

grafica

di Francesco Petroni

Ricominciamo da capo

(quarta parte)

La manipolazione delle figure

Ci occupiamo questo mese di un argomento generico, comune a tutte le problematiche di Computer Grafica: la manipolazione delle figure. Anche questa volta non si tratta di un argomento del tutto nuovo, in quanto è stato affrontato altre volte ma in questo numero lo vedremo da un'ottica differente.

Nel processo di visualizzazione tramite computer di un soggetto, indipendentemente dal processo logico matematico con il quale viene definito e identificato lo stesso, c'è un momento in cui questo viene tradotto in un formato adatto ad essere visualizzato (in genere sotto forma di coordinate bidimensionali conformi al formato di uscita (su video o plotter)).

Nel seguito intenderemo per figura un oggetto già definito e già espresso nella forma adatta alla visualizzazione.

Le manipolazioni che può subire una figura sono varie (traslazioni, rotazioni, ingrandimenti e rimpicciolimenti, zoom, scaling, ecc.) e sono operazioni che non comportano alterazioni "interne" della figura che rimane in un certo senso "rigida", qualsiasi operazione subisca.

Tratteremo questo argomento realizzando tre programmi "esercizio" che eseguono varie manipolazioni di una figura qualsiasi predefinita. I due programmi sono scritti in Applesoft. Il terzo programma/ esercizio è invece realizzato utilizzando la funzione WINDOW del Basic avanzato IBM, che permette, in un certo senso, di ottenere risultati analoghi ai precedenti non manipolando la figura ma la finestra in cui la figura viene visualizzata.

Nella seconda parte dell'articolo tratteremo un argomento "complementare" rispetto alla manipolazione delle figure e cioè gli sfondi. Nel senso che se consideriamo le "figure" come oggetti in "primo piano" dietro di questi ci deve essere sempre e comunque uno sfondo, al quale in generale non prestiamo molta attenzione.

Cominciamo quindi a ragionare di "sfondi" in questo numero anche se ci promettiamo di approfondire prossimamente l'argomento.

```
100 REM MANIPOLAZIONE DI UNA FIGURA - UNO
110 HOME
120 REM INIZIALIZZAZIONI
130 N = 16: DIM X(N),Y(N),X1(N),Y1(N),X2(N),Y2(N)
140 P = 3.14159:PP = P / 180:F = 1
150 FOR I = 1 TO 15: READ X(I),Y(I): NEXT I
160 HGR : HCOLOR= 3: H PLOT 0,0 TO 279,0 TO 279,159 TO 0,159 TO 0,0
170 REM DATA
180 DATA 0,0,14,0,14,26,40,0,50,0,50,10,28,32,50,54
190 DATA 50,70,36,70,36,56,20,40,14,40,14,70,0,70
200 REM SEQUENZA MANIPOLAZIONI
210 DATA TRASLAZIONE SEMPLICE
211 TX = 10:TY = 5: GOSUB 810: GOSUB 300: GOSUB 700
220 DATA TRASLAZIONE E ROTAZIONE
221 TX = 125:TY = 5:A = 45: GOSUB 810
222 GOSUB 400: GOSUB 830: GOSUB 300: GOSUB 700
230 DATA TRASLAZIONE E ZOOM
231 TX = 30:TY = 150:F = .6: GOSUB 810
232 GOSUB 300: GOSUB 830: GOSUB 500: GOSUB 700
240 DATA TRASLAZIONE ZOOM E INVERSIONE
241 TX = 220:TY = 75:A = 180: GOSUB 810
242 GOSUB 400: GOSUB 830: GOSUB 300: GOSUB 700
250 DATA TRASLAZIONE ZOOM E ROTAZIONE
251 TX = 210:TY = 90:A = 135:F = 1.2: GOSUB 810
252 GOSUB 400: GOSUB 830: GOSUB 300: GOSUB 830: GOSUB 500: GOSUB 700
260 DATA TRASLAZIONE E ZOOM
261 TX = 85:TY = 80: GOSUB 810
262 GOSUB 300: GOSUB 700
280 FOR K = 1 TO 999: NEXT K: END
300 REM TRASLAZIONE
310 FOR I = 1 TO N
320 X2(I) = X1(I) + TX:Y2(I) = Y1(I) + TY
330 NEXT I: RETURN
400 REM ROTAZIONE
410 C = COS (A * PP):S = SIN (A * PP): FOR I = 1 TO N
420 X2(I) = X1(I) * C - Y1(I) * S
430 Y2(I) = X1(I) * S + Y1(I) * C
450 NEXT I: RETURN
500 REM ZOOM
510 FOR I = 1 TO N
520 X2(I) = X1(I) * F:Y2(I) = Y1(I) * F
530 NEXT I: RETURN
700 REM PLOTTAGGIO
710 HOME : READ A*: VTAB (22): PRINT A*: PRINT
720 PRINT "TX="; INT (TX);" TY="; INT (TY);" A=";
730 PRINT INT (A);" F="; INT (F * 10) / 10;
740 FOR I = 1 TO N - 1
750 H PLOT X2(I),Y2(I) TO X2(I + 1),Y2(I + 1)
760 NEXT I:TX = 0:TY = 0:A = 0:F = 1: RETURN
800 REM PASSAGGIO VALORI
810 FOR I = 1 TO N:X1(I) = X(I):Y1(I) = Y(I): NEXT I: RETURN
820 FOR I = 1 TO N:X2(I) = X1(I):Y2(I) = Y1(I): NEXT I: RETURN
830 FOR I = 1 TO N:X1(I) = X2(I):Y1(I) = Y2(I): NEXT I: RETURN
```

Figura 1 - Manipolazione figura - prog. 1 - Listato

La figura è sempre la stessa, ed è definita con DATA interni al programma. Viene "manipolata" prima di essere visualizzata.



Figura 2 - Manipolazione figura - prog. 1 - Output
A piè pagina vengono riportati i valori correnti TX, TY, traslazione nelle due direzioni, R, angolo di rotazione e F fattore di scala.

Il primo programma di manipolazione

Il primo programma è scritto in Apple-soft (listato in fig. 1 e output in fig. 2) e illustra i tre tipi principali di manipolazione di una figura, combinandoli in vario modo e applicandoli ad una stessa figura.

La inizializzazione comprende la definizione di tre coppie di vettori poiché occorre conservare i valori iniziali e eseguire dei passaggi di valore intermedi in caso di manipolazioni successive.

In X(I), Y(I) vengono conservati i valori iniziali, X1(I), Y1(I) sono i valori iniziali di una manipolazione e X2(I), Y2(I) sono i valori finali. Quindi la routine di visualizzazione utilizza questi ultimi valori.

In riga 140 viene definito il P greco che servirà in fase di rotazione e viene settato F, fattore di scala, che vale 1 come default. In riga 150 vengono letti i DATA relativi alla figura che è un poligono irregolare che assomiglia ad un K.

Da riga 200 sono riportate varie manipolazioni che in pratica consistono nel definire i valori TX, TY (spostamenti nelle due direzioni), A (angolo di rotazione) e F (fattore di Zoom, e cioè per $F > 1$ ingrandimento, per $F < 1$ riduzione), e nel richiamare le routine specifiche.

Le routine sono la TRASLAZIONE (riga 300) la ROTAZIONE (riga 400) e lo ZOOM (riga 500). In quella di rotazione conviene precalcolare i valori seno e coseno prima di eseguire il loop in quanto, come noto, il calcolo delle funzioni trigonometriche rallenta molto il programma. La routine di ZOOM è semplicemente una moltiplicazione per un fattore F.

Come si vede ognuna di queste routine comporta una trasformazione dai valori X1, Y1 ai valori X2, Y2. Quindi in caso di successione di manipolazioni occorre un passaggio di valori da X2, Y2 a X1, Y1 per predisporre i dati alla successiva trasformazione.

Analogamente per la prima operazione occorre trasferire i valori da X, Y a X1, Y1, in modo da predisporre i dati per la manipolazione senza perdere i valori iniziali.

```

100 REM MANIPOLAZIONE DI UNA FIGURA - DUE
110 HOME
120 REM INIZIALIZZAZIONI
130 N = 16: DIM X(N), Y(N), X1(N), Y1(N), X2(N), Y2(N)
140 P = 3.14159: PP = P / 180: F = 1
150 FOR I = 1 TO N: READ X(I), Y(I): NEXT I
160 REM DATA
170 DATA 0, 0, 14, 0, 14, 26, 40, 0, 50, 0, 50, 10, 28, 32, 50, 54
180 DATA 50, 70, 36, 70, 36, 56, 20, 40, 14, 40, 14, 70, 0, 70, 0, 0
190 DATA 50, 70, 36, 70, 36, 56, 20, 40, 14, 40, 14, 70, 0, 70
200 REM SEQUENZA ROTAZIONI
210 FOR A = 0 TO 360 STEP 15
220 GOSUB 300: GOSUB 500: GOSUB 700
230 NEXT A: FOR K = 1 TO 999: NEXT K: END
300 REM ROTAZIONE
310 C = COS(A * PP): S = SIN(A * PP): FOR I = 1 TO N
320 X1(I) = X(I) * C - Y(I) * S
330 Y1(I) = X(I) * S + Y(I) * C
340 NEXT I: RETURN
500 REM SCALING
510 XN = 999: YN = 999: XM = - 999: YM = - 999
520 FOR I = 1 TO N
530 IF X1(I) < XN THEN XN = X1(I)
540 IF X1(I) > XM THEN XM = X1(I)
550 IF Y1(I) < YN THEN YN = Y1(I)
560 IF Y1(I) > YM THEN YM = Y1(I)
570 NEXT I
580 DX = XM - XN: SX = 278 / DX
590 DY = YM - YN: SY = 158 / DY
600 S = SX: IF SY < S THEN S = SY
610 FOR I = 1 TO N
620 X2(I) = (X1(I) - XN - DX / 2) * S + 139.5
630 Y2(I) = (Y1(I) - YN - DY / 2) * S + 79.5
640 NEXT I: RETURN
700 REM PLOTTAGGIO
710 HGR: HCOLOR = 3
720 HPLLOT 0, 0 TO 279, 0 TO 279, 159 TO 0, 159 TO 0, 0
730 HOME: VTAB(22): PRINT "ANGOLO "; A
740 FOR I = 1 TO N - 1
750 HPLLOT X2(I), Y2(I) TO X2(I + 1), Y2(I + 1)
760 NEXT I: RETURN

```

Figura 3 - Manipolazione figura - prog. 2 - Listato

La manipolazione questa volta consiste in una operazione prestabilita, cioè lo SCALING della figura. La figura viene tralata e ingrandita in modo da essere centrale rispetto al video e di riempire al meglio i margini.

Occorre quindi disporre di una serie di routine di passaggio valori, in riga 800 e seguenti, per tutte le operazioni possibili.

In realtà è possibile costruire un'unica routine di trasformazione, che comprenda tutte le possibili manipolazioni in un unico calcolo. Questo semplifica di molto la struttura del programma ma complica la unica routine che rimane.

Infine c'è la routine di plottaggio che è estremamente semplice in quanto si tratta di una figura chiusa e richiede un unico loop. Insieme alla routine di plot c'è una istruzione di PRINT che visualizza, al di sotto della cornice, i valori TX, TY, A, e F correnti.

Il secondo programma di manipolazione

Nel primo programma in pratica abbiamo definito una figura e l'abbiamo manipolata, cioè l'abbiamo spostata, ingrandita, ridotta, ruotata per farla vedere in un certo modo prestabilito, ovvero deciso da noi a priori.

L'operazione di scaling su video fa, in un certo senso, l'operazione inversa, cioè indipendentemente dal formato e dalla posizione della figura, la manipola in modo che appaia centrata e che occupi il massimo rispetto ad un prefissato formato in uscita. Ovvero la trasla, la ingrandisce/rimpicciolisce e (se fosse dato un angolo di rotazione tra sistema di riferimento dei dati della figura e del formato in uscita) la ruota.

Il programma, listato in figura 3 e output in figura 4, è suddiviso in più routine: — la inizializzazione con le consuete o-

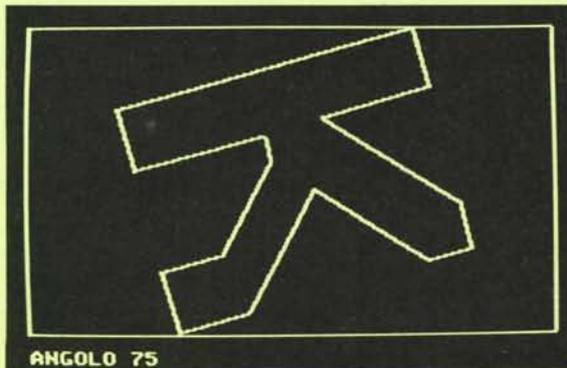


Figura 4 - Manipolazione figura - Prog. 2 - Output
Il loop fa ruotare la figura che, per ogni valore di A viene ingrandita in modo da occupare il formato uscita. Il programma produce una serie di immagini che la foto non può rendere.

perazioni (fino riga 140);

— il main program, nel quale viene eseguito il loop principale sull'angolo A assunto dalla figura (righe 200-230);

— la routine di rotazione, nella quale, per ogni angolo A, vengono calcolate le posizioni assunte dai punti che identificano la figura (righe 300-340);

— la routine di scaling vero e proprio (righe 500-640), che vedremo dopo;

— la routine di plot, che questa volta comprende anche la cancellazione del video e il disegno della cornice (righe 700-750).

La routine di scaling si compone di due parti.

La prima consiste in un loop tramite il quale vengono calcolati i valori massimi e minimi assunti dalle coordinate dei punti della figura (righe 500-570). La ricerca di massimo e di minimo è una operazione molto frequente in programmi di vario tipo. Nel nostro programma, poiché viene eseguita per ogni valore dell'angolo A, occorre settare (con valori -999, 999) XN, YN, XM, YM, rispettivamente minimi e massimi.

Definiti i valori minimi e massimi, si trovano gli intervalli DX e DY, i fattori di scala nelle due direzioni SX e SY, ottenuti facendo il rapporto con l'intervallo massimo voluto in uscita (che per la modalità HGR dell'Apple II è di 279 per 159 pixel).

Poi, per non variare la forma della figura, si sceglie il minore tra i due valori SX e SY, come fattore di scala unico nelle due direzioni. Con questi elementi XN, YN, S, determinati come visto solo dopo un loop su tutti i valori delle coordinate, si esegue un nuovo loop (righe 610-640) per calcolare i valori finali.

In pratica una operazione di scaling, se è manuale, comporta il passaggio di un fattore e quindi si esegue un unico loop per il calcolo dei punti da visualizzare. Se invece lo scaling è automatico, comporta un primo loop per ricercare "automaticamente" i valori XN, YN e S, trovati i quali si può eseguire il calcolo.

Il programma ZOOM

Il significato del termine ZOOM è ben noto ai fotoamatori ed ai cineamatori e deve esserlo anche ai "computergraficamatori".

Lo ZOOM consiste nell'ingrandimento o rimpicciolimento di un oggetto visualizzato. Nella realtà un oggetto varia, ai nostri occhi, la sua dimensione o se si avvicina o se siamo noi che ci avviciniamo.

L'obiettivo ZOOM ingrandisce il soggetto senza necessità di movimenti reciproci. Con il computer la trattazione deve essere tradotta in algoritmi geometrici.

Lo ZOOM è fondamentale in Computer Grafica ed infatti le apparecchiature professionali in genere ne dispongono a livello software di base o addirittura hardware, nel senso che in certi casi è possibile ingrandire o rimpicciolire una figura sul video

azionando direttamente delle apposite manopole. Nei microcomputer queste possibilità non esistono e quindi vanno realizzate via programma e questo significa innanzitutto che per ogni "zoomata" vanno ricalcolati tutti i dati che identificano la figura.

Il programma intitolato "ZOOM", listato in figura 5, riproduce quanto detto.

La variazione della figura è però ottenuta tenendola fissa in una finestra che cambia (riga 310), ed è realizzata tramite una routine che riconosce otto tasti "speciali" della tastiera IBM e cioè le quattro frecce che producono una traslazione nelle quattro direzioni (righe 460-490), PgDn e PgUp per ingrandire/rimpicciolire (righe 440-

```

100 REM          programma grafico di ZOOM
110 REM          realizzato mediante l' istruzione WINDOW
120 REM          del BASIC avanzato
200 REM          : SETTAGGIO INIZIALE
210 X0=-40:Y0=-25: REM ALTO A SINISTRA
220 X1=+40:Y1=+25: REM BASSO A DESTRA
230 S=1.2        : REM FATTORE DI ZOOM
240 T=4:V=3     : REM FATTORI DI MOVIMENTO
250 SCREEN 1
300 REM          DISEGNO INIZIALE
310 CLS:WINDOW (X0,Y0)-(X1,Y1)
320 CIRCLE (-4,-4),20,1:LINE (0,0)-(30,30),2,BF
400 REM          RICONOSCIMENTO TASTI
410 I$=INKEY$: IF LEN(I$)<2 THEN 410 ELSE K=ASC(MID$(I$,2,1))
420 IF K=71 THEN 200                                ? rem HOME
430 IF K=79 THEN CLS:SCREEN 0,0,0:WIDTH 80:END      ? rem END
440 IF K=73 THEN X0=X0*S:X1=X1*S:Y0=Y0*S:Y1=Y1*S   ? rem PAGUP
450 IF K=81 THEN X0=X0/S:X1=X1/S:Y0=Y0/S:Y1=Y1/S   ? rem PAGDOWN
460 IF K=77 THEN X0=X0-T:X1=X1-T                    ? rem LEFT
470 IF K=75 THEN X0=X0+T:X1=X1+T                    ? rem RIGHT
480 IF K=72 THEN Y0=Y0-V:Y1=Y1-V                    ? rem UP
490 IF K=80 THEN Y0=Y0+V:Y1=Y1+V                    ? rem DOWN
500 GOTO 300

```

Figura 5 - Manipolazione figura - prog. 3 - Listato

I tasti speciali della tastiera IBM (frecce, END, HOME, PGUP, ecc.) "passano" una stringa di due caratteri. Il primo dei quali è 0 mentre il secondo dipende dal singolo tasto. Il programma ne riconosce 8, vedi i REM da riga 420 in poi.

```

100 REM sfondo uno
110 SCREEN 3:CLS:PX=10:PY=4:P=3.14159:S=P/10:C=3:LINE (0,0)-(639,399),,B
120 PAINT (2,2),CHR$(0)+CHR$(85)
130 FOR A=0 TO 4*P STEP S:PX=PX+HX:HX=2+32*(ABS((SIN(A)^3)))
140 FOR B=0 TO 3*P STEP S:PY=PY+HY:HY=2+26*(ABS((SIN(B)^3)))
150 LINE (PX,PY)-(PX+HX-4,PY+HY-4),C,BF
160 REM alternanza colore
170 IF C=3 THEN C=0:GOTO 190
180 C=3
190 NEXT B:PY=4:HY=0:NEXT A
200 I$=INKEY$: IF I$="" THEN 200: ELSE SCREEN 0:CLS:END

```

Figura 6 - SFONDO UNO - Listato

L'andamento della larghezza della quadrettatura è data dalla funzione trigonometrica interna ai due loop (orizzontale e verticale, in righe 130-140).

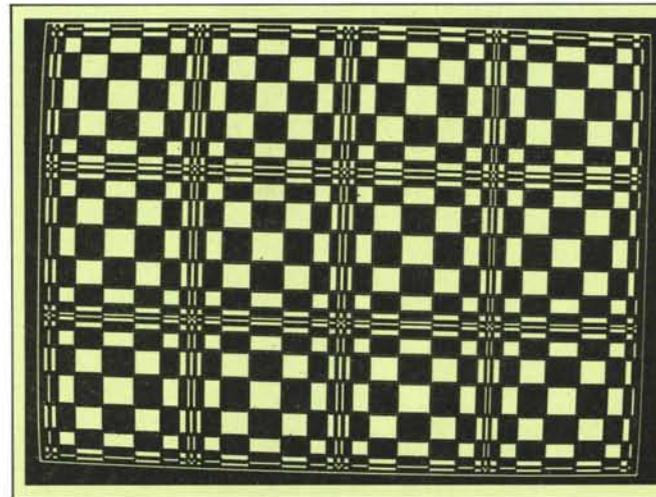


Figura 7 - SFONDO UNO - Output
Per sfondo si intende un disegno con andamento ripetitivo e quindi ripetibile teoricamente all'infinito.

450) END e HOME per la fine o il riinizio del programma.

La routine di disegno viene eseguita dopo ogni variazione delle condizioni di visualizzazione e comporta la cancellazione del video ed il settaggio della finestra (riga 310), nonché il disegno vero e proprio che nel nostro caso è semplicissimo: un cerchio blu e un quadrato rosso (riga 320).

Ovviamente c'è un settaggio iniziale dei valori della finestra e delle quantità di ciascuna variazione (righe 200-240).

Ricordiamo infine che la tastiera del PC IBM comprende codici lunghi 1 carattere e codici lunghi 2 caratteri, per i tasti speciali, e quindi se si vogliono riconoscere questi caratteri occorre individuare gli INKEY\$ di lunghezza superiore a 1 (riga 410).

Gli sfondi

A completamento dell'articolo tratteremo un argomento che definiremo "gli sfondi" che in un certo senso è complementare rispetto al precedente che trattava la manipolazione delle figure.

In generale un disegno comprende due tipi di elementi, le figure in un primo piano e gli sfondi. Se dobbiamo ad esempio disegnare una sfera occorre definire comunque uno sfondo in modo che la sfera venga evidenziata. La soluzione più semplice consiste nel definire un colore per lo sfondo e un colore per la figura in modo che siano adeguatamente in contrasto.

Ci sono delle applicazioni in cui lo sfondo assume molta importanza come ad esempio in taluni video giochi dove lo sfondo è lo scenario fisso o variabile in cui si muovono degli elementi. Un programma di giochi quindi comprende due tipi di routine grafiche, una legata allo scenario in cui si svolge l'azione, l'altra legata alle figure in azione.

In un disegno di tipo matematico o scientifico lo sfondo non riveste particolare importanza (come ad esempio quando si visualizza una superficie spaziale) oppure fa da griglia di riferimento come ad esem-

```

100 REM sfondo due
110 REM inizializzazione
120 CLS:SCREEN 3:WINDOW (12,9)-(-12,-9):LINE (12,9)-(-12,-9),,B
130 SCREEN 3:RH=30:D=40:TH=0:PH=.9
140 S1=SIN(TH):C1=COS(TH):S2=SIN(PH):C2=COS(PH):Z=0
200 REM loop principali
210 FOR X=-10 TO 10:FOR Y=-10 TO 10
220 GOSUB 500:IF Y=-10 THEN PSET(SX,SY):GOTO 240
230 LINE -(SX,SY)
240 NEXT Y:NEXT X
250 FOR Y=-10 TO 10:FOR X=-10 TO 10
260 GOSUB 500:IF X=-10 THEN PSET(SX,SY):GOTO 280
270 LINE -(SX,SY)
280 NEXT X:NEXT Y
300 REM paint
310 FOR Y=-9.5 TO 10:FOR X=-9.5+FL TO 10 STEP 2:GOSUB 500
320 PAINT(SX,SY),CHR$(170)+CHR$(85):NEXT X:IF FL=0 THEN FL=1:GOTO 340
330 IF FL=1 THEN FL=0
340 NEXT Y
400 REM fine
410 I$=INKEY$: IF I$="" THEN 410 ELSE CLS:END
500 REM da tridi a bidi
510 Z=X/4+Y/4 , REM piano inclinato
520 XE=-X*S1+Y*C1:YE=-X*C1*C2-Y*S1*C2+Z*S2:ZE=-X*S2*C1-Y*S1*S2-Z*C2+RH
530 SX=D*XE/ZE:SY=D*YE/ZE:RETURN

```

Figura 8 - SFONDO DUE - Listato

Il programma è tridimensionale. Variando la $Z = Z(X,Y)$ di riga 510 si ottiene una superficie nello spazio.

pio se si traccia una funzione sul piano e se ne vogliono fornire i valori.

Senza scendere nel dettaglio dell'argomento "sfondi", cosa che ci riserviamo di fare in seguito, presentiamo due programmi che realizzano due sfondi, il primo bidimensionale e il secondo tridimensionale.

Per sfondo in questo caso intendiamo un disegno con andamento ripetitivo e che quindi potrebbe essere ripetuto all'infinito. Se poi si tratta di motivi geometrici è relativamente facile trovare delle leggi matematiche che simulano questi motivi.

Debbo confessare che l'idea di questo articolo mi è venuta osservando una carta decorativa a motivi geometrici, con la quale era incartato un regalo di Natale.

Presentiamo due programmi scritti in GWBASIC su di un Olivetti M24 e che sfruttano le eccellenti qualità grafiche di questo computer.

Il primo, intitolato SFOND1, listato in figura 6 e il cui output è in figura 7, realizza una quadratura la cui andamento variabile è ottenuto utilizzando una funzione trigonometrica.

All'inizio vengono definite le costanti e viene disegnato lo schermo di lavoro che viene riempito con una campitura composta con l'istruzione PAINT X, Y, colore.

Poi vengono eseguiti i due loop (righe 130-190) verticale e orizzontale tramite i quali si definiscono i vertici e le larghezze dei singoli elementi rettangolari e che servono per il disegno realizzato tramite la istruzione LINE (X1,Y1) — (X2, Y2), C, BF. C è il colore, BF vuol dire rettangolo pieno. L'alternanza del colore è ottenuta tramite l'utilizzo della C come flag variabile tra 0 e 3.

Il secondo programma (SFOND2, listato in fig. 8 e output in fig. 9) si può considerare tridimensionale e potrebbe simulare la visualizzazione di un pavimento con mattonelle di colori alternati da un punto di osservazione sollevato rispetto al pavimento stesso.

Visto sotto un'altra ottica può considerarsi la vista tridimensionale di quella particolare curva dello spazio che è il piano. Per ottenere variazioni sul tema basta semplicemente inserire una funzione $Z = Z(X,Y)$ più complessa. Provate a vedere quello che succede ponendo $Z = X/10)^2 - (Y/10)^3$.

La funzione $Z = X/4 + Y/4$ rappresenta un piano inclinato rispetto alla base XY ed è un piano in quanto la Z varia linearmente con X e Y.

Non descriveremo il listato del programma in quanto ricalca il programma CAMPIONARIO pubblicato sul n. 37 pag. 85 di MCmicrocomputer al quale rimandiamo i lettori interessati.

MC

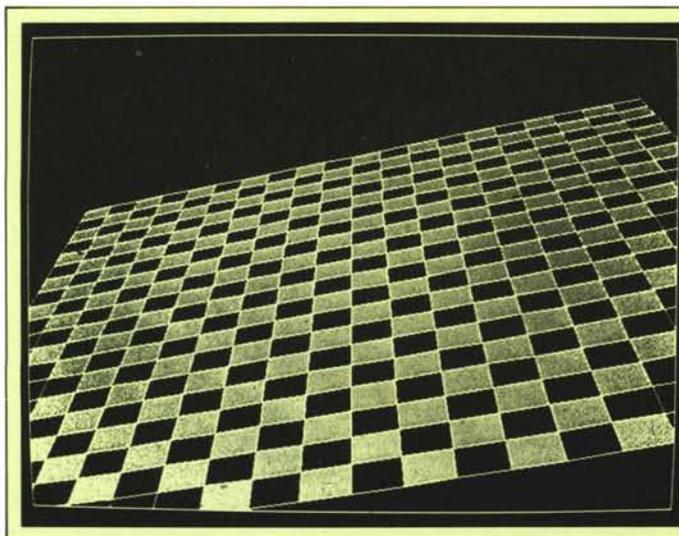


Figura 9 - SFONDO DUE - Output
L'effetto "pavimento" è dovuto alla visualizzazione di una superficie piana posta in posizione obliqua nello spazio. Se poi si varia la funzione Z il pavimento non sarà più orizzontale.

LA FAMIGLIA DEI PERSONAL COMPUTER OLIVETTI



FRIENDLY & COMPATIBLE

Questa famiglia di personal compatibili tra loro e con i più diffusi standard internazionali, non ha rivali per espandibilità e flessibilità. Prestazioni che su altri diventano opzionali, sui personal computer Olivetti sono di serie. Per esempio M24 offre uno schermo ad alta definizione grafica, ricco di 16 toni o di 16 colori e con una risoluzione di 600x400 pixel; mentre la sua unità base dispone di 7 slots di espansione, fatto questo che gli consente di accettare schede di espansione standard anche se utilizza un microprocessore a 16 bit reali (INTEL 8086). Ma ricchi vantaggi offrono anche tutti gli altri modelli.

Basti pensare che tutte le unità base includono sia l'interfaccia seriale che quella parallela. Oppure basti pensare all'ampia gamma di supporti magnetici: floppy da 360 a 720 KB o un'unità hard disk (incorporata o esterna) da 10 MB. La loro compatibilità, inoltre, fa sì che si possa far uso di una grande varietà di software disponibile sul mercato. Come, ad esempio, la libreria PCOS utilizzabile anche su M24. Come le librerie MS-DOS®, CP/M-86® e UCSD-P System®, utilizzabili sia da M20 che da M21 e M24.

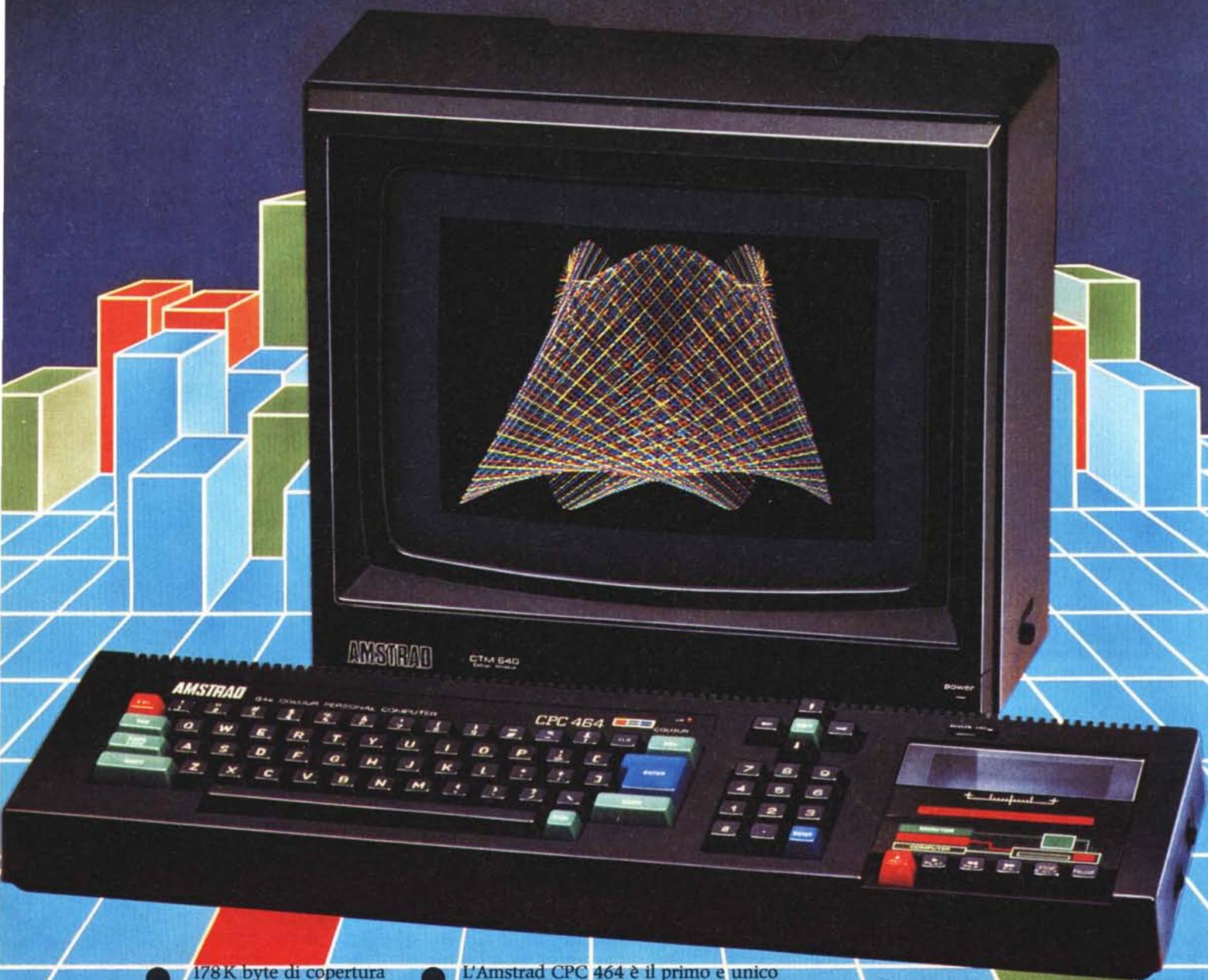
MS-DOS è un marchio Microsoft Corporation
 CP/M-86 è un marchio Digital Research Inc.
 UCSD-P System è un marchio Regents of the University of California

olivetti

Per maggiori informazioni inviare il coupon a: Olivetti, Divisione Personal Computer, Via Meravigli 12, 20123 Milano.

NOME
 INDIRIZZO
 CITTÀ
 TELEFONO

AMSTRAD CPC 464



● 178K byte di copertura per ogni faccia disco con possibilità di unire due Drive per ogni controller. Sistema operativo per floppy in CP/M 2.2 e AMSDOS (Digital Research inc.) Dimensioni mm 75x105x270. Peso kg. 1,6. Dischi 3 pollici.

● L'Amstrad CPC 464 è il primo e unico Personal Computer completo. Monitor a colori o a fosfori verdi. Basic esteso velocissimo. Grafica in alta soluzione (640x200). 80 colonne di scrittura. Suono (3 voci e 7 ottave). Z 80 A. 64K Ram. 32K Rom. Registratore incorporato con velocità di lettura selezionabile. Tastiera professionale (74 tasti). 27 colori utilizzabili.

● Stampa ad impatto monodirezionale. Matrice caratteri 5x7. Velocità 50 caratteri al secondo.

Prezzi

Con Monitor monocrom L. 739.000 + iva
Con Monitor a colori L. 1.030.000 + iva

ESCLUSIVISTA PER L'ITALIA

DEI s.p.a. Largo Porta Nuova, 14 - 24100 BERGAMO - Tel. 035/221031 (5 linee r.a.)