valori massimi, nello stesso modo nelle locazioni 3078-3080;

— quindi scrivete nella locazione 3081 il numero del joystick da considerare (1 o 2);

— adesso usate la locazione 3082 per indicare quali sprite muovere: ogni bit di questo registro indica se uno sprite è da usare (se vale 1) o da non usare (se vale 0). Il bit 0 corrisponde allo sprite 1, il bit 1 allo sprite 2 e così via. In generale per abilitare lo sprite S fate:

POKE 3082, (PEEK (3082) OR 2 ↑(S-1))

— infine «Pokate» nella locazione 3083 il valore che misura il tempo di accelerazione; il valore minimo è 1 (il puntatore andrà subito alla velocità massima), il massimo è 15;

— a questo punto un SYS 3306 abiliterà la gestione del puntatore, mentre un successivo SYS 3312 la disabiliterà.

Il tutto è molto più difficile da spiegare che da farsi comunque se avete dei dubbi date un'occhiata al listato del demo: questo programma illustra tutte le caratteristiche di Pointer; per usarlo posizionare la freccetta dove preferite e premete fuoco, da quel momento potete creare una finestra spostando ulteriormente il puntatore (vedrete che i contorni di questa seguono i movimenti della freccetta), per fissare la finestra premete ancora fuoco.

Spazi 05
Spazi 05
Spazi 05
Spazi 05
Spazi 05
Spazi 05

ad/italia

21-25 OTTOBRE 1988

**FIERA
DI TRIESTE**

QUINTA
RASSEGNA
DELLA RICERCA
E DELLA
TECNOLOGIA

RICERCA, ARTE
DIDATTICA
INFORMATICA

PIAZZALE DE GASPERI, 1
TRIESTE-TEL. (040)39.29.61/2/3
CAS.POST. 1509 - TELEX 460440 FAIRTS I