

Corso Rapido di Visual Basic destinato a chi non sa programmare

Nella prima puntata abbiamo studiato quasi tutti gli strumenti, detti Control, presenti nella Toolbox di Visual Basic, nella seconda ci siamo dedicati invece alle funzionalità interne, quelle che non hanno rapporto immediato con gli strumenti.

In questa terza puntata vi proponiamo qualche semplice programma completo, nel senso che il programma non è dedicato ad un singolo argomento. Nella prossima puntata, l'ultima, parleremo dei rapporti tra il Visual Basic e il resto del mondo Windows e quindi esploreremo le sue funzioni più vicine al Windows che non al Basic, di cui Visual Basic è una evoluzione.

Come al solito non vi proponiamo listati completi, da copiare passivamente, ma gli spunti, i suggerimenti e gli spezzoni più interessanti dei listati. Sarete voi, se l'argomento vi stimola, a completare e probabilmente a migliorare il risultato finale. Per quanto ci riguarda stiamo provvedendo a raccogliere tutto il materiale, testo e programmi, sviluppato in questo Minicorso per riproporverlo tutto insieme anche in altra forma

di Francesco Petroni e Gianni Catani

La Casa Ruotante

Si tratta di un programma ricorrente, perché lo riproponiamo ogni volta che troviamo uno strumento software programmabile ed in grado di gestire oggetti grafici.

Poiché i prodotti Windows sono comunque in grado di gestire elementi ed oggetti grafici, riteniamo interessante utilizzare alcuni temi standard per vedere «come si fa» a svilupparli con il nuovo prodotto.

Ed è probabile che vi rifileremo anche la versione per MS Access, appena uscito, ma in grado anche esso di

gestire PictureBox e di tracciare linee al suo interno.

Il problema è questo: come visualizzare, in modalità wireframe, vedendone cioè gli spigoli, un oggetto solido definito tramite i suoi punti nello spazio.

I punti, che sono i dati base e che, essendo spaziali, vanno definiti tramite le loro tre coordinate nello spazio cartesiano, vanno poi convertiti, a seconda della modalità di visualizzazione scelta e quindi delle routine di calcolo adottate, in punti schermo bidimensionali, che poi vanno a loro volta uniti.

La conversione da punti tridimensionali in punti bidimensionali si basa su

due angoli di visualizzazione, che rappresentano, grosso modo, la longitudine e la latitudine attraverso la quale l'osservatore vede l'oggetto oppure (l'effetto visuale è analogo) gli angoli di rotazione dell'oggetto rispetto all'osservatore.

Se leghiamo il valore, in gradi, dei due angoli a due Scroll Bar possiamo quindi «ruotare» il nostro oggetto agendo direttamente sulle due Scroll Bar.

In figura 1 vediamo alcuni «spezzoni» del programma e lo stesso in esecuzione. Nel Form c'è quindi una PictureBox, con la Cassetta, le due Scroll Bar e un Bottone che va pigiato solo all'inizio per provocare la lettura dei dati che sono memorizzati in tre vettori X(i), Y(i), Z(i).

In figura 2 vediamo lo stesso esercizio realizzato con Excel 4.0. Vi presentiamo tale seconda versione sia per dimostrare che si tratta di un problema (la visualizzazione in modalità WireFrame di un oggetto 3D) comunque risolvibile con un prodotto grafico, sia per fornirvi direttamente le coordinate dei punti da visualizzare.

Per semplificare la routine di visualizzazione i vari punti sono forniti in modo da essere collegati con una linea continua, e quindi la linea che li unisce può essere tracciata senza dover alzare la penna. In tale modo vengono «ripercorsi» alcuni segmenti, ma questo è meno gravoso del dover prevedere un meccanismo di «alzamento della penna».

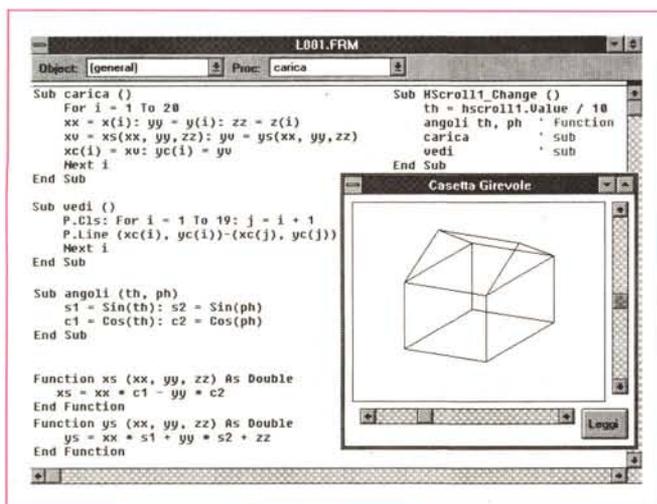


Figura 1 - Visual Basic - Barre per ruotare la figura. Il primo programma è esternamente, lato utente quindi, molto semplice. Nella PictureBox, quella con la figura, appare la vista assonometrica di una cassetta. Tale vista viene fatta ruotare agendo sulle due Scroll Bar che modificano i due angoli di rotazione dei vertici che individuano la cassetta rispetto ai loro valori iniziali. Quando cambia il valore di ciascuna Scroll Bar, varia uno degli angoli, vengono quindi ricalcolati i punti nello spazio, poi vengono ricalcolate le coordinate video, e infine, sulla base di queste ultime, viene rieseguito il disegno.

Sempre nell'esempio Excel 4.0, che fa uso di un tipo di grafico XY che Excel chiama «a dispersione», mostriamo anche alcune delle formule usate.

Tornando alla figura 1 vi indichiamo gli spezzoni di programma riportati.

— Sub Carica. Per caricare i dati dai vettori, tradurli in coordinate bidimensionali, sfruttando le due Function XS e YS.

— Sub Vedi. Che congiunge con dei segmenti le varie coppie di punti.

— Sub Angoli. Che, in funzione dell'angolo impostato con ciascuna Scroll Bar, ne calcola seni e coseni (che servono nelle successive Function).

— Function XS e YS. Cui vanno passate le tre coordinate spaziali e che restituiscono rispettivamente le due coordinate schermo.

— Sub HScroll Bar (manca quella verticale, ma il codice del sottoprogramma è analogo). L'evento Change della Scroll Bar provoca il ricalcolo degli Angoli, il ricalcolo dei Punti e la riesecuzione della Visualizzazione.

È evidente che le property Min e Max della Scroll Bar vanno scelte tra due angoli ragionevoli (es. — Pigreco e + Pigreco).

L'unica considerazione che vogliamo fare è che conviene, in tutte le applicazioni, fare uso di Subroutine e di Function, con le quali la programmazione viene razionalizzata in quanto viene frammentata in pezzetti «omogenei».

Applicazione Mini Draw

L'obiettivo è quello di realizzare un programma di disegno che lavori alla maniera di un programma Draw. Tale programma deve consistere in un Form

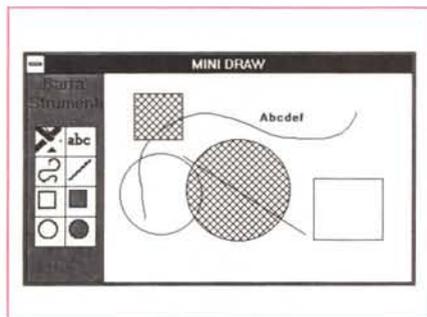
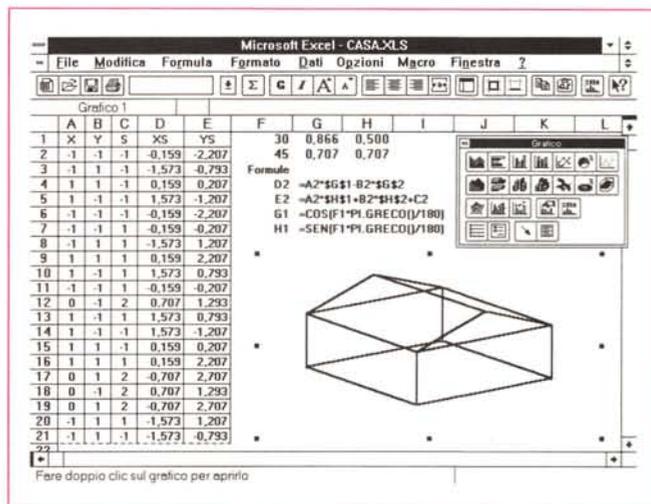


Figura 3 - Visual Basic - Un Mini Paint fatto in casa. In questa applicazione gli aspetti più interessanti sono due. Il primo è la gestione della Toolbox, la Barra con gli Strumenti sulla sinistra, il secondo è la gestione del foglio di disegno in cui vanno controllati, in maniera dipendente dallo strumento selezionato, gli eventi MouseDown, MouseMove e MouseUp, che sono quelli che permettono di iniziare a tracciare e di dimensionare i vari oggetti grafici.

Figura 2 - Visual Basic Coordinate della figura in Excel 4.0.

La figura, una casetta, è identificata da una serie di punti nello spazio (qui riportati per vostra comodità) uniti da linee. Per velocizzare il ricalcolo delle coordinate video abbiamo preferito collegare i vari punti con una linea continua a costo di dover ripercorrere segmenti già tracciati. Sicuramente un costo minore di quello richiesto da una routine che controlli i «salti» delle linee o che tracci individualmente i vari segmenti.



con sulla sinistra la Toolbox degli strumenti di disegno (lo vediamo in fig. 3) e sulla destra l'area per il disegno.

È evidente che ci porremo, per motivi «didattici» (nonché per motivi di spazio), obiettivi molto limitati, non tanto in termini di dotazione di strumenti, quan-

to in termini di inserimento di funzioni accessorie.

Non inseriremo routine di definizione del tipo, del colore e dello spessore delle linee, né quelle per la definizione del tipo di riempimento delle aree, né quelle per la scelta dei Font e dei loro

Figura 4 - Visual Basic Il Mini Paint - Evento Mouse Move.

Quando si clicca su uno degli strumenti della Toolbox vengono settati dei flag che individuano quale delle icone è stata scelta. Tutti i flag vanno a zero meno quello relativo allo strumento scelto. Poi occorre gestire le mosse del Mouse sul Form ovvero sul Foglio da Disegno. Con il primo click viene definito il vertice di partenza dell'oggetto. Poi, durante lo spostamento del Mouse sul Form, deve apparire l'effetto «scatola elastica». La figura viene continuamente ridisegnata, in dipendenza dello spostamento del secondo vertice (in pratica con la Sub Set_Col1 viene disegnata e con la Sub Set_Col2 viene disegnata con il colore opposto e quindi cancellata). È invece evento MouseUp che serve per «fissare» l'oggetto nella sua posizione finale.

```
Sub Form_MouseMove (B As Integer, S As Integer, X As Single, Y As Single)
  If TF Then
    Forml.MousePointer = 3 ' Mouse per Testo
  Else
    Forml.MousePointer = 2 ' Mouse per Grafica
  End If
  If B = 1 Then
    If DF Then ' Mano libera
      Line -(X, Y)
    End If
    If LF Then ' Linea
      Set_Col1
      Line (AX, AY)-(CurrentX, CurrentY)
      Line (AX, AY)-(X, Y)
      Set_Col2
    End If
    If BF Then ' Rettangolo Vuoto
      Set_Col1
      Fillstyle = 1
      Line (AX, AY)-(CurrentX, CurrentY), , B
      Line (AX, AY)-(X, Y), , B
      Set_Col2
    End If
    If BB Then ' Rettangolo Pieno
      Set_Col1
      Fillstyle = 7
      Line (AX, AY)-(CurrentX, CurrentY), , B
      Line (AX, AY)-(X, Y), , B
      Set_Col2
    End If
    If CF Then ' Cerchio Vuoto
      Set_Col1
      Fillstyle = 1
      RG = Sqr((AX - CurrentX) ^ 2 + (AY - CurrentY) ^ 2)
      Circle (AX, AY), RG
      RG = Sqr((AX - X) ^ 2 + (AY - Y) ^ 2)
      Circle (AX, AY), RG
      CurrentX = X: CurrentY = Y
      Set_Col2
    End If
    If CC Then ' Cerchio Pieno
      Set_Col1
      Fillstyle = 7
      RG = Sqr((AX - CurrentX) ^ 2 + (AY - CurrentY) ^ 2)
      Circle (AX, AY), RG
      RG = Sqr((AX - X) ^ 2 + (AY - Y) ^ 2)
      Circle (AX, AY), RG
      CurrentX = X: CurrentY = Y
      Set_Col2
    End If
  End If
End Sub
```

attributi. Vi daremo delle indicazioni utili per farlo da voi.

L'aspetto più interessante e più nuovo, per il nostro corso, di questa applicazione è la gestione degli eventi legati al mouse, con il quale si disegnano gli oggetti. Del mouse vanno gestiti tre eventi:

- il Click. Il primo click è quello che serve per posizionare il primo punto dell'oggetto da disegnare, il primo punto di un segmento o della linea a mano libera, un vertice di un rettangolo, o il centro della circonferenza;

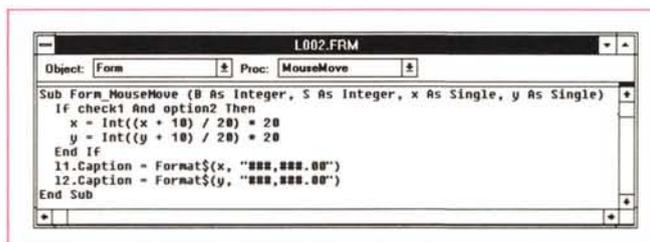
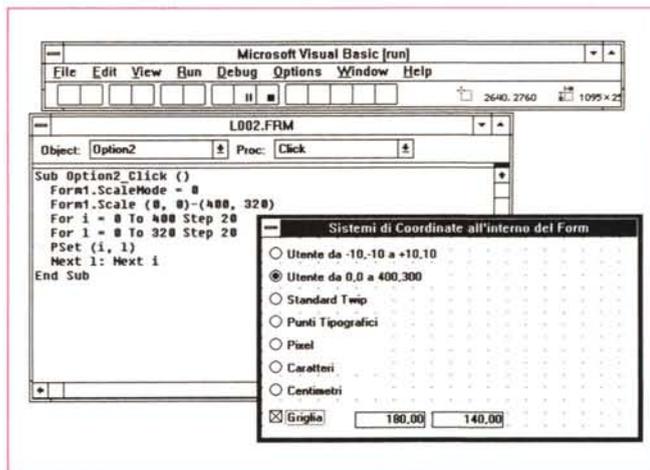
- il Move, che serve per visualizzare dinamicamente la posizione che via via assume il mouse. In questa fase occorre disegnare via via l'elemento, ad esempio il segmento o il rettangolo, e poi cancellarlo subito dopo, non appena il mouse viene spostato da un'altra parte. È infatti questo effetto di «scatola elastica» che permette il posizionamento «a vista» dell'oggetto (fig. 4);

- il Mouse Up, che coincide con la fine dell'operazione. L'oggetto viene definitivamente lasciato nella sua posizione finale.

Nella Toolbox invece i primi due strumenti sono la Gomma, che si limita ad eseguire una istruzione CLS che serve per pulire il Form, e il Text. In questo caso si usa l'istruzione KeyAscii, che individua il codice Ascii del tasto premuto, legata all'evento KeyPress nel Form.

In altre parole se si è scelto lo strumento Text basta portarsi con il Mouse sul Form, fare Click e cominciare a scrivere.

L'evento è KeyPress e i tasti vengono individuati dalla istruzione KeyAscii,



poi si scrive sul Form con l'istruzione Print Chr\$(KeyAscii).

Il tutto è molto semplice (occorre scrivere una mezza dozzina di istruzioni) a dimostrazione della predisposizione del Visual Basic per le modalità operative Windows.

Non abbiamo inserito istruzioni di definizione degli attributi del carattere (FontName, FontSize, FontBold) la cui attivazione, ad esempio via menu, o attraverso un Form che presenti un campionario grafico, o attraverso la Common Dialog Box di Windows (che come noto è a disposizione di tutti)

non presenta difficoltà.

DrawWidth, DrawStyle, FillColor, FillStyle, ecc. sono le funzioni da utilizzare nel caso si vogliano gestire anche le caratteristiche delle linee e dei riempimenti, che noi, per semplificare, abbiamo mantenuto fisse per tutta l'applicazione.

Non c'è il Bottone che consente la selezione di un oggetto, né quello per inserire una griglia «snap». Il primo è troppo «difficile» e quindi richiederebbe troppo tempo, del secondo ne parliamo nel prossimo esercizio.

Diamo ora uno sguardo ai listati nei

Figure 5, 6 - Visual Basic - Dimensionamento della Finestra di Lavoro.

Per posizionare correttamente gli oggetti in un Form, oppure per disegnare in un Form o in una Picture.Box, è opportuno conoscere i vari modi con i quali è possibile definire e gestire il sistema di coordinate interne ad un oggetto Visual Basic. In questa semplice applicazione scegliamo tra sette sistemi e controlliamo, tramite due Label, le coordinate relative alla posizione assunta via via dal Mouse. Inoltre attraverso una Check Box, attiviamo una griglia che riporta le coordinate lette dal mouse a valori interi e proporzionali a 20.

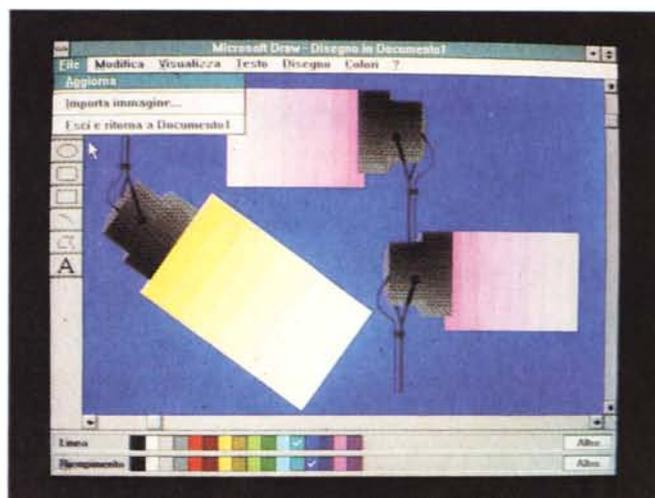
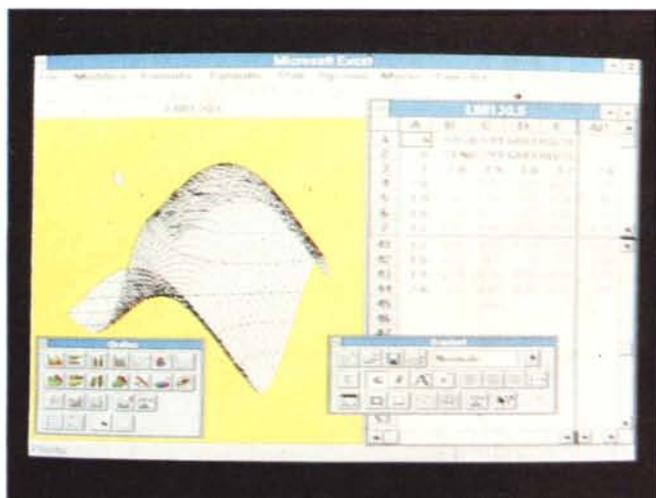
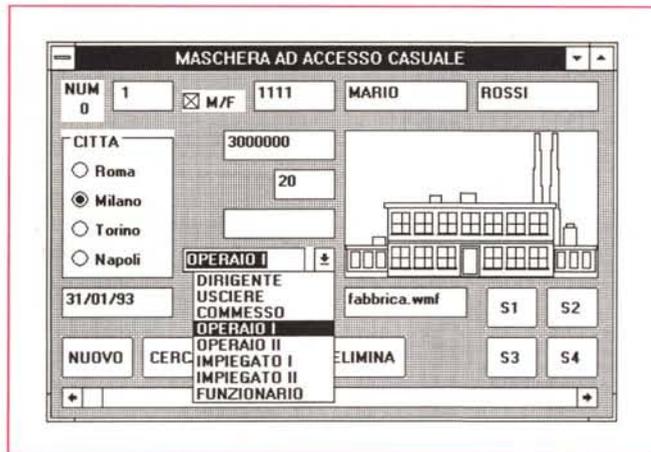


Figure 7, 8 - Microsoft Excel 4.0 e Draw - I due nostri modelli. È evidente che i nostri esercizi hanno solo finalità didattica. Esistono, per Windows, numerosi applicativi evoluti che risolvono in maniera completa tutti i problemi connessi con le viste grafiche tridimensionali e altri, con vario livello di sofisticazione, che permettono di eseguire disegni di tipo vettoriale.

Figura 9 - Visual Basic
Un archivio Plurimediale - Il Form.

Una delle prime cose che si cerca di fare con i prodotti per Windows è la costruzione di uno schedario Multimediale. Questo, con Visual Basic, è possibile in quanto è possibile inserire in un Form degli Oggetti OLE, quindi degli oggetti di qualsiasi tipo, provenienti da applicativi Windows di qualsiasi tipo. In questo primo esempio ci preoccupiamo solo di inserire una Picture.Box il cui contenuto grafico varia al variare del record dell'archivio.



quali presentiamo solo la parte più significativa, commentata nella didascalia.

Come Visual Basic gestisce le coordinate di un Form o di una Picture.Box

Nel precedente esercizio abbiamo accennato al problema della selezione degli oggetti, problema che pretende una gestione degli oggetti disegnati, che vanno in un qualche modo memorizzati, per poi poter essere, con una specifica routine, facilmente selezionati facendoci click sopra, o vicino, con il mouse.

Ma per poter memorizzare gli oggetti occorre conoscerne le coordinate, e conseguentemente occorre conoscere il sistema di coordinate del Form nel quale si disegna.

Ricordiamo che si può disegnare, ovvero utilizzare istruzioni di tracciamento, sia all'interno di una Picture.Box che all'interno di un Form.

Visual Basic propone più sistemi di coordinate che possono essere scelti a seconda delle varie necessità.

C'è ovviamente il tipo standard che è quello che abbiamo sfruttato, non sapendolo, nell'esercizio di prima, e ce ne sono altri probabilmente ancora più utili. Ne citiamo e sperimentiamo qualcuno in un programma di cui vi mostriamo sia il Form sia spezzoni di Listato (figg. 5, 6):

— Coordinate Utente. Si usa l'istruzione `Form.ScaleMode=0`, per selezionare il tipo di Scala. Poi la `Form1.Scale(-x1, -y1)-(x2,y2)` con la quale si definiscono gli intervalli desiderati, lungo l'asse X e lungo l'asse Y, delle coordinate utente.

— Coordinate Standard. L'unità di misura si chiama Twip. 1440 twpi sono un pollice.

— Coordinate Punti tipografici. I punti, ricordiamolo, sono una unità di misura di derivazione Tipografica, e corrispon-

dono ad un settantaduesimo del Pollice, o a un dodicesimo del Pica (che a sua volta quindi è un sesto di pollice).
— Coordinate Pixel. Qui giochiamo in casa nel senso che è l'unità di misura della scheda video. Nella VGA i pixel sono 640 per 480 e con questa unità di misura vengono direttamente indirizzati i pixel.

— Coordinate Caratteri. In realtà vengono considerati 80 caratteri di dimensione fissa distribuiti su 25 righe in uno schermo pieno.

— Coordinate Centimetri. Anche in questo caso di tratta di una unità di misura tradizionale addomesticata per il Visual Basic.

È chiaro che per i metodi che si basano sulle unità di misura lineari una unità, ad esempio un centimetro, viene riprodotta correttamente solo sulla stampa, mentre sul video la sua dimensione dipende anche dalla dimensione del monitor.

Il sistema di griglia

Nell'applicazione vedete anche una Check.Box con la scritta Griglia. Se si sceglie Utente da 0,0 a 400,320 e si attiva il Check Griglia, si visualizza una punteggiatura, e con un sistema di arrotondamento si fa in modo che vengano individuate solo coordinate che corrispondono ai punti della griglia.

Anche lo strumento Griglia è uno strumento standard nei programmi di disegno, al punto che è presente anche nel Visual Basic, quando si disegna il Form.

Nel programma (fig. 6) viene testato lo stato del Check Box e, se questo è in ON, vengono riportati ai valori resi obbligatori dalla griglia i valori X,Y via via letti dal Mouse che si muove sul Form.

Anche in questo caso l'evento «scatenante» è il `MouseMove` sul Form, che riporta lo stato dei Bottoni del Mouse e le Coordinate X,Y, che sono pronte per essere utilizzate.

Un Archivio Plurimediale

Nella prossima puntata, l'ultima, vedremo come sia possibile sfruttare dal Visual Basic tutte le funzionalità, anche le più evolute, di servizio di Windows.

La più potente è senza dubbio la possibilità di usare Visual Basic come «OLE Client» ovvero come destinatario di oggetti di vario tipo, oggetti tabellari, grafici bit-mapped, grafici metafile, testuali, suoni, ecc.

Sfruttando OLE è possibile inserire in

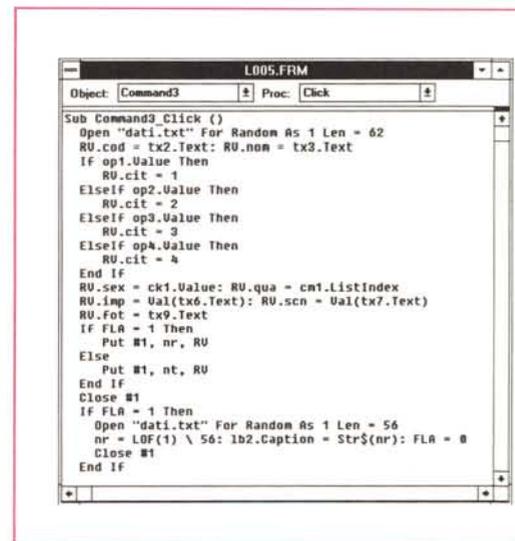


Figura 10 - Visual Basic - Un archivio Plurimediale
La routine di scrittura.

Sfruttiamo le routine di Input/Output viste nella scorsa puntata. Una delle novità è la gestione della Picture.Box in cui viene visualizzato un file grafico, il cui nome è memorizzato in una Text.Box. Usiamo l'istruzione `LoadPicture`, il cui compito è proprio quello di visualizzare il file nella box.

un Form elementi di qualsiasi tipo.

Nell'attesa vi proponiamo un esercizio, evolutivo rispetto a quelli proposti la settimana scorsa, in cui memorizziamo, in un File ad accesso Random, dati tradizionali e immagini di tipo WMF.

Si tratta in pratica di un archivio Plurimediale (ne vediamo il Form in figura 9).

In realtà nell'archivio non si memorizzano le figure ma, in un comune campo Testo, il Nome e l'Indirizzo del file con la Figura.

Poi per visualizzare il File nella Picture.Box si usa l'istruzione:

```
Picture1.picture=LoadPicture("nome.del.file")
```

che si occupa di tutto, meno che di controllare che il file, di cui è stato digitato il nome (nel caso illustrato si tratta di `fabbrica.wmf`), esista veramente.

Il buon programmatore deve anche inserire una istruzione per intercettare e «parare» eventuali errori.

Nel Basic, di qualsiasi tipo, esiste ed è la On Error Goto... Anche la trattazione dell'argomento Intercettazione degli Errori richiederebbe un paio di puntate.

Del programma, che non presenta ulteriori difficoltà rispetto a quelle superate... nel numero scorso, vi proponiamo uno stralcio che mostra la routine che trasferisce i dati digitati nel Form nella variabile generale RV, che racchiude tutti i campi (fig. 10).

Alla fine tale variabile viene scritta nel file DATI.TXT, nella posizione NR, se si tratta di un nuovo Record, in posizione NT, se si tratta di un Record che si sta solo modificando.

In pratica sfruttando il Flag FA abbiamo unificato la routine Nuovo Record con quella Aggiorna (fig. 10).

Una nuova calcolatrice

Riguardo questa nuova calcolatrice vi diamo solo alcune «dritte».

Va analizzata la sequenza logica delle operazioni.

— La prima è l'impostazione del Primo Numero, che si compone cliccando via via sui tasti con i numeri. In questa fase va gestito solo il «riempimento del display».

— La seconda inizia quando si clicca su un bottone che indica una Operazione. Con sistema di Flag vengono memorizzati il primo Numero, ormai completo, in una variabile e l'Operazione scelta.

— La terza operazione è la impostazione del Secondo Numero, che avviene nello stesso modo del primo, e la memorizzazione in una seconda variabile.

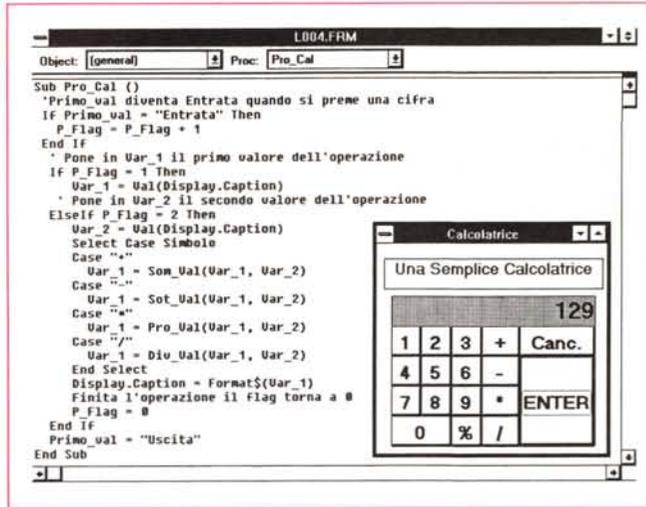
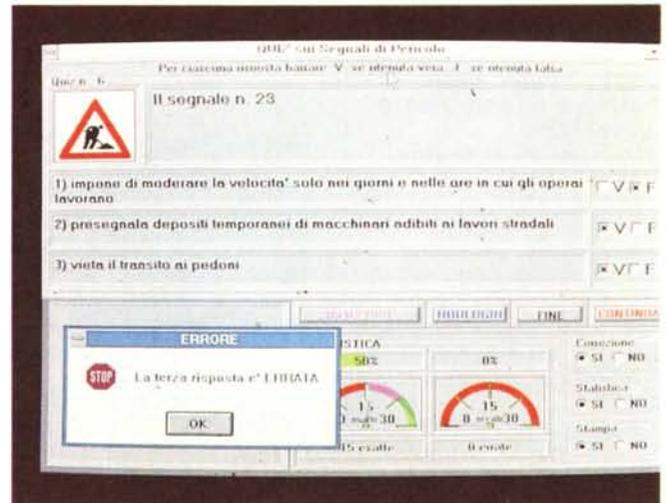


Figura 12 - Visual Basic - Un esempio di programma. La Patente Europea 1. È arrivato in redazione un programma particolarmente adatto al Visual Basic, e infatti dichiaratamente scritto in Visual Basic, per la realizzazione del quale è stato utilizzato anche il Professional Toolkit. Il programma si chiama Patente Europea e serve per addestrare l'aspirante automobilista a rispondere con sicurezza alle insidiose domande dei Quiz per il conseguimento della patente europea.



— L'ultima operazione è il Click sul tasto Enter che provoca il calcolo e la visualizzazione del risultato.

Alcune varianti sono costituite dall'uso del tasto Canc, che azzerava tutte le variabili e pulisce il Display, dall'uso del tasto «%», che è in pratica una ulteriore operazione che però si esegue su un solo Valore, dalla possibilità di eseguire operazioni in sequenza, facendo diventare il risultato della operazione primo Valore dell'operazione successiva.

La Patente Europea

È arrivato in redazione un programma, chiamato Patente Europea, scritto da un nostro lettore di Castel San Giorgio, provincia di Salerno (ing. Antonio Amendola, tel. 081/5162022), che ci sembra molto interessante perché riguarda una problematica particolarmente adatta ad essere trattata con il Visual Basic (figg. 11, 12).

Patente Europea è stato dichiaratamente scritto in Visual Basic, di cui è stato utilizzato anche il Professional Toolkit.

Il programma serve per addestrare l'aspirante automobilista a rispondere con sicurezza alle insidiose domande dei Quiz per il conseguimento della Patente Europea.

Si può accedere ai Quiz, e in tal caso l'utente deve scegliere le risposte Esatte tra quelle proposte dal Quiz stesso, oppure può accedere ai Segnali, ed in tale caso legge le definizioni esatte e quelle sbagliate proposte, per quel segnale, dai vari Quiz.

Il tutto è controllato da un sistema «grafico» di conteggio del punteggio conseguito e del tempo impiegato a rispondere.

L'applicazione Patente Europea è quindi adattissima al Visual Basic, sia lato utente, in quanto, dovendo rispondere ai Quiz, è possibile una sua totale interazione via mouse, sia lato programma, in quanto non necessita, al suo interno, di complesse strutture dei dati, che, come noto, non sono il forte del Basic.

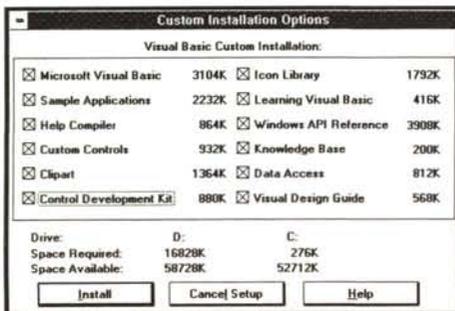


Figura 13 - Visual Basic - È uscito Professional Toolkit. Nel prossimo numero vi parleremo anche dell'ultimo nato, il Visual Basic 2.0 in edizione professionale. Questa che vi presentiamo in anteprima è la Dialog Box per l'installazione personalizzata, nella quale si scelgono i moduli da installare.

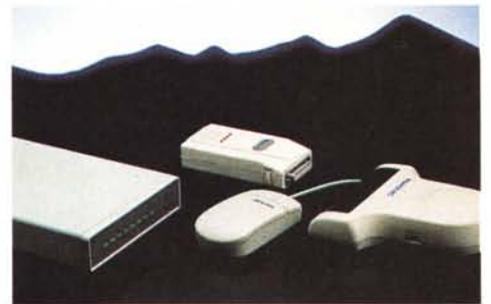
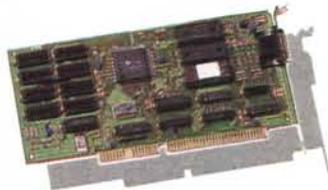
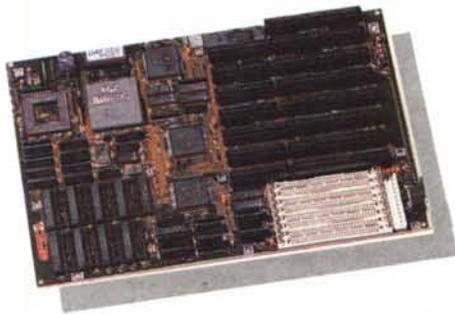
Made With Surgical Precision With Our Own Hands We Build Main Boards and Systems



Phonic Personal Computers Systems

CERCASI AGENTI PER ZONE LIBERE

Established in 1974, Phonic has been a manufacturer of computer products since the late '80s. Our two factories cover 9,000 square meters and are manned by 320 experienced employees. Output consists of a full array of motherboards, video cards, I/O cards, LAN cards, input devices, communication peripherals and complete PC systems. All items are made with surgically perfect precision. This makes them conform perfectly with industry standards, and they are brought to you at competitive prices.



System & Board Manufacturer

PHONIC[®] Computers Italia srl

REGGIO EMILIA - ITALY
Tel. 688334-688336 Fax. 0522/688322



System & Board Manufacturer

PHONIC[®] Corporation

P.O. Box 96-5, Taipei, Taiwan, R.O.C. - Telex 22564 FOTON
Phone 886-2-703-1933 (10 lines) - Fax 886-2-703-0191