

# Microsoft QuickBasic e Scheda Grafica EGA

Uno dei grandi misteri dell'informatica è quello legato alla nascita del PC IBM. Quando nacque, nel lontano 1982, presentava delle incongruenze che apparirono, all'epoca e in buona parte lo sono tutt'oggi, misteriose.

In fatti a fronte di una tecnologia evoluta dovuta all'utilizzo del microprocessore a 16 bit, alcune scelte Hardware e Software sembrarono limitative.

Apparve limitativa la scelta di una Grafica di tipo economico, in termini di modi grafici e di definizione in numero di pixel, apparve limitativa la scelta del Basic residente, che permette l'uso (del tutto teorico data la impossibilità di salvare File) della macchina senza unità a disco e DOS.

Le conseguenze alla prima limitazione sono state la proliferazione di schede grafiche non ufficiali, che se pur hanno allargato le possibilità grafiche della macchina hanno creato molta confusione, e in certi casi incompatibilità tra prodotti e hardware.

Altra conseguenza è stata lo stridente contrasto tra produzione di software grafico sempre più sofisticato, realizzato per il vasto pubblico degli utenti di PC IBM, ma costruito per un ambiente che però ne mortifica le potenzialità.

Oggi, dopo la uscita delle nuove schede IBM Enhanced Graphic Adapter (EGA) e Professional Graphic Adapter (PGA) lo scenario è sensibilmente cambiato anche se, e ne parleremo tra poco, la confusione resta.

Le conseguenze alla seconda limitazione, quella del Basic residente, sono state la non evoluzione del linguaggio

che in questi anni avrebbe potuto svilupparsi parallelamente all'ambiente Hardware, e il conseguente abbandono del Basic da parte degli utenti che programmano in proprio, a favore di linguaggi più scientifici di migliori prestazioni.

## IBM EGA

Oltre un anno fa l'IBM ha prodotto la scheda EGA, ed è stato un parto abbastanza laborioso in quanto la nuova scheda doveva essere, necessariamente, compatibile sia con tutte le schede già esistenti, e quindi le migliaia di pacchetti software che le utilizzano, e doveva essere compatibile, in alcune modalità d'uso, con i monitor tradizionali, in particolare quello Monocromatico e quello a Colori.

Conseguenza di questa novità hardware è stato un lento adeguamento dei vari pacchetti software, che oggi, in generale, prevedono tutti un driver di installazione su EGA, e, finalmente, la nascita di pacchetti direttamente ed esclusivamente per EGA.

Oggi, a più di un anno di distanza, la EGA è a sua volta diventata uno standard, a tal punto che la vecchia CGA (Color Graphic Adapter) è addirittura uscita di produzione e questo ha comportato anche la scomparsa delle schede «indipendenti». Unica eccezione è la scheda Hercules (monocromatica ad alta definizione) che a sua volta, standard tra le indipendenti, è anch'essa uscita in una nuova, e tecnologicamente rivoluzionaria, versione, provata su MC di dicembre 1986.

## Microsoft QuickBasic

Sul fronte Basic invece, non c'è stata nessuna novità di rilievo, se non l'uscita, avvenuta nel 1984, del Basic Compiler 2.0, della stessa IBM, che allineava il compilatore alla versione 2.0 dell'interprete e introduceva una serie di istruzioni necessarie in ambiente LAN.

Per quanto riguarda la produzione indipendente i Basic presenti sul mercato sono numerosi, ma nessuno, indipendentemente dalle sue prestazioni, è riuscito ad imporsi come standard di riferimento.

Oggi finalmente qualcosa si è mosso, è uscito il Microsoft Quick Basic (siamo alla versione 2.0), che presenta alcune caratteristiche tecniche e commerciali che ne fanno pronosticare un facile successo presso gli utilizzatori «attivi» del Personal Computer.

Le caratteristiche commerciali sono innanzitutto la famiglia di nascita, Microsoft, che costituisce un vero e proprio marchio di garanzia di qualità. La politica commerciale aggressiva, sia in termini di politica dei prezzi, sia in termini di campagne pubblicitarie.

La pubblicità Microsoft ha ormai da mesi, sulle riviste americane, assunto l'aspetto di Microsoft News, ovvero pagine di note tecniche operative sui singoli prodotti direttamente rivolte quindi agli interessati senza dover passare attraverso inutili (in questo caso) slogan pubblicitari.

Dal punto di vista tecnico le novità sono numerose.

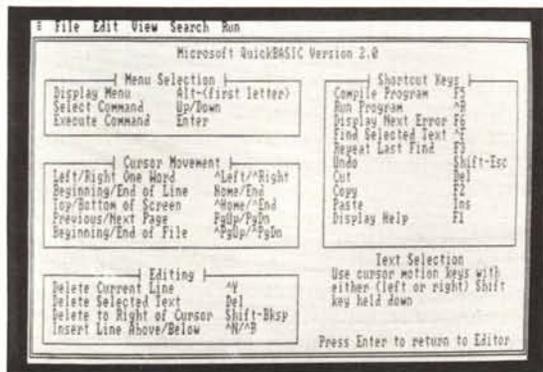
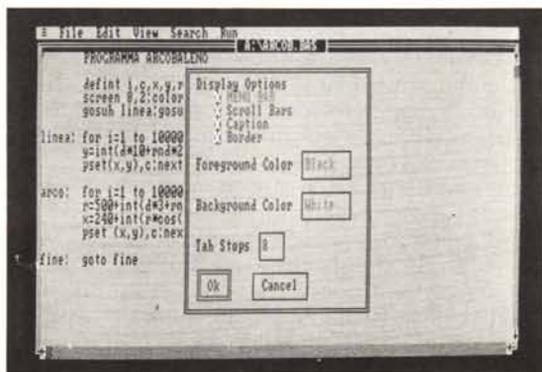


Fig. 1 - Videata di HELP del Microsoft QuickBasic. I comandi di editor sono riassunti in una videata di Help. Sono simili, ai comandi del più popolare dei Word Processor e cioè il WordStar.

Fig. 2 - Metodologia di lavoro a Finestre. Il lavoro col QuickBasic si svolge con la ormai solita tecnica delle finestre, per cui le varie funzionalità possono essere richiamate via menu pop-down.



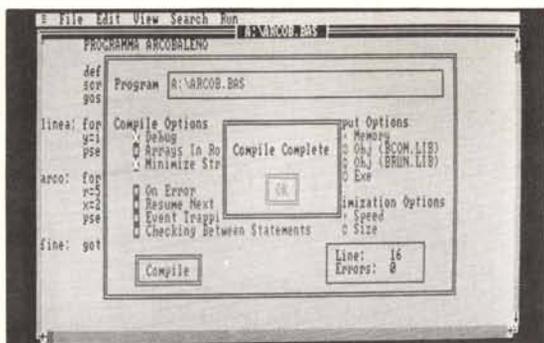
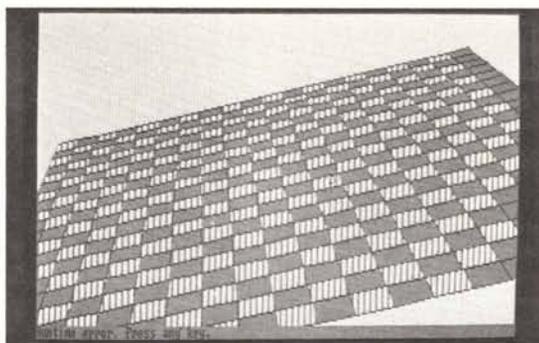


Fig. 3 - Le opzioni della Compilazione.

Scritto il testo del programma si passa alla compilazione, che può avvenire secondo svariate opzioni in termini di funzioni di Debug e di tipo di programma che si vuol ottenere.

Fig. 11 - Programma SFONDO Output. Altra rielaborazione in chiave EGA di un programma pubblicato tempo fa in questa rubrica.



Il QuickBasic è un compilatore, ma le varie opzioni di compilazione ne permettono un uso agile come quello di un interprete. Ad esempio in fase di debug si può, rapidamente, con un unico comando compilare ed eseguire un programma senza produrre file.

È del tutto compatibile con il Basic, per cui è possibile utilizzare come programmi direttamente compilabili ed eseguibili listati in Basic, ovviamente salvati come file di testo.

Ha rispetto al Basic alcune istruzioni in più, in termini di nuove funzioni, e in termini di parametri passabili a vecchie istruzioni.

Riconosce, per mezzo di nuove specifiche istruzioni, il Mouse, riconosce, per mezzo di parametri passabili alle vecchie istruzioni SCREEN e COLOR, la scheda EGA, nelle sue varie opzioni. Riconosce, anche in questo caso tramite nuove specifiche istruzioni, l'ambiente LAN.

Nelle figure 1, 2 e 3, presentiamo la videata principale di Help del QuickBasic, una tipica fase di lavoro, con menu a tendina in azione, e il menu delle opzioni relativo alla fase di compilazione.

## Prestazioni

Come compilatore il QB, offre un miglioramento delle prestazioni rispetto al Basic, variabile in funzione del programma da un massimo di una decina di volte per programmi di solo calcolo di numeri prevalentemente interi fino a un minimo di una volta e

mezza, per programmi senza calcoli ma che utilizzano routine base.

Ad esempio i classici Benchmark utilizzati per testare le macchine (MCmicrocomputer n. 50 pag. 53), qui usati sulla stessa macchina ma su due linguaggi differenti danno i seguenti risultati:

Bench 1 da 33.6 a 4.4 13.24%  
 Bench 2 da 91.0 a 41.9 46.11%  
 Bench 3 da 13.0 a 5.9 45.38%  
 Bench 4 da 8.6 a 1.5 17.90%

Al contrario, un programma di puro disegno, che utilizza il firmware della macchina, costituito da un loop da 1 a 100 e dal disegno di cento cerchi (istruzioni CIRCLE) si riduce solo di un terzo (da 9.71 a 6.32 secondi, pari al 65%).

Il QuickBasic, ovviamente, accetta listati senza numerazioni e conseguentemente utilizza Label per identificare le Righe richiamate da istruzioni di salto. La rinuncia alla numerazione non produce, ed è logico, alcun effetto sul risultato (programma Oggetto) della Compilazione e quindi nessun vantaggio in termini di prestazioni.

## Modalità di lavoro

Il QuickBasic possiede un Editor, per la stesura del programma, che risponde agli ultimi dettami della moda (menu top down), e, per la fase di compilazione, un'interfaccia molto semplificata rispetto ai compilatori tradizionali, tramite la quale sia la fase

di debug, che la fase finale di realizzazione del programma OBJ o EXE avviene via menu.

Il Debug si avvia direttamente dalla fase di Editor, e quindi viene eseguita una compilazione direttamente in memoria centrale. Il risultato, ovvero il programma oggetto, viene poi mandato in esecuzione, senza quindi produzione di alcun file.

Tale facilità di lavoro rende pressoché inutile l'uso di un interprete per preparare il prototipo del programma. Inoltre lavorando in QB si possono utilizzare direttamente le funzionalità che questo offre in più rispetto all'interprete.

Senza entrare nel dettaglio, dato che il nostro interesse è pur sempre relativo alle possibilità grafiche, citiamo la possibilità di definire funzioni (istruzione DEF FN(X) che occupano più di una riga, e la possibilità di costruire delle routine, identificate con label SUB NOME, richiamabili via comando CALL NOME, e alle quali vanno passate una serie di variabili.

Tali routine in pratica riconoscono solo le variabili che gli vengono passate, per cui in un certo senso possono vivere indipendentemente dal programma nel quale vengono utilizzate. Possono essere compilate a parte ed entrare in una libreria di routine.

In conclusione la facilità d'uso dell'editor e delle fasi successive, la presenza di comandi più sofisticati rendono pressoché inutile l'uso dell'interprete, almeno fino a quando non uscirà una nuova versione del Basic, a

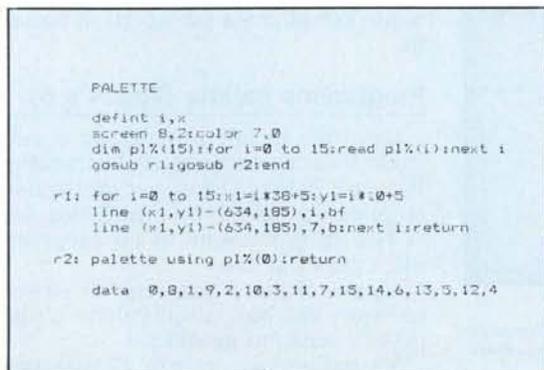
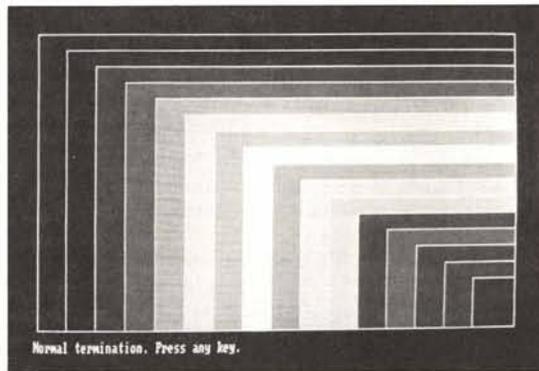


Fig. 4 - Programma PALETTE Listato.

Con l'istruzione Palette è possibile ridefinire le specifiche dei colori. Ad esempio assegnando il valore 4 al colore 3, tutto quello che nel disegno corrente era verde diventa rosso.

Fig. 5 - Programma PALETTE Output.

L'output non può mostrare appieno l'effetto dell'istruzione Palette, che è un effetto dinamico.



```

PROGRAMMA ARCOBALENO

defint i,c,x,y,r
screen 8,2,color 7,0:line (0,0) (639,199),5,b
gosub linea:gosub arco:end

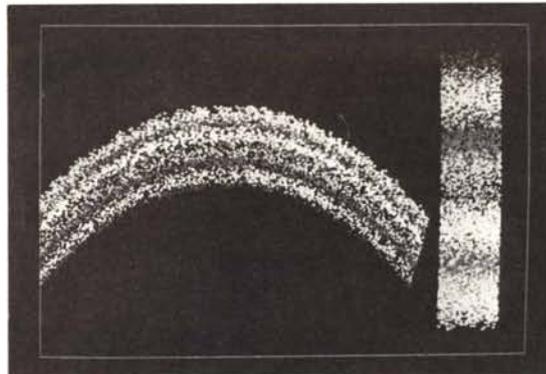
linea: for i=1 to 1000:d=rnd*16:c=int(d)
      y=int(d*10+rnd*20+5):x=int(rnd*80+520)
      pset (x,y),c:next i

arco:  for i=1 to 1000:d=rnd*16:c=int(d)
      r=500+int(d*5+rnd*6):a=1.07*rnd
      x=240+int(r*cos(a)):y=600-int(r*sin(a))
      pset (x,y),c:next i

fine: goto fine
    
```

Fig. 6 - Programma ARCOBALENO Listato. Con 16 colori, che in pratica però sono solo 8, non è possibile ottenere effetti di sfumatura se non ricorrendo a puntinature.

Fig. 7 - Programma ARCOBALENO Output. Sono prodotti due elementi. Un blocco rettangolare e un settore di arco in cui vengono un po' mischiati, con la funzione RANDOM, punti di vario colore, per ottenere un effetto sfumatura.



che ormai non risponde più alle necessità degli utenti più esigenti, che vogliono prodotti software allineati con le prestazioni hardware della propria macchina.

**Nuove istruzioni grafiche**

Innanzitutto le «vecchie» istruzioni SCREEN e COLOR si ampliano, in modo da comprendere i vari modi grafici:

- SCREEN 0 modo testo;
- SCREEN 1 modo grafico 320 per 200, 4 colori, CGA e EGA;
- SCREEN 2 modo grafico 640 per 200, 2 colori, CGA e EGA;
- SCREEN 7 modo grafico 320 per 200, 16 colori, fino a 8 pag. EGA;
- SCREEN 8 modo grafico 640 per 200, 16 colori fino a 4 pag. EGA;
- SCREEN 9 modo grafico 640 per 350, 16 colori tra 64 fino a 2 pag.;
- SCREEN 10 modo grafico 640 per 350 monocromo su EGA.

Per la EGA vanno fatte due considerazioni, la prima è che teoricamente sulla scheda si possono installare da 64 fino a 256 kbyte (teoricamente perché ormai hanno tutte 256 kbyte) e di conseguenza si possono avere più pagine in relazione alla modalità scelta.

La seconda considerazione è che gli ultimi due modi sono possibili solo se si dispone anche di monitor avanzati.

La sintassi del comando SCREEN è:

SCREEN modo, colore, n. pag. att. n. pag. vis. Modo è quello di cui sopra; Colore è il colore di foreground (primo piano); n. pag. att. è la pagina su cui si lavora; n. pag. vis. è la pagina visualizzata.

Lo diciamo per i non esperti. In questo modo è possibile disegnare sulla pagina che non si vede e poi finito il disegno farla apparire. È la tecnica che in genere si usa nelle animazioni, dove la «costruzione» dell'immagine deve venire nascosta.

Consequente all'istruzione SCREEN è l'istruzione COLOR con la quale, in relazione alla modalità scelta, si definiscono i colori di lavoro. Da notare che con la EGA non è più possibi-

le, al contrario di quanto avveniva con la CGA, definire il colore del bordo, ma solo foreground e background.

PALETTE con la quale è possibile riattribuire uno o una serie di colori. Questo permette in pratica di definire in un vettore le specifiche di colore di un disegno e di attribuirle in maniera parametrica. In altre parole si può realizzare un programma o una routine, che contiene delle specifiche, che poi in fase di esecuzione vengono cambiate, secondo le necessità.

WINDOW, VIEW, PUT, GET ecc. Le vecchie istruzioni sono utilizzabili nel nuovo ambiente, con le stesse caratteristiche e limitazioni d'uso già note. Ad esempio la GET, se si lavora con 16 colori, comporta un'occupazione di 4 bit per pixel.

Il comando PAINT (X, Y) chr\$(n)+chr\$(m)... con il quale è possibile costruire delle retinature, è rimasto, ma con le limitazioni di retinatura a due colori. In pratica si può costruire una trama composta con colore di sfondo e colore di primo piano.

Per quanto riguarda la paginazione delle pagine video, oltre allo SCREEN che permette il passaggio e l'accensione delle varie pagine esiste la funzione PCOPY, con la quale si ricopia una pagina su un'altra. Tale istruzione è indipendente dalla pagina attiva.

**Conversioni**

Il primo impulso è proprio quello di compilare i «vecchi» programmi che

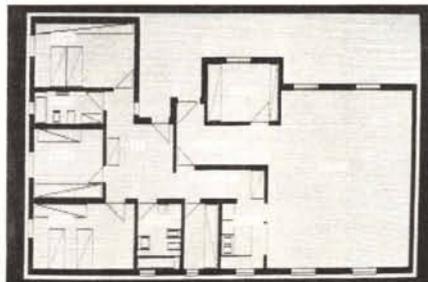


Fig. 8 - Programma CASA Output. In questo altro esempio il colore viene utilizzato per identificare le varie tipologie di elementi componenti il disegno.

già abbiamo scritto in Basica. E in questo non vi è nessuna difficoltà, se il programma originario è stato salvato in ASCII.

Il passo successivo è quello di adattare il listato al nuovo ambiente, innanzitutto eliminando la numerazione identificativa delle righe che ora non serve più. È però evidente che tutte le righe che vengono richiamate in istruzioni di salto, vanno identificate con una Label, che può essere alfabetica e quindi mnemonica.

Poiché il nostro interesse è nella utilizzazione della grafica EGA il lavoro, ma anche questo molto limitato, consiste nell'adattare tali istruzioni.

Nessuna difficoltà per quanto riguarda i formati, anzi nel nostro caso traduciamo programmi realizzati per lo SCREEN 2 (monocromatico 640 per 200) allo SCREEN 8 (16 colori sempre in 640 per 200).

L'unico vero problema sta nella scelta dei colori, il che comporta che ciascuna istruzione di tracciamento del tipo LINE, CIRCLE, PSET, SET va completata con specifica del colore. Inoltre ogni colore va scelto in funzione del fatto che serva al tracciamento di linee, o serva per campiture, e comunque va scelto in accordo con tutti gli altri colori dell'immagine.

Il colore in pratica, e la difficoltà aumenta con il numero dei colori, diventa un elemento compositivo in più, che aumenta il numero delle variabili in gioco.

Vediamo direttamente alcuni programmi, rielaborazione dei programmi molto semplici già presentati in passato.

**Programma Palette (figg. 4 e 5)**

Obiettivo di tale programma è mostrare il funzionamento dell'istruzione Palette e Palette Using, per mezzo della quale è possibile ridefinire, durante la fase di esecuzione di un programma, i colori in uso.

Viene tracciata una serie di strisce colorate, che poi, con il Palette Using pl%(0), vengono modificati.

Va definito un vettore PL%(i), che

