

Meccanismi Windows

Windows ha imposto numerose modalità operative a cui ci siamo tutti, piano piano, e in modo spesso inconsapevole, abituati. Ad esempio, a causa del mio lavoro, mi capita di vedere ogni mese almeno una dozzina di nuovi prodotti. Ebbene le procedure di installazione sono standard, l'organizzazione dei vari menu dei vari prodotti è standard, le modalità di navigazione interna al prodotto sono sempre le stesse, per cui anche l'approccio ad un nuovo prodotto è assolutamente facilitato.

Obiettivo di questo articolo è quello di ricostruire, e quindi di sperimentare usando il solito «piccolo chimico» di Windows che è il Visual Basic, alcuni di questi meccanismi, allo scopo sia di capirne meglio il funzionamento sia di mettere in grado i lettori di riprodurli nelle nostre eventuali applicazioni.

di Francesco Petroni

Usiamo il Visual Basic perché è un linguaggio ormai diffusissimo e noto a moltissimi dei nostri lettori, inoltre ci consente di fare della sperimentazione «pura» svincolandoci da contesti applicativi complessi. I vari listati dei programmi che vi proponiamo sono cortissimi, spesso una o due istruzioni, in nessun caso superano la videata. La documentazione è dunque costituita dalle varie schermate con le applicazioni e dalle schermate con il codice, in molti casi queste sono dei collage con i vari «pezzetti» di listato.

Gli argomenti trattati sono un approfondimento sulle tecniche del Drag and Drop, la ricostruzione di una tecnica di ricerca veloce in una lista, la costruzione di una Tastiera Virtuale, la sperimentazione dei comandi che richiamano funzioni DLL, ed altro. Altro ancora vedremo in successivi articoli.

I vari esercizi servono anche per parlare di altre tecniche e modalità, che in VB sono tantissime ed in certi casi alternative l'una alle altre. Vogliamo dire che in Windows e in Visual Basic è spesso possibile, per uno stesso problema applicativo, proporre diverse soluzioni. Se ne sceglie una rispetto alle altre, o perché è la più rapida da realizzare, o perché è quella che garantisce le prestazioni migliori, o perché è la sola che si sa realizzare, in quanto non si conoscono le altre strade. Con Visual Basic è comunque possibile ricostruire qualsiasi «situazione» Windows. Trattandosi di una mezza dozzina di semplici esercizi, con listati molto corti e comunque completamente riprodotti nelle illustrazioni, pensiamo di cavarcela in poco tempo.

Tastiera Virtuale (figure 1 e 2)

La tastiera è la più importante periferica

per l'immissione dei dati. Lo sarà, pensiamo, per un bel po'. È stata affiancata dal mouse, che è in grado di «accollarsi» un bel po' di compiti di input. Altre periferiche, vedi microfono, vedi pen, non hanno ancora avuto una diffusione tale da farli considerare come strumenti standard, come lo sono, invece, mouse e tastiera.

In certe situazioni operative, ad esempio pensiamo ad un PC lasciato a disposizione del pubblico in una anticamera di un ufficio, la tastiera può essere poco indicata, data la sua complessità. Risulta molto meno ostico uno strumento di puntamento, come il mouse, il cui funzionamento è più intuitivo in quanto legato ad un cursore che si muove sul monitor ed ai due tasti sullo strumento stesso.

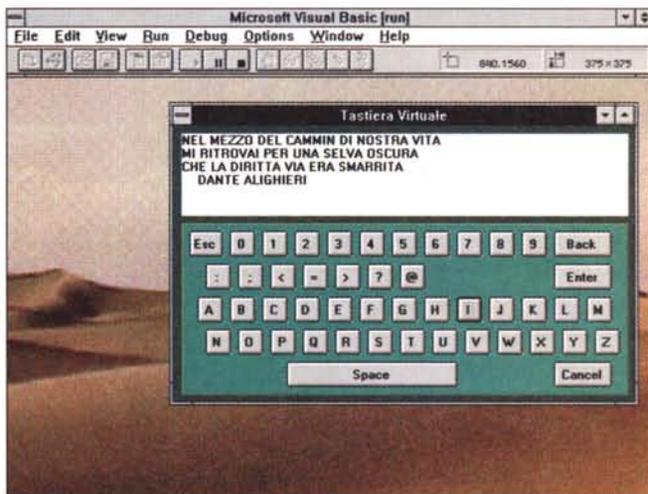
Il nostro obiettivo è quello di realizzare una tastiera a video, i cui tasti possano essere premuti con lo strumento di

puntamento. È chiaro che la velocità di digitazione sarà minima, pochi tasti al minuto, ma il suo uso è alla portata anche dell'utilizzatore occasionale.

Per realizzare la nostra tastiera virtuale, che vedete in figura 1, abbiamo sfruttato un Array di Controls, ovvero una unica famiglia di bottoni per i quali è previsto lo stesso codice.

Spieghiamo per i meno esperti come si può realizzare un array di bottoni in una Form: si crea il primo pulsante, di opportune dimensioni, al quale si cambia il nome, ad esempio lo si chiama C. Poi lo si copia. VB chiede se creare o meno un array di bottoni. Se si risponde NO il secondo bottone sarà indipendente dal primo e si chiamerà Command1. Se si risponde SI i due bottoni si chiameranno C(0) e C(1). A questo punto si possono selezionare i due e copiarli (e sono quattro), altri tre giri e si arriva a 32 bottoni, da 0 a 31.

Figura 1 - MS VB3 - L'aspetto della nostra Tastiera Virtuale. Il mouse può sostituire in tutto e per tutto la tastiera, tranne in alcune situazioni, ad esempio quando occorre digitare dei testi. In alcune situazioni, ad esempio quando occorre predisporre un «Kiosk», in pratica un PC riservato ad utenti occasionali, può essere opportuno togliere di mezzo la tastiera, lasciando a disposizione dell'utente solo un sistema di puntamento, come un mouse o un monitor «touch screen». Vogliamo realizzare una tastiera virtuale i cui tasti alfanumerici possano essere premuti con il mouse.



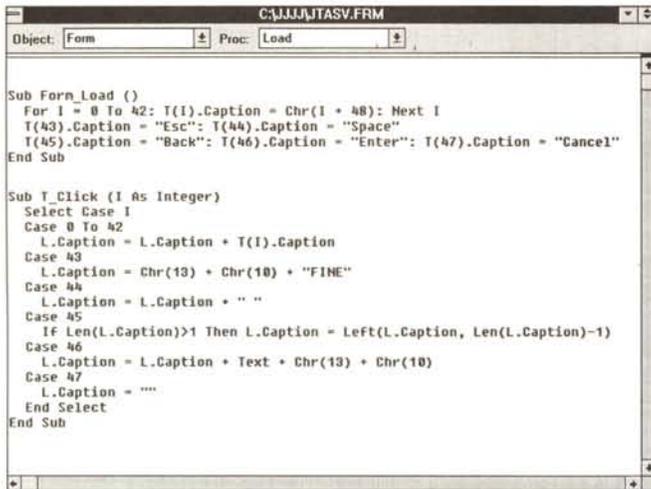


Figura 2 - MS VB3 - La gestione della nostra Tastiera Virtuale.

Abbiamo 48 tasti, alcuni normali, altri che attivano funzioni speciali. Non abbiamo previsto la possibilità di tasti premuti in coppia. Shift, Cntr, Alt, potrebbero essere facilmente gestiti con degli switch oppure usando il tasto destro del mouse. Vogliamo che la tastiera lavori in modo normale, si fa click su un tasto e appare il suo valore nella text box in alto. Vogliamo anche prevedere l'uso di tre o quattro tasti speciali. L'aspetto interessante del programma è che i vari tasti li abbiamo gestiti con un unico Array Control, cosa che semplifica il codice. Ne parliamo nel testo.

Figura 3 - MS VB3 - Applicazione Lista Dinamica - Digitare le iniziali della persona cercata.

Un meccanismo comodissimo quando si ha a che fare con delle liste lunghe è quello che permette di scrivere delle lettere che servono per puntare con maggiore esattezza l'elemento ricercato. Ad esempio una lista con 2.000 nomi, per quanto questi possano essere in ordine alfabetico, non è comodamente percorribile con la barra di scorrimento verticale. Con la nostra ricerca rapida, si preme M e la lista si posiziona al primo dei nomi che comincia con la M, poi si scrive MU e appare il primo che comincia con MU, e così via.

L'evento Click su uno qualsiasi dei bottoni attiva la stessa routine (è quindi una sola per tutti i tasti dell'array) alla quale viene passato un indice che indica quale dei 32 (se ci siamo fermati a 32) bottoni è stato premuto.

Nel nostro esempio, di cui vediamo in figura 2 tutti i listati, prevediamo di caricare automaticamente i Caption dei bottoni al verificarsi dell'evento Form Load. Se ci si accontenta di una tastiera ABC., invece che di una QWERTY, si può creare un ciclo da 0 a 31 per definire i Caption, sfruttando una sequenza di codici ASCII. Certo che se si vuole la QWERTY occorre ripiazzare tutti i bottoni a mano nel nuovo ordine, operazione lunga e noiosa, ma sicuramente non invidiosa.

Oltre ai tasti con i numeri, a quelli

con le lettere ed a quelli con alcuni segni di punteggiatura, abbiamo inserito, nella nostra tastiera virtuale, anche qualche tasto speciale, di cui parliamo dopo. Non abbiamo invece previsto né il tasto Shift, né il Ctrl, né l'Alt. Non essendo possibile lavorare con due mouse per gestire questi tasti dovremmo ricorrere ad interruttori, ed il programma si complica in quanto per ogni tasto dobbiamo prevedere cosa succede sia se è premuto da solo sia se è premuto quando sono attivi gli interruttori Shift, Cntr e Alt, da soli o in coppia.

I tasti speciali che gestiamo sono Esc che usiamo per terminare la digitazione, Cancel che serve per cancellare quanto scritto, Enter, che serve per dare Invio (a capo) durante la digitazione, BackSpace che serve per cancellare

l'ultimo carattere digitato, Space dall'ovvio significato.

Per digitare una frase occorre fare click sui vari tasti. Ai vari click la Label, che si chiama L, e che è settata come Multiline, si carica via via dei vari caratteri digitati che sono prelevati direttamente dalla Caption della Label. Questo a dimostrazione del fatto che le property dell'oggetto possono essere considerati come valori riutilizzabili direttamente.

Poiché non sono tutti caratteri e numeri, dobbiamo prevedere, per i vari tasti, comportamenti differenti, cosa che facciamo prevedendo i vari CASE nel sub Click del Bottone. Nell'impostare la struttura SELECT CASE conviene, come abbiamo fatto noi, analizzare prima i casi più frequenti, nel nostro caso la pressione dei tasti normali, in quanto i vari CASE sono mutuamente esclusivi ed in tale maniera viene minimizzato il numero dei CASE presi mediamente in considerazione.

Lista con scorciatoia (figure 3 e 4)

Uno degli strumenti operativi più comodi e diffusi in Windows è la Lista, che coincide, come noto, con un elenco di voci percorribile usando una barra di scorrimento verticale posta sulla destra. È uno strumento facile da usare se la Lista è corta, il suo uso si complica quando la Lista è molto lunga.

Uno dei sistemi per velocizzare il raggiungimento dell'elemento desiderato, utilizzato quando la lista è ordinata, è la ricerca per «chiave parziale», per dirla in

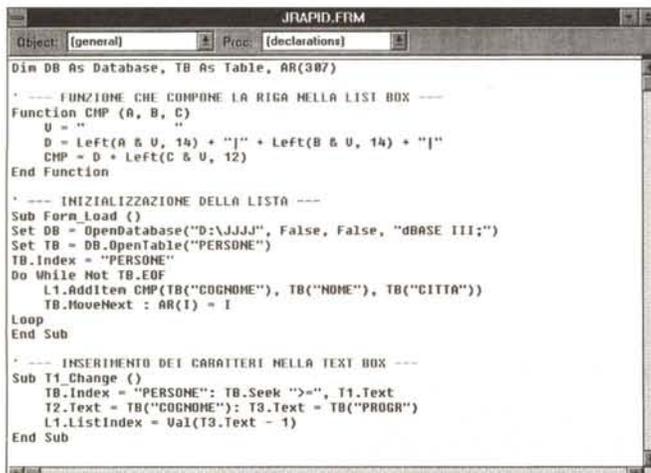


Figura 4 - MS VB3 - Applicazione Lista Dinamica - Listati.

Per potersi muovere nella lista abbiamo dovuto utilizzare come «Source», origine della lista stessa, una Table DBF, opportunamente ordinata ed indicizzata. Questo ci consente di usare il metodo Seek, che funziona in modo quasi uguale a quello del dBase. Quasi uguale perché ci sono delle differenze, che vanno «corrette». Ne parliamo nel testo.

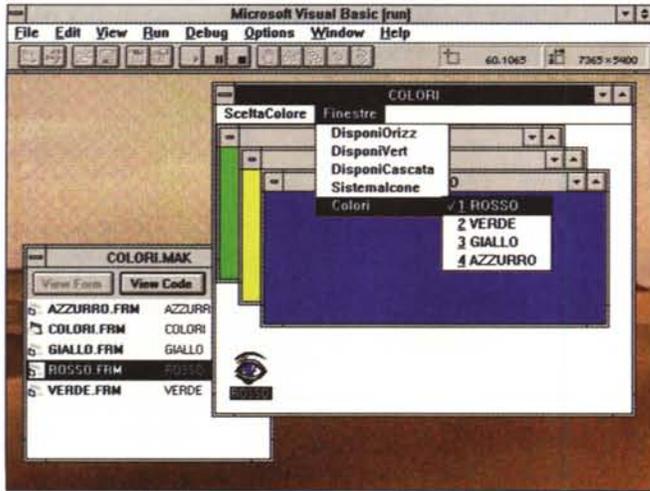


Figura 5 - MS VB3 - Applicazione MDI - Multiple Document Interface - Aspetto.

Uno degli strumenti organizzativi più comodi, qualsiasi sia il tipo di applicazione che si voglia realizzare con Visual Basic, è quello che gestisce le MDI Forms. Anche chi non le conosce... in realtà le conosce. Oggi tutte le applicazioni sono MDI, il che significa che c'è una finestra esterna, all'interno della quale si possono aprire più finestre (figlie) interne. Queste possono essere iconizzate, possono essere comunque dimensionate e posizionate, possono essere massimizzate fino a farle assumere la dimensione della finestra esterna (padre). Usando il comodo generatore di menu del Visual Basic e le istruzioni di gestione delle MDI Form, è possibile creare velocemente un'organizzazione standard per la nostra applicazione.

Figura 6 - MS VB3 - Applicazione MDI - Multiple Document Interface - Il generatore dei Menu.

Il nostro menu comprende solo due rami, il primo serve per aprire le quattro finestre MDI, il secondo per gestire il modo di apparizione delle finestre MDIchild nella finestra padre. Sono gli stessi esatti comandi che trovate ad esempio nel File Manager di Windows, in cui, data una serie di finestre aperte, è possibile vederle in cascata, oppure in orizzontale o in verticale. Altra necessità risolta dai comandi MDI è il passaggio da una finestra massimizzata ad un'altra sottostante.

modo tecnico. Si preme un tasto alfabetico e la lista si posiziona sul primo elemento con quella iniziale. Si preme un secondo tasto e la lista si posiziona sul primo elemento con quelle due iniziali. È il sistema usato per la funzione Cerca degli Help di Windows. Proviamo ad esempio con l'Help del Visual Basic e con la sua funzione Cerca. Premiamo L, ed appare Label, I ed appare Lib, S ed appare List.

Per ottenere lo stesso effetto sfrutteremo alcune funzionalità di tipo Database in Visual Basic. Ci serve una tabella esterna, noi ne usiamo una in formato DBF, e un suo Indice. Questo ci permette di usare il metodo SEEK. Altra necessità è quella di disporre nel file DBF anche di un campo numerico che ne indichi la posizione (in pratica di tratta di un progressivo). Questo serve perché, dopo aver «beccato» con il Seek l'elemento desiderato, dobbiamo anche posizionarci sull'elemento voluto nella lista, operazione che si può compiere con il metodo ListIndex che prevede però il passaggio del valore numerico del progressivo.

È un sistema un po' complicato, per il fatto che il file esterno che usiamo come origine della Lista deve essere adatto, o essere adattato, allo scopo.

Altra complicazione la porta il metodo SEEK che dà Errore in caso di non eguaglianza. Allora occorre usare come espressione di ricerca un'espressione di tipo maggiore od uguale.

In figura 3 vediamo il nostro esercizietto e nella successiva 4 il listato sottostante.

La lista viene caricata al verificarsi

dell'evento Load della Form. Poiché vengono caricati tre campi, realizziamo una Function (CMP) che costruisca una stringa che mostri in colonna i tre campi.

La ricerca vera e propria viene eseguita al verificarsi dell'evento Change della Text Box, in pratica ogni carattere che viene immesso.

Uso della tecnologia Multiple Document Interface (figure 5 e 6)

Tutte le applicazioni Windows devono essere MDI. Questo significa che l'applicazione apre una finestra esterna all'interno della quale si possono aprire più finestre, o più file, eseguiti all'interno dell'applicazione (tutti i Word Pro-

Figura 7 - MS VB3 -

Sfruttare le DLL - Out.

Lo dovrebbero sapere

anche gli utenti di Win-

dows meno esperti. Il

funzionamento di Win-

dows si basa sullo

sfruttamento delle DLL,

le famose librerie din-

amiche di Windows, che

contengono circa un

migliaio tra funzioni e

procedure, richiamate

dai vari applicativi come

veri e propri servizi di

Windows. In molti casi

sono i VBX di Visual

Basic che rendono più

semplice lo sfruttamento

di tali servizi (si pensi

alla Common Dialog

Box). Le stesse DLL

possono essere richia-

mate anche da Visual

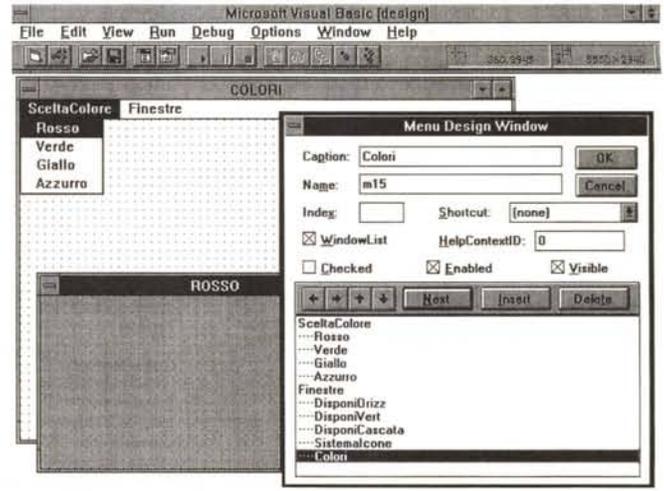
Basic, che dispone di

specifiche istruzioni, e

dagli altri linguaggi, a

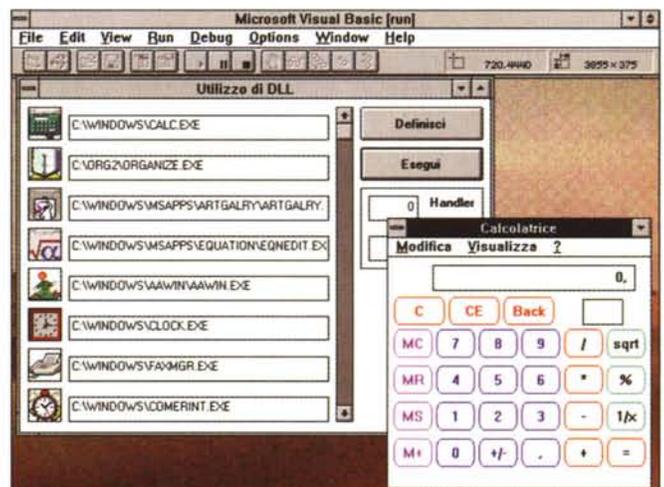
patto che se ne conosca

bene la sintassi.



cessor e gli Spreadsheet lavorano così), o più oggetti del file attivo. Ad esempio gli oggetti di MS Access corrispondono ciascuno ad una finestra. Una finestra MDI può essere iconizzata, dimensionata o massimizzata all'interno della finestra esterna. Al di là dell'aspetto esteriore un lavoro realizzato con finestre MDI risulta essere molto più ordinato ed organizzato e quindi molto più facile da usare da parte dell'utente.

Visual Basic permette di costruire con pochissimi comandi un'organizzazione a finestre MDI. I passi da compiere sono pochissimi, riferendoci al nostro esempio dobbiamo impostare quattro finestre normali di cui definiamo Caption e Nome, che sono rispettivamente Rosso, Blu, Giallo e Verde, e il colore di sfondo che rendiamo analogo al nome.



specifiche istruzioni, e dagli altri linguaggi, a patto che se ne conosca bene la sintassi.

Per ciascuna delle quattro scegliamo anche una Icona a piacere. Definiamo anche la property MDIChild come True. Questo significa che le quattro finestre sono figlie di una finestra padre. Questa va creata con il comando File New MDIForm. A questo punto il rapporto tra padre e figlie è stabilito.

Ora dobbiamo decidere come fare ad attivare le varie Form Child e come fare a gestire la visualizzazione delle varie finestre aperte.

Il modo classico è quello che fa uso del menu. Creiamo, con il generatore di menu del Visual Basic, una serie di voci, come da figura 6. Ogni voce del menu corrisponde ad una subroutine. Ad esempio quella per aprire la finestra Rosso sarà:

```
Sub Rosso
    Rosso Show
End Sub
```

A questo punto se si lancia l'applicazione e si clicca sulla voce di menu Colori Rosso appare la finestra, che si può, usando i classici pulsanti sulla cornice, iconizzare, massimizzare, oppure dimensionare a piacere, sempre all'interno della finestra padre, che abbiamo chiamato Colori.

Per chiudere la finestra Rosso l'unica possibilità, non avendo previsto strade alternative, è costituita dal tasto in alto a sinistra.

Allo stesso modo si aprono e movimentano le altre tre finestre colorate.

In ogni applicazione Windows che permette di aprire più finestre c'è sempre un ramo di menu, chiamato Finestre, che serve proprio per definire il modo di visualizzare le varie finestre. Quindi nel menu inseriamo le voci classiche che servono per disporle in modo Orizzontale, Verticale e a Cascata. Altre tipiche funzioni del menu Finestre sono quelle che ridispongono le Icone e quelle che permettono di passare da una finestra ad un'altra, se aperte e se massimizzate, quando la prima copre completamente le altre. Di queste varianti di visualizzazione si occupa il metodo Arrange che si applica alla finestra padre e che ha quattro varianti. Ad esempio:

```
Colori.Arrange 0
```

significa che le finestre interne vengono riorganizzate a cascata. Il metodo Arrange accetta valori da 0 a 3.

Sfruttare le DLL (figure 7 e 8)

Il funzionamento di Windows e dei vari applicativi che girano dentro Windows si basa sullo sfruttamento delle DLL, le famose librerie dinamiche, che contengono circa un migliaio tra funzioni e procedure, richiamate dai vari appli-

```
Object: [general] Proc: [declarations]
Declare Function ExtractIcon% Lib "shell.dll" (ByVal E%, ByVal NS%, ByVal F%)
Declare Function DrawIcon% Lib "User" (ByVal hDC%, ByVal X%, ByVal Y%, ByVal I%)

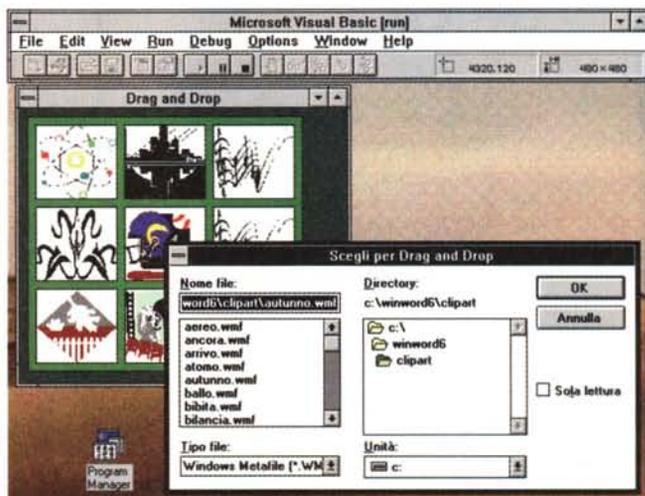
Sub Form_Load ()
    N.Text = 0: U.Value = 0
End Sub

Sub U_Change ()
    N.Text = U.Value
End Sub

Sub C_Click ()
    FS.DialogTitle = "Scegli un eseguibile Windows"
    FS.Filter = "Eseguibili Windows (*.EXE)|*.EXE"
    FS.Action = 1
    K = Val(N.Text): NF(K).Text = UCase(FS.FileName)
    I% = ExtractIcon(0, NF(K).Text, 0)
    H.Text = I%
    If (I%) Then
        J% = DrawIcon%(P(K).hDC, 0, 0, I%)
    End If
End Sub

Sub E_Click ()
    X = Shell(NF(Val(N.Text)), 1)
End Sub
```

Figura 9 - MS VB3 - Drag and Drop vero e/o simulato - Un programma di visualizzazione e di gioco con le figurine. La tecnica del Drag and Drop permette di eseguire, con il minor numero di passaggi possibile, alcune operazioni che coinvolgono due oggetti. Visual Basic permette di gestire sia l'evento Drag (mouse premuto sull'elemento origine dell'operazione) che l'evento Drop (mouse rilasciato sull'elemento di destinazione).



cattivi come veri e propri servizi di Windows.

Questo aspetto interessa i programmatori professionisti che usano linguaggi di basso livello, come ad esempio il C++, per scrivere programmi per Windows che eseguono continuamente richiami a funzioni DLL.

Anche i linguaggi di alto livello (più vicini all'utente e meno vicini alla macchina ed al sistema operativo) permettono di richiedere i servizi delle DLL, solo che in genere tali linguaggi dispongono di proprie funzioni ed oggetti (nel caso del Visual Basic) in cui già sono risolti i problemi del rapporto con Windows. In altre parole se nella nostra applicazione VB serve una Scroll Bar non dobbiamo ricorrere alle DLL in quanto è Visual Basic che ce le propone come oggetto preconfezionato.

In ogni caso le DLL possono essere richiamate anche da Visual Basic, che dispone di specifiche istruzioni, e dagli altri linguaggi di alto livello (anche dalle Macro di Excel), a patto che se ne conosca bene la sintassi.

Il problema è proprio questo. Occor-

Figura 8 - MS VB3 - Sfruttare le DLL - Listati.

Usiamo due funzioni, presenti nella libreria SHELL.DLL e nella USER.EXE, che servono per estrarre l'icona inserita in un file eseguibile Windows e per disegnare l'icona stessa da un'altra parte. Avendo questa necessità è probabile che sia proprio quella dello sfruttamento di tali due funzioni DLL la strada più veloce per raggiungere l'obiettivo. Vogliamo realizzare una sorta di elenco di file eseguibili, con tanto di icona, e lanciabili con un tasto.

re la documentazione che non fa parte né di quella standard Windows, né di quella cartacea di Visual Basic. Esistono per il VB, e vengono scaricati in fase di installazione, degli Help File con molte indicazioni sull'argomento. Si tratta comunque sempre di indicazioni individuali, nel senso che si occupano di una singola funzione DLL.

L'esercizio che vi proponiamo usa due funzioni, presenti rispettivamente nella libreria SHELL.DLL e nella USER.EXE, che servono per estrarre l'icona presente in un file eseguibile Windows, e per disegnare l'icona stessa da qualche altra parte. Non ci sono, con VB, altre strade per eseguire questa estrazione. In pratica vogliamo realizzare una sorta di elenco di file eseguibili, con tanto di icona, e lanciabili con un tasto. Ci limitiamo ad otto file, scelti attraverso una Scroll Bar che va da 1 a 8.

Nelle 8 PictureBox e nelle 8 TextBox carichiamo Icone e nomi dei programmi, completi di path.

Caricato l'elenco degli otto programmi, selezionatone uno, con la Scroll Bar,

Figura 10 - MS VB3 - Drag and Drop vero e/o simulato - Il listato monopagina.

Il click con il tasto destro lancia la Common Dialog Box «File Apri», nella quale si può scegliere un file grafico che poi appare nella PictureBox, quella, tra le nove, puntata. Un primo click con il tasto sinistro e viene memorizzata la picture attiva, che viene copiata nella picture di destinazione, che è quella su cui si esegue il secondo click.

```

JDED.FRM
Object: P1() Proc: MouseDown
DIM F
Sub P1_MouseDown (I As Integer, B As Integer, S As Integer, X As Single, Y As Single)
    IF B = 2 Then
        CD.DialogTitle = "Scegli per Drag and Drop"
        CD.FileName = "Windows MetaFile (*.WMF)|*.WMF|Windows Bitmap (*.BMP)|*.BMP"
        CD.Filter = CDF
        CD.FilterIndex = 1
        CD.InitDir = "D:\VB3\METAFILE\BUSINESS"
        CD.Action = 1
        P1(I).Picture = LoadPicture(CD.FileName)
    Else
        IF F = 0 Then
            F = 1
            Clipboard.Clear
            Clipboard.SetData P1(I).Picture, 3
        Else
            F = 0
            P1(I).Picture = Clipboard.GetData()
        End IF
    End IF
End Sub

```

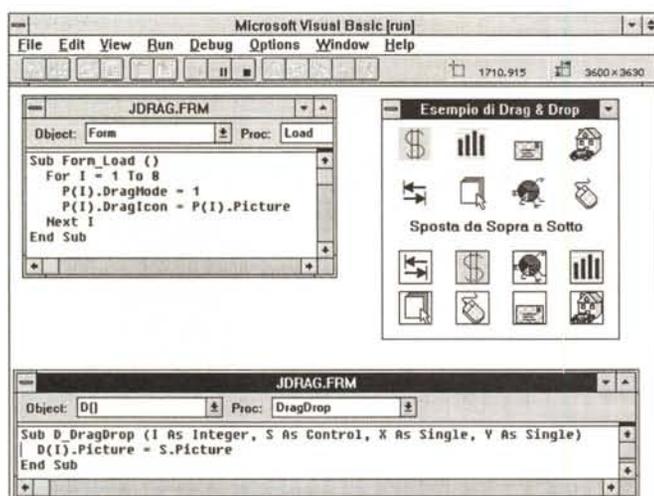


Figura 11 - MS VB3 - Drag and Drop in una sola istruzione.

Eccezionale la sinteticità di certi comandi che servono per eseguire operazioni anche complicatissime. Nella Form abbiamo 8 PictureBox, con 8 figurine (file di tipo ICO) e 8 PictureBox vuote. Gestiamo il Drag and Drop da sopra a sotto. Gli eventi coinvolti sono la dichiarazione di disponibilità ad essere «origine» di un Drag and Drop, da parte delle 8 figurine, e il Drop nelle 8 picture box vuote.

elementi. Gestiamo il solo evento MouseDown. L'unica precauzione che abbiamo preso è la definizione, nella sezione General, di una variabile F, che ci servirà da Flag.

Se facciamo click con il tasto destro apriamo la Common Dialog Box «File Apri», che ci serve per prelevare immagini di tipo grafico (o BMP o WMF), che facciamo apparire nella PictureBox puntata. Per ricopiare un'immagine da una Box ad un'altra basta fare click sulla prima (e viene portato a 2 il Flag F) e poi click sulla seconda. È il flag che serve per far capire all'evento MouseDown se è la prima volta che viene cliccato su una PictureBox o la seconda.

Per portare l'immagine da una Box all'altra scomodiamo la Clipboard e usiamo i comandi Clipboard.SetData e Clipboard.GetData(), sempre utili quando occorre portare qualcosa da una parte all'altra. Si poteva, più direttamente, scrivere l'istruzione:

```
P1(dest.).Picture = P1(orig.).Picture
```

Con l'accortezza di assegnare a dest. il valore numerico dell'elemento di destinazione ed a orig. quello di quello di origine.

Drag and Drop quello vero (figura 11)

Prepariamo due array di picture. La prima la chiamiamo P(x) e la seconda D(y). Nella prima piazziamo otto figurine, ad esempio otto file ICO.

Per rendere draggabili le otto icone dobbiamo impostare il rispettivo valore DragMode a 1. Se poi vogliamo che durante l'operazione il mouse assuma l'aspetto dell'oggetto trascinato dobbiamo impostare anche il valore DragIcon che in questo caso è uguale a quello della PictureBox puntata.

Una volta definita la disponibilità di ciascuna Picture ad essere draggata occorre prevedere la conclusione dell'operazione, che corrisponde all'evento DragDrop sull'elemento di destinazione. È importante il secondo parametro della Sub_DragDrop, che indica quale è il Source dell'operazione, nel nostro caso la PictureBox.

In pratica le istruzioni per eseguire l'operazione di Drag and Drop sono tre (riducibili a due se ce ne freghiamo dell'aspetto del cursore del mouse).

Drag and Drop dei nomi dei file (figura 12 e 13)

Si può draggare un elemento di una lista. Se questa è una lista che contiene nomi di file si può individuare il nome del file e poi decidere cosa farci.

Nell'esercizio che vi proponiamo rea-

basta un click sul bottone Esegui per eseguire il programma proprio del progressivo scelto.

Per quanto riguarda la proprietà hDC, gestita da Windows, si rimanda alla documentazione VB.

Drag and Drop - Vero e simulato

Per chi usa frequentemente Windows la tecnica del Drag and Drop è diventata un automatismo, al punto che, anche quando usa prodotti che non conosce, esegue spontaneamente dei Drag and Drop tra gli oggetti.

Se il prodotto è fatto nel rispetto delle regole Windows, il meccanismo funziona.

La caratteristica principale del Drag and Drop consiste nel fatto che vengono coinvolti due oggetti, quello di origine e quello di destinazione, quello da cui comincia l'operazione, premendo il tasto del mouse e tenendolo premuto, e quello in cui finisce l'operazione su cui si arriva con il mouse e si «molla» la presa.

Nei vari prodotti il Drag and Drop ser-

ve per operazioni di spostamento e/o di copia (si pensi ad un comune Word Processor o a uno Spreadsheet, o al solito File Manager, che contiene tutti i meccanismi Windows).

È chiaro che esistono numerose varianti a seconda di cosa si trascina da una parte all'altra (ad esempio un nome di un file grafico in una PictureBox in cui vogliamo vedere l'immagine) ed esistono numerose possibilità di errore ad esempio se si trascina qualcosa di grafico in un oggetto che non può ricevere oggetti grafici.

Vi proponiamo tre esercizi. Il primo è una simulazione del Drag and Drop. Nel secondo invece facciamo il vero unico autentico Drag and Drop. L'ultimo esercizio invece coinvolge, come detto prima, dei nomi di file grafici e testuali, che poi vengono riversati, via D&D, o in PictureBox o in TextBox.

Tutto con un Click (figura 9 e 10)

Prepariamo una Form con nove PictureBox, anzi il solito Array con nove

lizziamo, con i tre oggetti Path, Dir e ListFile, un sistema per listare una serie di file dai quali eseguire la parte iniziale di un Drag and Drop. Si usano questi tre oggetti, anziché il più comodo File Api, usato in altri esercizi, per il fatto che quest'ultimo non permette il Drag and Drop.

Come destinazione dell'operazione prevediamo o una PictureBox, che visualizzi il file grafico, o una TextBox, che visualizzi il file testuale.

Non si tratta di un vero Drag and Drop in quanto sull'oggetto di destinazione viene eseguita una routine che

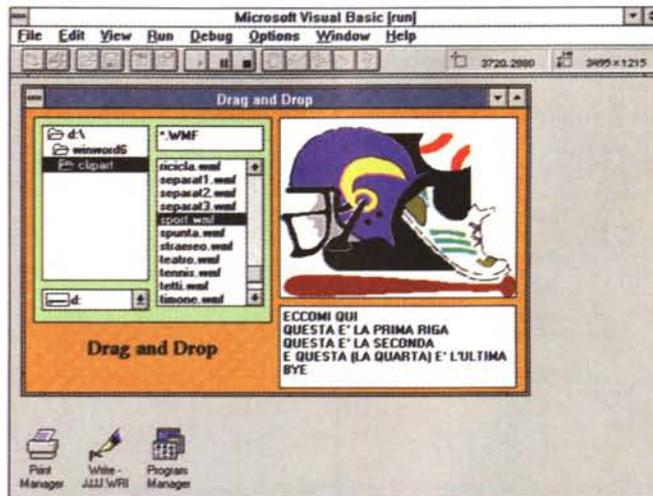


Figura 12 - MS VB3 - Drag and Drop con i nomi dei file - Out. L'ultima variante del Drag and Drop è quella che passa per il nome di un file. In questo caso vogliamo «draggarlo» il nome di un file da una lista di file ad una picture, oppure ad una Text Box, nel caso di file testuale. Abbiamo utilizzato i tre Controls, Path, Dir, FileList, della Toolbox standard, anziché la Common Dialog Box, perché in questo caso dobbiamo eseguire il «Drag» del nome del file.

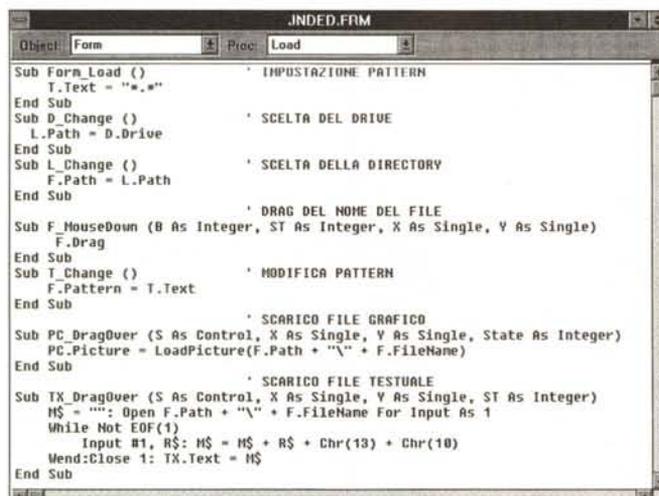


Figura 14 - MS VB3 - Visualizzazione dei modi Draw Mode.

Questo esercizio serve a sperimentare il concetto di DrawMode legato agli oggetti grafici. In pratica abbiamo piazzato un disegno qualsiasi su una Form e poi su questo abbiamo piazzato sedici rettangolini «pieni». A seconda del DrawMode impostato il rettangolo copre, o interagisce, con la figura sottostante. Le 16 Picture sono una Array e il programma è solo quello, cortissimo, che si vede in secondo piano.



Figura 13 - MS VB3 - Drag and Drop con i nomi dei file - Listato. Il Drop comporta l'uso del nome del file evidenziato nella lista. Se il file è di tipo grafico il Drop viene fatto nella PictureBox, se il file è di tipo testuale il Drop viene fatto nella Text-Box. Se si sbaglia destinazione il programma va in errore. È chiaro che il nostro obiettivo non è quello di proporre programmi completi ma spunti da approfondire. La gestione dell'errore è ovviamente possibile ma ci porterebbe a trattare un argomento fuori tema.

sequenziale. Ipotizziamo che il file contenga un testo a righe e che ogni riga sia un record. Allora bisogna prevedere una routine di lettura di file a tutti gli effetti.

Il DrawMode (figura 14)

L'ultimo esercizio è il più leggero e serve per sperimentare la property DrawMode, che ha 16 varianti.

Con DrawMode si definisce il modo con il quale un oggetto grafico viene disegnato sullo sfondo. Prepariamo una Form, con un'immagine colorata di sfondo, e poi la ricopriamo con 16 Shape rettangolari (anzi una array di 16 shape), per ognuna delle quale impostiamo lo stesso colore e lo stesso FillMode a Transparent.

Poi con un'istruzione che esegue più volte un ciclo da 1 a 16 impostiamo al volo un nuovo DrawMode, uno per ciascuna shape. Alcuni modi provocano la sovrapposizione della shape rispetto allo sfondo, altri provocano delle operazioni booleane tra i pixel della shape e i pixel dello sfondo, di cui appare il risultato. Per la descrizione vi rimandiamo al manuale che spiega uno per uno i 16 tipi.

L'istruzione DoEvents, di cui parleremo un'altra volta, permette la visualizzazione dell'effetto grafico rallentando il pratica la velocità del ciclo.

Conclusioni

Abbiamo visto una serie di meccanismi Windows e abbiamo potuto constatare come sia facile riprodurli con VB. Di altri a cui abbiamo pensato, ad esempio l'uso delle scorciatoie da tastiera, oppure l'uso massiccio dello strumento Outline, oppure ancora la definizione di finestre volanti, parleremo prossimamente.

MC

legge il file e lo riversa dentro e non un riversamento diretto.

I due meccanismi, quello per il file grafico e quello per il file testuale, sono

assolutamente differenti. Nel primo caso si usa la funzione LoadPicture, nel secondo invece occorre eseguire una lettura vera e propria di un file testuale

Per avere il massimo da un personal, basta poco.



Tanto così.

È finita l'era delle acrobazie tecniche, delle perdite di tempo e dei tavoli ingombri di attrezzi e dischetti. Oggi per installare schede aggiuntive nel vostro PC bastano due dita ed una sola parola: PCMCIA.

Utilizzare una card PCMCIA, infatti, è così incredibilmente facile che non solo i portatili ma anche i desktop hanno aperto loro le porte.

Oggi nelle ridotte dimensioni di una carta di credito potete spaziare dai



fax/modem agli scanners, dall'HI-FI digitale all'acquisizione video, dalla connettività al GPS e persino agli hard disks.

Kernel ha in catalogo cards PCMCIA per tutte le esigenze, e punti vendita dove è disponibile l'intera gamma e dove è possibile provare le ultime novità: il massimo in fatto di PCMCIA.

I prezzi? Conti alla mano, piccoli piccoli.



PIEMONTE - Omega, via Bonafous 4a TORINO • **LOMBARDIA** - Supertronic, via S.G.B. De La Salle 4 MILANO • **FRIULI V. GIULIA** - Media Source, via Trieste 67c STARANZANO (Go) • **VENETO** - Cartocontabile, via S. Pio X 165 CASTELFRANCO V. (Tv) • **LAZIO** - Gam Computer, via delle Cave 131 ROMA • **M. Electronic**, via Tazio Nuvolari centro commerciale "I Granai" ROMA • **Redwood Microlink**, viale Tirreno 207 ROMA • **P.S. Italia**,

PK
Punto affiliato

via A. Gramsci 9 S.M. DELLE MOLE (Rm) • **CAMPANIA** - Informatica Meridionale, corso V. Emanuele 700a NAPOLI • **Know How**, via Luca Giordano 51 NAPOLI • **General Computer**, corso Garibaldi 56 SALERNO • **PUGLIA** - Progetti Informatici, viale Magna Grecia 108b TARANTO • **SICILIA** - Delta Inform, via A. Siciliana 116 PALERMO • **Data Studio**, via Massimo D'Azeglio 34/36 BAGHERIA (Pa) • **Zuleima**, viale Italia 22 ALCAMO (Tp)