

Le Basi della Computer Grafica

Qualche giorno fa mi è capitato di scorrere alcuni numeri «storici» di MC (risalenti a parecchi anni fa) e quindi di leggere alcuni vecchi articoli della rubrica di Grafica, nei quali venivano presentate tecniche di calcolo e di disegno con il computer. In questi articoli venivano teorizzati metodi di calcolo e di disegno e venivano proposti dei programmi che li mettevano in pratica. Allora, parliamo dei tempi in cui non era ancora nato, o stava appena nascendo, il PC classe IBM, non esistevano ancora i pacchetti integrati, né tantomeno esistevano applicativi grafici. Occorreva fare tutto da sé, utilizzando direttamente i pochi linguaggi di programmazione, ancora rudimentali, allora disponibili

di Francesco Petroni

Preso dalla nostalgia ho pensato di riproporre, in versione riveduta e corretta, qualcuno di quei programmi, riscrivendoli con uno dei linguaggi oggi più utilizzati: il Visual Basic della Microsoft.

Questa ripresa è interessante anche per il fatto che, paradossalmente, il linguaggio, il Basic, è sempre lo stesso, anche se ora la situazione ambientale, quella offerta da Windows, è del tutto

diversa e la modalità di lavoro, quella offerta dalla programmazione Object Based, Event Driven, è addirittura rivoluzionata. L'articolo risulterà interessante a tutti quelli che si occupano, a vario titolo, di Computer Grafica per il fatto che parla delle tecniche di base, le stesse che poi vengono incorporate in tutti gli applicativi evoluti che siamo oggi abituati ad usare.

I comandi di tracciamento e la predisposizione del foglio di lavoro

Nella prima figura presentiamo una tabellina che mostra i vari comandi Visual Basic da usare in applicazioni grafiche. I comandi di tracciamento sul Form, di Punti e di Linee, che via via useremo, fanno riferimento ad un siste-

| Categoria | Metodo Proprietà Funzione | Parametri | Tipo | Descrizione |
|--|---------------------------|--|-----------|---|
| Definizione caratteristiche del foglio di lavoro | Scale(X1,Y1)-(X2,Y2) | X1 vertice in alto a sinistra Y1 vertice in alto a sinistra X2 vertice in basso a destra Y2 vertice in basso a destra | Metodo | Personalizzazione del sistema di coordinate del Form |
| | ScaleMode | 0 Custom da usare con ScaleWidth, ecc. 1 Twips (1440 per pollice logico) 2 Punti (72 per pollice logico) 3 Pixel 4 Caratteri | Proprietà | Imposta l'unità di misura del Form |
| Comandi di Tracciamento | PSet(X,Y),C | X posizione X Y posizione Y C colore | Metodo | Traccia un Punto nel Form, nella posizione indicata |
| | Line(X1,Y1)-(X2,Y2),C,BF | X1,Y1 posizione primo punto X2,Y2 posizione secondo punto C colore B rettangolo F pieno | Metodo | Traccia una Linea tra due Punti oppure una Linea fino ad un Punto |
| | Circle(X,Y),R,C,.... | X,Y posizione del centro R raggio C colore | Metodo | Traccia una Circonferenza, dato il Centro ed il Raggio |
| Modalità di Tracciamento | DrawMode num | 0 Ricopre 1 Inverso | Proprietà | Definisce la modalità di tracciamento |
| | DrawStyle num | 0 Solido 1 Tratteggiato 2 Puntinato ... | Proprietà | Definisce il tipo di linea |
| Impostazione del Colore | DrawWidth num | | Proprietà | Definisce spessore della linea |
| | RGB(r,g,b) | red valore componente rosso (0-255) green valore comp.nite verde (0-255) blue valore comp.nite blu (0-255) | Funzione | Definisce un colore tra 16 milioni |
| | QBColor(col) | 0 Nero 1 Blu 2 Verde 3 Cyan 4 Rosso 5 Magenta 6 Giallo 7 Bianco | Funzione | Definisce un colore tra 16 |

Figura 1 - Tabella con Oggetti, Metodi, Proprietà e Funzioni Grafiche di MS Visual Basic. Traceremo i nostri disegni nei Form di Visual Basic per i quali imposteremo Proprietà e sui quali agiremo sfruttando Metodi, ad esempio Metodi che eseguono il tracciamento di linee rette e curve. Molto interessanti, le abbiamo trattate più volte nei nostri articoli, sono le due funzioni con cui si definiscono i colori. La più sofisticata è quella che permette di definire uno dei 16 milioni di colori di Windows.

ma di coordinate che va tenuto sotto controllo.

Il tutto è regolato dalla proprietà ScaleMode del Form, che serve per stabilire quale sia l'unità di misura da utilizzare e che deve essere rispettata dai vari comandi di tracciamento.

Facciamo tre esempi, ipotizzando di disporre di un Form abbastanza grande:

- ScaleMode 1 scala in Twips
(1.440.mi di
pollice)
- PSet(100,100) il punto è posto
quasi sul vertice
in alto a sinistra
del Form
- ScaleMode 3 scala in Pixel
(es. in caso di
scheda VGA i pixel
sono 640 per 480)
- PSet(100,100) il punto è posto
ben distante
dal vertice in alto
a sinistra del Form
- Scale(0,0)-(100,100) il Form ha un
sistema
di riferimento
personalizzato
indipendente dalla
sua dimensione
e posizione sul
video
- PSet(100,100) il punto
corrisponde
al vertice inferiore
destro del Form

Nelle applicazioni grafiche orientate al Bitmap converrà, in generale, lavorare con lo ScaleMode 3. Nelle applicazioni grafiche in cui si tracciano delle linee, magari calcolate con processi matematici, conviene personalizzare, con il Metodo Scale, le dimensioni logiche del Form. Tali dimensioni rimangono le stesse anche se si variano le dimensioni fisiche del Form stesso.

Come primo esercizio, totalmente controllabile nella figura 2 (listato ed output), mostriamo un modo per definire le caratteristiche dimensionali di un Form, in pratica un modo per preparare un foglio di disegno. Usiamo due variabili: la prima C, che poniamo pari a 100, e la seconda D, che poniamo pari a 101. Con il metodo Scale inoltre definiamo

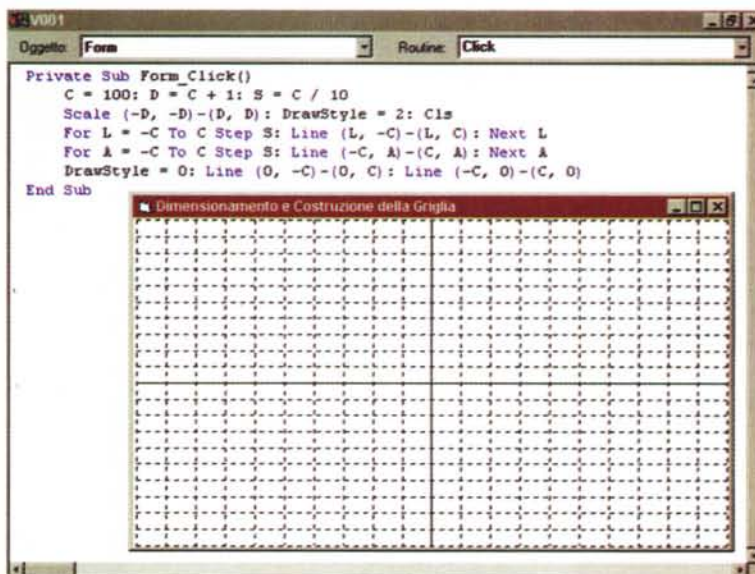


Figura 2 - Predisposizione del foglio di lavoro sul quale disegnare. Quando si disegna, qualsiasi sia lo strumento che si sta utilizzando, occorre, come prima cosa, predisporre il foglio di lavoro. Nelle nostre applicazioni Visual Basic il foglio di lavoro è il Form. Useremo una serie di istruzioni, che potete analizzare nella tabella di figura 1, che servono per definire i limiti del foglio e l'unità di misura che si vuole utilizzare al suo interno. Per le nostre applicazioni grafiche utilizzeremo, come unità di misura, i pixel, che fanno corrispondere un punto del Form ad un punto del video, oppure lavoreremo su un foglio personalizzato personalizzandone le dimensioni con il metodo Scale.



Figura 3 - Sperimentazione sulle istruzioni grafiche di MS Visual Basic. Abbiamo utilizzato, come sfondo del nostro Form, la bitmap con le nuvolette di Windows 95. Su questo sfondo abbiamo tracciato una serie di elementi grafici mettendo in pratica un po' delle istruzioni di tracciamento proposte nella tabella 1. È, tra le altre cose, interessante sperimentare il DrawMode con il quale si definisce il rapporto tra linee tracciate e immagine di sfondo.

come limiti del Form i valori $-D, -D$ e $+D, +D$.

Poi tracciamo una quadrettatura tratteggiata con un passo 10 per 10 e tracciamo i due assi principali con una linea continua.

La routine di Scaling e di Tracciamento della quadrettatura viene eseguita al click sul Form. Questo consente di ingrandire il Form e, con il click, di ristabilire la quadrettatura nella nuova dimensione.

Notare l'uso del DrawStyle per ottenere il tratteggio e l'uso del Metodo Line per tracciare la quadrettatura.

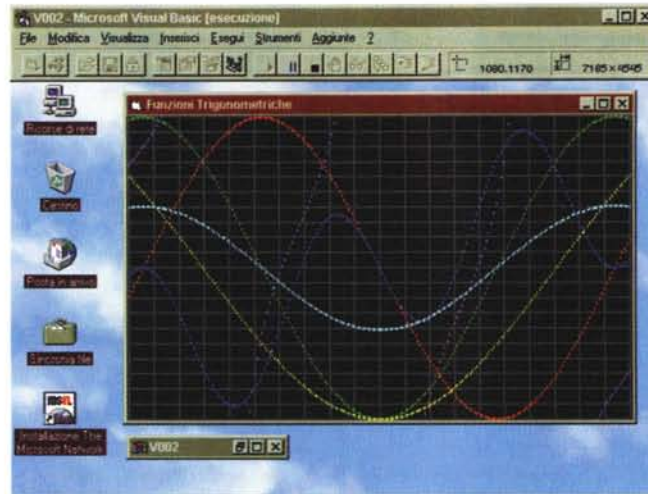


Figura 4 - Curve trigonometriche - Output. Una delle... cose più facili da tracciare sono le Funzioni, in cui i vari punti che individuano le linee generate sono ottenuti con processi di calcolo relativamente semplici. Tra le curve le più facili da costruire e più spettacolari da vedere sono quelle che sfruttano le funzioni trigonometriche, tradizionalmente disponibili sin dai primi Basic.

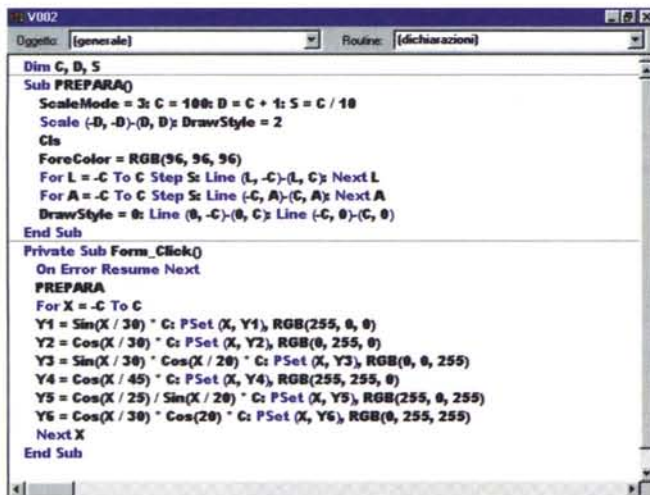


Figura 6 - Una famiglia di curve trigonometriche - Listato ed Output.

Se la funzione da tracciare contiene anche un parametro numerico fisso, la si può tracciare più volte modificando via via proprio questo parametro. Si ottiene in pratica una famiglia di curve. Nel nostro esempio abbiamo utilizzato un semplice contatore per generare la famiglia e per cambiare il colore di ciascuna curva tracciata.

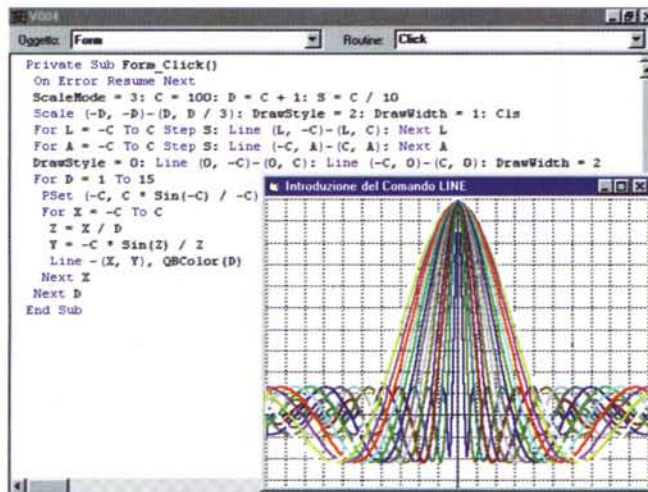


Figura 5 - Curve trigonometriche - Listato. Una funzione di una variabile si esprime nella forma $Y=Y(X)$. Il programma che la visualizza, in pratica, deve fare solo due cose: deve calcolare una serie di coppie di valori X,Y, ad esempio per 100 valori di X 100 valori di Y, e deve visualizzarle come punti P(X,Y), in genere convertendoli nell'unità di misura del Form.

vuol dire dimensione, tipo di tratteggio e tipo di effetto sullo sfondo (proprietà DrawMode).

Poi le due Funzioni con le quali si stabilisce il colore delle linee tracciate. La «vecchia» QBColor, che accetta i 16 colori classici del Basic classico, il QuickBasic. La «nuova» RGB, che consente di dosare le tre quantità dei tre colori fondamentali per ottenere uno tra i 16 milioni di colori possibili.

Sperimentiamo i due Metodi che servono per tracciare Circonferenze o Archi di Circonferenza, e Linee o Rettangoli, vuoti o pieni.

Non sperimentiamo, anche perché non se ne vedrebbe l'effetto, il metodo PSet(X,Y) che traccia un singolo punto nel sistema di riferimento scelto. Cogliamo l'occasione per citare anche il Metodo, che è complementare rispetto al PSet, e che serve per leggere il colore del Punto: Point(X,Y).

Una buona sperimentazione consiste nel provare, con dei cicli For... Next, tutti i possibili parametri delle varie proprietà.

Ad esempio la proprietà DrawMode, che serve per stabilire il comportamento del tracciamento di una linea rispetto all'immagine sottostante, ha ben 16 varianti. Questa abbondanza non consente di prevedere l'effetto che quindi può essere verificato solo sperimentando.

Per tracciare un segmento tra due punti si usa il Metodo Line, che presenta due varianti:

- Linea Da... .. A
- Linea

Nel primo caso servono due coppie di coordinate. Nel secondo caso serve solo un punto di arrivo, in quanto il punto di partenza è l'ultimo punto precedentemente tracciato, o con il precedente comando Line o con un semplice comando PSet.

Quindi quando si tracciano una serie di segmenti occorre decidere se trattare una coppia di punti, oppure un solo punto da collegare al precedente.

A proposito del Metodo Line

Nella figura 3 vediamo il risultato di un esercizio semplicissimo.

Prendiamo un Form nel quale impostiamo come sfondo (proprietà Picture

del Form) la BitMap con le nuvolette di Windows 95. Su questo sfondo tracciamo Cerchi, Linee e Rettangoli sperimentando varie cosette.

Innanzitutto le tre Proprietà che servono per definire il tipo di linea, il che

Un po' di linee classiche Le linee trigonometriche

Il modo più semplice per tracciare linee è quello con il quale i suoi vari punti sono calcolati da apposite routine di calcolo che magari elaborano semplici funzioni matematiche, il cui compito diventa quello di fornire coppie di punti da unire con un segmento.

Il processo è abbastanza elementare. Si crea un ciclo su una variabile, ad esempio sulla X, e poi con tale X si calcola direttamente la funzione $Y=Y(X)$. Poi i casi sono due: o la coppia di punti così determinati è già dimensionata per il Form e allora può essere tracciata direttamente, oppure va convertita con un'operazione di scaling e di traslazione.

Nell'esercizio, cui si riferiscono le due figure 4 (output) e 5 (listato), vediamo un fascio di sei curve tracciate con sei funzioni matematico-trigonometriche.

La routine Prepara è quella che si occupa di predisporre il foglio da disegno e di tracciarne la quadrettatura.

Costituisce una grossa semplificazione il fatto che le routine di calcolo utilizzino gli stessi valori delle coordinate. In altre parole il Form ha un suo sistema di riferimento impostato da -100,-100 a 100,100 e le routine di calcolo eseguono proprio i 201 passi. La variabile utilizzata per contare viene direttamente sfruttata nelle varie formule trigonometriche come valore angolare.

Nella figura 6 vediamo come sia facile costruire una famiglia di curve, un fascio di linee, semplicemente usando un'ulteriore variabile con la quale costruire un ulteriore ciclo. Nel nostro caso la linea viene tracciata 15 volte. Il parametro D, che conta da 1 a 15, viene anche usato per stabilire il colore della linea e come parametro nella funzione trigonometrica. Variando il parametro le linee subiscono anche successive e piccole modifiche di forma.

L'evento MouseMove

I programmi visti fino ad ora generano una serie di linee sulla base di processi di calcolo e sfruttano istruzioni di tracciamento diretto. Il tutto avviene senza nessuna forma di interazione tra programma ed utilizzatore.

Altre tipologie di programmi grafici si basano proprio sulla interazione tra programma ed utilizzatore che opera sulla tastiera oppure interagisce usando il mouse direttamente con il video.

Visual Basic riconosce tantissimi eventi legati all'uso del Mouse, della Tastiera e dei vari oggetti presenti sul Form.

Figura 7 - Evento MouseMove - Serve per il tracciamento a mano.

Uno degli «eventi» più interessanti che riguardano il Mouse è il MouseMove. Quando sul Form si muove il mouse, si cliccano i suoi pulsanti e/o si pigiano i tasti Shift, Ctrl e Alt della tastiera, vengono realizzate delle variabili, legate all'evento MouseMove, che specificano la posizione del Mouse, quale o quali dei suoi tasti sono stati premuti, quale o quali tasti della tastiera sono stati premuti. Si può dire, in generale, che, combinando tra di loro le variabili che indicano lo «stato» del mouse e della tastiera, è possibile definire svariate situazioni operative in base alle quali pilotare un programma grafico.

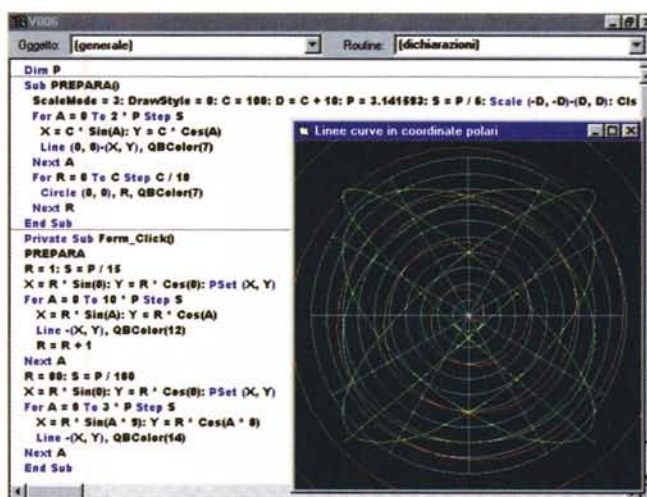
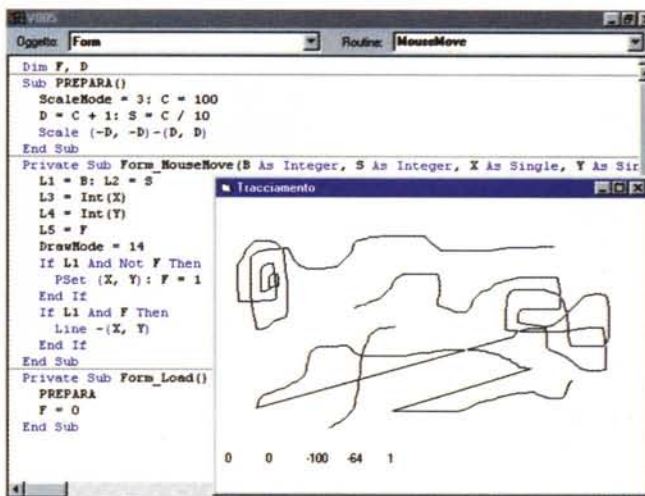


Figura 8 - Uso delle Coordinate Polari - Il ritorno della curva. Esistono funzioni matematiche o trigonometriche in cui ad ogni valore X corrispondono più valori Y. Ne esistono di altre in cui accade il viceversa, ovvero la linea generata passa più volte sullo stesso punto X. In genere queste funzioni vengono meglio rappresentate in un riferimento polare, in cui si utilizzano non una X e una Y, ma un angolo A ed un raggio R. Nella figura vediamo una spirale semplice, che si ottiene cercando di tracciare una circonferenza che... non si chiude mai, perché il suo raggio aumenta via via che

aumenta l'angolo. Un altro tipo di curve intrecciate è quello generato da punti i cui valori X e Y sono calcolati da funzioni trigonometriche, indipendenti tra di loro.

Uno degli eventi più interessanti legati all'uso del Form e del Mouse è il MouseMove. Non è un evento singolo, come il Click su un Pulsante, ma un evento continuo, che si verifica per il semplice fatto che qualcuno sta muovendo il Mouse sul Form. Questo evento produce quattro parametri:

la posizione X del mouse, nel sistema di riferimento del Form
la posizione Y del mouse, nel sistema di riferimento del Form
lo stato dei tasti del Mouse
(0 nessun tasto pigiato, 1 tasto di sinistra, 2 tasto di destra, 3 tutti e due)

lo stato dei tasti speciali della Tastiera: Shift, Ctrl e Alt.

(0 nessun tasto pigiato, 1 tasto Shift, 2 tasto Ctrl, ...).

Intercettando le coordinate relative alla posizione del Mouse e intercettando lo status dei tasti del mouse o della tastiera, è possibile creare semplici programmi di tracciamento diretto, con un minimo di varianti operative. Ne vediamo un semplice esempio in figura 7.

Le linee che ritornano

Non è un titolo di un film thrilling ma un classico problema di tracciamento.

Quando, un paio di esercizi fa, abbiamo tracciato delle linee riferite a funzioni trigonometriche abbiamo creato un contatore per il valore X, e, per ciascun

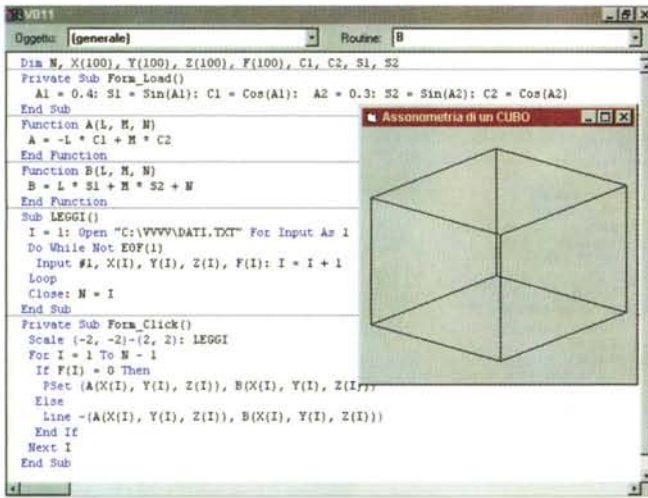
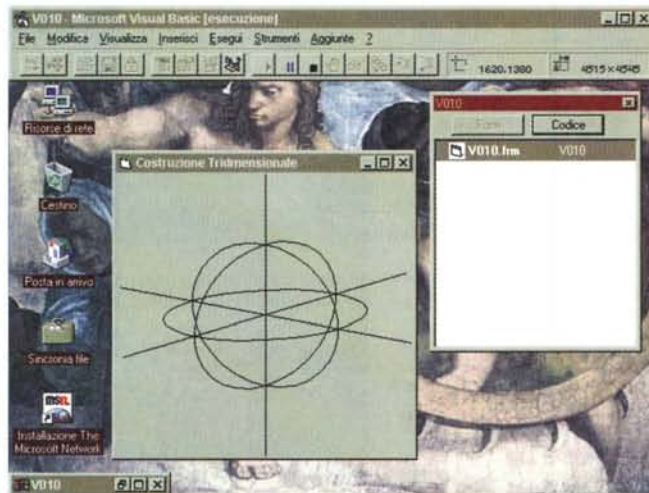


Figura 9 - Primi passi nel 3D - Un cubo in assonometria.

Facciamo una puntatina nella terza dimensione. Molti degli articoli «storici», che abbiamo usato come punto di partenza per i nostri esercizi, sono infatti dedicati alla grafica tridimensionale. In ogni caso occorre trasformare gli elementi, identificati da coordinate spaziali $P_s(X_s, Y_s, Z_s)$, in elementi sul video, che è «solo» bidimensionale e quindi ha solo due coordinate $P_f(X_f, Y_f)$. Vedremo come costruire un'Assonometria, più semplice ma meno realistica, e come costruire una Prospettiva, più complessa ma vicina al modo di vedere dell'occhio umano.

Figura 10 - Entriamo nella terza dimensione - Output.

Tracciamo i tre Assi Cartesiani X,Y,Z e tre Circonferenze poste sui tre Piani XY, XZ e YZ, tutte con centro nell'Origine degli Assi. I punti «spaziali» della circonferenza posta sul piano XY, hanno il valore Z sempre uguale a 0. Le altre due circonferenze si ottengono azzerando la Y e poi la X. Viene costruita una semplice assonometria che ha come soggetti le tre circonferenze.



valore assunto dalla X, abbiamo calcolato un valore Y. Si trattava di casi semplici in quanto ad ogni valore X corrisponde un solo valore Y.

Esistono tipi di linea più complicati, ad esempio quando si «passa» più volte su uno stesso valore X, oltre che più volte sullo stesso valore Y. In tal caso va sicuramente modificato il processo di calcolo.

Uno dei sistemi di calcolo più diffusi è quello che si basa sulle coordinate polari, in cui a variare non sono un valore X e un valore Y di un riferimento cartesiano, ma un valore di un Raggio e il valore di un Angolo.

Un Form Visual Basic lavora sempre in coordinate cartesiane per cui se si debbono eseguire i calcoli in coordinate polari è necessario eseguire anche la conversione tra i due sistemi, che si ba-

sa su due espressioni semplicissime: dato un punto espresso in coordinate polari $P(R,A)$ il suo valore X è $R \cdot \sin(A)$ il suo valore Y è $R \cdot \cos(A)$.

Nell'esercizio di figura 8 prepariamo, in un Form Visual Basic, una griglia polare, con una serie di Raggi (con passo 30 gradi) e una serie di Circonferenze concentriche (con passo 10).

In questo sistema tracciamo due linee, una Spirale e una Linea Intrecciata.

La Spirale più semplice è quella in cui si usano le formule di tracciamento di una circonferenza, solo che via via che aumenta l'Angolo aumenta pure il Raggio e la circonferenza non si «chiude» mai. Anche la linea intrecciata, proposta nello stesso esercizio, si basa sulle formule di tracciamento di una circonferenza. In questo caso basta utilizzare

due valori differenti per l'angolo nel calcolo della X e nel calcolo della Y.

Nel nostro caso usiamo un angolo che varia con un passo di un centesimo di P-greco e che moltiplichiamo per 9 per calcolare la X e per 8 per calcolare la Y. Per ottenere altre linee della stessa famiglia basta cambiare questa coppia di valori.

I problemi della terza dimensione

Facciamo ora una puntatina nella terza dimensione, presentando una serie di esercizi, rielaborati da altri esercizi «storici», che mostrano anche in questo caso le tecniche fondamentali. Li vediamo nelle figure che vanno dalla 9 alla 12. La prima cosa da dire è che mentre nella grafica bidimensionale gli elementi che si tracciano hanno coordinate bidimensionali e si tracciano in un «foglio da disegno» bidimensionale, e quindi non esiste un problema di conversione di coordinate, lo stesso non accade nella grafica 3D. Gli elementi di partenza hanno coordinate tridimensionali ma vengono tracciati su un foglio bidimensionale. Il problema più grande da risolvere diventa proprio quello della conversione delle coordinate.

Scendendo più nel tecnico: per ogni punto caratteristico di ciascun elemento (spaziale) da disegnare, identificabile con le sue tre coordinate $(P_s(X_s, Y_s, Z_s))$, riferite al suo sistema di riferimento, va calcolata la posizione sul foglio da disegno $(P_f(X_f, Y_f))$, che ha tutt'altro sistema di riferimento. Le tecniche di conversione sono materia di studio nelle scuole superiori e sono tantissime in funzione del risultato che si vuol raggiungere. Parleremo delle tre tecniche fondamentali: la Proiezione Ortogonale, l'Assonometria, la Prospettiva.

Le Proiezioni Ortogonali

È la tecnica più semplice in quanto la conversione consiste nel far assumere alle coordinate bidimensionali del foglio da disegno solo una coppia di coordinate tridimensionali, ignorando completamente la terza. Poiché la coordinata «ignorabile» è una su tre, ci sono tre tipi di proiezioni ortogonali:

Se si ignora la coordinata Z: $P_f(X_f, Y_f) = P_s(X_s, Y_s)$ e si ha una proiezione dall'alto

Se si ignora la coordinata Y: $P_f(X_f, Y_f) = P_s(X_s, Z_s)$ e si ha una proiezione di fronte

Se si ignora la coordinata X: $P_f(X_f, Y_f) = P_s(Y_s, Z_s)$ e si ha una proiezione di lato.

Volendo riferire questi sistemi di calcolo al modo di vedere dell'occhio umano

no è come se si guardasse il soggetto da un punto infinitamente distante, posto rispettivamente sull'asse Z, sull'asse Y, sull'asse X. Se si trattasse del disegno di una casa si direbbe pianta, prospetto frontale, prospetto laterale.

Non presentiamo esercizi in quanto possiamo considerare le Proiezioni Ortogonali come un caso particolare di Assonometria o di Prospettiva.

L'Assonometria

Anche nella vista Assonometrica si guarda il soggetto da un punto infinitamente distante, solo che tale punto non è posto su uno degli assi, ma è posto in una posizione qualsiasi nello spazio caratterizzata da due angoli (longitudine e latitudine) che lo localizzano esattamente (dato che la distanza è infinita) nel sistema di riferimento.

In questo caso la conversione si poggia su due formule che calcolano le due coordinate nel foglio Xf e Yf, in funzione della posizione del punto nello spazio Ps(Xs,Ys,Zs). È evidente che in tali due formule entrano i due angoli citati prima.

Altra cosa da dire è che poiché, anche in questo caso, si tratta di un sistema matematico semplificato (nella realtà non esiste un punto a distanza infinita), la vista che si ottiene non è realistica. In particolare le dimensioni dei vari elementi che si vedono non vengono falsate (come invece avviene nella prospettiva). Nel nostro esercizio un lato posteriore del Cubo appare di uguale lunghezza rispetto ad un lato anteriore.

L'aspetto più interessante di tutto il discorso consiste indubbiamente nella realizzazione della formula che, una volta standardizzata, può essere usata per tutti gli esperimenti che si vuole.

Detti A1 e A2 i due angoli, le formule di conversione diventano:

$$X_d = -X_s * \cos(A1) + Y_s * \cos(A2)$$

$$Y_d = X_s * \sin(A1) + Y_s * \sin(A2) + Z_s$$

Nelle routine di calcolo i valori trigonometrici (SIN(A1), SIN(A2), COS(A1), COS(A2)) rimangono sempre gli stessi per cui possono essere «parcheggiati» in variabili precalcolate allo scopo di velocizzare l'esecuzione.

In figura 9 vediamo una prima applicazione di questa formula riferita al tracciamento di un Cubo nello spazio «assonometrico».

Dobbiamo però precisare la modalità con cui sono stati indicati i valori delle coordinate che identificano il Cubo.

Le Coordinate degli elementi tridimensionali

Come nel caso dei grafici bidimensionali esistono due possibilità: che i dati

Figura 11 - Entriamo nella terza dimensione. Listato.

Lo scopo di questi esercizi è introduttivo anche perché vengono affrontati solo alcuni dei problemi connessi con la produzione di disegni tridimensionali. I nostri «oggetti» sono molto semplici, e vengono tracciati in modalità «wireframe», e quindi si vedono linee e non superfici. Un altro dei problemi classici, che non affrontiamo, è costituito dall'eliminazione delle linee nascoste, che l'occhio umano non vede perché poste dietro rispetto agli elementi in primo piano. Questo implica la necessità di lavorare per superfici.

```

V010
Oggetto: Ferra
Routine: Click

Dim P, C1, C2, S1, S2
Function AdL, M, N
  A = -L * C1 + M * C2
End Function
Function BdL, M, N
  B = L * S1 + M * S2 + N
End Function
Sub PREPARA
  X = 2: Y = 0: Z = 0: PSet (AOC, Y, Z), BOC, Y, Z: X = -2: Line -(AOC, Y, Z), BOC, Y, Z ' ASSE X
  X = 0: Y = 2: Z = 0: PSet (AOC, Y, Z), BOC, Y, Z: Y = -2: Line -(AOC, Y, Z), BOC, Y, Z ' ASSE Y
  X = 0: Y = 0: Z = 2: PSet (AOC, Y, Z), BOC, Y, Z: Z = -2: Line -(AOC, Y, Z), BOC, Y, Z ' ASSE Z
End Sub
Private Sub Form_Click()
  Scale (-2, -2, 2, 2): PREPARA
  X = Sin(Q): Y = Cos(Q): Z = 0: PSet (AOC, Y, Z), BOC, Y, Z ' CERCHIO SU X,Y
  For Q = 0 To 2 * P Step P / 15: X = Sin(Q): Y = Cos(Q): Z = 0
  Line -(AOC, Y, Z), BOC, Y, Z: Next Q
  X = Sin(Q): Y = 0: Z = Cos(Q): PSet (AOC, Y, Z), BOC, Y, Z ' CERCHIO SU X,Z
  For Q = 0 To 2 * P Step P / 15: X = Sin(Q): Y = 0: Z = Cos(Q)
  Line -(AOC, Y, Z), BOC, Y, Z: Next Q
  X = 0: Y = Sin(Q): Z = Cos(Q): PSet (AOC, Y, Z), BOC, Y, Z ' CERCHIO SU Y,Z
  For Q = 0 To 2 * P Step P / 15: X = 0: Y = Sin(Q): Z = Cos(Q)
  Line -(AOC, Y, Z), BOC, Y, Z: Next Q
End Sub
Private Sub Form_Load()
  P = 3.14159: A1 = 0.3: S1 = Sin(A1): C1 = Cos(A1): A2 = 0.2: S2 = Sin(A2): C2 = Cos(A2)
End Sub
    
```

```

V013
Oggetto: Ferra
Routine: Click

Private Sub Form_Click()
  Scale (-1.2, -1.2, 1.2, 1.2): C1s
  X = 0: Y = 0: Z = -2: PSet (AOC, Y, Z), BOC, Y, Z) ' ASSI
  X = 0: Y = 0: Z = 2: Line -(AOC, Y, Z), BOC, Y, Z), QBColor(15)
  For M = 0 To 2 * P Step P / 15 ' PARALLELI
  X = Cos(M) * Cos(Q)
  Y = Cos(M) * Sin(Q)
  Z = Sin(M)
  PSet (AOC, Y, Z), BOC, Y, Z)
  For N = 0 To 2 * P Step P / 15 ' MERIDIANI
  X = Cos(M) * Cos(N)
  Y = Cos(M) * Sin(N)
  Z = Sin(M)
  Line -(AOC, Y, Z), BOC, Y, Z)
  Next N: Next M
  For M = 0 To 2 * P Step P / 15 ' MERIDIANI
  X = Cos(M) * Cos(N)
  Y = Cos(M) * Sin(N)
  Z = Sin(M)
  Line -(AOC, Y, Z), BOC, Y, Z)
  Next N: Next M
End Sub
    
```

Figura 12 - Una Sfera in assonometria - Listato ed Output.

Abbiamo assemblato il motore di calcolo dei punti in assonometria con una... carrozzeria sferica. Per calcolare la sfera abbiamo usato due cicli che calcolano i due angoli, M e N, variabili da 0 a 2 P-Greco, e che rappresentano Longitudine e Latitudine del generico punto. Con tre formule abbiamo convertito i valori P(R,M,N) in valori Ps(Xs,Ys,Zs). A questo punto abbiamo eseguito l'aggancio con le routine viste prima.

da tradurre in disegno provengano da calcoli, che i dati provengano da un file.

Per tracciare il cubo ipotizziamo questa seconda evenienza.

In pratica abbiamo memorizzato i valori delle coordinate dei vari vertici del cubo in un file (li vedete in figura 14) che poi viene letto con le classiche istruzioni Basic per la lettura di file sequenziali (OPEN...). Il file, per ciascun vertice del cubo, prevede quattro valori: le tre coordinate X, Y e Z ed un flag che indica se quel punto è collegato al punto precedente (il valore è 1) o se costituisce l'inizio di una nuova linea (il valore è 0).

In questa maniera si simula una penna che traccia, punto dopo punto, la linea, si solleva se la linea finisce e si abbassa se inizia un'altra linea.

I valori delle coordinate sono sempli-

cissimi (0,1,2,-1,-2) per semplificare al massimo il calcolo e il foglio da disegno è già adeguatamente dimensionato.

Assonometrie su linee calcolate

Nei successivi due esercizi proponiamo assonometrie ottenute con routine di calcolo.

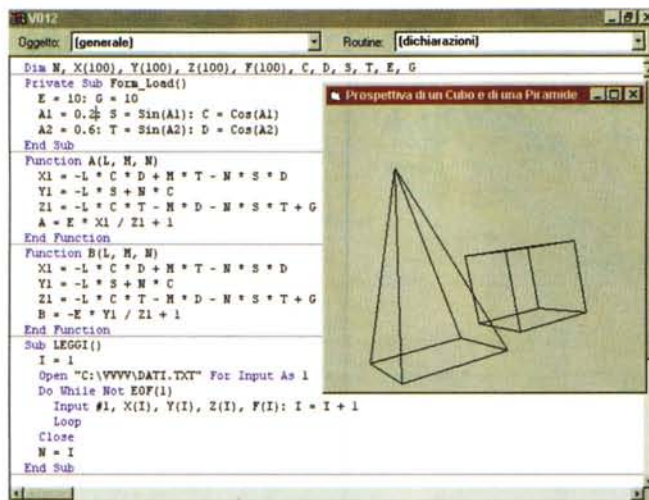
Il primo, che vediamo in figura 10 e 11, è elementare, e mostra i tre assi cartesiani e tre circonferenze, che hanno come centro l'origine del sistema di riferimento e che giacciono ciascuna su ogni piano, XY, XZ e YZ.

Rispetto all'esercizio di prima rimangono inalterate le routine di calcolo delle coordinate sul foglio di disegno, date le coordinate del singolo punto e dati i due angoli dell'assonometria. Le routine sono state, per comodità, inglobate in

due funzioni. Nel secondo esercizio (figura 12) viene costruita una sfera, con il sistema dei meridiani e dei paralleli. Attenzione che nella figura, nella parte con il listato, non appaiono le funzioni, che però sono le stesse create nell'esercizio precedente.

L'assonometria mostra meglio le dimensioni reali degli oggetti, ma falsa il loro aspetto che non è realistico come quello mostrato in una prospettiva.

Se si visualizza una sfera, che è piena di curve, la differenza tra le due modalità di visualizzazione non è molto evidente.



no le varie distanze tra osservatore, piano del disegno e soggetto disegnato.

Figura 13 - La Prospettiva è un'altra cosa - Una vista.

Il calcolo della Prospettiva, al contrario di quello dell'Assonometria, tiene conto della posizione dell'osservatore rispetto a quella dell'oggetto visto. In altre parole l'oggetto appare tanto più piccolo quanto più è distante dall'osservatore. Nella figura vediamo le due funzioni A e B, che vanno sostituite a quelle che producono l'assonometria. In queste due funzioni, che non dettagliamo dal punto di vista trigonometrico, entrano in gioco anche due parametri, che abbiamo chiamato E e G, che indica-

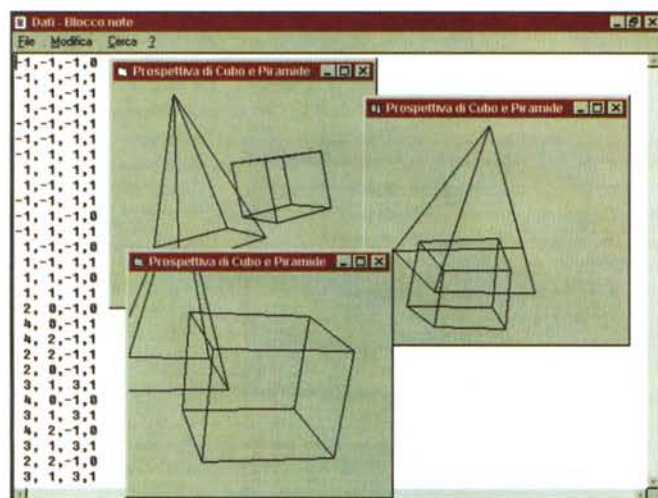


Figura 14 - La Prospettiva è un'altra cosa - I dati e tre altre viste.

Dato un soggetto, nel nostro caso un Cubo e una Piramide, lo si può vedere in infiniti modi in relazione alla distanza dell'osservatore, alla direzione e all'orientamento verso il quale l'osservatore punta lo sguardo (potrebbe anche guardare da un'altra parte, oppure mettersi a testa in giù). Proponiamo un collage che mostra tre viste del soggetto. Sullo sfondo invece i dati che costituiscono il soggetto. Come leggerli dal file testuale in cui risiedono è descritto nel testo.

cazione, rispetto al calcolo dell'Assonometria, cambiano solo le funzioni di conversione che diventano leggermente più complesse.

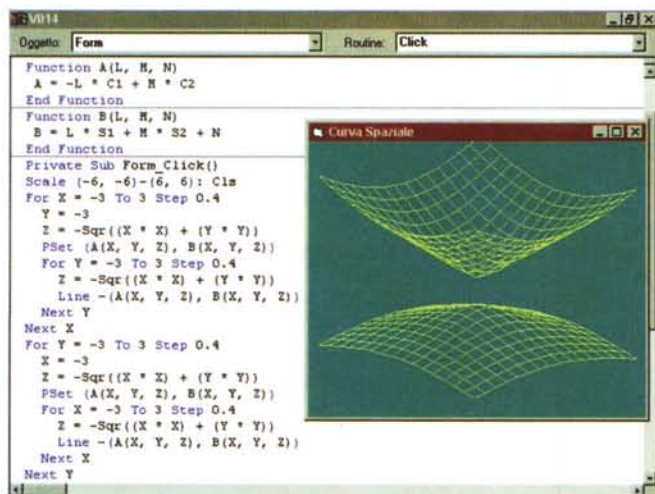
Risulterà banale, una volta che si abbiano ambedue i due gruppi di funzioni, creare un programma in cui al momento della visualizzazione si scelga quale dei due metodi usare.

Le funzioni per il calcolo della Prospettiva Semplificata le vedete in figura 13, e non le commentiamo.

Le applichiamo allo stesso disegno di prima (il cubo) al quale abbiamo aggiunto una piramide.

Nella figura 14, un collage, vediamo il nostro soggetto «ripreso» da tre punti di vista differenti e, sullo sfondo, i dati numerici con le coordinate dei punti.

Figura 15 - Realizzazione di Curve più ardite. Il passo fondamentale è la messa a punto delle routine di visualizzazione (assonometria, prospettiva semplificata, prospettiva completa) e delle routine di lettura dei dati, se il soggetto non è esprimibile con una formula, o delle routine di calcolo della formula. Il passo successivo è quello di trovare una funzione che produca un bel disegno oppure quello di caricare un volume maggiore di dati che rappresentino le coordinate del nostro soggetto.



La prospettiva è un'altra cosa

Come appena visto, le formule che calcolano i punti dell'Assonometria sono semplici, sono facilmente inglobabili

in due funzioni, sono facilmente riutilizzabili.

Il passo successivo consiste nel realizzare la Prospettiva, ed è bene chiarire subito che, dal punto di vista dell'appli-

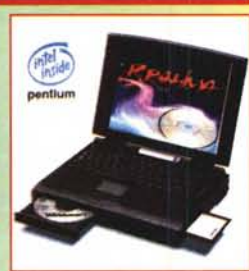
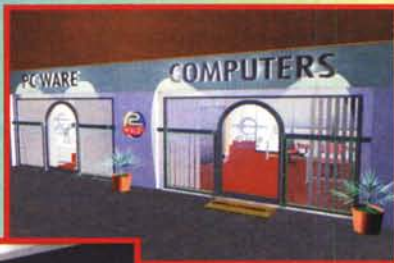
Conclusioni

Queste sono le procedure di base, le stesse che troviamo, già incorporate ed opportunamente implementate, in tutti i prodotti grafici di tipo vettoriale, sia in quelli bidimensionali che in quelli tridimensionali.

Dal canto nostro possiamo creare disegni più complessi usando o funzioni più complicate (ad esempio la superficie mostrata in figura 15) o maneggiando più punti, ma sfruttando sempre lo stesso «motorino» per la visualizzazione.

Esistono ovviamente altre tecniche di base, che non abbiamo trattato, ad esempio quelle che servono per lavorare per superfici, e non per linee, oppure quelle che servono per individuare, e quindi eliminare dal disegno, le linee nascoste in una visualizzazione 3D.

Si tratta di tecniche di secondo livello, molto più complesse di quelle viste in questo articolo, il cui scopo è spesso quello di realizzare una vista «realistica» del soggetto, e che quindi cercano di simulare il più possibile il funzionamento dell'occhio umano.



NOTEBOOK OYSTER

KALA' - LCD COLOR 10,5" 640x480x256C - AUDIO 8 BIT - TRACK POINT

| CPU | HD420MB | HARD DISK 810MB | |
|------------|---------|------------------------------|---------------|
| 486DX2-66 | 2.440 | ESPANSIONE MEMORIA 4MB / 8MB | + 250 / + 400 |
| 486DX4-100 | 2.490 | ESPANSIONE MEMORIA 16MB | + 790 |

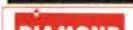
**BRAHMA - LCD COLOR 11,4" 800x600x256C - 8MB - AUDIO 16 BIT
CD-ROM 4X INTEGRATO - TOUCH PAD - MICROSOFT WINDOWS 95 CD**

| CPU | HD540MB | HARD DISK 810MB | |
|-------------|---------|--------------------------------------|---------|
| PENTIUM 75 | 4.540 | HARD DISK 1GB | + 350 |
| PENTIUM 100 | 4.590 | HARD DISK 1,2GB | + 650 |
| PENTIUM 120 | 4.790 | ESPANSIONE MEMORIA + 8MB (2 x 4MB) | + 440 |
| PENTIUM 133 | 4.990 | ESPANSIONE MEMORIA + 16MB (2 x 8MB) | + 770 |
| | | ESPANSIONE MEMORIA + 32MB (2 x 16MB) | + 1.540 |
| | | LCD COLORE TFT 10,4" | + 700 |

GARANZIA 2 ANNI - ASSISTENZA IN 72 ORE

SCHEDE VIDEO PCI

| | |
|--|------------|
| AL-2302 1MB ESP. 2MB PnP W95 MPEG SW | 90 |
| S3-TRIO64V+ 1MB/2MB EDO PnP W95 MPEG SW | 140 |
| S3-TRIO64V+ 1MB/4MB EDO PnP W95 MPEG HW | 240 |
| DIAMOND STEALTH 64 2120 1MB DRAM ESP 2MB | 240 |
| ESPANSIONE 1MB DRAM PER S3 / DIAMOND | 50 |
| DIAMOND STEALTH 64 3240 2MB VRAM ESP 4MB | 420 |
| ESPANSIONE 2MB VRAM PER DIAMOND STEALTH | 290 |
| DIAMOND STEALTH EDGE 3D 2120 1MB EDO RAM | 490 |
| DIAMOND STEALTH EDGE 3D 3240 2MB VRAM | 740 |
| MATROX MILLENNIUM 2MB | 470 |
| MATROX MILLENNIUM 4MB | 670 |
| ESPANSIONE 2MB VRAM PER MATROX | 290 |
| ESPANSIONE 4MB VRAM PER MATROX | 550 |
| MIRO VIDEO 20SV 2MB VRAM | 440 |
| NUMBER 9 FX VISION 330 2MB DRAM W95 PnP | 340 |



MODEM / FAX

| | |
|-----------------------------|---------|
| 14.4 VOICE INTERNO/ESTERNO | 120/140 |
| 28.8 INTERNO/ESTERNO | 270/290 |
| 28.8 VOICE ESTERNO/33.6 INT | 350/320 |
| 14.4 POCKET | 200 |
| 14.4 PCMCIA/+VOICE | 220/240 |
| 28.8 PCMCIA | 420 |

U.S. ROBOTICS

| | |
|--------------------------|---------|
| SPORTSTER 14.4 EST VOICE | 260 |
| SPORTSTER 28.8 INT/EST | 340/390 |
| COURIER 28.8 ESTERNO | 690 |

FAST MOVIE LINE

| | |
|-------------------------|-----------|
| MOVIE MACHINE II/FPS 60 | 1.190/940 |
| MPEG/MJPEG | 470/840 |
| AVIATOR SPEED | 490 |
| AV MASTER | 1.540 |

SCANNER MUSTEK 600 II SP

A4 600dpi/4800dpi 16 MILIONI DI COLORI
SINGOLO PASSAGGIO - DRIVERS WIN95
IN OFFERTA SPECIALE A L. 620.000

PC PENTIUM 100-166 MHZ



CONFIGURAZIONE BASE : CASE DESKTOP O MINITOWER CE
MAIN BOARD PENTIUM 75-200 MHZ - CHIPSET INTEL TRITON
256KB CACHE SINCRONA ON BOARD ESPANDIBILE A 512KB
RAM 8MB ESP 128MB - CTRL PCI EIDE MULTI I/O ON BOARD
VGA PCI 1MB ESP 2MB - DRIVE 1,44MB - TASTIERA WINDOWS 95 - MOUSE

| CPU | HD 850MB | U | P | G | R | A | D | E |
|-------------|----------|-------------------------|---|---|---|---|---|------|
| PENTIUM 100 | 1.240 | HARD DISK 1,2GB | | | | | | +70 |
| PENTIUM 120 | 1.340 | HARD DISK 1,7GB | | | | | | +140 |
| PENTIUM 133 | 1.470 | HARD DISK 2.0GB | | | | | | +240 |
| PENTIUM 150 | 1.670 | HARD DISK 2.5GB | | | | | | +340 |
| PENTIUM 166 | 1.920 | ESPANSIONE RAM + 8MB | | | | | | +140 |
| | | S3 TRIO64V+ 1MB MPEG SW | | | | | | +50 |
| | | ESPANSIONE CACHE 256KB | | | | | | +50 |

TUTTI I PREZZI SONO X 1.000 ESCLUSA IVA 19%

MONITOR SAMTRON (2 ANNI DI GARANZIA)

| | |
|--|-------|
| - 14" 1024x 768 NI 0,28 DOT PITCH - MPRII | 440 |
| - 15" 1280x1024 NI 0,28 DOT PITCH - MPRII DIGITALE OSD | 650 |
| - 17" 1280x1024 NI 0,28 DOT PITCH - MPRII DIGITALE OSD | 1.090 |
| - 17" 1280x1024 NI 0,26 DOT PITCH - MPRII DIGITALE OSD | 1.390 |
| - 20" 1280x1024 NI 0,30 DOT PITCH - MPRII DIGITALE OSD | 1.990 |
| - 15" CPD-15SX 1280x1024x0,25 (60Hz) DIGITALE | 750 |
| - 15" CPD-15SF2 1280x1024x0,25 (60Hz) DIGIT - OSD | 850 |
| - 17" CPD-17SF2 1280x1024x0,25 DIGITALE - OSD | 1.740 |
| - 17" CPD-17SE2 1600x1280x0,25 DIGIT - OSD | 2.290 |
| - 20" CPD-20SF2 1600x1280x0,30 DIGIT - OSD | 2.890 |
| - 20" GDM-20SE 1600x1280x0,30 | 3.790 |
| - 20" GDM-20SH 1600x1280x0,25 | 4.490 |



| | | | | | |
|------------------|-----|----------------|-------|--------------------|-------|
| 4MB 72PIN | 70 | 486DX4-100 AMD | 70 | 850MB QUANTUM | 300 |
| 8MB 72PIN | 140 | 586DX4-133 AMD | 120 | 1,2GB QUANTUM | 370 |
| 16MB 72PIN | 280 | PENTIUM 100 | 230 | 1,7GB QUANTUM | 440 |
| 4MB 72PIN EDO | 100 | PENTIUM 120 | 330 | 2,1GB SEAGATE | 540 |
| 8MB 72PIN EDO | 200 | PENTIUM 133 | 460 | 2,5GB QUANTUM | 640 |
| 16MB 72PIN EDO | 440 | PENTIUM 150 | 640 | 2,5" 810MB IBM | 690 |
| 256KB CACHE SYNC | 50 | PENTIUM 166 | 880 | 2,1GB SCSI QUANTUM | 1.170 |
| 512KB CACHE SYNC | 100 | PENTIUM 200 | 1.170 | 4,3GB SCSI QUANTUM | 1670 |

MAIN BOARDS 486-586 / PENTIUM

| | |
|---|-----|
| -486-586DX4 256KB CACHE CTRL PCI-4 EIDE ON BOARD + MULTI I/O 16550 | 150 |
| -PENTIUM 75-200MHZ CHIPSET INTEL TRITON - 256KB ESP 512KB CACHE SYNC ON BOARD | 220 |
| CONTROLLER PCI-EIDE MULTI I/O ON BOARD (2 SER 16550+PAR BIDIREZIONALE) | |
| -PENTIUM 75-200MHZ "PRIDE FREEWAY II" - C/S | 390 |

CD ROM IDE / SCSI

| | |
|---|-----|
| - HITACHI CDR-7730B 4X (600KBsec/170 msec) | 100 |
| - PIONEER DR-UA124X 4X (600KBsec/150 msec) | 120 |
| - TOSHIBA CDR XM5522B 6X (900KBsec/170msec) | 150 |
| - TOSHIBA CDR XM5602B 8X (1200KBsec/150 msec) | 270 |
| - WEARNES CDD-820 8X (1200KBsec/150 msec) | 270 |
| - TEAC CD-56S SCSI 6X (900KBsec/110 msec) | 290 |
| - TOSHIBA 3701 SCSI 6,7X | 340 |

MASTERIZZATORI

| | |
|---------------------------|-------|
| SONY 920-S 4Xread/2Xwrite | 1.270 |
| PHILIPS CDD2000 4Xr/2Xw | 1.190 |
| JVC XRW-2010 4Xr/2Xw | 1.140 |
| YAMAHA CDR-102 4Xr/2Xw | 1.190 |
| YAMAHA CDR-100 4Xr/4Xw | 1.790 |
| RCD 5040 2Xr/4Xw INTERNO | 1.390 |
| RCD 5040 2Xr/4Xw ESTERNO | 1.590 |

FINANZIAMENTI DA 12 A 36 MESI (CREDITCON)

PAGAMENTO RATEALE CON BOLLETTINI DI C/C
SVOLGIMENTO PRATICHE FINANZIARIE IN SEDE
O PER CORRISPONDENZA TRAMITE POSTA O FAX.



PC WARE srl

VIA CARLO PIRZIO BIROLI, 60
00043 CIAMPINO - ROMA



06/791.21.21 - 791.55.55 FAX 791.06.43



CONSEGNA GRATUITA A DOMICILIO
COMPUTERS, MONITORS E ACCESSORI
PER ROMA E PROVINCIA.

CONSEGNA GRATUITA COMPUTERS IN
TUTTA ITALIA TRAMITE CORRIERE ESPRESSO
CON PAGAMENTO IN CONTRASSEGNO.



VISITATE IL NOSTRO COMPUTER SHOP
(CATALOGO GENERALE PRODOTTI ON LINE)

<http://www.webcom.com/pcware>

E-mail: pcware@cdc.it

OFFERTE SPECIALI E PROMOZIONI IN TEMPO REALE.

LA PC WARE SRL PER IL POTENZIAMENTO DELLA
PROPRIA STRUTTURA DI VENDITA SU TUTTO IL
TERRITORIO NAZIONALE SELEZIONA
UN NUMERO LIMITATO DI PUNTI VENDITA QUALIFICATI
PER COSTITUIRE UNA RETE DI NEGOZI AFFILIATI.