

AmigaBasic: menu & interrupt

di Andrea de Prisco

■ *Sempre a proposito dell'AmigaBasic, questo mese parleremo dei menu a discesa e delle interruzioni. Coi primi sarà possibile "abbellire" i propri programmi dotandoli, con poco sforzo, di una interfaccia intuitiva e, tramite mouse, anche molto veloce. Col meccanismo delle interruzioni potremo controllare facilmente, a programma, alcuni eventi esterni. Il motivo per cui trattiamo insieme i due argomenti è presto detto: uno degli eventi gestibili come interrupt è appunto la scelta da menu a discesa...* ■

Prologo

Alcune settimane fa, sono arrivate in redazione un paio di "amighevoli" lettere nelle quali i rispettivi autori contestavano il fatto che su MCmicro-computer si spreca spazio editoriale per parlare dell'AmigaBasic, secondo loro, ampiamente trattato già sul manuale fornito con la macchina.

A dire il vero, qualche mese fa anche il sottoscritto la pensava alla stessa maniera, e il ciclo di articoli relativi all'AmigaBasic ha preso vita solo "a gentile richiesta" di numerosi lettori che si trovavano in difficoltà con questo linguaggio. Preso poi seriamente in mano il manuale, bisogna riconoscere che alcune cose sono spiegate particolarmente male e si riesce a venirne a capo solo dopo svariati tentativi falliti...

Menu a discesa

I menu a discesa definibili da AmigaBasic funzionano esattamente come

MENU 1,0,1,"Menu uno "	MENU 1,0,1,"Menu uno "
MENU 1,1,1,"scelta 1 "	MENU 1,1,1,"scelta 1 "
MENU 1,2,1,"scelta 2 "	MENU 1,2,1,"scelta 2 "
MENU 1,3,1,"scelta 3 "	MENU 1,3,1,"scelta 3 "
MENU 2,0,1,"Menu due "	MENU 2,0,1,"Menu due "
MENU 2,1,1,"scelta 1 "	MENU 2,1,1,"scelta 1 "
MENU 2,2,1,"scelta 2 "	MENU 2,2,1,"scelta 2 "
MENU 2,3,1,"scelta 3 "	MENU 2,3,1,"scelta 3 "
MENU 3,0,0,""	MENU 3,0,0,""
MENU 4,0,0,""	MENU 4,0,0,""
WHILE 1	ON MENU GOSUB Scelta
I=I+1	MENU ON
PRINT I	WHILE 1
m=MENU(0)	I=I+1
IF m<>0 THEN GOSUB scelta	PRINT I
WEND	WEND
scelta:	Scelta:
PRINT "menu";m;	PRINT "menu";MENU(0);
PRINT "scelta";MENU(1)	PRINT "scelta";MENU(1)
RETURN	RETURN
Listato 1	Listato 2

quelli del Workbench. Premendo il tasto destro del mouse facciamo apparire la barra superiore contenente i nomi dei singoli menu, in questa condizione, pertanto il mouse su uno di questi nomi vedremo le varie opzioni che abbiamo programmato. Come accade nel Workbench, alcune di queste (o un intero menu) potranno essere non attive, nel tal caso appariranno un po' meno leggibili.

L'istruzione che permette la creazione di un nuovo menu è MENU, nella sintassi:

MENU numero,item,stato,stringa

dove "numero" è il numero del menu (si conta da sinistra verso destra cominciando da 1); "item" è la posizione nel menu appena indicato (la posizione 0 è il nome del menu); "stato" indica appunto se l'item o l'intero menu è

che restituiscono rispettivamente il numero del menu e (all'interno di questo) il numero dell'item selezionato. Quando l'operatore non sta selezionando niente, MENU(0) vale 0 dunque è possibile eseguire dei cicli di attesa finché... non succede qualcosa.

Il listato 1 è una prima, banale, applicazione dei comandi testé discussi. Come è facile notare, la prima parte del listato serve per definire due menu a discesa. Seguono due istruzioni per far "scompare" i menu "Run" e "Window" del Basic. Nel ciclo infinito While...Wend dopo aver stampato su video la variabile I, incrementata ogni volta, viene interrogata la funzione MENU(0). Si usa la variabile temporanea "m" essendo interessati a congelare la lettura (riutilizzeremo tale valore all'interno della subroutine). Così, se MENU(0) è diverso da zero saltiamo alla subroutine che non fa altro

ginate un normale programma in esecuzione. Cosa stia facendo in quel momento il programma a noi non interessa: diciamo che "gira" e basta. L'interruzione più banale è, non a caso, il comando di stop del programma: quando "questo meno se l'aspetta" un bel Control C fa terminare anzitempo l'esecuzione. Da sottolineare l'asincronicità dell'evento, occorso non in un particolare momento dell'esecuzione ma in un istante qualsiasi (traduzione: ricominciando tutto daccapo e dando nuovamente il Control C, in generale, fermeremo l'esecuzione del programma in un punto differente dal precedente).

Questo come caso banale. Molto probabilmente, infatti, non siamo interessati solo ad interruzioni "totali", ma anche a interruzioni meno "dolorose": al verificarsi di un determinato evento, desideriamo che sia eseguita una particolare procedura detta appunto di manipolazione dell'interruzione. Terminata tale procedura, il corso dell'elaborazione dovrà continuare dallo stesso punto dove era stato interrotto, in un certo senso come se nulla fosse successo, a meno che l'interruzione (o meglio, la routine di manipolazione dell'interruzione) non abbia provocato essa stessa un cambiamento di stato.

Le interruzioni controllabili da AmigaBasic sono in tutto 5 e riguardano la scelta da menu, la pressione del tasto sinistro del mouse, la collisione tra sprite (che non tratteremo in questa sede), il già citato Control C (o stop da menu) e il timer interno col quale è possibile implementare un mini multitasking all'interno di un programma Basic.

Per abilitare il meccanismo di rilevamento e trattamento delle interruzioni si definisce inizialmente il "da farsi" in caso di interrupt. Quando, in un certo senso, siamo pronti a trattare gli interrupt, nel nostro programma utilizzeremo il comando di abilitazione vera e propria.

Per definire il "da farsi" si usa il comando:

ON <evento> GOSUB <subroutine>

che sta proprio ad indicare: in caso di <evento> devi saltare a <subroutine>. Per abilitare il rivelamento useremo il comando:

<evento> ON

per disabilitare:

<evento> OFF

```
ON BREAK GOSUB stampa
BREAK ON
WHILE 1
  i=i+1
  PRINT i
WEND

stampa:
  PRINT "non mi fermi!"
RETURN
```

Listato 3

```
ON MOUSE GOSUB leggi
MOUSE ON
WHILE 1
  i=i+1
  PRINT i
WEND

leggi:
  a=MOUSE(0)
  PRINT "posizione";MOUSE(1);MOUSE(2)
RETURN
```

Listato 4

attivo (1) o non attivo (0); "stringa", infine, è la stringa da associare a quella determinata posizione del menu, o il titolo del menu stesso.

Per cambiare stato ad un menu già definito indicheremo solo i primi tre valori. Ad esempio, per disabilitare la terza scelta del secondo menu (precedentemente definito) scriveremo:

MENU 2,3,0

Ricordando che lo schermo di default è il normale schermo di output dell'AmigaBasic se dopo aver giocato un po' coi menu vogliamo ritornare alle condizioni iniziali, digiteremo:

MENU RESET

Le funzioni di lettura

Per conoscere, da programma, cosa ha selezionato l'operatore esistono due funzioni, MENU(0) e MENU(1),

che stampare, utilizzando anche MENU(1), la scelta effettuata. L'effetto finale sarà dunque uno "scrollare" di numeri via via crescenti intervallati dalle nostre scelte se utilizzeremo i menu a discesa. Non serve assolutamente a niente ma funziona...

Le interruzioni

Giacché siamo in tema di menu, e come abbiamo detto la scelta da menu può essere trattata come un interrupt, entriamo immediatamente in argomento modificando il programmino 1 secondo il nuovo schema.

Il tutto è mostrato nel listato 2: le modifiche sono minime, ma rendono tutto il listato notevolmente più leggibile. Specialmente una volta chiarito il significato dei comandi ON MENU e MENU ON...

A proposito, siamo sicuri che tutti sanno cos'è una interruzione? Imma-

<evento> potrà essere una delle quattro parole chiave: MENU, MOUSE, TIMER e BREAK, mentre <subroutine> è il nome o il numero linea di una subroutine posta in qualsiasi punto del programma e come tale (mi raccomando) dotata di RETURN finale. Nella subroutine in questione scriveremo le linee di programma che implementano il vero e proprio trattamento dell'interruzione.

Tornando al listato 2 possiamo ora interpretare correttamente il significato di quelle linee di programma. Con:

```
ON MENU GOSUB Scelta
```

diciamo che (quando avremo dato l'ordine di "ascoltare" gli interrupt) ad ogni accesso ai menu a discesa dovremo saltare alla subroutine "Scelta". Con:

```
MENU ON
```

abilitiamo il rilevamento degli interrupt. Si noti come il ciclo infinito While...Wend questa volta si preoccupi solo di stampare i numerini via via crescenti: è il sistema che provvederà automaticamente a fermare la stampa in caso di selezione da menu (deve infatti essere eseguita la subroutine "Scelta").

Control C

Che il Control C sia un interrupt l'abbiamo già appurato un po' di righe fa. Da AmigaBasic è possibile controllarlo e trattarlo come qualsiasi altra interruzione. Il listato 3 ne è un esempio: l'evento si chiama BREAK, e come prima, usiamo un comando per definire dove saltare in caso di interrupt:

```
ON BREAK GOSUB stampa
```

e un secondo comando per abilitare il rilevamento:

```
BREAK ON
```

Il risultato di tutto il programma sarà il solito scrollare di numeri, intervallati da un messaggio all'operatore quando questo tenta di fermarne l'esecuzione. Per la cronaca, l'unico modo per fermarlo sarà chiudere tutte le finestre relative al lancio dell'AmigaBasic.

Mouse e affini

Il terzo tipo di interrupt riguarda la pressione del tasto sinistro del mouse. Grazie a questa possibilità diventa molto facile utilizzare l'interfaccia

mouse & pointer anche per i nostri programmi. Ad esempio selezionare col topo alcuni oggetti sullo schermo, effettuare spostamenti (sì, pensate pure ad una partita a scacchi) o altro. Il metodo è sempre lo stesso: scriviamo da qualche parte nel nostro programma la subroutine, che dovrà trattare l'interruzione da mouse, e con i due comandi, di dichiarazione e di attivazione, diamo inizio al "trattamento".

Come per i menu, esiste una funzione MOUSE() che, a seconda dell'argomento passato, restituisce lo stato del dispositivo. Senza scendere troppo nei dettagli, che comunque troverete nel manuale fornito con la macchina, diremo che MOUSE(0) è usato per conoscere lo stato del pulsante sinistro, MOUSE(1) e MOUSE(2) la posizione attuale del mouse, MOUSE(3) e MOUSE(4) la posizione iniziale di un movimento, MOUSE(5) e MOUSE(6)

quella finale. Tutti questi valori sono aggiornati solo dopo ogni lettura di MOUSE(0), sarà quindi questa la prima operazione da compiere prima di leggere le varie coordinate.

Il programma di listato 4 è una prima applicazione: dando RUN a questo vedremo i soliti numerini che scrollano, intervallati dalla posizione attuale del mouse ogni volta che premiamo il tasto sinistro del mouse che provoca l'interrupt. Le coordinate del mouse sono sempre espresse in pixel, con punto d'origine nell'angolo superiore sinistro dello schermo. Da notare la prima linea delle subroutine "leggi":

```
a=MOUSE(0)
```

indispensabile per leggere correttamente le coordinate alla linea successiva. Il listato 5 mostra un'applicazione dei rimanenti valori passati alla funzione MOUSE. Questa volta, tenendo premuto il pulsante, tracciando col mouse un percorso qualsiasi, il programma traccierà una linea retta tra i punti iniziali e finali del movimento.

Timer

L'ultima interruzione che tratteremo riguarda il timer interno all'Amiga. Da AmigaBasic potremo programmare il timer affinché ci invii interrupt ad intervalli di tempo prefissati. L'indicazione del quanto di tempo, espresso in secondi, avviene nello stesso momento in cui dichiariamo di voler utilizzare il timer. Nel listato 6, ad esempio, il timer è settato per intervalli pari ad un secondo, nel listato 7 l'intervallo tra un interrupt e l'altro è fissato pari a mezzo secondo. Nel listato 6 i soliti numerini saranno intervallati dall'indicazione "interrupt!" ogni secondo, nel listato 7 abbiamo inserito l'emissione di una nota random (prendano "nota" coloro i quali sono interessati a inserire musicchette nei loro giochi) ogni mezzo secondo.

Una volta dato il RUN ad uno di questi due programmi, provando a cambiare la dimensione della finestra in cui avviene lo scroll dei numerini noterete, come era prevedibile, che le interruzioni non dipendono dal numero di numerini stampati (come noto con finestre più piccole lo scroll avviene più velocemente e quindi anche i programmi vanno più veloci quando devono stampare qualcosa), ma solo e soltanto dal tempo.

Grazie e arrivederci.

```
ON MOUSE GOSUB traccia
MOUSE ON
WHILE 1 : WEND

traccia:

a=MOUSE(0)
x1=MOUSE(3):y1=MOUSE(4)
WHILE MOUSE(0)--1 : WEND
x2=MOUSE(5):y2=MOUSE(6)
LINE (x1,y1)-(x2,y2)
```

```
RETURN
```

Listato 5

```
ON TIMER(1) GOSUB stampa
TIMER ON
WHILE 1
i=i+1
PRINT i
WEND
```

```
stampa:
PRINT "interrupt!"
RETURN
```

Listato 6

```
ON TIMER(.5) GOSUB musica
TIMER ON
WAVE 1,SIN
WHILE 1
i=i+1
PRINT i
WEND

musica:
PRINT "musica!"
SOUND RND(1)*4000+50,2,100,1
RETURN
```

Listato 7