

di Francesco Petroni

Uso avanzato dei Caratteri Grafici in modalità Screen 2 su PC IBM (e compatibili)

Chi segue, anche saltuariamente, questa rubrica di grafica da parecchi anni (magari dal numero 1) avrà notato come da una rubrica di tipo sperimentale in cui venivano suggerite e analizzate problematiche di computer grafica sviluppate poi praticamente sul microcomputer che allora andava per la maggiore (il glorioso Apple II), si sia trasformata in una rubrica anche di analisi di prodotti grafici soprattutto per la macchina che «oggi» va per la maggiore e cioè l'IBM.

Questo perché il mondo del microcomputer si sta trasformando da un mondo di pochi programmatori appassionati in un mondo in cui la maggioranza della popolazione è costituita da utilizzatori di programmi, il cui hobby al limite si è trasformato in un tipico collezionismo di pacchetti con i quali fare a cambio, magari alla ricerca dell'ultima novità fresca di copiatura, pacchetti che poi, per oggettiva mancanza di tempo, non verranno mai esaminati a fondo come ciascun pacchetto in genere merita.

È però indubbio che ambedue le attività, quella di sperimentatore di altrui programmi e quella di fare sperimentazione scrivendo in proprio i programmi, sono interessanti e anche a noi piace alternarle, nel preparare gli articoli.

Per venire al sodo, obbiettivo della puntata è quello di rielaborare delle note tecniche apparse su TECH PC di agosto '86 relative alla modalità di gestire i caratteri alfanumerici sullo SCREEN 2 del PC IBM.

È noto che il PC IBM possiede due modalità di visualizzazione dei caratteri, in una matrice 9 per 14 pixel (modalità monocromatica) e una matrice 8 per 8 (in modalità grafica). È altresì noto che le modalità grafiche sono due, lo SCREEN 1 con una definizione di 320 per 200 pixel, in cui si posso-

no quindi visualizzare 40 per 25 (uguale 1000) caratteri e lo SCREEN di 640 per 200 pixel e quindi in grado di visualizzare le tradizionali 25 righe di 80 caratteri.

Se si lavora in modo grafico sullo SCREEN 2 il sistema più semplice di scrivere testi è quello di usare i classici comandi Basic PRINT e LOCATE. In questo modo però i caratteri possono essere scritti solo nella matrice di 2000

posizioni e possono essere scritti in un solo formato e con una sola orientazione.

La sintassi del LOCATE Y, X accetta valori compresi tra 1 e 24 e tra 1 e 80. Non è quindi possibile utilizzare posizioni intermedie.

Le altre limitazioni insite nell'uso delle istruzioni classiche consistono nell'impossibilità di usare caratteri di differenti formati e nell'impossibilità

```
100 SCREEN 2:KEY OFF:CLS:LINE (0,0)-(639,199),,B
110 A$="Caratteri Orizzontali":X=100:Y=20:FOR A=1 TO LEN(A$)
120 L$=MID$(A$,A,1):GOSUB 200:X=X+8:Y=Y+0:NEXT A
130 A$="Caratteri Verticali":X=65:Y=30:FOR A=1 TO LEN(A$)
140 L$=MID$(A$,A,1):GOSUB 200:X=X+0:Y=Y+8:NEXT A
150 A$="Caratteri Grandi":Y=80:FOR A=1 TO LEN(A$)
160 X=150+A*16:L$=MID$(A$,A,1):GOSUB 300:NEXT A
170 A$="Caratteri Ruotati":X=560:FOR A=1 TO LEN(A$)
180 Y=200-A*8:L$=MID$(A$,A,1):GOSUB 400:NEXT A
190 END
200 REM semplice
210 XC=X MOD 640:IF XC<0 THEN XC=XC+640
220 S=2^(8-(XC MOD 8)):L=INT(XC/8):N=(L+1) MOD 80
230 FOR I=Y-7 TO Y:YC=I MOD 200:IF YC<0 THEN YC=YC+200
240 DEF SEG=&HF000:K=PEEK(&HFA6E+ASC(L$)*8+I-Y+7)*S
250 DEF SEG=&HB800:IF YC MOD 2=0 THEN D=0 ELSE D=&H2000
260 M=INT(YC/2)*80
270 POKE D+M+L,PEEK(D+M+L) OR INT(K/256)
280 POKE D+M+N,PEEK(D+M+N) OR (K-INT(K/256))*256)
290 NEXT I:RETURN
300 REM raddoppio formato
310 DEF SEG=&HF000
320 FOR I=Y-7 TO Y:XC=X MOD 640:IF XC<0 THEN XC=XC+640
330 YC=I MOD 200:IF YC<0 THEN YC=YC+200
340 K=PEEK(&HFA6E+ASC(L$)*8+I-Y+7)
350 FOR L=1 TO 8:M=K MOD 2:N=XC+16-2*L
360 IF M<>0 THEN PSET(N MOD 640,YC):PSET((N+1) MOD 640,YC)
370 K=INT(K/2):NEXT L:NEXT I:RETURN
400 REM rotazione 90 gradi
410 DEF SEG=&HF000:FOR I=1 TO 8
420 K=PEEK(&HFA6E+ASC(L$)*8+I-1)
430 FOR L=1 TO 8:M=K MOD 2
440 XC=(X-16+2*I-1) MOD 640:IF XC<0 THEN XC=XC+640
450 YC=(Y-8+L) MOD 200:IF YC<0 THEN YC=YC+200
460 IF M<>0 THEN PSET(XC,YC):PSET(XC+1,YC)
470 K=INT(K/2):NEXT L:NEXT I:RETURN
```

Figura 1 - Listato programma 1 (Campionario). Il carattere viene prelevato dalla locazione di memoria in cui è allocato (segmento F0000, locazione FA6A e seguenti) e poi o visualizzato direttamente via POKE, oppure manipolato e visualizzato via istruzione grafica PSET.

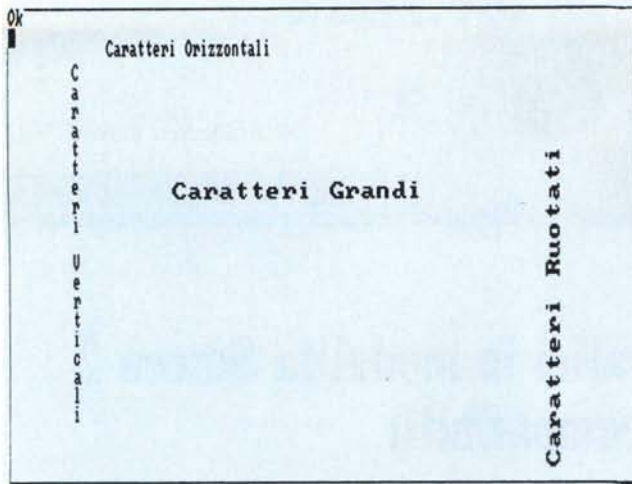


Figura 2 - Output programma 1. Sono raccolte tre routine che permettono di manipolare, in fase di visualizzazione, il set standard di caratteri, oltre l'uso delle istruzioni standard Basic.

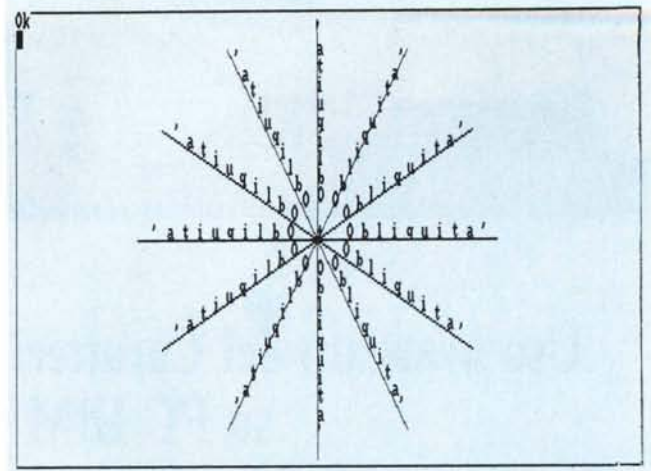


Figura 4 - Output programma 2. È evidente che data una linea obliqua questa può essere percorsa in due direzioni. Nel nostro esempio sono due angoli che differiscono di 180 gradi.

di «ruotare» le scritte nelle varie direzioni.

Se si utilizzano pacchetti grafici specifici, in qualsiasi linguaggio questi siano scritti, si utilizzano le funzionalità di scrittura testi da questi ultimi offerte, in genere ben superiori a quelle offerte dal Basic.

Se invece si lavora in Basic bisogna organizzarsi, o ricorrendo a Tool di generazione e utilizzo caratteri, oppure costruendosi proprie routine TEXT, più o meno complesse.

Quello che proponiamo è il sistema più economico per uscire dalle limitazioni delle istruzioni di Print, in quanto consiste in una serie di routine molto semplici e corte e quindi utilizzabili con facilità in qualsiasi programma grafico.

In pratica si tratta di andare a prelevare i vari caratteri nelle porzioni di memoria in cui sono allocati nella mappa della memoria del PC IBM (e quasi tutti i compatibili) e sottoporli a piccole routine di trasformazione.

Le routine che suggeriamo consentono di posizionare in qualsiasi posizione le scritte, di ingrandirle di un fattore 2, e di ruotarle nelle quattro direzioni, e in più di invertirle, in modo speculare.

Compresa la filosofia delle routine, che sono scritte in Basic e sono particolarmente corte, se ne possono realizzare delle altre per ottenere altri effetti. Il limite residuo è quello di continuare ad utilizzare solo un set di caratteri quello di base nel PC IBM.

Elementi fondamentali

Lo SCREEN 2 del PC IBM permette di visualizzare 640 per 200 pixel. Il SET di caratteri risiede in ROM, alla

locazione F000 e consiste in 128 caratteri, definiti ciascuno per mezzo di 8 byte, ciascuno dei quali rappresenta una riga di otto pixel, per un totale di 64 pixel, in una matrice 8 per 8.

Quindi il problema di prelevare il carattere è risolto da una routine che esegue in sequenza 8 PEEK, nella esatta locazione dove risiede il campione del carattere.

La visualizzazione del carattere invece può essere eseguita con due modalità differenti. La prima per mezzo dell'istruzione POKE, la seconda manipolando il valor ritornato dalla PEEK in modo da «leggere» lo stato dei singoli pixel costituenti le matrici.

La prima soluzione comporta la necessità di «spezzare» in due il byte da visualizzare, proprio per permettere di piazzarlo in qualsiasi posizione sulla griglia 640 per 200 pixel e non solo

sulla griglia di 80 per 25 caratteri.

La seconda soluzione comporta la necessità di elaborare il byte prelevato dal set di caratteri in modo da isolare il singolo pixel. Se questo è acceso si accende anche il pixel in scrittura, nella posizione finale calcolata, e viceversa. Lavorando sul singolo pixel è possibile sofisticare la routine di uscita sottoponendo il pixel stesso a pesanti elaborazioni.

Passiamo a esaminare i programmi presentati.

Primo programma - Campionario listato in figura 1 e output in figura 2

Il primo programma contiene tre routine di scrittura. La prima permette di scrivere il carattere in qualsiasi delle 640 per 200 posizioni permesse dal-

```

100 SCREEN 2:KEY OFF:CLS:LINE (0,0)-(639,199),,B
110 A$="+ Obliquita":X0=320:Y0=100:P=3.1416/180
120 FOR A=0 TO 360 STEP 30
130 A1=A*P:S=2*SIN(A1):C=COS(A1):XS=B*S:YS=B*C:X=X0:Y=Y0
140 FOR B=1 TO LEN(A$):C#=MID$(A$,B,1)
150 GOSUB 170:X=X+XS:Y=Y+YS:NEXT B
151 LINE (320,100)-(X,Y)
152 NEXT A:END
160 END
170 REM semplice
180 XC=X MOD 640:IF XC<0 THEN XC=XC+640
190 S=2^(8-(XC MOD 8)):L=INT(XC/8):N=(L+1) MOD 80
200 FOR I=Y-7 TO Y:YC=I MOD 200:IF YC<0 THEN YC=YC+200
210 DEF SEG=%HF000:K=PEEK(%HFA6E+ASC(C#)*8+I-Y+7)*S
220 DEF SEG=%HB000:IF YC MOD 2=0 THEN D=0 ELSE D=%H2000
230 M=INT(YC/2)*80
240 POKE D+M+L,PEEK(D+M+L) OR INT(K/256)
250 POKE D+M+N,PEEK(D+M+N) OR (K-INT(K/256))*256
260 NEXT I:RETURN

```

Figura 3 - Listato programma 2 (Scritte Oblique). Routine con la quale data una scritta (A\$), un angolo espresso in gradi (A, variabile tra 0 e 360) e una posizione iniziale (X0, Y0), questa viene visualizzata seguendo l'orientamento.

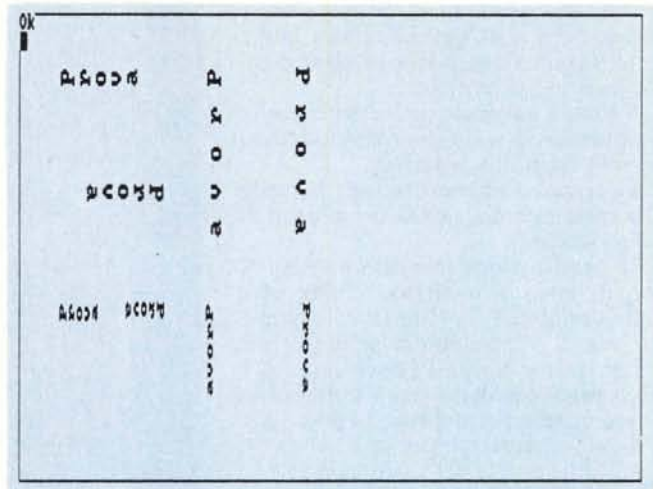


Figura 6 - Output programma 3. L'effetto corsivo è ottenuto facendo scorrere di una certa entità, l'una rispetto all'altra, le varie linee che compongono il carattere. Il risultato, non essendo tali caratteri nati per essere trattati in tale maniera, rimane accettabile solo per piccoli scorrimenti.

Figura 8 - Output del Programma 4. Il carattere base è 8 per 8 pixel, il video è 640 per 200 pixel, per cui, data la routine di cambio formato, è possibile stabilire il numero di caratteri visualizzabili.

```

100 SCREEN 2:KEY OFF:CLS:LINE (0,0)-(639,199),,B
110 A$="Caratteri Corsivi":Y=60
120 FOR B=-.4 TO .4 STEP .2:FOR A=1 TO LEN(A$)
130 X=50+A*20:L$=MID$(A$,A,1):GOSUB 150
140 NEXT A:Y=Y+20:NEXT B:END
150 REM corsivo
160 DEF SEG=&HFA000: SX=0
170 FOR I=Y-7 TO Y:XC=X MOD 640:IF XC<0 THEN XC=XC+640
180 YC=I MOD 200:IF YC<0 THEN YC=YC+200
190 K=PEEK(&HFA6E+ASC(L$)*8+I-Y+7)
200 FOR L=1 TO 8:M=K MOD 2:N=XC+16-2*L
210 X1=N MOD 640+SX:X2=(N+1) MOD 640+SX
220 IF M<>0 THEN PSET(X1, YC):PSET(X2, YC):SX=SX-B
230 K=INT(K/2):NEXT L:NEXT I:RETURN

```

Figura 5 - Listato programma 3 (Scritte in Corsivo). Una volta prelevato il carattere dalla memoria e inserito in variabili, è possibile eseguire qualsiasi tipo di elaborazione su tali variabili, tendente a raggiungere i risultati voluti.

```

100 SCREEN 2:KEY OFF:CLS:A$="Prova":S=1:FL=0:LINE (0,0)-(639,199),,B
110 I1=1:I2=8:I3= 1:L1=1:L2=8:L3= 1:X= 50:Y=30 :SX=16:SY=0 :GOSUB 230
120 I1=8:I2=1:I3=-1:L1=1:L2=8:L3= 1:X=200:Y=30 :SX=0 :SY=16:GOSUB 230
130 FL=1
140 I1=1:I2=8:I3= 1:L1=8:L2=1:L3=-1:X=350:Y=30 :SX=0 :SY=16:GOSUB 230
150 I1=8:I2=1:I3=-1:L1=8:L2=1:L3=-1:X=500:Y=80 :SX=16:SY=0 :GOSUB 230
160 S=0:FL=0
170 I1=1:I2=8:I3= 1:L1=1:L2=8:L3= 1:X= 50:Y=130:SX=8 :SY=0 :GOSUB 230
180 I1=8:I2=1:I3=-1:L1=1:L2=8:L3= 1:X=200:Y=130:SX=0 :SY=8 :GOSUB 230
190 FL=1
200 I1=1:I2=8:I3= 1:L1=8:L2=1:L3=-1:X=350:Y=130:SX=0 :SY=8 :GOSUB 230
210 I1=8:I2=1:I3=-1:L1=8:L2=1:L3=-1:X=500:Y=130:SX=8 :SY=0 :GOSUB 230
220 END
230 FOR B=1 TO LEN(A$):C$=MID$(A$,B,1)
240 GOSUB 250:X=X+SX:Y=Y+SY:NEXT B:RETURN
250 REM rotazione 90 gradi
260 DEF SEG=&HFA000:FOR I=11 TO I2 STEP I3
270 K=PEEK(&HFA6E+ASC(C$)*8+I-1)
280 FOR L=L1 TO L2 STEP L3:M=K MOD 2
290 XC=(X-B+(1+S)*I-1) MOD 640:IF XC<0 THEN XC=XC+640
300 YC=(Y-B+L) MOD 200:IF YC<0 THEN YC=YC+200
310 IF FL=0 THEN IF M<>0 THEN PSET(XC, YC):PSET(XC+S, YC)
320 IF FL=1 THEN IF M<>0 THEN PSET(640-XC, YC):PSET(640-(XC+S), YC)
330 K=INT(K/2):NEXT L:NEXT I:RETURN

```

Figura 7 - Listato programma 4 (Formato e Direzioni). Parametrizzando tutti gli elementi variabili è possibile impostare sia formati (che partono da una matrice 8 per 8), sia direzioni (invertendo X e Y), sia la specularità (prendendo il complemento ad un valore fisso dei valori X o Y).

lo SCREEN 2 del PC IBM, la seconda raddoppia il formato orizzontale del carattere, la terza ruota il carattere invertendo i valori X con quelli Y.

Ciascuna delle tre routine accetta in entrata in valori X, Y che rappresentano le coordinate nelle quali posizionare il carattere, e poi la variabile LS, che contiene il carattere da visualizzare. In ciascuna delle tre routine è presente un LOOP che esegue 8 volte una lettura, per mezzo della PEEK, diretta di ciascuno degli otto byte costituenti il «campione» del carattere.

Poiché i 128 caratteri sono in sequenza, per individuarne l'esatta locazione basta utilizzare il valore numerico fornito dal suo carattere ASCII. Inoltre caratteristica comune alle tre routine è la correzione dei valori X, Y che, se esterni al range (0-639 per la X, 0-199 per la Y), vengono ridotti a valori accettabili.

Esaminiamo le tre routine, che differiscono nella modalità di utilizzazione dei byte prelevati.

Nella prima, il carattere viene visualizzato direttamente nella memoria video (allocata, come noto, a partire dal segmento HB800), per mezzo di due istruzioni POKE. Questo perché ciascun byte iniziale deve essere ripartito tra due locazioni contigue della memoria video (le routine di calcolo possono essere seguite sul listato).

Inoltre poiché i 16.384 byte della memoria video sono suddivisi in strisciate contigue separate da 8192 byte occorre correggere il valore Y risultante aggiungendo tale valore se la Y indica una posizione «dispari».

La seconda routine, prelevato il valore K, con la citata PEEK, lo sottopone, mediante otto successive divisioni per 2, ad un'analisi tendente a leggere

lo stato di ciascun pixel. Se tale pixel è acceso (se il resto della divisione non è 0) lo si può riaccendere in altra posizione, o in più posizioni.

Variante sul tema può essere quella di invertire lo stato dei pixel, ottenendo una scritta in negativo.

La terza ed ultima routine provvede alla rotazione dei caratteri prelevati al solito modo.

In pratica viene invertito il senso X con il senso Y mediante l'inversione delle variabili I, L. Con la I, variabile tra 1 e 8, viene letto il singolo byte orizzontale e con la L, anch'essa tra 1 e 8, vengono scorsi i singoli bit. Poi la I viene usata per definire la posizione X finale e viceversa per la Y.

Secondo programma

Scritte oblique

listato in figura 3 e output in figura 4

Una semplice applicazione della prima routine del primo programma, mediante la quale data una stringa e un suo orientamento è possibile scriverla con la direzione e l'orientamento voluto.

In questi casi l'uso delle istruzioni LOCATE e PRINT potrebbe non seguire perfettamente l'inclinazione delle linee che si vogliono seguire con risultati estetici scadenti. Infatti rispetto a queste ultime condizioni si passa da una definizione di 2.000 posizioni a una di 128.000.

Terzo programma

Scritte in corsivo

listato in figura 5 e output in figura 6

Capito il meccanismo è facile eseguire ulteriori elaborazioni, quella proposta consiste nell'inclinare i caratteri semplicemente facendo scorrere ciascuna riga di 8 bit rispetto a quella inferiore.

L'entità dello scorrimento, e quindi dell'inclinazione del corsivo, può essere resa variabile, in quanto è anch'esso un elemento di calcolo (la grandezza B, variabile tra -0,4 a 0,4 con intervallo di 0,2).

Quarto programma

Variazioni di formato

listato in figura 7 e output in figura 8

Per finire un'altra variazione sul tema, con varie manipolazioni dei caratteri che vengono stirati, accorciati, rivoltati, ecc.

Per concludere questi sistemi hanno il vantaggio di essere facilmente inseribili in qualsiasi programma Basic senza appesantirlo né in termini di righe di programma, né in termini di routine di caricamento di set di caratteri esterni.

Da 10 anni, in queste tre



lettere c'è una grande esperienza.

Gruppo Ethos

SHR

Software.

Questa "S" risolve ogni giorno centinaia di problemi come il tuo, perché significa soluzioni innovative garantite da migliaia di installazioni in Italia.

Hardware.

Questa "H" ti dà la soluzione che non trovi altrove, perché ti offre una gamma completa del miglior hardware con la massima garanzia e assistenza, oltre ogni tua aspettativa.

Ricerca.

Questa "R" significa ricerca: cioè un enorme impegno per essere avanti, con soluzioni sempre aggiornate e sicure nel tempo.

