

Appunti di programmazione

La programmazione del Macintosh

Come costruire una interfaccia utente ben realizzata e conforme ai canoni Mac.

Ho ritrovato nella casella di MC-Link un messaggio di un certo dr Ambrogio Rossi (non credo sia un siciliano) che con estremo garbo mi parlava a lungo dei suoi tentativi di programmare in maniera decente il suo SE/30, con risultati certo eccellenti nella soluzione dei problemi che si proponeva, ma senza poter forse saper realizzare quelle meravigliose interfacce proprie di tutti, o quasi tutti i pacchetti presenti sul mercato

Le parole che concludono il suo lungo messaggio sono più o meno queste: «Insomma, non pretendo certo di costruire due o trecento shortcut come quelli di Word o Word Perfect, o una finestra con dieci menu (magari qualcuno gerarchico) e venti bottoni, ma avrei voluto almeno fare un poco di bella figura con me stesso e premere o cliccare qualcosa ottenendo il risultato che desideravo e non la solita bomba di sistema».

Ho deciso di rispondere a questo messaggio in quanto mi pare significativo sia dagli argomenti che stiamo trattando, sia di uno dei più grossi problemi che un utente Mac è generalmente chiamato ad affrontare. E spero, con questo di poter chiarire uno dei problemi che più di tutti affliggono l'ambiente Mac: la creazione di una interfaccia Mac-Like.

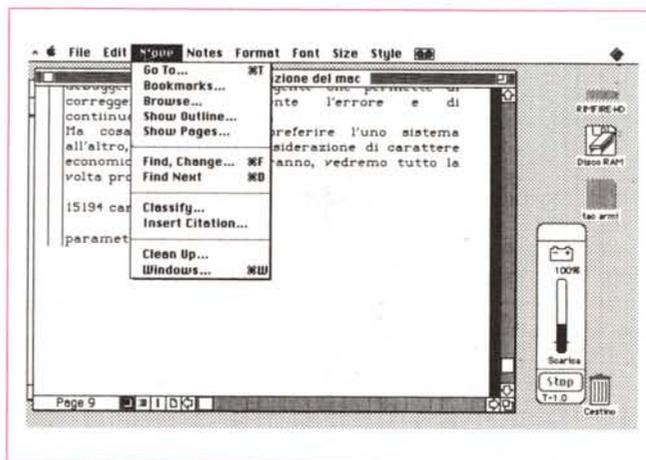
La costruzione di una interfaccia «Click-and-go»

I miei primi tentativi di produrre un poco di software «serio», in particolare alcuni programmi specifici per la mia professione, li ho fatti, intorno alla fine degli anni Settanta su alcune macchine

HP, tra cui anche il fantastico 85-87 e l'ancora più potente 9816, macchina a 16 bit in un mondo in cui appena balbettavano gli Apple II. Ovviamente a quei tempi parlare di interfaccia utente era come discutere di brioches e crostini a Bassora durante la guerra del Golfo. Gli input erano realizzati quasi esclusivamente attraverso domande e risposte, direttamente battute alla tastiera (il classico INKEY\$ era addirittura un lusso, visto che sull'87 non esisteva e occorreva simularlo attraverso una strana circonlocuzione programmatica).

Al mio primo acquisto del Mac pensai di poter ripetere i miei exploit programmatici («beatus monoculus in terra caecorum»), ma mi resi conto quasi immediatamente di avere di fronte un osso duro da rodere.

In effetti Mac è una delle macchine più complesse da programmare; intendiamoci, se si desidera costruire un programma non eccessivamente elegante, le difficoltà sono né più né meno quelle che si incontrerebbero con un PC; si avrebbe in questo modo il classico srotolo di domande-risposte sullo schermo e la cosa finirebbe lì. Ma se solo solo si desidera entrare nell'affascinante mondo delle finestre, dei bottoni,



Il menu a tendina, un classico dell'interfaccia Mac; nel particolare il menu Move di FullWrite Professional.

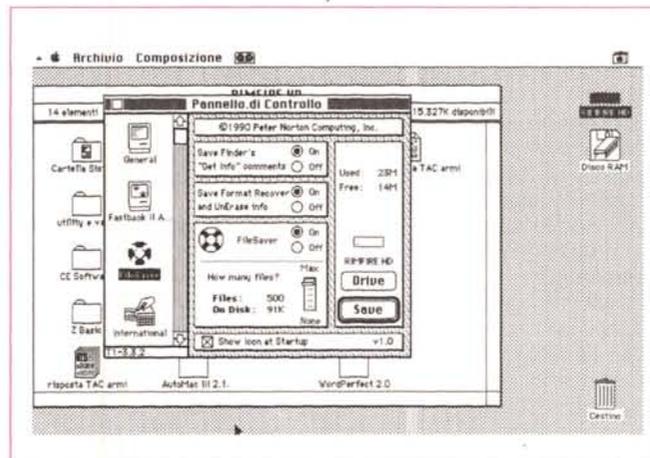
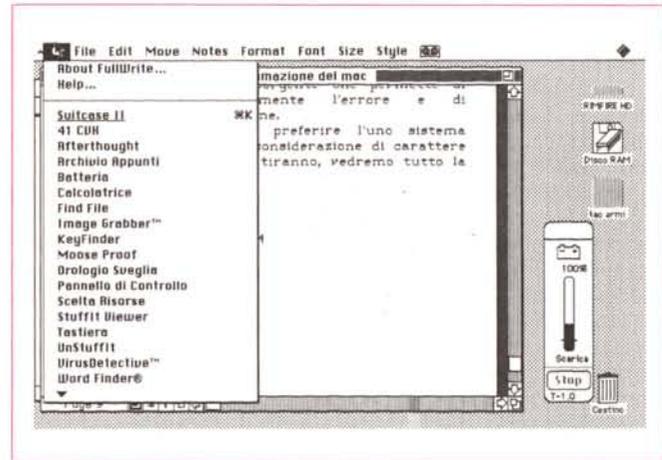
dei menu, delle chiamate ad altri programmi o file, magari lavorando in Multifinder, le cose divengono immediatamente complesse!

E questo purtroppo è un problema specie se si decide che il nostro prodotto ha sfondato tante barriere del sapere da poter essere distribuito. Il fatto è, lo abbiamo ripetuto un sacco di volte, che applicazioni sofisticate come Word, Draw, Excel ci hanno abituato a interfacce così complesse ma altrettanto facili da usare che dover rispondere a domande del tipo «Sei sicuro dei dati immessi?» battendo la risposta alla tastiera ci sembra per lo meno umiliante. Fatto sta che, solo per capire e gestire in maniera corretta il ToolBox, Apple ha realizzato ben sei volumi di Inside Macintosh, ognuno dello spessore di un elenco telefonico.

Il tutto si traduce in una sorta di frustrazione diffusa, quando ci si accorge che l'applicazione fa tutto tranne quello che desideriamo, che premendo un bottone così pazientemente costruito abbiamo innescato una bella bomba di sistema, e così via. Il fatto è che Macintosh è un ambiente di programmazione estremamente sofisticato e complicato. Conoscerlo, e divenire un buon programmatore, può comportare tempi lunghi (anni, laddove per una macchina più semplice come un PC l'impegno si può rappresentare in mesi o addirittura settimane); gestire una complessa interfaccia come quella del sistema operativo è cosa che richiede pazienza e profonda conoscenza delle chiamate (se sanno qualcosa, oggi, quelli che timidamente stanno accennando a programmare in Windows, anche se questo ambiente non può essere certo paragonato, per complessità ed efficienza, a quello Mac) al ToolBox e al pacchetto QuickDraw. E così l'aver costruito una o più routine affidabili e riutilizzabili diviene qui più che un lusso una esigenza, rivelando quanto sia già in tale pratica compresa l'assenza principale della programmazione orientata all'oggetto.

Infine e non è da poco, Apple ha deciso di adottare come linguaggio d'elezione per tutta la bibliografia destinata al Mac una versione del Pascal; non ci è parsa una scelta molto felice, visto che, in fondo, la stella di questo linguaggio non ha mai brillato prepotentemente e la presenza sempre più imperiosa del «C» lo sta trasformando in un nobile decaduto destinato alla sorte cui si è ridotto il Fortran o il Cobol. Ma tant'è; per fortuna tutti i costruttori di linguag-

Ancora due dei cardini dell'interfaccia Mac; i DA e i CDEV (v. testo).



gi, dal più agevole Basic al più sofisticato dei C hanno pensato bene di bypassare il collo di bottiglia delle chiamate al ToolBox adottando specifici statement per quasi tutte le Call e le Function presenti su ROM; un esempio è il potentissimo ZBasic che nella sua ultima sofisticata versione (la 5.1) permette di accedere a tutte le routine di sistema, non solo, ma di crearsene delle proprie.

Immaginiamo quindi di voler costruire una applicazione Mac destinata al mercato, con tanto di interfaccia utente efficiente e ben costruita e progettata.

Innanzitutto quale macchina si desidera adottare come base per lo sviluppo? Oggi il panorama Mac è estremamente ampio, dal Classic fino all'FX (stavo, per dire all'F15), con una altrettanto ampia panoramica di periferiche e device speciali. Alcune generiche applicazioni, come un word processor o un database, devono poter girare su ogni

tipo di macchina, senza alcun riguardo alla configurazione o al tipo; e devono funzionare allo stesso modo sulla macchina prima e ultima della classe, senza differenze per l'utente (ma spesso con grandi differenze per il programmatore). Così se si desidera sviluppare una applicazione generica occorre tenere presenti tutte le caratteristiche dei Mac presenti sul mercato, con un occhio anche a quelli oggi non più in produzione.

Le basi dell'interfaccia Mac

Con alcune rare eccezioni (come ad esempio Zedcor C) tutte le applicazioni Mac lavorano in un ambiente interattivo al massimo grado. Per lanciare un programma, ad esempio, occorre «cliccare» l'applicazione, ma, cosa del tutto inconcepibile in MS-DOS, è possibile lanciare anche un file dati. Una volta che il programma è stato lanciato esso

interagisce con l'utente attraverso una serie di menu, finestre, dialoghi; con un poco di memoria in più è possibile usare MultiFinder, una applicazione di base che permette la esecuzione di diversi programmi contemporaneamente.

La filosofia della gestione dell'interfaccia è, a questo punto, per così dire obbligatoria. La pressione del tasto command e della lettera O deve aprire, in qualunque applicazione, un documento o un file dati; non attenersi a queste regole significa cadere nel caos più completo. È a questo punto che interviene il concetto di applicazione; essa è un ambiente che agisce indipendentemente da tutto quanto presente nel più generale ambiente Mac. Ciononostante l'ambiente generale Mac dispone di una serie di «piece» che la stessa applicazione può utilizzare e, a cui, talvolta, è indissolubilmente legata.

Fanno parte dell'ambiente generale Mac una serie di categorie di software che funzionano in maniera trasparente o vengono chiamate direttamente dall'utente; vediamo qualcuna.

Principi delle applicazioni laterali sono i DA, di essi si sa troppo per poter pretendere di dire qualcosa di più in queste righe. C'è solo da far presente che, a quanto pare, con il nuovo sistema operativo [7.0] sarà possibile gestire questi anche in modo gerarchico.

Ci sono poi i Device Driver che, gestiti dal Device Manager assolvono a una serie di compiti «oscuri» del programma stesso; sovente sono inseriti all'interno di un INIT che provvede ad installarli. Dei Device Driver probabilmente pochi utenti Mac conoscono l'esistenza e la vera funzione.

Una RISORSA (Resource Code) è un pezzo di software che viene chiamato, alla bisogna, per eseguire una certa funzione in un programma; delle risorse (che possono essere di sistema o specifiche per ogni programma) ci si rende conto se si pensa che un programma come Claris Cad o Word Perfect, della grandezza di circa un mega, riesce a funzionare anche su macchine che, tra sistema operativo e Finder, possono solo mettere a disposizione qualche centinaio di K. Ci si accorge della loro presenza, ad esempio, sul portatile, quando l'azione del programma che stiamo utilizzando si ferma per consentire il caricamento di questa o quella routine che, sebbene facente parte del file della applicazione stessa, non è fisicamente presente nella RAM della macchina.

Un INIT è un programma, generalmente (ma non sempre) di piccole dimensioni sistemato nella cartella di sistema ed eseguito dallo stesso sistema

operativo durante il bootstrap. Si tratta di un terreno fertilissimo di programmazione, tanto che oggi si è arrivati a una vera e propria giungla inestricabile; occorre tener conto che per avere proprie risorse definite nel resource fork dello stesso init, alcuni di essi non funzionano in combinazione con altri, nell'ambito di certe applicazioni, o addirittura in occasione di cambiamenti del sistema operativo, tanto da rendere praticamente inutilizzabili gli stessi programmi o da entrare in conflitto con relativa mandata in bomba del sistema.

Simili nelle funzioni, ma programmabili nelle loro esigenze, sono i CDEV (acronimo di Control Device), accessori del pannello di controllo che permettono di regolare (e talora modificare) le caratteristiche dell'ambiente di sistema. La loro presenza è segnalata nella finestra a scorrimento laterale del pannello di controllo, dove si presentano con la loro icona e che, di qui, possono essere settati nelle loro regolazioni. Un esempio è il Keyboard, il Color, il Sound, forniti colla dotazione di base, ma anche l'Adole Type Manager, alcuni intercettatori di virus, Disk Express, Mirror di Mac Tools, di cui parliamo su queste pagine, e così via.

Infine, ma non ultimo, il MPW, Mac Intosh Programmers Workshop. Si tratta dell'ambiente di sviluppo messo a punto e supportato dalla stessa Apple. Tutti i compilatori di linguaggio, come C, C++, Pascal, e i relativi tool di supporto, come linker e librerie, girano in questo speciale ambiente e con esso si sinergizzano. Ne parleremo tra poco.

Nella complessità dell'ambiente Mac

Con la release 7.0 del suo sistema operativo, già disponibile negli Stati Uniti e prossimamente presente anche da noi, sono stati eliminati molti dei problemi presenti nel labirintico sistema operativo 6. Ciononostante le cose non divengono per questo meno facili e agevoli.

Se si è giunti a questo punto a pensare che ci si trova in un ambiente di programmazione molto poco confortevole, ebbene, non si è affatto lontani dal vero. Sebbene Mac metta a disposizione una interfaccia magnifica per quanto attiene all'interattività dell'utente, questa non è purtroppo molto facile da programmare. Tanto poi per aggiungere un di più molti termini della programmazione Mac (ad esempio risorsa) sono nuovi e richiedono una spiegazione essi stessi; e tanto per colmare la misura, Apple non ha fatto mai alcuno sforzo per fornire all'utente un glossario

esauriente e ragionato della sua terminologia. Così la più completa e affidabile fonte di informazione rimane Inside Macintosh, il ben noto (e monumentale) lavoro di documentazione dell'ambiente del sistema operativo e del ToolBox. Ciononostante non si tratta di un lavoro né piacevole né facile, anche perché gli autori non fanno nulla o quasi nulla per mettere in condizione l'utente di capire (tant'è che Addison Wesley ha pensato bene di proporre una sua collana, basata sullo stesso argomento, ma molto più chiara nella esposizione). Inoltre la collana è estremamente povera di esempi e organizzata, a livello concettuale, in maniera a dir poco ostile.

Per chi desidera cominciare a programmare in maniera efficiente è consigliabile precedere l'acquisto dell'Apple Macintosh Programmers Workshop, e del MPW C, ambedue disponibili, sul mercato USA, a 150 \$ ognuno (e molto più cari, in proporzione in Italia). Consigliabile anche l'Atzec C, già nominato, e il Think C, ambedue (specie il primo) più costosi.

MPW è un programma-applicazione che crea un ambiente di sviluppo molto simile a quello UNIX, e, come questo, mette a disposizione uno shell efficiente, con relativi alias e variabili di sistema, la somiglianza al potente UNIX è tanto forte da comprendere redirezione, pipe, script di sistema, storie e espressioni. Ai comandi specifici di tastiera di UNIX si accompagna uno specifico subambiente, (Comando), che fornisce una serie di statement guidati da menu e mouse, per una più semplice programmazione. Per chi sa ben programmare in UNIX, il passaggio a MPW sarà facile e non privo di piacevoli sorprese. Inoltre MPW C è un eccellente compilatore che genera programmi oggetto utilizzabili da altre parti di MPW.

Symantec fornisce il suo Think C, un compilatore velocissimo (fino a 100.000 linee al minuto), specie sulle piattaforme più avanzate. Esso inoltre fornisce un potente editor multiwindow, e una serie di facility di programmazione molto interessanti. Cosa molto comoda, Think C non segnala solo, come la maggior parte dei compilatori, gli errori incontrati e loro ubicazione, ma, alla scoperta dell'errore, ferma la compilazione e sposta l'azione su un debugger a livello di sorgente che permette di correggere immediatamente l'errore e di continuare la compilazione.

Ma cosa dovrebbe far preferire l'uno sistema all'altro, a parte pure considerazioni di carattere economico? Lo spazio è tiranno, vedremo tutto la volta prossima.

ME

Atari News

ATonce Plus Emulatore 80286-16 MHz

ATonce e' una piccola scheda che inserita all'interno del vostro computer Atari ST lo trasforma in un vero PC- AT Compatibile a 16 MHz.

Con emulazioni grafiche: CGA, Hercules, Olivetti, EGA e VGA mono.

Completa gestione della memoria (Protetta/espansa)

Velocita': Norton Si 8.0 e MIPS-TEST 108%

Completo di Tools.

L.479.000+Iva

Sono disponibili anche gli adattatori per l'installazione all'interno dei Mega ST ed Atari 1040STE senza saldature.

1st Card **CRAZY**

Intelligenza artificiale ed Iper testo
Database testuale, shell, grafica
ed Intelligenza artificiale

**MULTIMEDIALITA' ANCHE
PER ATARI ST**

L. 220.000+Iva

Testatene le molteplici possibilita' con la versione ridotta completa del manuale originale a sole

L. 30.000+Iva

**SIAMO PRESENTI PRESSO
NUMEROSI PUNTI VENDITA
ATARI IN TUTTA ITALIA.
CONTATTATECI PER L'INDIRIZZO DEL
NOSTRO RIVENDITORE PIU' VICINO.**

CRAZY DOTS

**La scheda grafica per
Atari Mega ST**

1MB di Ram, Compatibilita' software Gem e SM 194, Slot per Applicazioni Video

Risoluzioni da 320x200 fino a 1664x1200 pixel con 256 colori su 16,7 milioni

Possibilita' di visualizzare 1664x1200 anche su monitor 800x600 con la tecnica dell'Overscrolling

Compatibile anche con monitor VGA

L. 1.350.000+Iva

Anche in versione per TT



UMANA
SYSTEM & SOFTWARE

TEL. & FAX 011-99.69.552

Strada Cuorgne' 57, 10070 MAPPANO (To)

I signori rivenditori sono invitati a contattarci

NeoDesk 3

Il nuovo desktop per ST/E e TT

La rivoluzionaria interfaccia grafica per Atari che rimpiazza il desktop originale dell'Atari.

Per ulteriori informazioni vi invitiamo a leggere la recensione apparsa su MC di Aprile 1991 a pag. 234, o richiedete il nostro demo.

L. 85.000+Iva

**E' ora disponibile nella versione 3.02
in ITALIANO.**

OMIKRON.

**Easy
Base**

Il Database Flessibile

**Flessibile come le
vostre Idee**

E' finalmente disponibile il rivoluzionario database per Atari ST prodotto dalla OMIKRON Software.

Per verificarne le reali potenzialita' e possibilita' richiedeteci la versione Demo.

In ITALIANO

L. 185.000+Iva

OMIKRON MORTIMER

Il multiaccessorio per Atari ST

Una volta installato la pressione dei tasti CONTROL+ALTERNATE vi fanno accedere a tutte le possibilita' che Mortimer vi offre. Text Editor, Ram Disk dinamica, Spooler, Macro, Snapshot, gestione del discoil tutto resisterne anche al reset del computer.

L. 55.000+Iva

MASTER

**La rivista su disco per Atari ST in Italiano.
Contattateci per ricevere la vostra copia !!**

tutti i marchi sono registrati dai legittimi proprietari