Ingrandire o ridurre una parte dello schermo

http://www.vbsimple.net/howto/ht_036.htm

Difficoltà: 3 / 5

Prendendo spunto dall'How-To per determinare il colore di un pixel sullo schermo sviluppperemo un semplice progettino che funga da lente d'ingrandimento dello schermo. Sarà inquadrata una porzione dello schermo indicata dal cursore del mouse e in base al fattore di riproduzione sarà mostrata la sezione grafica.

Il progetto consiste di un solo form contenente quattro controlli:

1. una grande *PictureBox* di nome **Immagine**
2. una *CheckBox* di nome **Invertito**
3. un *Timer* di nome **Tempo** con la proprietà *Interval* impostata a ***100*** e la proprietà *Enabled* impostata a ***True***
4. una *ComboBox* di nome **Fattore** con la proprietà *Style* impostata a ***2 - Dropdown List***.

La PictureBox Immagine conterrà la sezione grafica estratta dallo schermo; la ComboBox conterrà l'elenco dei fattori di riproduzione della grandezza ed il Timer provvederà ad aggiornare regolarmente l'immagine estratta.
La CheckBox servirà esclusivamente per vedere l'immagine in maniera invertita, con i colori al contrario. Il codice non è molto difficile e contiene molte parti in comune con il codice per determinare il colore di un pixel sullo schermo:

```
 1. Option Explicit
 2.
 3. Private Type POINTAPI
 4.     x As Long
 5.     y As Long
 6. End Type
 7.
 8. Private Declare Function GetCursorPos Lib "USER32" (lpPoint As POINTAPI) As Long
 9. Private Declare Function GetDC Lib "USER32" (ByVal hwnd As Long) As Long
10. Private Declare Function ReleaseDC Lib "USER32" (ByVal hwnd As Long, ByVal hdc As
    Long) As Long
11. Private Declare Function StretchBlt Lib "GDI32" (ByVal hdc As Long, ByVal x As
    Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As
    Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal nSrcWidth As Long, ByVal
    nSrcHeight As Long, ByVal dwRop As Long) As Long
12.
13. Private ScreenDC As Long
14.
```

Alle righe 3-6 definiamo un nuovo tipo di dati di nome **POINTAPI** che consiste di due dati semplici: x e y. Esso viene utilizzato da alcune funzioni API.

Alla riga 8 è presente la dichiarazione della funzione API ***GetCursorPos*** vista nell'altro HowTo. Essa ritrova la posizione del cursore del mouse e richiede la struttura **POINTAPI**.

Le righe 9 e 10 contengono le dichiarazioni delle funzioni ***GetDC*** e ***ReleaseDC***. La prima serve per ottenere il DC (Device Context) di una finestra, mentre la seconda serve per rilasciare (deallocare) un DC.

La riga 11 contiene la dichiarazione di una nuova funzione API: ***StretchBlt***, letteralmente Stretch Bit Block Transfer. Essa permetterà di allargare e stringere un'immagine in base ai dati passati. Richiede un handle DC, le coordinate di sorgente e destinazione ed il tipo di disegnamento da effettuare, valore preso dall'enumerazione RasterOpConstants.

Prima di vedere il codice vero e proprio abbiamo dichiarato una variabile di nome **ScreenDC** che verrà utilizzata per contenere l'handle del Device Context dello schermo.

```
15. Private Sub Form_Load()
16.     ScreenDC = GetDC(0)
17.     Fattore.AddItem "50%"
18.     Fattore.AddItem "75%"
19.     Fattore.AddItem "100%"
20.     Fattore.AddItem "125%"
21.     Fattore.AddItem "150%"
22.     Fattore.AddItem "200%"
23.     Fattore.AddItem "250%"
24.     Fattore.AddItem "300%"
25.     Fattore.ListIndex = 2
26.     Immagine.ScaleMode = vbPixels
27. End Sub
28.
```

Al caricamento del form verrà innanzitutto ritrovato l'hande DC dello schermo tramite funzione ***GetDC*** e salvato nella variabile *ScreenDC* (riga 16). La finestra 0 indicata da GetDC è lo schermo grafico.

Alle righe 17-24 viene riempita la ComboBox **Fattore** con dei valori percentuali che saranno utilizzati per determinare il fattore di riproduzione. Riempita la ComboBox sarà selezionato automaticamente il terzo elemento (l'array *List* ha base 0) ovvero il 100%.

Alla riga 26 viene invece impostato il fattore di conversione delle dimensioni in pixel poiché le nostre funzioni API utilizzeranno i pixel e non i twips.

```
29. Private Sub Form_Unload(Cancel As Integer)
30.     Call ReleaseDC(0, ScreenDC)
31. End Sub
32.
```

Prima di chiudere il programma sarà necessario deallocare l'handle DC ScreenDC riservato precedentemente. Ciò viene fatto mediante utilizzo della funzione ***ReleaseDC*** passandole come numero di finestra lo schermo ovvero 0 e l'handle DC da rilasciare ovvero *ScreenDC*.

La funzione di estrazione dell'immagine dallo schermo è nella routine richiamata all'evento Timer di **Tempo**. Ad intervalli regolari sarà richiamata la funzione di aggiornamento dell'immagine e sarà calcolata la grandezza dell'immagine in base al fattore di riproduzione selezionato nella *ComboBox*.


```
33. Private Sub Tempo_Timer()
34.     Dim COORD As POINTAPI
35.     Dim DIMENSIONE As POINTAPI
36.     Call GetCursorPos(COORD)
37.     DIMENSIONE.x = Immagine.ScaleWidth / Val(Fattore.Text) * 100
38.     DIMENSIONE.y = Immagine.ScaleHeight / Val(Fattore.Text) * 100
39.     If Invertito.Value = vbChecked Then
40.         Call StretchBlt(Immagine.hdc, 0, 0, Immagine.ScaleWidth,
    Immagine.ScaleHeight, ScreenDC, COORD.x, COORD.y, DIMENSIONE.x, DIMENSIONE.y,
    vbNotSrcCopy)
41.     Else
42.         Call StretchBlt(Immagine.hdc, 0, 0, Immagine.ScaleWidth,
    Immagine.ScaleHeight, ScreenDC, COORD.x, COORD.y, DIMENSIONE.x, DIMENSIONE.y,
    vbSrcCopy)
43.     End If
44. End Sub
```

Alle righe 34 e 35 sono dichiarate due variabili di tipo ***POINTAPI*** e di nome **COORD**, **DIMENSIONE**; la prima servirà per ritrovare le coordinate del cursore del mouse (riga 36) mentre la seconda sarà utilizzata per calcolare la dimensione dell'area da inquadrare e mostrare nella *PictureBox* (righe 37 e 38).

Per calcolare la grandezza dell'area da inquadrare utilizzeremo una semplice formula di conversione: grandezza originale (quindi quella della PictureBox) diviso il fattore di conversione in percentuale. L'istruzione Val ignorerà il segno di percentuale e pertanto sarà necessario effettuare la moltiplicazione manualmente. La formula pertanto è data da ***GrandezzaOriginale / Fattore * 100***.

Possiamo adesso richiamare la funzione API StretchBitBlt in base al valore della CheckBox **Invertito**. Se essa è selezionata l'immagine sarà in negativo altrimenti sarà normale.

Diamo uno sguardo più approfondito alla chiamata alla funzione ***StretchBitBlt***:

```
40.         Call StretchBlt(Immagine.hdc, 0, 0, Immagine.ScaleWidth,
    Immagine.ScaleHeight, ScreenDC, COORD.x, COORD.y, DIMENSIONE.x, DIMENSIONE.y,
    vbNotSrcCopy)
```

La funzione ***StretchBitBlt*** richiede due serie di parametri:

1. il primo di questi è il DC di destinazione in ovvero Immagine.hdc;
2. il secondo è la coordinata X del punto in cui iniziare il disegnamento. Nel nostro caso sarà 0, ovvero il lato sinistro di **Immagine**;
3. il terzo è la coordinata Y del punto in cui iniziare il disegnamento. Sarà 0 poiché vogliamo disegnare l'immagine nell'angolo superiore sinistro della PictureBox;
4. la larghezza dell'area in cui disegnare è indicata nel quarto parametro, specificato mediante la proprietà ScaleWidth di Immagine (ricordiamo che ScaleMode è impostato su vbPixels);
5. lo stesso discorso varrà per l'altezza dell'area in cui disegnare (proprietà ScaleHeight);

Definiti i parametri di destinazione dei dati immagine da trasferire sarà necessario specificare lo stesso genere di parametri per l'immagine sorgente:

```
6. il DC di origine è indicato nella nostra variabile ScreenDC, ovvero il DC dello
   schermo;
```


7. la coordinata X del punto da cui selezionare l'immagine da ricopiare è indicato nella struttura **COORD**, indicante la posizione del cursore del mouse;
8. la coordinata Y dello stesso punto è indicata nella struttura **COORD**;
9. la larghezza dell'area da copiare nell'immagine di destinazione è indicata in **DIMENSIONE**;
10. anche l'altezza è indicata in **DIMENSIONE**;

L'ultimo parametro è il tipo di disegnamento da effettuare. Le due chiamate differiscono soltanto per l'ultimo parametro: *vbNotSrcCopy* per i colori in negativo e *vbSrcCopy* per i colori normali; i due valori sono prelevati entrambi dall'enumerazione RasterOpConstants in base al valore della CheckBox **Invertito**.

La prova del programma è d'obbligo ma l'utilizzo semplicissimo. Basta scegliere il corretto fattore di riproduzione dalla *ComboBox* ed attivare o disattivare l'unica *CheckBox*.

Figura 2

Figura 3

Il codice è alquanto semplice; l'unica riga che può creare qualche perplessità è quella riguardante la funzione ***StretchBitBlt***. Basta studiare il significato dei parametri richiesti ed il suo utilizzo sarà immediatamente chiaro.

Fibia FBI
17 Maggio 2001

Torna all'indice degli HowTo

