



di Francesco Petroni

La Grafica di presentazione in ambiente PC IBM

Uno dei settori in cui si è maggiormente sviluppata la Computer Grafica è quello della «presentazione» intesa come strumento per costruire singole immagini o per organizzare e presentare insieme di immagini.

Gli ambiti applicativi sono infiniti, anche perché in genere non hanno nulla in comune con la Informatica. Si pensi per fare degli esempi al campo dei Congressi dove vengono presentate e proiettate diapositive dai vari relatori, oppure alle Mostre, anche queste riguardanti qualsiasi settore, dove occorre «presentare» attività e/o prodotti di una certa Azienda.

Oppure nel campo dell'istruzione, sia da parte di chi insegna, che può utilizzare una serie di slide come materiale didattico di facile «presa» sull'allievo, sia da parte dell'allievo stesso che può preparare i propri compiti utilizzando tali supporti. Pensate ad una tesi di laurea preparata su slide, invece delle solite tesi in quattro copie, rilegate, in due ore, su carta telata, battute a macchina alla bell'e meglio, e che poi nessuno leggerà mai.

Nella grafica di presentazione esistono due fasi assolutamente distinte tra di loro, quella di preparazione delle singole immagini, e quella di «montaggio» delle stesse in una sequenza logica ed eventualmente temporizzata. Tale seconda attività diventa complessa, quando, come spesso accade, le immagini sono molte e sono state realizzate un po' alla rinfusa.

La prima fase invece consiste sia nella predisposizione di singole immagini originali, sia nella ricerca di altre imma-

gini qua e là nei vari software che ci capita di vedere, da recuperare in toto o in parte, per adattarle alle nostre necessità.

Questa diventa un'attività creativa e anche produttiva, infatti inserire un'immagine grafica di grande effetto, anche non strettamente in tema con l'argomento trattato, in una presentazione ha come risultato quello di tenere desta l'attenzione dell'uditorio.

Per quanto riguarda il software con il quale sviluppare tale attività, come al solito esiste la possibilità di utilizzare prodotti specifici, oppure i più esperti possono realizzarlo in proprio usando il Basic.

Il Basic, ormai non più produttivo nel campo gestionale, è tuttora il linguaggio più adatto ad applicazioni grafiche, in quanto è il linguaggio che sfrutta al meglio le caratteristiche hardware della macchina, infatti i pacchetti di software di presentazione (tipo l'Execuvision o l'Energraphic) sono scritti in Basic compilato.

La trattazione dell'argomento, che prenderà due puntate, comprenderà due problematiche differenti, in quanto da una parte svilupperemo qualche programma in Basic che risolva le singole funzioni necessarie in un software di Grafica di Presentazione, dall'altra daremo un'occhiata ad alcuni pacchetti di Grafica di Presentazione presenti sul mercato, cercando come al solito di trarne spunti per esperimenti.

Per quanto riguarda questi ultimi c'è in atto una evoluzione. I prodotti più diffusi sono Execuvision e Energraphics, citati prima, ma sono annunciati i

GEM, vera e propria collana di prodotti grafici e l'IBM Storyboard Software.

Quest'ultimo si preannuncia molto interessante, e comprende quattro moduli. Il Picture Taker, l'«acchiappa immagini» dai vari ambiti software, il Picture Maker, programma Autore per la composizione di immagini originali, il Picture Editor per organizzare la presentazione, il Picture Teller per «raccontarla».

Il nome «Storyboard» deriva dal campo cinematografico e significa «copione». Per realizzare una presentazione la cosa più impegnativa non è infatti la realizzazione della singola immagine, ma proprio la scrittura del copione.

Per quanto riguarda i programmi in Basic partiremo da programmi semplici, che, come detto, risolvono ciascuno un solo problema, poi svilupperemo temi un po' più complessi, senza però poter arrivare alla costruzione di un package di presentazione in quanto non basterebbero 10 articoli. Si pensi che i programmi presenti nei pacchetti professionali sopra citati superano le centinaia di Kbyte.

Tipologia e Modalità Tecniche di Presentazione

Il software di presentazione viene utilizzato per realizzare le immagini. Il problema di come presentare le varie immagini preparate è successivo e può essere risolto sostanzialmente in due modi differenti a seconda che si possa utilizzare o meno un computer.

Se non si può utilizzare un computer occorre tradurre le varie immagini in diapositive, proiettabili con un qualsiasi proiettore su qualsiasi super-

ficie chiara. L'organizzazione e la temporizzazione riguardano quindi l'operatore al proiettore.

Per la riproduzione su diapositiva dell'immagine video occorre scattare delle foto del monitor, e questo per un fotografo, anche di poca esperienza, non presenta difficoltà. Se è poi necessaria una alta qualità dell'immagine prodotta si deve ricorrere ad apparecchiature specifiche come il Polaroid Palette (presentato su MC n. 42) o sul riproduttore Kodak, che garantiscono soprattutto la correzione del difetto dovuto alla curvatura della superficie del monitor.

Se nell'ambiente dove viene svolta la Presentazione si può disporre di un computer, sono necessarie altre apparecchiature per ingrandire le immagini, se debbono essere mostrate in una sala, o per replicarle in più monitor, se si debbono proiettare in più ambienti separati tra di loro. È chiaro che questo è un problema che deve venir risolto da strutture specializzate, che risolvono in pochissimo tempo anche tutti i problemi di connessione, stendimento dei cavi, ecc.

Ormai sono moltissime le società che operano nella organizzazione e predisposizione di congressi e mostre,

```

100 REM cerchio
110 REM inizializzazione
120 CLS:SCREEN 2:DIM A%(500)
130 CIRCLE (300,100),30
140 PAINT (300,100),CHR$(85)
150 GET (270,85)-(330,115),A%
200 REM sfondo
210 CLS:X=0:Y=0:S=2
220 LINE (0,0)-(639,199),,B
230 FOR I=-300 TO 640 STEP 20
240 LINE (I,0)-(I+299,199)
250 LINE (I,199)-(I+299,0):NEXT I
300 REM main
310 FOR K=0 TO 580 STEP 2
320 PUT (X,Y),A%,X1=X:Y1=Y
330 X=K:Y=K*170/610
340 PUT (X1,Y1),A%:NEXT K

```

Figura 1 - Listato Programma Cerchio. Il doppio tratteggio viene ottenuto con un unico loop che calcola coordinate esterne all'area del video.

che sono in grado di prendersi in carico la soluzione di questo problema.

Nel caso di disponibilità di un computer occorre organizzare via software lo scorrimento delle immagini, o mediante un temporizzatore che fermi per un certo numero di secondi ciascuna immagine o mediante un comando che permetta l'avanzamento delle immagini, a richiesta.

Programmi più evoluti, specializzati nel Computer Aided Instruction, prevedono percorsi logici tra le immagini a seconda delle scelte eseguite e delle risposte fornite sulle immagini precedenti.

Differenza tra Grafica di Presentazione e Animazione

La differenza sostanziale è che la grafica di presentazione richiede poche immagini e ciascuna di esse deve essere poco complessa, al contrario

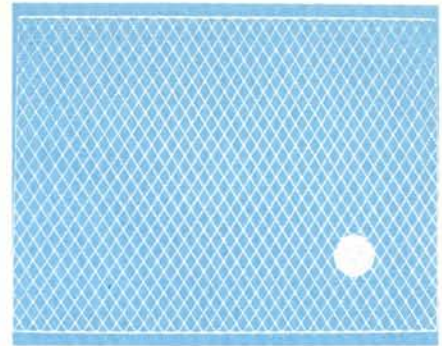


Figura 2 - Output del Programma Cerchio. In questo esempio viene utilizzato lo SCREEN 2 del PC IBM, la cui definizione è di 640 per 200 pixel in bianco e nero.

l'animazione richiede migliaia di immagini tutte molto complesse.

È poiché questo si traduce in termini di prestazioni del computer si può dire che si può fare grafica di presentazione anche con un Home Computer, mentre per fare animazione occorrono apparecchiature professionali o personal Computer particolarmente implementati sia come hard che come soft.

Caratteristiche di un Software per Realizzare Presentazioni

Esistono, come detto, numerosi pacchetti di Software per Presentazione, per il PC IBM, in genere scritti in Basic Compilatore, e che contengono due tipi fondamentali di funzionalità, quella che permette di realizzare la singola immagine (tramite un programma che si definisce «autore») e quella che permette di organizzare l'ordine logico delle immagini, indipendentemente dall'ordine di predisposizione.

L'assoluta indipendenza tra le due fasi consente anche ad un utente poco esperto di farsi in casa un software di presentazione. Infatti è possibile costruire in un modo qualsiasi singole immagini, memorizzarle come file di tipo picture, che fa la «fotocopia» su

```

1000 REM TESTO
110 SCREEN 1:COLOR 1,0:CLS:REM
120 NN=27:DIM A%(NN):FOR I=1 TO NN:READ A%(I):NEXT I
130 LOCATE 25,1:INPUT "Coordinate Iniziali ";X,Y:PSET (X,Y)
140 LOCATE 25,1:PRINT "
150 Q$=INKEY$:IF Q$="" THEN 150
160 IF Q$="*" THEN CLS:SCREEN 0:WIDTH 80:END
170 IF Q$="-" THEN 130
180 IF Q$="+" THEN Q=27:GOTO 200
190 Q=ASC(Q$)-64:IF Q<1 OR Q>26 THEN 150 '
200 DRAW A%(Q):GOTO 150
10000 REM
1010 DATA "U4 E2 F2 D4 U3 L3 BM+5,+3"
1020 DATA "U6 R3 F1 D1 G1 F1 D1 G1 L3 BR6"
1030 DATA "BR3 L2 H1 U4 E1 R2 BM+2,+6"
1040 DATA "U6 R3 F1 D4 G1 L3 BR6"
1050 DATA "U6 R3 L3 D3 R2 L2 D3 R3 BR2"
1060 DATA "U6 R3 L3 D3 R2 BM+3,+3"
1070 DATA "BR3 E1 U1 D1 G1 L2 H1 U4 E1 R2 BM+3,+6"
1080 DATA "U6 D3 R3 U3 D6 BR2"
1090 DATA "U6 BM+2,+6"
1100 DATA "BU6 R3 D5 G1 L1 H1 BM+5,+1"
1110 DATA "U6 D4 E4 G3 F3 BR2"
1120 DATA "BU6 D6 R3 BR2"
1130 DATA "U6 F2 E2 D6 BR2"
1140 DATA "U6 F4 U4 D6 BR2"
1150 DATA "BU1 U4 E1 R2 F1 D4 G1 L2 H1 BM+6,+1"
1160 DATA "U6 R3 F1 D1 G1 L2 BM+5,+3"
1170 DATA "BU1 U4 E1 R2 F1 D4 G1 F1 H1 L2 BR5"
1180 DATA "U6 R3 F1 D2 G1 L1 F2 BR2"
1190 DATA "R3 E1 U1 H1 L2 H1 U1 E1 R2 BM+3,+6"
1200 DATA "BR2 U6 R2 L4 BM+6,+6"
1210 DATA "BU6 D5 F1 R2 E1 U5 BM+2,+6"
1220 DATA "BU6 D4 F2 E2 U4 BM+2,+6"
1230 DATA "BU6 D4 F2 E1 F1 E2 U4 BM+2,+6"
1240 DATA "U1 E4 U1 D1 G2 H2 U1 D1 F4 D1 BR1"
1250 DATA "BR2 U3 E2 U1 D1 G2 H2 U1 BM+6,+6"
1260 DATA "BR4 L4 U2 E4 L4 BM+2,+6"
1270 DATA "BR4"

```

Figura 3 Listato del Programma Testo. Utilizzazione massiccia del metalinguaggio DRAW, con il quale si realizzano figure identificando gli spostamenti di un pixel che le disegni, avanzando sul video, nelle varie direzioni.



Figura 4 - Output del Programma Testo. Con una definizione di 320 pixel si possono ottenere al massimo sessanta caratteri per riga, riducendo il formato dei caratteri al limite della leggibilità.

```

100 REM finestra
110 SCREEN 2:CLS:LINE (0,0)-(639,199),,B
120 FOR R=0 TO 300 STEP 20:CIRCLE (320,100),R,,,.3:NEXT R
130 L=50:H=30:X=320:Y=100:S=5:X1=X-L:X2=X+L:Y1=Y-H:Y2=Y+H
140 LINE (X1,Y1)-(X2,Y2),1,B
150 I$=INKEY$:IF I$="" THEN 140
160 IF ASC(I$)>48 AND ASC(I$)<58 THEN S=VAL(I$):GOTO 150
170 IF LEN(I$)<2 THEN 150
180 LINE (X1,Y1)-(X2,Y2),0,B
190 X3=X1:X4=X2:Y3=Y1:Y4=Y2
200 IF ASC(MID$(I$,2,1))=80 THEN Y=Y+S:GOTO 290
210 IF ASC(MID$(I$,2,1))=72 THEN Y=Y-S:GOTO 290
220 IF ASC(MID$(I$,2,1))=75 THEN X=X-S:GOTO 290
230 IF ASC(MID$(I$,2,1))=77 THEN X=X+S:GOTO 290
240 IF ASC(MID$(I$,2,1))=115 THEN L=L-S:GOTO 290
250 IF ASC(MID$(I$,2,1))=116 THEN L=L+S:GOTO 290
260 IF ASC(MID$(I$,2,1))=118 THEN H=H-S:GOTO 290
270 IF ASC(MID$(I$,2,1))=132 THEN H=H+S:GOTO 290
280 LINE (X3,Y3)-(X4,Y4),0,B
290 X1=X-L:X2=X+L:Y1=Y-H:Y2=Y+H
300 GOTO 140

```

Figura 5
Listato Programma Finestra. In questo tipo di programmi è sempre indispensabile riconoscere tutti i tasti speciali (freccie, tasti funzione, control, ecc.).



Figura 6 - Output Programma Finestra. Lo spostamento e il cambiamento di dimensioni della finestra provoca alcune modificazioni al disegno sottostante.

disco della memoria video, e poi realizzare un semplice programma che legge via via i file e poi li visualizza.

Il discorso vale per qualsiasi ambiente hardware, ci limitiamo però all'ambiente PC IBM in quanto è quello che dispone di un maggior numero di pacchetti professionali.

I conti in tasca ad un software di presentazione

La memoria video grafica del PC IBM è di 16 Kbyte, e quindi su un singolo dischetto possono essere immagazzinate fino a 22 immagini, il che non è molto. Nessun problema se si dispone di un Hard Disk. Per risparmiare spazio, ovvero per immagazzinare un po' più di immagini su un solo dischetto, si deve ricorrere ad altre

modalità di costruzione delle stesse.

Ad esempio se una immagine è composta solo di un testo anche a più colori è più economico, in termini di byte, inserire nel programma di presentazione una subroutine che produca la scritta, oppure tutte le scritte, piuttosto che immagazzinare una Picture che, anche se di contenuto scarno, comunque occupa 16.000 byte.

In questo caso però è il programma di presentazione che si appesantisce. Quindi come al solito la soluzione ottimale va ricercata caso per caso, e per questo è indispensabile che il realizzatore del programma sia il più possibile esperto delle varie tecniche grafiche.

I programmi presentati

Presentiamo sei programmi, ciascuno che risolve una specifica funzionalità.

Un programma Autore completo prevede decine di funzioni, ciascuna delle quali collegata con le altre. È evidente che la complessità del programma nel suo insieme non è direttamente proporzionale al numero delle funzioni ma varia, secondo una legge esponenziale, con esse.

Le funzionalità che presentiamo sono:

- Spostamento di un Oggetto Pre-disegnato sul Video, senza disturbo del disegno sottostante (programma CERCH);

- Utilizzazione di un CHARAC-

```

100 REM color
110 SCREEN 1:CLS:BLOAD "CAMION.PIC"
120 CA=0:CB=0:DEF FN CC(X)=ABS(X-INT(I/16)*16)
130 COLOR CA,CB
140 I$=INKEY$
150 IF I$="" THEN 140 ELSE IF LEN(I$)<2 THEN 140
160 J=ASC(MID$(I$,2,1))
170 IF J=80 THEN CA=CA+1:CA=FN CC(CA)
180 IF J=72 THEN CA=CA-1:CA=FN CC(CA)
190 IF J=75 THEN CB=CB+1:CB=FN CC(CB)
200 IF J=77 THEN CB=CB-1:CB=FN CC(CB)
210 IF J=79 THEN COLOR 1,0:SCREEN 0:WIDTH 80:CLS:END
220 GOTO 130

```

Figura 7
Listato del Programma Color. L'istruzione COLOR, che permette la modifica al volo dei colori dello schermo accetta parametri che variano tra 0 e 15.

TER SET Personale (programma TESTO);

- Localizzazione e Dimensionamento di Una Finestra Rettangolare su Video (programma FINES);

- Cambio dei Colori di Primo Piano e di Sfondo di una Immagine già costruita (programma COLOR);

- Creazione di una Matrice nella quale memorizzare una Porzione di Video, sua memorizzazione in un File sequenziale (programma ISOLA);

- Riutilizzazione di una Matrice nella quale è memorizzata una Porzione di Video (programma PESCA).

Programma CERCHIO

Il programma, listato di figura 1 e output di figura 2, ha come obiettivo quello di illustrare come si memorizza, in una matrice di appoggio, un disegno qualsivoglia e come possa, questo stesso disegno, essere utilizzato in una parte qualsiasi del video.

Ovvero è possibile designare un oggetto e poi posizionarlo «a vista» facendolo muovere sul video con opportuni comandi di «guida», e una volta raggiunto il punto ottimale di destinazione, rilasciarlo.

Il problema principale è quello di non nuocere, durante il movimento, al disegno sottostante, e di questo si occupa una coppia di istruzioni del BASIC grafico.

La GET <coordinate>, <vettore> e la PUT <coordinate>, <vettore>, <azione>. Con la prima si memorizza in un vettore opportunamente dimensionato una porzione del video in cui è posto il disegno che ci interessa. È critico il dimensionamento in quanto deve essere esatto, perché si corre il rischio di «perdersi» parte del disegno o di «portarsi appresso» porzione di video che non interessano.

Con la seconda si ritrasferisce il contenuto grafico del vettore nella zona di destinazione sul video, e poiché la PUT accetta parametri <azione> può essere stabilito il risultato della sovrapposizione con il disegno sottostante.

Il vettore può essere utilizzato all'interno dello stesso programma, ma può anche essere trattato come qualsiasi vettore di dati numerici, in particolare può essere trasferito su di un file sequenziale, che ne comprenda anche le caratteristiche dimensionali, per poter

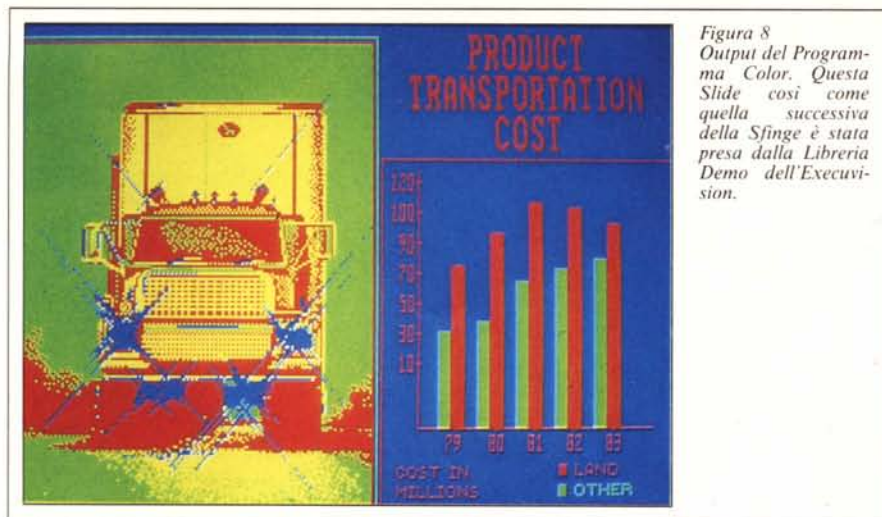


Figura 8
Output del Program
ma Color. Questa
Slide così come
quella successiva
della Sfinge è stata
presa dalla Libreria
Demo dell'Execu-
vision.

venire disegnato e poi cancellato quando viene disegnato nella posizione successiva. Le righe del programma vanno dalla 300 alla fine.

Il programma TESTO

Ogni programma Autore dispone di più set di caratteri e di una serie di istruzioni per mezzo delle quali i caratteri del set possono essere ruotati, colorati, ingranditi e rimpiccioliti.

I set di caratteri a disposizione non debbono risiedere in memoria principale ma vengono, all'occorrenza, richiamati da file specifici (in Execuvision esistono circa 10 character set). Il sistema più economico anche quando si realizza un proprio programma autore è quello di utilizzare i Character Set che si trovano nei vari software in circolazione.

Il programma TESTO utilizza un Character Set realizzato con il comando DRAW, la cui caratteristica è quella di essere i più piccoli caratteri possibili in una utilizzazione nella modalità grafica SCREEN 1 del Basica IBM.

Tale modalità è lo standard della grafica a colori IBM e permette 320 pixel in orizzontale e 200 in verticale. Dispone, di base, di un set di caratteri esteso ma che ne permette solo 40 per riga. Economizzando e rendendo la scrittura proporzionale si arriva anche a 60 caratteri per riga, utili in caso di scritte su disegni.

Il programma listato in figura 3 e output in figura 4, presenta un set di caratteri (solo alfabetici e solo maiuscoli da riga 1000 in poi). Ha solo tre funzionalità, lo space, ottenuto con il carattere «+», il posizionamento ottenuto, attraverso una richiesta di coordinate digitando il carattere «-». Infine premendo il carattere «*» si ottiene la fine del programma.

Programma FINESTRA

Un'altra funzionalità sempre presente nei programmi autore è quella che permette di identificare direttamente sul video una finestra. Tale funzionalità di base può servire per varie funzionalità specifiche, come lo spostamento, la copia, la cancellazione, la memorizzazione.

Nel programma presentato utilizziamo l'istruzione LINE <coordinate>, <parametri> che permette il disegno

essere riutilizzato anche in fase di realizzazione di altri disegni.

Il programma si divide in tre fasi. La prima, da riga 100 a riga 150, consiste nella preparazione dell'oggetto e nella sua memorizzazione nel vettore A%. L'oggetto qui è un semplice cerchio riempito con un tratteggio.

La seconda fase consiste nella pre-disposizione dello sfondo, costituito

da un tratteggio obliquo, e realizzato tramite un loop che genera valori di coordinate anche esterne al video, in modo da poter produrre linee oblique, con un solo loop. Da riga 200 a riga 250.

L'ultima fase è costituita da un loop che provoca il trascinamento dell'oggetto sul video, lasciando inalterato lo sfondo. Per ottenere ciò l'oggetto deve

```

100 REM isola
110 REM definizione area
120 DEF FNA(X,Y)=4+Y*INT((2*X+7)/8)
130 X1=160:Y1=22:X2=300:Y2=116:X3=X2-X1+1:Y3=Y2-Y1+1
140 OPTION BASE 1:D=FNA(X3,Y3)/2:DIM DD%(D)
200 REM caricamento picture
210 SCREEN 1:COLOR 1,0
220 CLS:BLOAD "sfinge.pic"
230 LINE (X1,Y1)-(X2,Y2),,B
240 GET (X1,Y1)-(X2,Y2),DD%
300 REM salva
310 OPEN "sfinge.dat" FOR OUTPUT AS 1
320 WRITE #1,D
330 FOR I=1 TO D:WRITE #1,DD%(I):NEXT I
340 CLOSE #1:CLS
400 REM scaricamento
410 PUT (0,0),DD%,XOR
420 PUT (10,99),DD%,PRESET
430 PUT (159,0),DD%,PRESET
440 PUT (169,100),DD%,OR
450 I$=INKEY$:IF I$="" THEN 450
460 CLS:SCREEN 0:WIDTH 80

```

Figura 9
Listato Programma
Isola. Viene letta
una Picture, dalla
quale si estrae un
rettangolo, che viene
memorizzato in un
file sequenziale,
DAT.

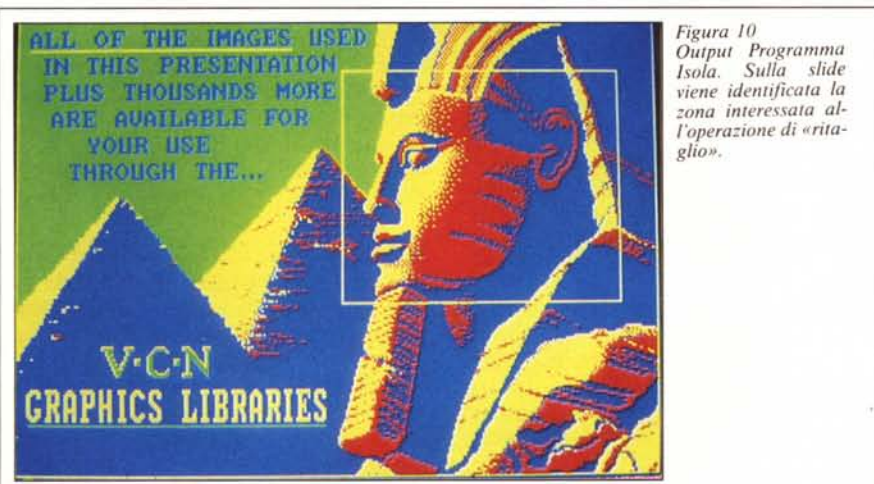


Figura 10
Output Program
Isola. Sulla slide
viene identificata la
zona interessata al-
l'operazione di «rita-
gio».



Figura 12
Output Programma Pesca. Una volta a disposizione il vettore DD%, può essere utilizzato in vari modi, a seconda anche del disegno sottostante.

```

100 REM pesca
110 LOCATE 24,2:PRINT"attendere"
120 OPTION BASE 1
130 OPEN "sfinge.dat" FOR INPUT AS 1
140 INPUT #1,D:DIM DD%(D)
150 FOR I=1 TO D:INPUT #1,DD%(I)
160 NEXT I:CLOSE #1
170 REM scaricamento
180 SCREEN 1:COLOR 1,0
190 PUT (0,0),DD%,OR
200 PUT (10,99),DD%,PRESET
210 PUT (159,0),DD%,PRESET
220 PUT (169,100),DD%,OR
230 I$=INKEY$:IF I$="" THEN 220
240 CLS:SCREEN 0:WIDTH 80

```

Figura 11 - Listato Programma Pesca. Viene letto il file SFINGE.DAT e caricato su un vettore DD%. Poi questo viene visualizzato con il comando PUT.

di un rettangolo. Purtroppo non esiste la possibilità di muovere questo rettangolo senza provocare «danni» sul disegno sottostante, poiché quando lo spostiamo in un'altra posizione lo dobbiamo cancellare dalla precedente.

Il listato (fig. 5 e out in fig. 6) si divide in tre parti. La prima inizia con un disegno qualsiasi, data la finalità DEMO del programma, e cioè una serie di ellissi concentriche (righe 100-120). La seconda dimensiona e disegna, su questa serie di ellissi, un rettangolo (riga 130-140).

La parte principale è costituita dalla routine che gestisce il dimensionamento e il posizionamento del rettangolo. Queste operazioni avvengono per mezzo di 8 tasti speciali. Le quattro frecce per i movimenti. Due control-freccia per l'allargamento o restringimento in senso orizzontale. Due control-PgUp e Pg-Dn per l'analogia operazione nell'altro senso.

L'entità dello spostamento può essere determinata premendo un tasto numerico, e quindi può andare da 1 a 9. Il riconoscimento del tasto numerico è in riga 160, quello dei tasti speciali, da riga 200 in poi.

Programma COLOR

Un'altra funzionalità essenziale è

quella che permette la definizione dei colori di un disegno «a posteriori». Ovvero si realizza un disegno e solo in fase finale di montaggio della presentazione se ne decide il colore.

Questa è un'opportunità importante in quanto è sempre preferibile uniformare i colori di sfondo di tutte le slide, per dare uniformità anche «cromatica» al discorso.

Il programma COLOR, listato di figura 7 e output di fig. 8, carica, nella video memory a colori del PC IBM, una slide preparata precedentemente. Poi tramite i tasti freccia si modificano i colori background e foreground. Col tasto END si finisce.

I tasti freccia, riconosciuti nella routine da riga 140 a riga 210, modificano i valori numerici CA e CB che vengono passati all'istruzione COLOR CA, CB, che permette il cambio dei colori al volo. I valori numerici possono variare solo tra 0 e 15 per cui lo scorrere dei corrispondenti CA e CB viene ridotto sempre a tale intervallo per mezzo della funzione FN CC(X) di riga 120.

Programma ISOLA

È spesso necessario utilizzare porzioni di slide in altre slide e quindi fare un'operazione di collage prendendo pezzetti qua e là e ricompensandoli. In pratica occorre caricare una slide, identificare la zona rettangolare che interessa, trasferirla su un vettore tramite l'istruzione GET. Poi il vettore viene trasferito su un file sequenziale, il cui primo elemento sia la sua lunghezza.

A questo punto la porzione di disegno è a disposizione per essere prelevata. Il programma ISOLA esegue la prima fase, fino alla scrittura del file sequenziale. Il successivo programma PESCA lo legge e ne riutilizza il contenuto.

Nel programma (listato in fig. 9 e

output in fig. 10) il rettangolo viene identificato direttamente impostandone le coordinate dei vertici (riga 130). Viene calcolata dinamicamente la dimensione necessaria per il vettore tramite una DEF FNA (X,Y) di riga 140.

Come detto il dimensionamento è importantissimo, specie nel caso si utilizzino lo stesso vettore per eseguire vari trasferimenti. È quindi opportuno stabilire al volo quanto deve essere grande il vettore in funzione dell'estensione del disegno.

Viene caricata una Picture (SFINGE.PIC) realizzata in precedenza (è una Sfinge presa da una DEMO dell'Execuvision) sulla quale si verifica la posizione del rettangolo che identifica la porzione (riga 230).

Si esegue subito la GET e il trasferimento su un file sequenziale (SFINGE.DAT) in cui come primo elemento mettiamo la lunghezza del file, necessaria in fase di lettura (righe da 300 a 340). Infine si controlla il buon esito dell'operazione facendo delle PUT del vettore.

Programma PESCA


È il programma complementare al precedente in quanto legge il file sequenziale e ne visualizza il contenuto (listato in fig. 11 e output in fig. 12).

Il dimensionamento del vettore viene eseguito all'interno della routine di lettura file, in quanto necessita del valore dimensione che è il primo valore nel file. La lettura continua con un loop da 1 alla lunghezza del file che produce, elemento per elemento, il caricamento del vettore.

Caricato il vettore può essere riutilizzato con la PUT <coordinate>, <vettore>.

Conclusioni

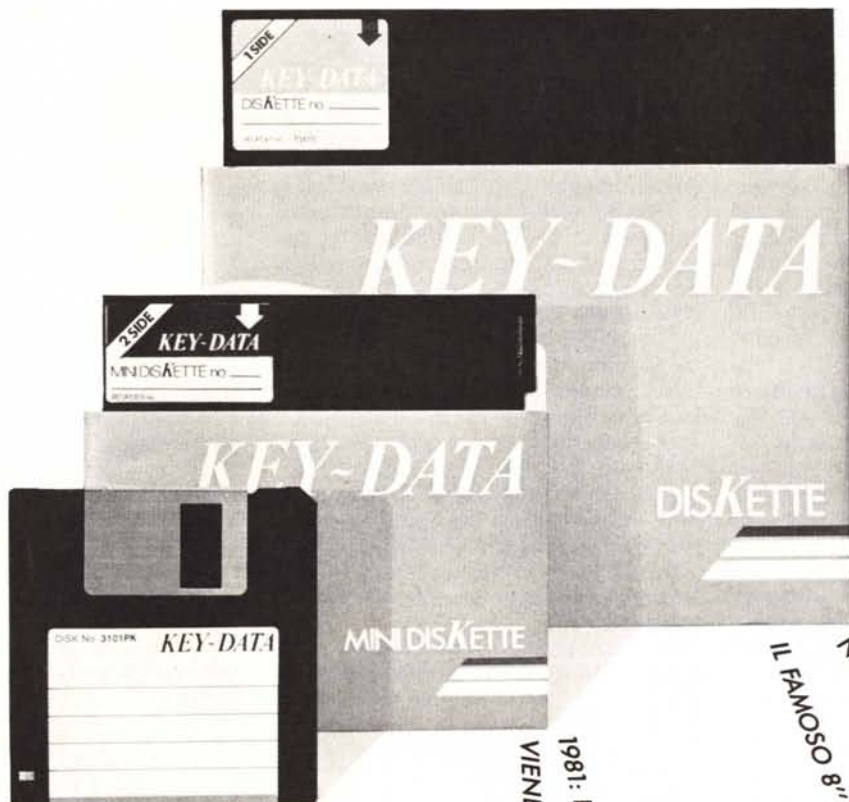
È evidente il legame che unisce i vari programmi presentati, esemplificando tutte funzionalità che servono in un programma autore, che servono addirittura, nella stessa sequenza operativa.

Nella prossima puntata continueremo la trattazione affrontando problematiche un po' più complesse. 

gierre informatica
presenta

K KEY-DATA

L'EVOLUZIONE CHE GIRA NEL TEMPO



IL 3,5" KEY-DATA

NASCE IL PICCOLO MICROFLOPPY

1984: CONTINUA L'EVOLUZIONE

1981: LA RIVOLUZIONE DEI PERSONAL COMPUTERS
VIENE INTRODOTTI SUL MERCATO IL 5 1/4 KEY-DATA

1981: LA RIVOLUZIONE DEI PERSONAL COMPUTERS

NEL LONTANO 1980
NASCEVA IL PRIMO FLOPPY-DISK
IL FAMOSO 8" KEY-DATA



I MINIDISKETTE 5 1/4 KEY-DATA SONO DISPONIBILI
IN VARIE COLORAZIONI

GI-ERRE INFORMATICA s.r.l.
42100 REGGIO EMILIA VIA UMBRIA 36/A TEL. 0522 38655 • 512345
70125 BARI VIA MONTE S. MICHELE 2/B TEL. 080 415975
95100 CATANIA P.ZZA GALATEA 2 TEL. 095 375222