

## Paroliamo

di Marco Iori - Roma

Il programma, che gira su uno Spectrum 16K RAM, intende riproporre l'affermato gioco televisivo Paroliamo, che pur essendo di sicuro effetto non è il classico Space Battle. Si tratta di battere il proprio avversario (o avversari, dato che si può giocare fino a sei) a suon di parole, il che rende la competizione istruttiva oltre che eccitante.

Il programma si spiega da solo grazie alle numerose REM inserite, e non ha dunque bisogno di troppi commenti. Usa in quantità discreta le operazioni sulle stringhe, e il suo nucleo è organizzato in tre loop nidificati.

Una cosa importante da notare è che il controllo effettuato sulle parole non riguarda l'esistenza del termine in italiano ma solo l'effettiva appartenenza di tutte le lettere usate dal giocatore a quelle estratte dal computer: questo comporta che il rispetto della lingua venga affidato ai giocatori, e in caso di dubbio al vocabolario del quale si saranno equipaggiati.

Che Paroliamo non sia Space Battle, oltre a farci piacere, ce ne consente la pubblicazione senza i tradizionali rimorsi del redattore che non voglia incanalarsi in un filone sfruttatissimo (ormai in agonia) come quello degli invasori. D'altronde ben vengano giochi di società sul computer, il cui uso intanto permette di assicurarsi imparzialità, e poi evita la perdita dei vari attrezzi del gioco — per quello presentato stavolta le carte con le lettere, ad esempio.

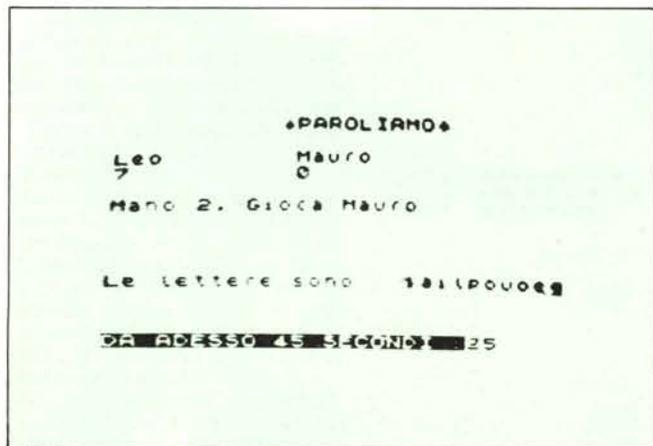
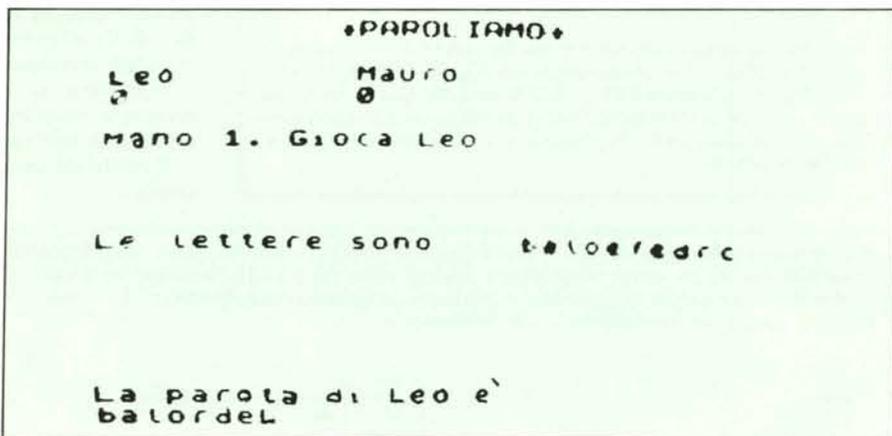
Passiamo al gioco. Come il lettore noterà, abbiamo sfrondato le due schermate di istruzioni, consci della popolarità del pasatempo ma anche della naturale repulsione per la digitazione di troppe chiacchiere. Per chi non conoscesse il gioco di Tele Montecarlo, ecco un sunto delle regole più importanti.

Si tratta, date dieci lettere a caso (nella fattispecie estratte dal computer tramite il ciclo 230-310, annunciato dalla REM in 229), di comporre entro un tempo prefissato (qui di 45 secondi) la parola italiana più lunga possibile. Due o più giocatori, fino a sei nel caso nostro, si alternano alla composizione dell'insieme di lettere su cui agire: il controllo consentito è, di volta in volta, la scelta vocale-consonante. Giunti a dieci elementi inizia il conteggio del tempo, al termine del quale ognuno deve pre-

sentare la propria parola ed inserirla nel programma. Per evitare che il secondo concorrente dichiari la stessa parola dell'avversario se è più lunga della sua consigliamo di scrivere volta per volta le parole su un foglietto (ma forse non tutti sono malfidati come noi...).

Poi il programma presenta una scelta discutibile, anche se allo scopo di semplificare la vita. Chiunque abbia scelto le lettere, l'ordine di digitazione delle proprie parole è sempre lo stesso con cui i nomi dei concorrenti sono stati inizialmente inseriti, e ciò — come verificherete da soli risulta un po' scomodo. Tra l'altro, come citato dall'autore, nessuna verifica viene eseguita sulla correttezza della parola, per cui fate attenzione e munitevi di un opportuno vocabolario.

Questa zona del programma è quella che più si presta a modifiche. Chi lo desiderasse potrà appesantirla con routine grafiche di presentazione, eventuali sottoprogrammi — assai onerosi — di segnalazione di errori nella parola (attenzione, non di esistenza della stessa) o soluzioni alternative all'inserimento: una soluzione al problemi-



Una schermata del gioco proposto. La presentazione grafica è essenziale, e ciò contribuisce parecchio alla linearità del listato: sarà così piuttosto agevole per chiacchierata modificare la veste estetica del Paroliamo in modo personale.

Aggiungiamo una cosa non citata nel testo dell'articolo. Chi vince se tutti i concorrenti mettono parole sbagliate (nel senso che comprendono lettere non estratte dal computer) ovvero parole nulle (praticamente digitando direttamente l'ENTER)? Dato che in entrambi i casi viene assegnato il punteggio zero — quindi il giocatore non viene squalificato, come sarebbe più plausibile — il vincitore è il concorrente di mano. Che poi sia una vincita platonica, visto che il punteggio è pari al numero di lettere della parola vincente, ciò poco importa ai decubertiniani lettori di MCmicrocomputer ...

### Variabili usate nel programma

- r\$, contiene gli input letterali
- ng, è il numero di giocatori
- i,j,y, sono contatori d'anello nei cicli secondari
- m, è la mano di gioco attuale e il contatore dell'anello più esterno
- g, è il giocatore di mano e il contatore dell'anello intermedio
- f, indica i giocatori durante gli input, nel controllo di correttezza delle parole e nell'assegnazione di vittoria
- a\$, contiene le 21 lettere dell'alfabeto italiano
- b\$, contiene le lettere via via estratte
- cr cc, sono i contatori di riga e di colonna durante la stampa dei punteggi
- v\$, contiene il nome del vincitore del gioco
- h, variabile di comodo per assegnare il vincitore del gioco
- t\$, variabile di comodo nella verifica della parola in input
- c, conta le lettere presenti nella parola in input
- e\$, 32 blank per cancellare una riga
- v, punti dei giocatori durante la classifica

### Matrici usate

- n\$(ng,10), contiene i nomi dei giocatori — lunghi al massimo 10 lettere
- f\$(ng,10), contiene le parole dei giocatori
- l(ng), è la lunghezza delle parole dei giocatori
- s(ng), contiene il punteggio dei giocatori

no di fiducia sollevato pocanzi potrebbe essere la sostituzione dell'attuale routine 360-587 più sub 3500-3540 con un'altra che renda manifeste le varie parole solo quando tutti i giocatori avranno digitato i rispettivi termini.

L'autore della trovata più lunga (ancorché corretta) si vedrà assegnare tanti punti quanti sono i caratteri utilizzati: le mani

del gioco sono il doppio del numero dei giocatori, come indicato dal massimo valore assunto dalla variabile di ciclo *m* inizializzata in linea 160.

Il listato è sufficientemente leggibile e scorrevole: l'autore non ha complicato la semplice organizzazione del gioco, riconducendo alle capacità dei giocatori tutti quei controlli impossibili da farsi a questo

livello. Le documentazioni tramite REM sono in numero adeguato, ma forse leggermente troppo stringate per la comprensione da parte di successivi programmatori.

Sarebbe interessante vedere una versione computer di Scarabeo, un po' più lungo da programmare ma non certo più difficile. Se son rose..., e mandate anche la casetta!

MC

```

20 PRINT AT 12,11; FLASH 1;"PAROLIAMO": PAUSE 100
69 REM *INTESTAZIONE*
70 CLS : PRINT AT 0,10;"*PAROLIAMO*"
79 REM *QUANTI GIOCATORI*
80 IF INKEY$("<") THEN GO TO 80
90 INPUT "Quanti siete a giocare (max 6)? ";ng
95 IF ng>6 THEN GO TO 90
99 REM *DIM MATRICI*
100 DIM n$(ng,10): DIM f$(ng,10): DIM l(ng): DIM s(ng)
109 REM *NOMI GIOCATORI*
110 FOR i=1 TO ng
120 INPUT "Come si chiama il giocatore ";i);" ?" LINE r$
130 LET n$(i)=r$
140 NEXT i
149 REM *LOOP DI INGRESSO*
150 LET a$="aeioubcdfghilmnpqrstvz"
160 FOR m=1 TO ng*2: FOR g=1 TO ng
165 LET b$=""
169 REM *STAMPA PUNTEGGI*
170 LET cr=2: LET cc=0: LET i=1
180 PRINT AT cr,cc;n$(i);AT cr+1,cc;s(i): LET i=i+1
190 IF i>ng THEN GO TO 220
195 LET cc=cc+11
200 IF cc>22 THEN LET cr=cr+2: LET cc=0
210 GO TO 180
219 REM *GIOCATORE DI MANO*
220 PRINT : PRINT "Mano ";m;" Gioca ";n$(g)
229 REM *LETTERE*
230 FOR i=1 TO 10
235 PRINT AT 10,0;"Le lettere sono : "; BRIGHT 1;b$
240 PRINT AT 12,0;"Cosa vuoi (Cons./Voc.)?"
250 IF INKEY$("<") THEN GO TO 250
260 IF INKEY$="" THEN GO TO 260
270 LET r$=INKEY$
280 IF r$="v" OR r$="V" THEN LET b$=b$+a$(INT(1+RND*5)): GO TO 310
290 IF r$="c" OR r$="C" THEN LET b$=b$+a$(INT(6+RND*16)): GO TO 310
300 GO TO 250
310 NEXT i: PRINT AT 10,18; BRIGHT 1;b$
320 PRINT AT 12,0;"
329 REM *45 SECONDI*
330 PRINT AT 14,0; INVERSE 1;"D'ADDESSO 45 SECONDI .". BEEP 1,1
340 FOR i=0 TO 45: PRINT AT 14,22;i: BEEP .25,0: PAUSE 36: NEXT i
350 BEEP 3,10: PRINT AT 14,0;"FINE TEMPO": PAUSE
150
355 PRINT AT 14,10;"
359 REM *INGRESSO PAROLE*
360 FOR f=1 TO ng
380 GO SUB 3500
420 INPUT "La parola di ";(f$);" e' : " LINE r$
430 LET f$(f)=r$: LET t$=r$
439 REM *PAROLA CORRETTA?
440 LET c=0
450 IF r$="" THEN GO TO 520
460 FOR j=1 TO 10
470 FOR y=1 TO LEN r$
480 IF b$(j)=r$(y) THEN LET c=c+1: LET r$=r$(1 TO y-1)+r$(y+1 TO LEN r$): LET y=10
490 NEXT y: NEXT j
499 REM *PAROLA ERRATA*
500 IF LEN t$("<")c THEN PRINT AT 14,10; INVERSE 1;"NON VA BENE": PAUSE 100: PRINT AT 14,10;"": LET l(f)=0: GO TO 530
509 REM *PAROLA CORRETTA*
510 PRINT AT 14,12;"VA BENE": PAUSE 100: PRINT AT 14,12;"
520 LET l(f)=c
530 NEXT f
539 REM *VINCITORE GIOCO*
540 LET h=g
545 LET v$=n$(h)
550 IF g=ng THEN GO TO 580
560 FOR f=g+1 TO ng
570 IF l(f)>l(h) THEN LET v$=n$(f): LET h=f
580 NEXT f
583 FOR f=1 TO g-1
585 IF l(f)>l(h) THEN LET v$=n$(f): LET h=f
587 NEXT f
590 PRINT AT 14,0;"In questo gioco vince ";v$: PRINT : PRINT "on la parola ";f$(h): PRINT : PRINT "Lunga ";l(h);" lettere.": PAUSE 500
600 PRINT AT 10,18;"
610 LET c$=""
620 PRINT AT 14,0;c$: PRINT : PRINT c$: PRINT : PRINT c$
630 LET s(h)=s(h)+l(h)
640 NEXT g: NEXT m: REM *PROSSIMO GIOCO E MANO*
649 REM *FINE E VITTORIA*
650 CLS : PRINT AT 0,10;"*PAROLIAMO*"
660 PRINT AT 2,2;"PROCLAMAZIONE DEL VINCITORE"
670 PRINT AT 6,0;"Vince ";
680 FOR y=1 TO ng
690 GO SUB 3000
700 PRINT r$;" con punti ";v
710 LET s(f)=0
719 REM *CLASSIFICA*
720 IF y<ng THEN PRINT : PRINT "Seguito da ";
730 NEXT y
740 PRINT AT 20,13; FLASH 1;"FINE": PAUSE 300: STOP
1000 CLS
1070 PRINT : PRINT TAB 1;"PREMI UN TASTO PER CONTINUARE"
2999 REM *TROVA IL PIU' GRANDE*
3000 LET f=1: LET v=s(f)
3010 FOR j=1 TO ng
3020 IF s(j)>v THEN LET v=s(j): LET f=j
3030 NEXT j
3499 REM *ELIMINA SPAZI INUTILI*
3500 FOR i=10 TO 1 STEP -1
3510 IF n$(f,i)("<") THEN GO TO 3530
3520 NEXT i
3530 LET r$=n$(f, TO i)
3540 RETURN

```