

# Microsoft Basic 3.0

*Parliamo ancora per una volta di Basic, la lingua inglese della programmazione. Niente di strano in ciò, visto che le prime donne meritano sempre attenzione e riguardi. E, dopo la visita di dovere a J.G. Kenemy e T.E. Kurtz, numi tutelari del Basic, ecco la visita d'obbligo in real casa Microsoft, a vedere da vicino l'ultima versione di questo linguaggio, la 3.0, che la pubblicità roboante ed un po' kitch delle riviste statunitensi rivela come la «most», il tutto seguito da decine di aggettivi più o meno incensieri. Vediamo cosa è cambiato in questa versione, e vediamo se queste modifiche hanno sanato qualche piccolo problema che la versione 2.1 aveva lasciato aperto.*

*D'altro canto da Microsoft non ci si può aspettare meno che il massimo, visto a che cosa ci ha abituato con prodotti come Word 3 e MS File (peccato che questo pacchetto, peraltro già nato adulto, si sia fermato nel suo sviluppo, lasciandosi sopravanzare da più agguerriti e moderni DB). A curiosità verso questa nuova release ci spingeva anche il prezzo, particolarmente basso, e più che allineato con la più recente tendenza alla riduzione dei prezzi (leggi Borland). In ogni caso MS Basic, che è stato all'inizio il solo tool di sviluppo, in questo idioma (ricordate la rugginosa e deboluccia release 1.0, ancora con i numeri di linea?), dimostra, dopo tanto tempo, di sapersi ringiovanire continuamente; vediamo fin dove ci è riuscito, anche con la concorrenza che ormai è scesa in campo, e che conta avversari agguerriti come il True visto l'altra volta, e come lo Z Basic, macchina quasi perfetta.*

## Il MS Basic

Una gran messe di persone usa il linguaggio di programmazione Basic per differenti ragioni. Alcune di esse non

sono programmatori di professione, ma utilizzano programmi redatti da altri; ma la maggior parte degli utilizzatori acquista questo linguaggio per utilizzarlo per produrre programmi di proprio disegno; accanto ad essi esistono ancora numerosi studenti, che utilizzano la loro macchina per affrontare problemi scolastici. Gran parte dei programmi verticali in circolazione devono la loro essenza a questo idioma, anche se, non si sa per quale ragione (probabilmente per puro snobismo) nessuno oserebbe professarsi sostenitore di questo linguaggio in una riunione di programmatori «seri» (anche se probabilmente molti di loro celano sotto le spoglie di un compilatore molti listatoni in Basic). Eppure basta accostarsi a linguaggi ben organizzati, come questo MS 3, per rendersi conto di quanto sia stupida questa remora, e che questi laudatores di esoterici idiomi sono in palese mala fede (o forse soffrono di un complesso di superiorità, o come si suol dire dalle mie parti, di puzza al naso).

Lasciamo perdere questi ipocriti e tiriamo avanti, confortati dal fatto che, ormai, anche grosse macchine di case prestigiose, come la serie 1000 e 9000 dell'HP, tanto per dirne una, cominciamo a presentare le loro creature corredate del loro bel Basic (ve lo immaginate cosa è capace di fare un linguaggio, che occupa, anche se in segmenti, quasi un mega di RAM; probabilmente basta parlare per farsi ubbidire).

Microsoft Basic per il Macintosh è un Basic molto standard, vale a dire che rispetta lo standard Microsoft, in modo tale che programmi scritti con altre macchine sotto le stesse condizioni (ad esempio redatti in Basic su un IBM) possono girare con pochissime modifiche anche su Mac. Ciononostante la particolare filosofia di Macintosh non poteva essere mortificata, e MS Basic 3.0 supporta tutti i tool e le feature che





hanno reso questa macchina così potente, diversa, ed amichevole verso l'utente.

Attraverso 3.0 è possibile redigere applicazioni particolarmente potenti, orientate soprattutto sull'uso del mouse. Una funzione specifica, MOUSE, consente al programma di interagire con l'utente attraverso l'input da mouse, appunto; inoltre, con estrema facilità, è possibile redigere ed accedere a menu a tendina, rendere tutto ancor più interattivo attraverso finestre di dialogo, utilizzare grafica estremamente potente ed avanzata, ecc.

3.0 (da questo momento consentiteci di chiamarlo così) si presenta, come già la versione 2, articolata in due versioni poco diverse dal punto di vista dell'utilizzo, ma dalla differente tipologia operativa. La differenza sta, soprattutto, nelle modalità di utilizzo ed articolazione dell'aritmetica in virgola mobile. È possibile utilizzare la versione decimale, meglio compatibile con problemi di tipo finanziario, data l'assenza d'arrotondamento della rappresentazione BCD (questa versione accetta, inoltre, tutto il software sviluppato nella versione 1), o la versione IEEE, meglio utilizzabile in applicazioni scientifiche, dove la precisione di certi sottomultipli può essere sacrificata alla velocità d'insieme, velocità che può anche notevolmente rallentarsi utilizzando certe funzioni trascendenti. Nella maggior parte dei casi, comunque, non esiste differenza rilevante tra le due realizzazioni: in caso di lancio di programma già esistente, il Finder si incaricherà di caricare automaticamente la versione del Basic idonea. Anche i file sono generalmente leggibili indifferente dalle due versioni, tranne che per certi campi numerici formattati in condizioni particolari; inoltre se lo stesso programma viene lanciato sotto differenti versioni, i risultati numerici ottenuti possono essere un po' diversi; ma la differenza è, sovente, insignificante.

### Le modalità operative del programma

MS Basic è formato da due dischetti e da due manuali; essi rappresentano a tutti gli effetti un efficace tutorial sia del linguaggio Basic che della particolare implementazione. Ciononostante, con una tecnica finora non vista in altri manuali Microsoft, e comunque come accadeva con i precedenti manuali di questo linguaggio, il manuale di riferimento dell'interprete non può essere compreso, in tutta la sua interezza, da un utente sprovvisto. Molte nozioni sono sottintese, e sovente anche un programmatore esperto si trova in difficoltà

ad intendere certi concetti (un esempio è il capitolo che riguarda la manipolazione del Clipboard File o delle Picture, da cui non è facile venir fuori con le idee chiare). Al contrario abbiamo trovato molto esauriente la trattazione dei file, che, secondo una tecnica propria della Microsoft, ed ormai divenuta uno standard de facto, adotta la tecnica di «seraggio delle file» con i comandi LSET e simili. Un capitolo intero è destinato alla compatibilità ed alla trasmissione dati tra Basic ed altri programmi, come Multiplan, Mac Paint, la maggior parte dei word processor.

Numerose pagine introduttive sono dedicate alla illustrazione di una delle feature più utili e potenti dell'idioma; i sottoprogrammi, che, all'atto pratico, sono in tutto e per tutto simili alle procedure pascaliane. Questa struttura (che comunque, non ha subito alcuna variazione rispetto alla versione 2.0) appare dotata delle caratteristiche più avanzate, come variabili locali e globali, passaggio di parametri, ecc. Inoltre, cosa molto pratica e di notevole effetto per quanto attiene alla chiarezza, i sotto programmi, così come le procedure in

Pascal e le Funzioni in «C» possono essere chiamate per nome e, ancora meglio, direttamente da tastiera, in modo immediato, indipendente dal programma che le utilizza, purché siano, ovviamente, presenti nel programma esistente in memoria.

Buona parte del manuale di riferimento è, come in tutti i casi del genere, rappresentato da un indice dei comandi e funzioni disponibili (circa 200): notiamo, tra l'altro, ancora una volta la fastidiosa mancanza di possibilità di lavorare direttamente in gradi (ammessi solo i radianti), mentre viene ampiamente risolta l'esigenza di realizzare programmi secondo il particolare look Mac; comandi come BUTTON, MENU, MOUSE, FILE\$, DIALOG, DEF, LPAGE, DEF LPRINT, WINDOW, ecc. consentono di realizzare con facilità programmi dotati di quelle caratteristiche «inventate» da Mac. Tutte cose anche queste, comunque, che avevamo già visto nella edizione 2. Il manuale continua con numerose appendici: la classica tabella ASCII, una tavola di riferimento degli errori, una lista di comandi, alcune note tecniche sulle modalità di rappresentazione inter-

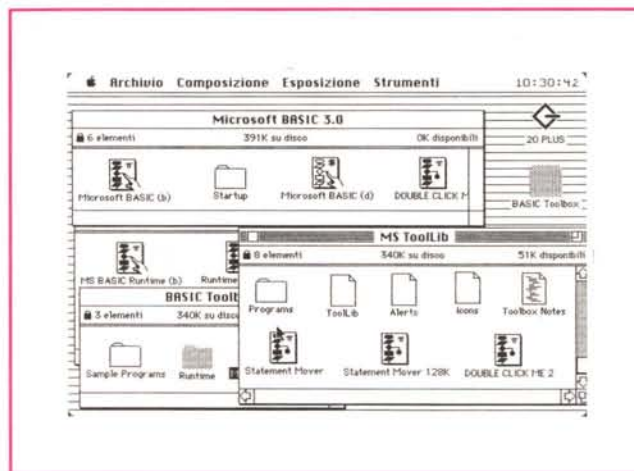
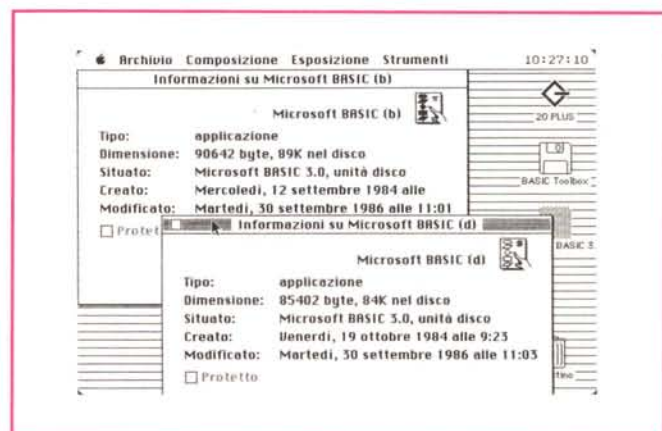


Figura A  
Il contenuto dei due dischetti, forniti col linguaggio.

Figura B  
Le due versioni del linguaggio; sopra la binaria, sotto la BCD.



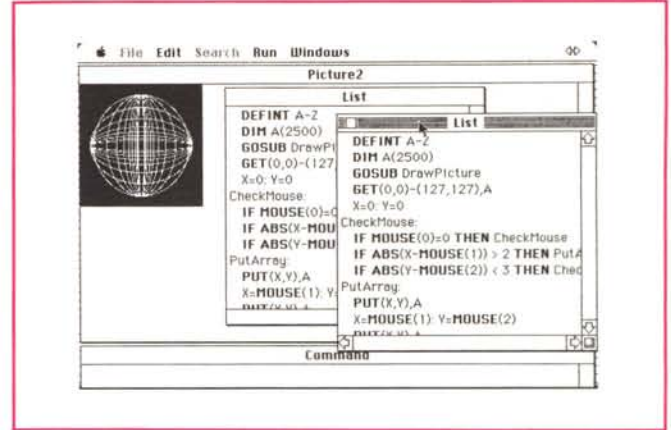


na e sulla possibilità di ricavare funzioni matematiche particolari, non previste dal linguaggio. L'appendice F parla dell'accesso al toolbox, (anche qui tutto identico alla vers. 2), con le CALL alla formattazione del testo, alle caratteristiche della penna e delle linee, ai retini di disegno, ai poligoni (in verità descritti in maniera abbastanza veloce ed oscura), alle routine di manipolazione del mouse. Il tutto sa di affrettato e, oltre tutto, non è neanche supportato da esempi. In poche parole ci è parso che, almeno in questo, più che di versione 3.0 si sarebbe potuto parlare, al massimo di versione 2.2 (l'ultima vista in circolazione, la 2.1, risale a circa due anni fa). Potremmo sbagliarci, ma si tratta dello stesso manuale, utilizzato nella precedente versione, tal quale. Fin qui, quindi, almeno nella documentazione, poche le differenze.

**Il toolbox reference manual**

Quanto abbiamo detto finora, comunque, è ben poco. Sta di fatto che MS Basic è accompagnato da un manuale aggiuntivo di circa 80 pagine dedicato

Figura C  
La classica scrivania di editing di MS Basic; visibili le finestre di RUN, COMMAND e le due di listing.



ad una collezione di routine di programmazione, incluse nel secondo dischetto fornito. Si tratta di un potente blocco di librerie realizzate dalla Clear Lake (5615 Morningside #127, Houston, Texas) e destinate a consentire un più facile lavoro programmatore. Si può guardare ad una libreria come ad una raccolta di subroutine, del tutto simili a quelle scritte in Basic, con la sola differenza che

esse sono redatte in linguaggio macchina e risiedono esternamente al linguaggio stesso. Ogni routine nella libreria ha un nome, e invocata per nome nel programma, entra automaticamente in azione. Tutto ciò ha l'effetto di estendere il linguaggio Basic a livelli nettamente superiori (si pensi che le librerie fornite aggiungono circa 150 nuovi statement al linguaggio, e che la Clear Lake, in fondo al manuale, offre altre 3 librerie acquistabili a parte).

Vediamo in cosa consistono queste librerie. Chi pensasse di avere, con esse, acquistato nuovi statement estranei alla filosofia Mac (come ad esempio la tanto desiderata opzione DEG) si sbaglia. Queste librerie (e d'altro canto è più che giusto che sia così), sono destinate all'apertura ed alla piena utilizzazione del toolbox, con mezzi e facilità maggiori della manipolazione di VARPTR% e simili diavolerie. Il pacchetto si presenta presentato da un demo di grande efficacia in cui appare la potenza delle utility (lo ripetiamo, si raddoppiano praticamente le istruzioni di base del linguaggio); attraverso questo programma, che è possibile esplorare alla ricerca di routine di gran pregio, è visibile come definire macro con combinazioni di tasti tipo Command-Key, utilizzare gli scroll bar, analizzare figure, pattern, bottoni di «Alert», «clickare» e «dragare» parti dello schermo, cambiare forma al cursore. Il package viene fornito in un unico blocco-libreria, ma, per evitare di caricarsi di routine inutili, vengono forniti due programmi aggiuntivi, gli Statement Mover, destinati a macchine da 128K e 512K, che consentono di definire librerie più piccole, personalizzate, includenti solo i tool necessari, con evidente risparmio di tempo e memoria.

La libreria viene inclusa attraverso l'istruzione:

Library "nome-della-libreria"

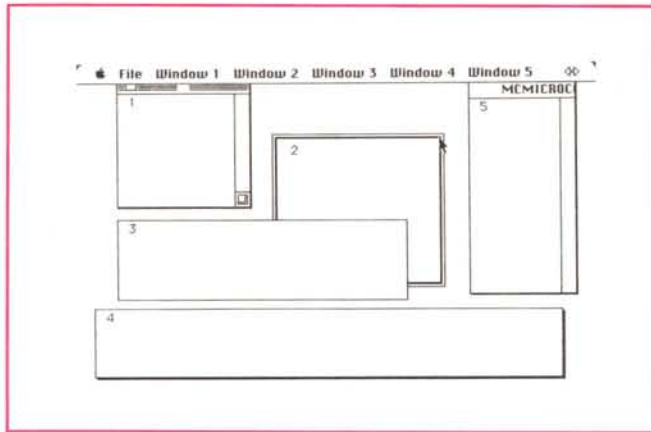
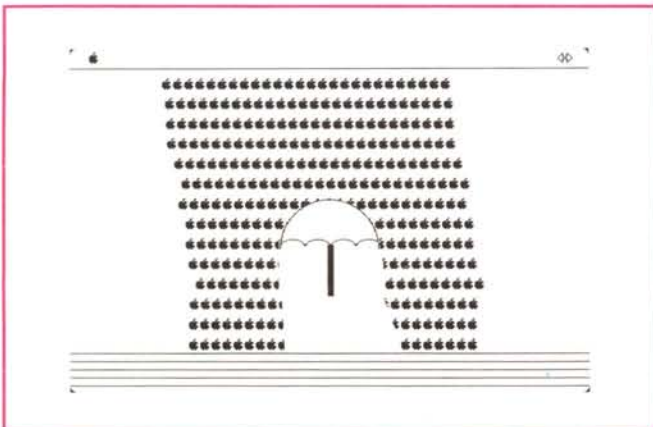


Figura D  
Tipologia di creazione delle finestre da software; si noti come siano o accessibili tutti i formati più comuni, presenti nella maggior parte degli altri programmi.

Figura E  
Pioggia di mele su un ombrello; pur se non velocissima, la grafica si difende bene, con discreta efficienza.



eventualmente aggiungendo nomi di directory e disco. Dopo di che i nuovi statement sono in linea, e vanno utilizzati né più né meno come quelli già noti del Basic.

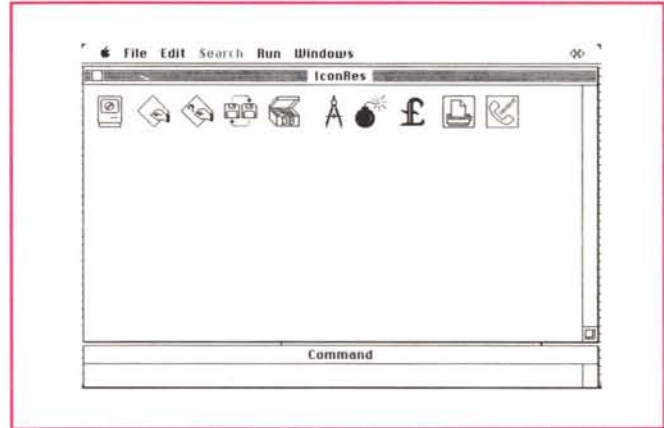
Diamo una rapida scorsa ad alcune routine più interessanti; una, estremamente utile è [CmdKey] (acronimo di Command-Key), che permette di creare macro. Ad esempio, battendo nel programma:

MENU 1,1,1, "Esci"  
 CmdKey 1,1, "Q"

l'opzione "Esci" del menu a tendina sarà accompagnata dal blocco Command-Q, e a tale combinazione sarà sensibile. Un'altra, la [Getmouse m%(0)] inserisce le coordinate (in pixel) del mouse negli elementi m(0) ed m(1) dell'array m( ).

ChangeCursor mette a disposizione una strada facile e veloce per cambiare la forma del mouse tra alcune già predefinite; altri cursori è possibile ottenere tramite la solita array VarPtr%, in maniera agevole anche se un po' lunga (si vedano in figura, i risultati, migliorati con un programma di evidenziazione

Figura F  
 Accediamo alle librerie della Clear Lake; una fase dell'analisi delle risorse di icona presenti nel pacchetto.



delle operazioni). In questo caso esiste addirittura una utility che consente di riempire una array senza la solita trafila READ-DATA.

Il blocco delle routine di QuickDraw viene attinto tramite una serie di operazioni che, superate finalmente le pastoie dei record-puntatori del Pascal, permette di agganciare tutte le routine di disegno semplicemente avendo cura di

definire una array, ordinatamente, in precedenza. Poligoni, regioni, cerchi, ovali, rettangoli (anche con curvatura di vertice a scelta) divengono davvero semplici da disegnare e «dragare».

Un capitolo estremamente importante è dedicato alle risorse, tool unico del Mac; una particolarità importante di Mac è quella di utilizzare file di risorse, un sofisticato sistema che consente di salvare e richiamare non solo testo e numeri, ma «oggetti» complessi, come disegni, icone, box di avviso, cursori, ecc. Queste parti possono essere considerate elementi unici, immediatamente raggiungibili ed indirizzabili, così che è possibile dire al programma, in una maniera più o meno informale: «Prendi l'alert box 3, metti l'icona 2, metti l'avviso n. 12 ed evidenzia un cursore tipo clessidra che mostra lo scorrere del tempo». Tutto viene eseguito non più tramite istruzioni particolareggiate, ma tramite accesso a blocchi specializzati (risorse) che possono risiedere nel programma stesso, ma che è molto più logico sistemare in file separati, accessibili eventualmente da altri programmi.

Ogni risorsa è caratterizzata da una sua tipologia ed è rappresentata da un massimo di quattro lettere maiuscole. Alcune sono:

- CURS cursore
- GNAL di utilizzo generico
- PICT disegno
- AL alert box
- ICON icona
- PAT pattern
- STR stringa
- DLOG dialog

ed il mezzo principe per la costruzione di esse è il Resource Editor, prodotto dalla Apple (ma niente impedisce di usare prodotti similari, che negli States sono di «general domain», distribuiti generalmente attraverso network, gratuitamente).

Figura G  
 Costruiamoci un cursore personalizzato; si noti, a destra, la rappresentazione numerica del pattern a sinistra, ivi comprese le coordinate dell'«hot pixel».

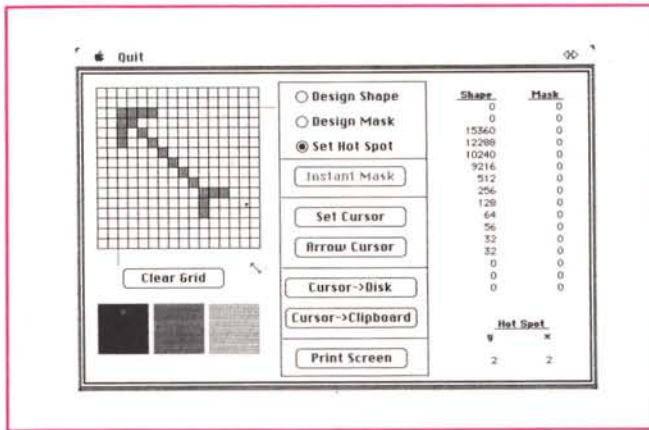
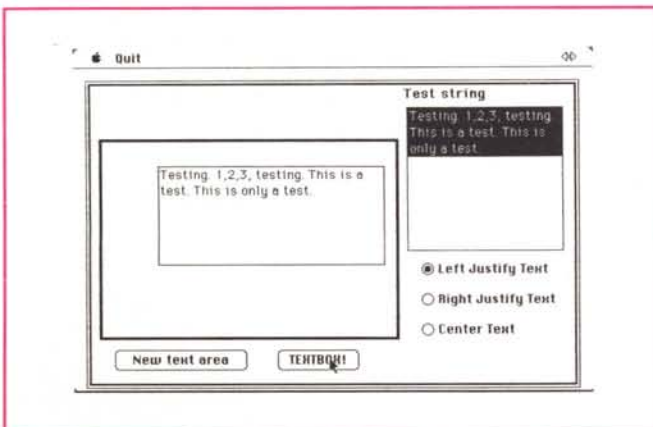
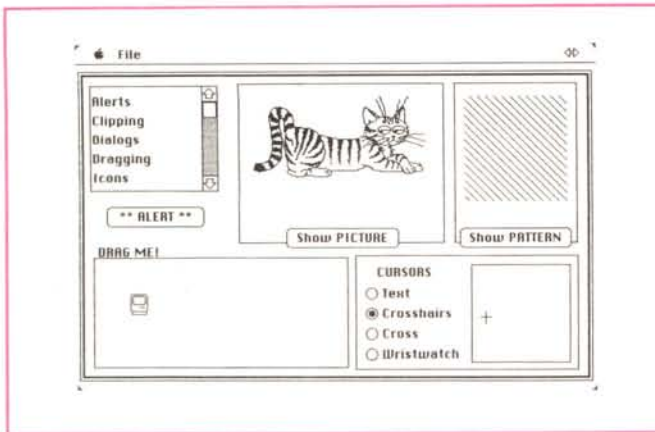


Figura H  
 Utilizzo delle finestre di dialogo, con tipologia di centratura del testo.







Alcune delle figure presentate, tratte dai demo presenti sui dischetti, sono state realizzate e sono proprietà di John O. Brooks, Sharon Zardetto Aker, e della Clear Lake Res.

Figura 11, 12  
Miscellanea di tipologie d'utilizzo delle risorse, finalizzate a diversi scopi: finestre di alert, dialog, costruzione di pattern, trascinalamento di figure, ecc.

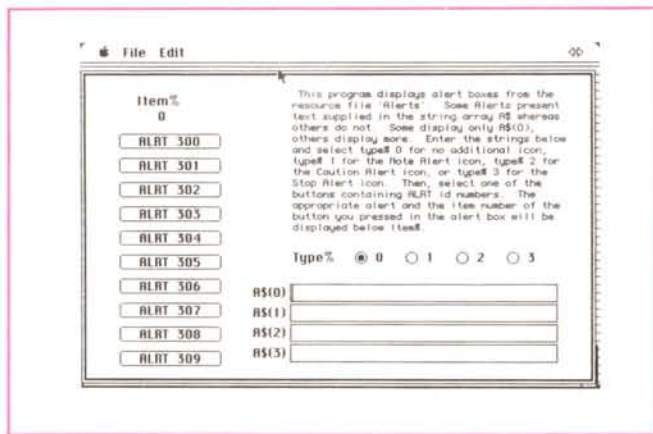


Figura L  
Costruiamoci delle «fork» per l'aggancio di routine interne; evidenziate, in figura, le chiamate all'ora ed alla posizione del mouse.



Un file di risorsa è accessibile attraverso un puntatore, parola difficile che indica, in questo caso semplicemente un numero di riferimento.

Questo numero viene automaticamente assegnato dal programma quando si apre il file di risorsa con il comando:

OpenResFile "nome-del-file", numero-di-rif%

Una risorsa è chiamata da un file inserendola in una variabile Basic od in un'area di memoria chiamata «heap», rispettivamente attraverso i comandi LOAD o GET; al contrario una risorsa

viene salvata usando i comandi ADD e SAVE.

Ancora notizie interessanti sui cursori. Di per sé il sistema possiede già 5 cursori precostituiti che possono essere chiamati direttamente con il comando

CALL CHANGECURSOR(n)

dove [n] è il numero del cursore desiderato. Ciononostante è possibile costruirsi un cursore personale con una tecnica abbastanza semplice (ne vedete un esempio in figura); un cursore è rappresentato, a tutti gli effetti, da una array di 34 elementi in cui i primi 16 sono la rappresentazione numerica intera del pattern che ha generato il cursore stesso, i successivi 16 individuano il «mask», la maschera di base in cui si trasforma il cursore quando si sposta su aree già colorate (generalmente viene eseguito uno scambio a NOT dei pixel); infine gli ultimi due valori sono relativi alle coordinate (si ricordi l'assurda notazione, imposta da Microsoft, dell'origine delle coordinate in alto a sinistra). Allo stesso modo è possibile costruirsi una icona e manipolarla tramite le opzioni GETICON, SAVEICON e DRAWICON.

Un intero capitolo dell'appendice è destinato ai dialoghi di ALERT, di cui si vedono, in figura, alcuni demo; anche qui nulla di speciale; ogni ALERT viene manipolato da un ID number, un numero di riferimento che consente, tra l'altro, di «agganciare», nel box di alert, anche icone esterne, da aggiungere alla finestra stessa.

Continuare con questa descrizione richiederebbe numerose pagine, e, forse non avrebbe molto senso; meglio andare a braccio, evidenziando una serie di opzioni e comandi di cui avevamo intuito l'esistenza, utilizzando altri programmi, ma che ci erano parse diavolerie irraggiungibili; vediamo il comando BringToFront, che mette in primo piano qualcosa di nascosto (non è altro che quello che facciamo quando «clickiamo» una finestra in secondo piano; ricordate il metti-dietro o porta-in-primo-piano di Mac Draw e simili?); ancora il ClosePgon, che chiude automaticamente il poligono iniziato con un altro comando, OpenPgon. Ancora, che meraviglia il CmdKey che permette di creare comandi equivalenti a quelli dei menu a tendina; e poi comparazioni di rettangoli, tracciamento e cancellazione di poligoni, riempimento di regioni con pattern, interrogazioni sullo stato del mouse, del cursore, dei pixel dello schermo, rovesciamento di coordinate e figure, scroll delle stesse con refreshing dello schermo, resettaggio delle coordinate della finestra attiva, costruzione della penna, autocentratura del