a cura di Francesco Petroni

Software per la tavoletta grafica di MCmicrocomputer

Posizionamento e riconoscimento di menu

In generale ogni applicazione di Computer Grafica utilizza routine, algoritmi, sottoprogrammi comuni ad altre applicazioni. Si veda ad esempio il problema dello scaling, comune a tutte le routine di visualizzazione dei programmi grafici.

Quindi noi nel pubblicare questi articoli, confidiamo in una certa fedeltà da parte dei lettori, che se trovano applicata per la decima volta una routine, ma non ne trovano la spiegazione, la vanno a cercare sui numeri precedenti della rivista.

È anche vero poi che quando si studia un programma o una procedura e ci si trova di fronte ad un problema specifico, in genere per studiarlo a fondo si astrae dalla procedura e si generalizza allo scopo di cercare soluzioni valide anche per altre applicazioni.

Nascono così tutta una serie di programmi, routine, che di per se stesse non servono a molto ma che sono facilmente inseribili in programmi e procedure più vaste.

Già abbiamo presentato parecchie di queste routine (calcolo dell'arcotangente, intersezione tra due segmenti, scaling di una figura, ecc.); in questo numero ne presenteremo altre che poi ci serviranno per la realizzazione del programma di visualizzazione dell'Archivio Grafico, la cui trattazione è cominciata nel numero scorso.

Il nostro obiettivo finale è quello di realizzare una routine generalizzata per il posizionamento in un punto qualsiasi e quindi per l'utilizzazione di un menu rettangolare a 12 opzioni sulla Tavoletta Grafica di MCmicrocomputer.

Articoli sulla tavoletta sono stati pubblicati su numerosi numeri di MC ed a questi rimandiamo chi voglia approfondire l'argomento.

Ricordiamo che la Tavoletta è fornita con un software su disco molto potente ed efficace specie per la elaborazione diretta sul video dell'immagine. Con questi articoli ci ripromettiamo di ampliare la utilizzazione anche ad altri campi della Computer Grafica.

Il nostro problema è esaminato graficamente nella figura 1. La tavoletta va comunque tarata con il programma di calibrazione fornito con il software di base, in modo da fornire valori X, Y compatibili con il formato della pagina grafica dell'Apple II nella zona a quadretti della sua superficie.

Risultano fuori scala le zone menu in quanto questa zona non è utilizzata per la visualizzazione.

Noi vogliamo utilizzare tutto il formato fisico della tavoletta, che se non modifichiamo i dati di taratura, permette una definizione di 280 punti in orizzontale e 230 punti in verticale. E vogliamo posizionare poi il menu rettangolare in un punto qualsiasi della tavoletta, che non si sovrapponga agli altri dati grafici.

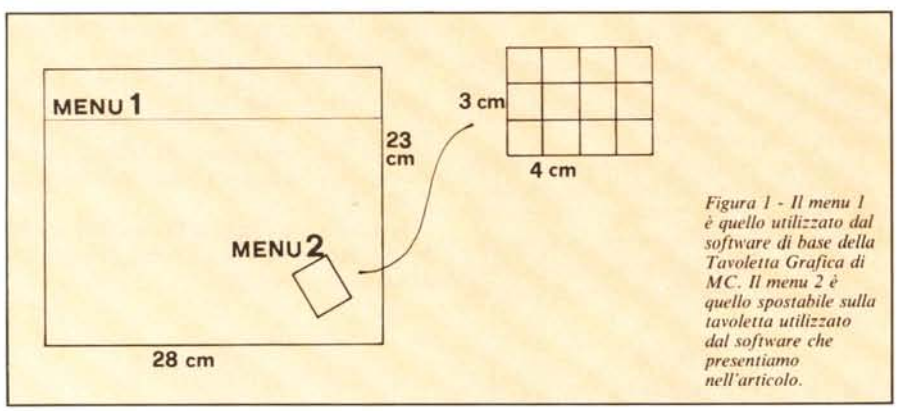

Figura 1 - Il menu 1 è quello utilizzato dal software di base della Tavoletta Grafica di MC. Il menu 2 è quello spostabile sulla tavoletta utilizzato dal software che presentiamo nell'articolo.

Il problema è quello di individuare quando viene puntato il menù e quando viene puntata la zona di immissione dati.

Le routine che presentiamo ci permettono di individuare quando il punto è dentro e quando è fuori del rettangolo, comunque questo sia posizionato.

Se il rettangolo fosse sempre orientato nello stesso modo della tavoletta il problema dell'individuazione del dentro/fuori del menu sarebbe elementare.

La figura 5 è il listato di un programma DEMO che mostra, sul monitor grafico dell'Apple II, un rettangolo le cui dimensioni sono richieste via input e che genera punti casuali sul video.

Esegue un test di dentro/fuori e se il punto casuale è interno al rettangolo esegue un beep.

In questo caso, essendo il rettangolo orientato come gli assi di riferimento del monitor, la routine di test è semplicemente la riga 170, in cui viene verificata la condizione che il punto casuale P(X0,Y0) abbia $X1 < X0 < X2$ e $Y1 < Y0 < Y2$. Se si verificano queste condizioni il punto è interno.

Il problema generalizzato per individuare se un punto è interno od esterno ad un poligono generico ha numerose soluzioni.

Ad esempio si possono contare le intersezioni che una semiretta uscente dal punto P0 forma con i segmenti del poligono. Se le intersezioni sono 0 o un numero pari il punto P0 è esterno, viceversa, se le intersezioni sono un numero dispari il punto è interno.

Questo procedimento è valido anche nel caso di poligoni convessi (vedi fig. 2). L'algoritmo non è molto macchinoso e noi stessi in questi articoli abbiamo più volte trattato l'intersezione tra due rette e/o segmenti.

Un altro metodo è quello di eseguire la somma degli angoli sotto cui è visto ciascun segmento del poligono. Se tale somma è un angolo giro il punto è interno (vedi fig. 3).

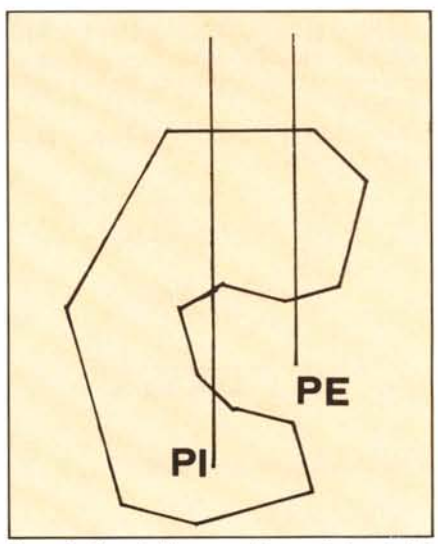

Figura 2 - Metodo del numero di intersezioni per individuare se un punto è esterno (numero 0 o numero pari) o interno ad un poligono (numero dispari).

Questo metodo presenta la solita difficoltà del calcolo dell'arcotangente per quei BASIC che non hanno la funzione Z=ATN(X,Y), che fornisce il quadrante esatto.

Noi non utilizzeremo questi metodi in quanto, dovendoli applicare ad un poligono semplice come il rettangolo, sono eccessivamente macchinosi.

Il nostro metodo per la individuazione della posizione di un punto

(interno/esterno) a un rettangolo comunque posto sul piano consiste nella rotazione del rettangolo e del punto attorno ad un centro arbitrario, eseguita in modo da orientare gli assi come gli assi di riferimento del monitor grafico e di eseguire il test nel caso semplice già visto precedentemente.

Per avvicinarsi ulteriormente al problema abbiamo realizzato un programma che visualizza la rotazione attorno ad un suo vertice, di un rettangolo al variare dell'angolo di rotazione. A noi interessano rotazioni da -90 a +90 gradi, in quanto escludiamo nell'uso pratico che il menù sia messo a rovescio sulla tavoletta. (fig. 4).

Ci interessa il viceversa, ovvero, posizionata la tavoletta e puntati i punti della base superiore, ne è individuata la posizione sul

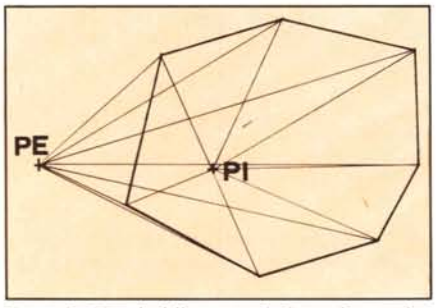

Figura 3 - Metodo della somma degli angoli per individuare se un punto è interno ($\Sigma A = 2.\pi$) o è esterno ($\Sigma A \neq 2.\pi$) a un poligono.

piano ed è possibile trovare i punti dentro/fuori.

Infine abbiamo applicato la routine precedente ad un programma che fa uso della tavoletta grafica.

Praticamente occorre disegnare un rettangolo di 4 cm per 3 cm su un foglio lucido, e suddividerlo in 12 quadratini di 1 cm di lato, come in figura 1.

Posizionato il disegno da digitare sulla tavoletta, coprendo, come detto, anche l'intero formato di 28 cm per 23 cm, si posiziona il menu trasparente in una posizione qualsiasi, che però non interferisca con i dati grafici da immettere.

Ora bisogna comunicare al computer la posizione del menu in modo che siano individuati i punti interni al menu da quelli esterni. La operazione va fatta puntando i due vertici superiori del rettangolo.

Il programma è puramente dimostrativo, ovvero appaiono sul monitor le coordinate dei punti immessi e se il punto è interno al menu appare la scritta Settore n... con il numero del settore individuato.

Sarà il vostro programma applicativo che utilizzerà le 12 opzioni offerte dal menu.

Posizionando il flag G = 1 in riga 100, il nostro programma produrrà una uscita, puramente dimostrativa di quello che succede, sul monitor APPLE II.

Tale uscita fa vedere la traccia del menu e del punto immesso sia in posizione reale, sia nella posizione ruotata cioè nella posi-

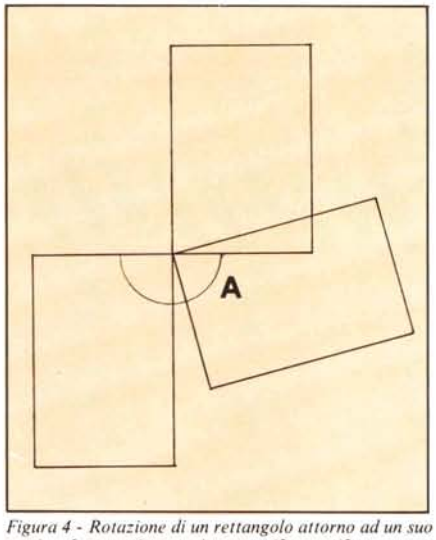

Figura 4 - Rotazione di un rettangolo attorno ad un suo vertice. La rotazione varia tra - $\pi/2$ e $+\pi/2$.

zione usata per la individuazione del dentro/fuori.

Programma rettangolo

Dopo la inizializzazione grafica e il tracciamento della cornice (riga 100), vanno immesse le coordinate del vertice superiore sinistro e del vertice inferiore destro del rettangolo (righe 110 - 120).

Nelle righe 130 - 140 c'è il controllo della posizione reciproca dei punti e il disegno del rettangolo.

Vengono generati e poi tracciati punti casuali sul video (righe 150 - 160). Sulla riga 170 c'è il test dentro/fuori che se positivo genera un beep.

Infine il ritorno alla routine che genera il punto casuale. Listato in figura 5 e output su video in figura 6.

Programma di rotazione

Il programma genera un rettangolo che comprende un reticolo, e lo ruota attorno ad un suo vertice. (Listato fig. 7).

Il programma non ha input, utilizza i dati di riga 10 che rappresentano: P pigreco, P1 lo step di rotazione, X%(1),Y%(1) il primo vertice del rettangolo attorno a cui ruota il rettangolo stesso, L,H le sue due dimensioni.

Nelle righe 20 - 30 vengono calcolati anche gli altri punti del rettangolo.

Nella riga 40 c'è il loop sull'angolo A di rotazione che va da — P/2 a + P/2. Gli angoli A(I) sono gli angoli sotto i quali ciascun punto del rettangolo è visto dal vertice di rotazione.

Il valore dell'angolo è scritto in gradi sotto il disegno (riga 50) e dalla riga 60 alla riga 80 c'è il loop che calcola le posizioni del punto che originariamente aveva coordinata X%(I),Y%(I) e che ruotato di un angolo A assume i valori M%(I),N%(I).

Il disegno del rettangolo e del reticolo è eseguito dalla apposita routine (righe 100 - 150) che consiste nel calcolo degli incrementi X1,X2,Y1,Y2 da dare alle coordinate dei vertici per individuare i punti che delimitano i segmenti del tratteggio. (Output fig. 8).

La riga 90 è alternativa alla 80 e non utilizza la routine 100 - 150, ma traccia solo il perimetro del rettangolo.

Programma menu su video

Prima di applicare tutte le nostre elucubrazioni alla Tavoletta grafica, realizzeremo una applicazione con input da tastiera e output su video. (Listato fig. 9).

Le inizializzazioni (righe 100-110) riguardano i valori L1,L2 lati del rettangolo, il Pigreco e il B$ che produce il beep.

```
100  HGR : HCOLOR= 3: HPLOT 0,0 TO 279,0 TO 279,159 TO 0,159 TO 0,0
110  HOME : VTAB (22): INPUT " PUNTO ALTO A SINISTRA        ";X1,Y1
120  INPUT " PUNTO IN BASSO A DESTRA      ";X2,Y2: HOME
130 B$ =  CHR$ (7): IF X1 =  > X2 OR Y1 =  > Y2 THEN 110
140  HPLOT X1,Y1 TO X2,Y1 TO X2,Y2 TO X1,Y2 TO X1,Y1
150 X0 =  INT ( RND (1) * 270 + 4):Y0 =  INT ( RND (1) * 150 + 4)
160  HPLOT X0 - 3,Y0 TO X0 + 3,Y0: HPLOT X0,Y0 - 3 TO X0,Y0 + 3
170  IF X0 > X1 AND X0 < X2 AND Y0 > Y1 AND Y0 < Y2 THEN  PRINT B$
180  GOTO 150
```

Figura 5 - Listato del programma RETTANGOLO: la condizione dentro/fuori è data dalla riga 170 in quanto i lati del rettangolo sono equiorientati rispetto agli assi di riferimento.

Il posizionamento del menu (righe 120-160) consiste nel puntare il vertice superiore sinistro del menu e il vertice superiore destro. In realtà il secondo punto serve solo per dare la indicazione della direzione, in quanto la distanza tra i due punti è nota e vale L1.

Nella riga 170 c'è il controllo che il menu assuma una angolazione consentita, cioè tra —P/2 e +P/2. Se l'angolazione è errata si ricomincia da capo.

Nelle righe 180-190 c'è il calcolo dell'angolo A che assume il lato principale del rettangolo rispetto agli assi di riferimento.

Nella riga 200 il disegno della cornice.

Nelle righe 210-230 c'è il calcolo delle coordinate dei vari vertici del rettangolo, calcolabili una volta noti L1,L2,A e le coordinate del primo vertice.

Nella riga 240 c'è il disegno del rettangolo.

A questo punto inizia il calcolo dei punti casuali (riga 250) che vengono tracciati (riga 260) e che subiscono nella routine di riga 280 il test dentro/fuori.

Il test dentro/fuori consiste, come detto,

nella rotazione del rettangolo e del punto casuale attorno al primo vertice e quindi il test è analogo a quello del programma Rettangolo.

In riga 290 viene calcolato il punto X5,Y5 che indica il vertice inferiore destro del rettangolo ruotato. Dalla riga 300 alla riga 340 c'è l'individuazione della posizione del punto casuale X0,Y0 rispetto al centro di rotazione X1,Y1. In riga 350 c'è il calcolo dell'angolo di cui va ruotato il punto X0,Y0 per renderlo compatibile con la rotazione del rettangolo.

In riga 360 c'è il calcolo di X,Y nuova posizione del punto casuale, su cui, in riga 370, c'è il test dentro/fuori. La subroutine finisce in riga 380. (L'output del programma è in fig. 10).

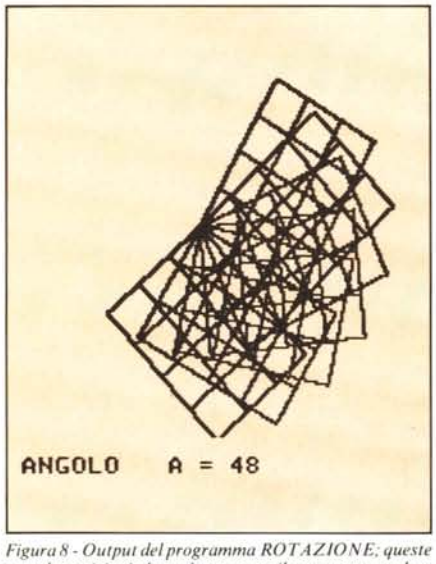

Figura 8 - Output del programma ROTAZIONE; queste sono le posizioni che può assumere il menu rettangolare sulla tavoletta.

Programma menu su tavoletta

Il programma è simile al precedente tranne per il fatto che tutti gli input vanno eseguiti da tavoletta.

Innanzi tutto segnaliamo le routine che permettono l'uso della tavoletta. (Vedi listato fig. 11).

Righe 450-480: routine di inizializzazione che comprende anche il caricamento del Paddle.Code. Tale file binario contiene la routine di lettura della tavoletta nonché i dati di calibrazione ottenuti dal programma di CALIBRAZIONE presente nel software su disco fornito con la tavoletta.

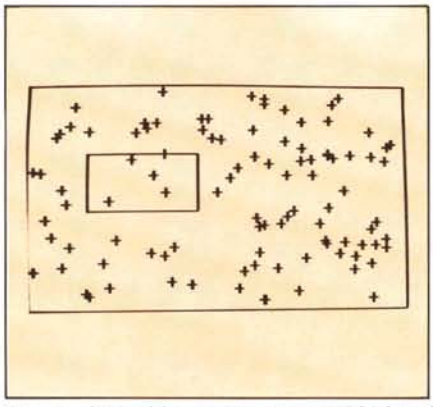

Figura 6 - Output del programma RETTANGOLO; se il punto generato casualmente cade entro il rettangolo viene prodotto un beep.

Righe 490-500: c'è la routine che preleva costantemente i dati dalla tavoletta; se viene premuto il pulsante si esce dalla riga 500 e vengono calcolati, tramite le formule trigonometriche, X%,Y% che sono le coordinate del punto immesso (righe 510-520).

Nelle righe 530-540 c'è la routine di lettura Paddle che richiama la CALL caricata con il Paddle.Code.

Non approfondiamo l'argomento lettura dati da tavoletta in quanto è stato più volte affrontato nei numeri scorsi della rivista.

Nelle righe 110-150 c'è la routine per la immissione della posizione del menu sulla tavoletta e per il controllo della stessa, routine del tutto simile a quella del programma precedente.

Ogni riga che comincia con IF G THEN..., come la riga 160, comporta l'uscita su video grafico, se il flag G è posto uguale a 1 in riga 100.

Una volta definita la posizione del menu, vengono calcolati i suoi quattro vertici e viene disegnato il rettangolo, non ruotato, se G = 1.

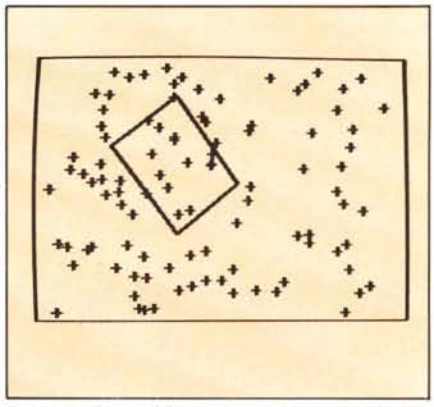

Figura 10 - Output del programma MENU SU SCHERMO. L'output è simile al programma Rettangolo, solo che il menu può assumere qualsiasi inclinazione.

A questo punto può essere richiamata la routine 350-400 che ruota il menu (e lo visualizza) e che calcola le coordinate dei punti centrali dei dodici settori del menu e li carica sulla matrice C%(12,2) e li visualizza.

Tramite tali baricentri, una volta che viene puntato l'interno del menu, viene individuato quale settore dei dodici è stato puntato.

Poiché la precisione del sistema computer-tavoletta è comunque arrotondata all'unità, è bene che si centri il più possibile il

```
10  HGR : HCOLOR= 3:P = 3.1416:P1 = P / 10:X%(1) = 120
    :Y%(1) = 74:L = 72:H = 54
20 X%(2) = X%(1) + L:X%(3) = X%(2):X%(4) = X%(1):X%(5)
    = X%(1)
30 Y%(2) = Y%(1):Y%(3) = Y%(1) + H:Y%(4) = Y%(3):Y%(5)
    = Y%(1)
40 A(1) = 0:A(2) = 0:A(3) =  ATN (H / L):A(4) = P / 2:
    FOR A =  - P / 2 TO P / 2 STEP P1
50  HOME : VTAB (22): PRINT "      ANGOLO   A = "; INT
    (A * 180 / P + .5)
60  FOR I = 1 TO 5:R =  SQR ((X%(I) - X%(1)) ^ 2 + (Y%
    (I) - Y%(1)) ^ 2)
70 M%(I) = X%(1) + R *  COS (A + A(I)):N%(I) = Y%(1) +
    R *  SIN (A + A(I)): NEXT I
80  FOR I = 1 TO 4: GOSUB 100: NEXT A: END
90  FOR I = 1 TO 4: HPLOT M%(I),N%(I) TO M%(I + 1),N%(
    I + 1): NEXT I: NEXT A: END
100  REM  ROUTINE DI TRATTEGGIO
110 X1 = L *  COS (A + A(I)) / 4:Y1 = H *  SIN (A + A(
    I)) / 3
120 X2 =  - L *  SIN (A + A(I)) / 4:Y2 = H *  COS (A +
    A(I)) / 3
130  FOR J = 0 TO 4: HPLOT M%(1) + J * X1,N%(1) + J *
    Y1 TO M%(4) + J * X1,N%(4) + J * Y1: NEXT J
140  FOR J = 0 TO 3: HPLOT M%(1) +  + J * X2,N%(1) + J
     * Y2 TO M%(2) + J * X2,N%(2) + J * Y2: NEXT J
150  RETURN
```

Figura 7 - Listato del programma ROTAZIONE; il programma ha due uscite differenti, quella di riga 80 che genera un reticolo, quella di riga 90 che disegna solo il perimetro del rettangolo che ruota.

```
100  REM  INIZIALIZZAZIONI
110  TEXT : HOME :L1 = 80:L2 = 60:P = 3.14159:B$ =  CHR$ (7)
120  REM  ROUTINE POSIZIONAMENTO MENU
130  VTAB (18): PRINT "PUNTA IL VERTICE IN ALTO A SINISTRA  "
140  INPUT " X1,Y1 ";X1,Y1: HPLOT X1,Y1
150  PRINT " PUNTA IL VERTICE IN ALTO A DESTRA"
160  INPUT " X2,Y2 ";X2,Y2: HOME : HGR2 : HCOLOR= 3
170  IF X2 < X1 AND Y2 < Y1 THEN 110
180  IF X2 - X1 = 0 THEN A = P / 2: GOTO 200
190 A =  ATN ((Y2 - Y1) / (X2 - X1))
200  HPLOT 0,0 TO 279,0 TO 279,191 TO 0,191 TO 0,0
210 X2 = X1 + L1 *  COS (A):Y2 = Y1 + L1 *  SIN (A)
220 X3 = X2 - L2 *  SIN (A):Y3 = Y2 + L2 *  COS (A)
230 X4 = X1 - L2 *  SIN (A):Y4 = Y1 + L2 *  COS (A)
240  HPLOT X1,Y1 TO X2,Y2 TO X3,Y3 TO X4,Y4 TO X1,Y1
250 X0 =  RND (1) * 270 + 5:Y0 =  RND (1) * 181 + 5
260  HPLOT X0 - 3,Y0 TO X0 + 3,Y0: HPLOT X0,Y0 - 3 TO X0,Y0 + 3
270  GOSUB 280: GOTO 250
280  REM  TEST DENTRO/FUORI
290 X5 = X1 + L1:Y5 = Y1 + L2
300 DX = X0 - X1:DY = Y0 - Y1: IF DX = 0 THEN DX =  .00001
310 R0 =  SQR (DX ^ 2 + DY ^ 2)
320 A0 =  ATN (DY / DX)
330  IF DX <  = 0 AND DY <  = 0 THEN A0 = A0 + P
340  IF DX <  = 0 AND DY > 0 THEN A0 = A0 + P
350 A1 = (A - A0)
360 X = X1 + R0 *  COS (A1):Y = Y1 - R0 *  SIN (A1)
370  IF X > X1 AND X < X5 AND Y > Y1 AND Y < Y5 THEN  PRINT B$
380  RETURN
```

Figura 9 - Listato del programma MENU SU SCHERMO.
Il programma chiede di posizionare il rettangolo in un punto qualsiasi dello schermo, fornendo i suoi due vertici superiori. Genera poi i soliti punti casuali che se sono interni generano un beep.

settore scelto del menu ad evitare errori di margine.

Ritornati dalla subroutine c'è la immissione dei punti che avviene in modo alfanumerico e che produce l'ouput di figura 12. I punti vengono numerati in successione e l'input da tavoletta avviene con la solita subroutine di riga 500.

Ogni punto immesso subisce il test dentro/fuori di righe 260-340 e se il test è positivo viene eseguita la routine di individuazione settore.

Di ritorno dai due test (riga 250) viene eseguito un beep e si ritorna alla riga di immissione punto.

Il test dentro/fuori menu è del tutto simile a quello del programma precedente. Produce comunque la visualizzazione del punto e se è positivo rimanda alla subroutine di individuazione del settore.

Questa routine viene eseguita solo quando il punto immesso è interno al menu. L'individuazione del settore è riconducibile ad un problema di minimo. Ovvero si cerca quale dei baricentri dei dodici settori è più vicino al punto, calcolando le dodici distanze e prendendo la più piccola DD e il suo indice KI. Finito il loop su K (in riga 420) il valore KI indica quale settore è stato puntato. Tale informazione viene scritta accanto alle coordinate del punto (riga 440).

Il programma, come detto, ha anche una uscita grafica se il flag G viene posto uguale a 1. Tale uscita, puramente dimostrativa, non è stata formattata. Occorre allora individuare quale porzione della tavoletta non dà errori di formato e lavorare su tale porzione. Vengono visualizzati (vedi fig. 13) il rettangolo originario e quello ruotato, nonché i punti originari e quelli ruotati distinguibili rispettivamente da una x e da un +. MC

```
100 G = 1:S = 1.5: GOSUB 450: REM  FLAG
110  REM  ROUTINE POSIZIONAMENTO MENU'
120  TEXT : HOME : PRINT L$: PRINT "PUNTAMENTO DEL MENU'": PRINT
     : PRINT L$: PRINT : PRINT "PUNTA IL VERTICE IN ALTO A SINIST
     RA": PRINT
130  PRINT " X1,Y1 ";: GOSUB 490:X1 = X%:Y1 = Y%: PRINT  TAB( 12)
     X1; TAB( 18);Y1: PRINT : PRINT "PUNTA IL VERTICE IN ALTO A D
     ESTRA": PRINT : PRINT " X2,Y2 ";: GOSUB 490:X2 = X%:Y2 = Y%:
      PRINT  TAB( 12)X2; TAB( 18);Y1
140  IF X2 < X1 AND Y2 < Y1 THEN 160
150  IF X2 - X1 = 0 THEN A = P / 2: GOTO 180
160  IF G THEN  HGR2 : HCOLOR= 3
170 A1 =  ATN ((Y2 - Y1) / (X2 - X1))
180 X2 = X1 + L1 *  COS (A1):Y2 = Y1 + L1 *  SIN (A1):X3 = X2 - L
     2 *  SIN (A1):Y3 = Y2 + L2 *  COS (A1):X4 = X1 - L2 *  SIN (
     A1):Y4 = Y1 + L2 *  COS (A1)
190  IF G THEN  HPLOT S * X1,S * Y1 TO S * X2,S * Y2 TO S * X3,S *
     Y3 TO S * X4,S * Y4 TO S * X1,S * Y1
200  GOSUB 350
210  REM  IMMISSIONE PUNTI
220  HOME : PRINT L$: PRINT "IMMISSIONE PUNTI  ": PRINT : PRINT L
     $: PRINT : POKE 34,6
230 H = H + 1: PRINT "PUNTO N. ";H; TAB( 10);" ";: GOSUB 490: PRINT
      TAB( 13)X%; TAB( 18);Y%;
240  IF G THEN  HPLOT S * X% - 2,S * Y% - 2 TO S * X% + 2,S * Y% +
     2: HPLOT S * X% - 2,S * Y% + 2 TO S * X% + 2,S * Y% - 2
250 X0 = X%:Y0 = Y%: GOSUB 260: PRINT  CHR$ (7): GOTO 230
260  REM  TEST DENTRO/FUORI
270 DX = X0 - X1:DY = Y0 - Y1: IF DX = 0 THEN DX = .00001
280 R0 =  SQR (DX ^ 2 + DY ^ 2):A0 =  ATN (DY / DX)
290  IF DX <  = 0 AND DY <  = 0 THEN A0 = A0 + P
300  IF DX <  = 0 AND DY > 0 THEN A0 = A0 + P
310 A2 = (A1 - A0):X6 = X1 + R0 *  COS (A2):Y6 = Y1 - R0 *  SIN (
     A2): IF  NOT G THEN 330
320  HPLOT S * X6 - 2,S * Y6 TO S * X6 + 2,S * Y6: HPLOT S * X6,S
      * Y6 - 2 TO S * X6,S * Y6 + 2
330  IF X6 =  > X1 AND X6 <  = X5 AND Y6 =  > Y1 AND Y6 <  = Y5 THEN
      GOSUB 410
340  RETURN
350  REM  BARICENTRI SEZIONI DEL MENU'
360 X5 = X1 + L1:Y5 = Y1 + L2
370  IF G THEN : HPLOT S * X1,S * Y1 TO S * X5,S * Y1 TO S * X5,S
      * Y5 TO S * X1,S * Y5 TO S * X1,S * Y1
380 L3 = L1 / 4:L4 = L2 / 3: FOR L = 1 TO 3: FOR I = 1 TO 4:K = (
     L - 1) * 4 + I:C%(K,1) = X1 - L1 / 8 + L3 * I:C%(K,2) = Y1 -
     L2 / 6 + L4 * L: NEXT I,L
390  IF G THEN : FOR K = 1 TO 12: HPLOT S * C%(K,1),S * C%(K,2): NEXT
     K
400  RETURN
410  REM  INDIVIDUAZIONE SETTORE
420 DD = 10000: FOR K = 1 TO 12:D =  SQR ((C%(K,1) - X6) ^ 2 + (C
     %(K,2) - Y6) ^ 2)
430  IF D < DD THEN DD = D:KI = K
440  NEXT K: PRINT  TAB( 25)"SETT.  "KI; CHR$ (7);: RETURN
450  REM CARICAMENTO DATI CALIBRAZIONE
460 L$ = "----------------------------------------": DIM C%(12,2)
     :P = 3.14149:L1 = 40:L2 = 30
470  PRINT  CHR$ (4)"BLOAD PADDLE.CODE": DEF  FN PK(I) =  PEEK (I
     ) + 256 *  PEEK (I + 1)
480 Z0 =  FN PK(797):Z1 =  FN PK(799):V0 =  FN PK(801):V1 =  FN P
     K(803):PY = 3.14159 / V1:PZ = 3.14159 / V0: RETURN
490  REM  PRELIEVO DATI DALLA TAVOLETTA
500  GOSUB 530: IF  PEEK (49251) > 127 THEN 500
510  REM FORMULE TRIGONOMETRICHE
520 A = (Z0 - P0) * PZ:B = (P1 - Z1) * PY - A:X% = 150 * ( COS (B
     ) -  COS (A)):Y% = 150 * ( SIN (B) +  SIN (A)): RETURN
530  REM LETTURA PADDLES
540  POKE 779,100: CALL 768:P0 = 256 *  PEEK (13) +  PEEK (12): FOR
     T = 1 TO 50: NEXT : POKE 779,101: CALL 768:P1 = 256 *  PEEK
     (13) +  PEEK (12): FOR T = 1 TO 50: NEXT : RETURN
```

Figura 11 - Listato del programma MENU SU TAVOLETTA. Se si pone il flag G = 1 il programma ha una uscita dimostrativa grafica, altrimenti per G = 0 si ha una uscita alfanumerica.

```
----------------------------------------
IMMISSIONE PUNTI
----------------------------------------

PUNTO N.1    -106 76      SETT.  1
PUNTO N.2    5    118
PUNTO N.3    -35  167
PUNTO N.4    11   176
PUNTO N.5    -85  92      SETT.  8
PUNTO N.6    -102 167
PUNTO N.7    82   192
PUNTO N.8    75   133
PUNTO N.9    75   133
PUNTO N.10   -97  99      SETT.  11
PUNTO N.11   20   194
PUNTO N.12   78   122
PUNTO N.13   -39  139
PUNTO N.14   -52  140
PUNTO N.15   -107 85      SETT.  5
PUNTO N.16   18   138
PUNTO N.17   -5   156
PUNTO N.18
```

Figura 12 - Output alfanumerico del programma MENU SU TAVOLETTA. Se il punto immesso ricade all'interno del menu, viene individuato e visualizzato il settore relativo.

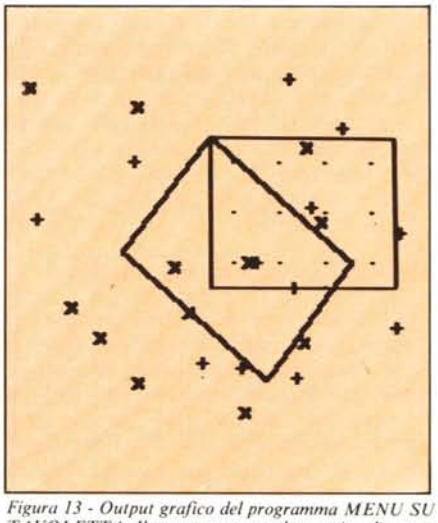

Figura 13 - Output grafico del programma MENU SU TAVOLETTA. Il programma non ha routine di correzione di formato e quindi funziona solo per piccole porzioni della tavoletta. Il segno + indica il punto reale, il segno x il punto ruotato insieme al menu attorno al centro di rotazione.