GRAFIK und ANIMATION mit VISUAL BASIC

Hermann KÖBERL DSK-536\\AVI

Im Zeitalter von Pentium-Prozessoren, Super-VGA-Karten und jede Menge an RAM ist das Arbeiten mit bunten Grafiken, Sound und Animationen für jeden PC-Freak ein leicht erfüllbarer Wunsch geworden.

Das Einbinden von Grafikdateien in VisualBasic-Anwendungen gehört wohl zu den ersten „Gehversuchen" jedes VB-Anfängers. Es ist nicht einmal Programmcode erforderlich; es genügt die gewünschte Bilddatei interaktiv auszuwählen und als `Picture`-Eigenschaft der Form, eines Bildfeldes oder einer Anzeige zu definieren. Die Größenanpassung erfolgt durch die Eigenschaften `Autosize` bzw. '`Stretch`'. Das Ausdrucken der kompletten Form erledigt die '`PrintForm`'-Methode. Es besteht aber auch die Möglichkeit zur Laufzeit eine beliebige Grafik mit Hilfe der `LoadPicture(„Grafikdatei")`-Methode einzubinden.

Der einzige Schönheitsfehler der bisherigen VB-Versionen war wohl die Beschränkung auf Ikonen-, Bitmap- und Metafiles. Für das Frühjahr 97 ist Visual-Basic-Version 5.0 angekündigt und nach verläßlichen Informationen können neben vielen anderen Neuerungen damit auch endlich JPEG, GIF, PCX, TIFF und CGM-Grafikformate verarbeitet werden. Mit den zusätzlichen Internet-Features sind dem kreativen Multimedienfreund kaum noch Grenzen gesetzt.

Wer selbst gerne Grafiken entwirft, findet mit den zahlreichen Objekteigenschaften bzw. VB-Methoden für Koordinatensystem, Farbgestaltung und Zeichnen (Punkt, Linie, Kreis, Rechteck) eine Vielzahl von Möglichkeiten vor.

Aber mit dem Essen kommt bekanntlich auch der Appetit: Bunte Bilder sind zwar schön - aber Soundeffekte und bewegte Bilder machen viele Anwendungen erst richtig attraktiv. Videoclips, Musik oder Sprachausgabe in ein selbsterstelltes VB-Programm einbauen, das ist der Traum vieler Hobbyprogrammierer. Zum Anfertigen eines individuellen Videoclips, z. B. als Geburtstagsüberraschung oder zu Werbezwecken, ist eine Videoausrüstung und eine Video-Capture-Card für den PC (Kosten ca. 5000 S) erforderlich. Wie kann aber die Wiedergabe in ein VB-Projekt eingebaut werden?

Eine einfache Möglichkeit dafür sind OLE-Container. Es ist jedoch immer ein entsprechendes Windows-Programm erforderlich, welches als sogenannter OLE-Server dient, d.h. das eigene VB-Programm ruft z.B. eine Multimedienanwendung auf, ein WAV-Datei wird abgespielt und das ausführende Programm wird wieder deaktiviert.

Ein anderer Weg besteht im Einsatz des MultiMedien-CustomControls der professionellen Ausgabe von VB. Damit kann jeder sofort ein komplettes Videogerät am Bildschirm simulieren.

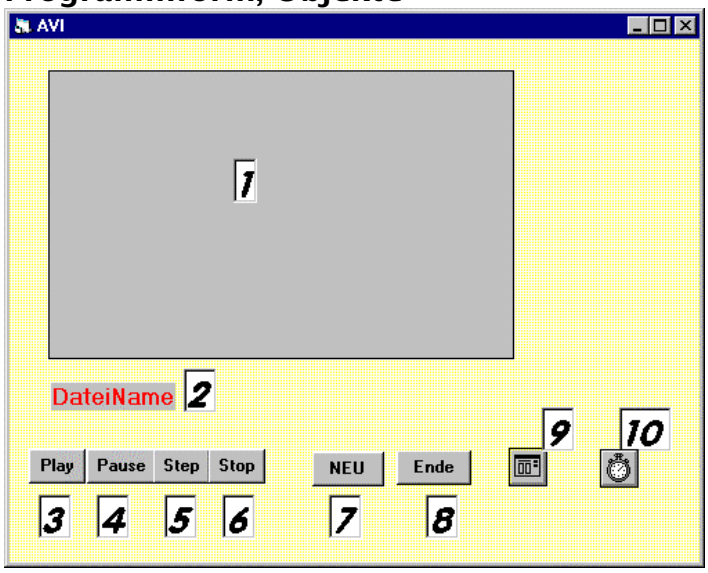
Im hier vorgestellten Projekt soll eine ganz einfache Lösung demonstriert werden: Der Aufruf einer einzigen Bibliotheksfunktion (API) ermöglicht das Abspielen von Videoclips (AVI) in einem bestimmten Bildfeld. Dafür benötigt man nur die Windows-Multimedien-Library namens `WINMM.DLL`, eine 32-Bit-Bibliothek, die mit Windows95 mitgeliefert wird.

Das nachfolgende Bild zeigt die Oberfläche mit der AVI-Datei `BRIDGEFLIGHT`. In Farbe sieht das Ganze natürlich attraktiver aus.

Die Befehlsschaltfläche *NEU* öffnet das StandardDialogfeld zur Dateiauswahl und eine beliebige AVI-Datei kann angeklickt werden. Mit *PLAY* wird diese abgespielt, *PAUSE* , *STEP* und *STOP* dienen zur Steuerung.

Systemvoraussetzung für dieses Projekt sind lediglich eine VGA- und Soundkarte und natürlich Windows95.

Programmform, Objekte



Nr.	*Objekt*	*Name*	*Eigenschaften*	*Ereignisse/ Methoden*
0	Form	AVIDEMO	Borderstyle, Backcolor, Caption,	Load
1	Bildfeld	Picture1	BackColor, hWnd, GotFocus	Cls, SetFocus
2	Bezeichnungsfeld	DateiName	Autosize, Caption, ForeColor, Font..	
3	Befehlsschaltfläche	Play	Caption, Font, Name	Click, SetFocus
4	Befehlsschaltfläche	Pause	Caption, Font, Name	Click
5	Befehlsschaltfläche	Step	Caption, Font, Name	Click
6	Befehlsschaltfläche	Stop	Caption, Font, Name	Click
7	Befehlsschaltfläche	Neu	Caption, Font, Name	Click
8	Befehlsschaltfläche	Ende	Caption, Font, Name	Click
9	Standarddialog	Auswahl	Filter, FileName	ShowOpen
10	Zeitgeber	Zeit1	Interval	

Programmablauf:

Alle MCI-Aktionen des Projektes werden mit einer einzigen DLL-Funktion durchgeführt. Es ist dies die Funktion `mciExecute` aus der Bibliothek `WINMM.DLL`. Bei Aufrufen sind nur die gewünschten Kommandos, eventuell mit Zusätzen, als String zu übergeben.

Die API-Funktion wird in den allgemeinen Definitionen in der Programmzeile

```
Private Declare Function mciExecute Lib „winmm.dll“
(ByVal LpstrCommand As String) As Long
```

definiert. Der Rückgabewert entspricht einer Long-Integer Variablen. Die Bibliothek `WINMM.DLL` ist nach jeder W95-Installation standardmäßig vorhanden - es sind daher keine zusätzlichen OCX-Module nötig.

Zunächst muß eine passende Datei ausgewählt werden. Danach erscheint der aktuelle Dateiname samt Pfad über den Steuerknöpfen.

Nach Betätigen der *PLAY*-Schaltfläche wird der Fokus auf das Bildfeld gesetzt und dessen Windows-Zugriffsnummer an die API-Funktion übergeben. Die AVI-Datei läuft nun in diesem markierten Feld ab. Es sind unterschiedliche Bildgrößen zu erwarten, daher ist es ratsam das Bildfeld mit der selben Hintergrundfarbe wie die Form zu versehen und als Borderstyle 0 (kein Rahmen) zu wählen.

Für die Zusatzfunktionen (*PAUSE, STEP, STOP, NEU* und *ENDE*) werden jeweils nur die erforderlichen Befehlssequenzen an die Funktion `mciExecute` übergeben, sowie durch die Enable-Eigenschaft unsinnige Kombinationen der Steuerflächen verhindert.

Im Übergabestring sind Dateinamen, `ALIAS` und `Action` jeweils getrennt, um den Einbau in andere Projekte zu erleichtern. Für die Wiedergabe von „WAV“-Dateien für Soundeffekte müßte z.B. nur als `ALIAS` der Text `TALK` übergeben werden.

Das nachfolgende Programm ist ausführlich kommentiert und nach Bedarf einfach zu ändern. Viel Spaß mit Visual Basic!

Programmcode:

```
VERSION 4.00
'Variable auf Modulebene
'DLL-Funktionen deklarieren

'Sendet Kommando-Strings zum MIDI-Treiber
Private Declare Function mciExecute Lib "winmm.dll" (ByVal
lpstrCommand As String) As Long
Dim mciFile As String    'AVI-Dateiname
Dim AliasName As String  'Zusatz bei API-Aufruf

Dim HwndFrame As Integer 'Speichert die Zugriffsnummer
                         '(hWnd) der Bildfläche
Dim Paused As Boolean    'Anzeige, wenn auf Pause geschaltet
Dim Warten As Integer    'Schalter für Warteposition
Dim I As Long            'API-Rückgabewert
Dim Offen As Boolean     'AVI-Datei offen? J/N
```

```
Private Sub Neu_Click()
   ' Stoppt das Video, schließt das  MCI device
   If Offen = True Then
       Action$ = "Close all"
       I = mciExecute(Action$)
       Offen = False
       DateiName.Caption = ""
   End If

  'Videobild löschen
  Picture1.Cls
  'neue Auswahl, Verzeichnis bleibt
  Auswahl.Filter = "AVI-Dateien|*.AVI"
  Auswahl.ShowOpen
  mciFile = Auswahl.filename
  DateiName.Caption = mciFile
  Play.SetFocus  'Fokus auf Play-Button
End Sub
```

```
Private Sub Play_Click()
  ' Alle geöffneten MCI devices schließen
  If Offen = True Then
    Action$ = "Close all"
    I = mciExecute(Action$)
    Offen = False
  End If
  ' Ausgesuchte Datei mit ALIAS=VID öffnen
  AliasName = "VID"
  Action$ = "open " + mciFile + " alias " + AliasName
  I = mciExecute(Action$)
  Offen = True
  DateiName.Caption = mciFile
  ' Focus auf das Bildfeld, wo das Video laufen soll
  ' Picture1 GotFocus gibt die windows-"handle #" (hWnd)
  ' als HwndFrame zurück
  Picture1.SetFocus  'Focus auf Bildfeld
  Zeit1.Interval = 500 'Zeitmesser einstellen
  Warten = 1         'um Vorgang zu bremsen
  Do
       DoEvents   'Ereignisse abfragen (Timer)
  Loop Until Warten = 0

  HwndFr$ = CStr(HwndFrame) 'Umwandlung in String
  'Übegabe der Handle-Nummer für AVI-Darstellungsbereich
  Action$ = "window " + AliasName + " handle " + HwndFr$
  I = mciExecute(Action$)

  'Video abspielen
  Action$ = "Play VID"
  I = mciExecute(Action$)
End Sub
```

```
Private Sub Pause_Click()
   ' Wenn noch keine Pause , Pause-Kommando
   If Paused = False Then
      Action$ = "Pause " + AliasName
   End If
   ' Wenn gerade Pause, weiterspielen
   If Paused = True Then
      Action$ = "Play " + AliasName
   End If

   ' Pause-Flag immer umschalten
   If Paused = True Then
      Paused = False
   Else
      Paused = True
   End If

   I = mciExecute(Action$)  'API-Befehl ausführen
End Sub
```

```
Private Sub Step_Click()
' Single step video , 1 Frame vorwärts
   Action$ = "step " + AliasName + " by 1"
   I = mciExecute(Action$)
End Sub
```

```
Private Sub Stop_Click()
  ' Stoppt das Video, schließt das  MCI device
  If Offen = True Then
    Action$ = "Close all"
    I = mciExecute(Action$)
    Offen = False
    DateiName.Caption = ""
  End If
End Sub
```

```
Private Sub Ende_Click()
  If Offen = True Then
    ' MCI-Kanal schließen
    Action$ = "Close all"
    I = mciExecute(Action$)
    Offen = False
  End If
  Unload Me
  End
End Sub
```

```
Private Sub Picture1_GotFocus()
  ' Speichert die  hWnd "handle number" vom Picture1
  HwndFrame = Picture1.hWnd
End Sub
```

```
Private Sub Zeit1_Timer()
  Warten = 0  'Ende der Wartezeit
  Zeit1.Interval = 0 'Uhr inaktivieren
End Sub
```