



Programmeren in Microsoft Visual Basic 6.0, lessenserie voor het voortgezet onderwijs HAVO/VWO

© David Lans, Emmauscollege, Marnix Gymnasium
Rotterdam, maart 2005

Hoofdstuk 9: Menu's toevoegen

9.0 Leerdoel

In dit negende hoofdstuk van de cursus "Programmeren in Visual Basic" leer je:

- kennis maken met de volgende onderdelen van een Visual Basic project:
 - menu-items toevoegen
 - submenu-items toevoegen
- nieuwe gebeurtenissen in een project beschrijven met programmacode, zoals:
 - programmacode aan menu-items koppelen

We doen dat aan de hand van de voorbeeldprojecten:

- De Schuifpuzzel
- De Schuifpuzzel met menu
- De Schuifpuzzel met submenu

Met dit hoofdstuk ben je, afhankelijk van je tempo, ongeveer 2 lesuren van 50 minuten bezig.

9.1 De schuifpuzzel (zonder menu).

In veel computerprogramma's is een menubalk aanwezig met daarin voor het programma belangrijke menu-opties. In Visual Basic zijn deze menu's eenvoudig toe te voegen



Open het project "De schuifpuzzel".



Bij een druk op de startknop cmdStart worden de puzzelstukjes verwisseld. Het spel begint en de puzzelstukjes, geïndexeerde opdrachtknoppen (commandbuttons cmdStukje(i)) moeten weer in de oorspronkelijke positie worden teruggeschoven.

De programmacode die bij de startknop is:

```
Private Sub cmdStart_Click()  
Dim beurt, richting, plaats As Integer  
Dim stukje As String  
' Er wordt 100 keer een willekeurig stukje met de lege plaats verwisseld  
For beurt = 1 To 100  
' selecteer een richting at random  
richting = Int(4 * Rnd)  
' de richting bepaalt de plaats van het stukje dat met de lege plaats wordt verwisseld  
Select Case richting  
Case 0 ' omhoog  
    plaats = legeplaats - 4  
Case 1 ' omlaag  
    plaats = legeplaats + 4  
Case 2 ' naar links  
    plaats = legeplaats - 1  
Case 3 ' naar rechts  
    plaats = legeplaats + 1  
End Select  
' check of random bepaalde richting een geldig blokje is  
If (plaats >= 0) And (plaats <= 15) Then  
' zo ja, verwissel stukjes  
' verwissel Visible  
cmdStukje(legeplaats).Visible = true  
cmdStukje(plaats).Visible = false  
' verwissel Caption  
cmdStukje(legeplaats).Caption = cmdStukje(plaats).Caption  
cmdStukje(plaats).Caption = ""  
' de nieuwe lege plek is de plaats waarmee de lege plaats is verwisseld  
legeplaats = plaats  
End If  
Next beurt  
End Sub
```

Nadat de puzzel met 100 verwisselingen door elkaar is geschoven, kan de speler de stukjes weer in de juiste posities gaan verschuiven. Dat doet de speler door op een stukje (opdrachtknop) naast de lege plaats te drukken. De programmacode die bij het drukken op de (geïndexeerde) puzzelstukjes hoort is:

```
Private Sub cmdStukje_Click(Index As Integer)
Dim i As Integer

' aangeklikte knopje is aangrenzend aan het lege? Wissel dan om!
If (Index = legeplaats - 4) Or (Index = legeplaats + 4) Or (Index = legeplaats - 1) Or (Index = legeplaats + 1) Then
    cmdStukje(legeplaats).Caption = cmdStukje(Index).Caption
    cmdStukje(Index).Visible = False
    cmdStukje(legeplaats).Visible = True
    legeplaats = Index
End If
' Controleer of alle stukjes op de goede plaats liggen
For i = 0 To 14
If Val(cmdStukje(i).Caption) <> i + 1 Or (legeplaats <> 15) Then
    Exit Sub
End If
Next
MsgBox "Het is gelukt!!!"

End Sub
```

Omdat ook bij dit programma de random-functie een belangrijke rol speelt dient bij het laden van het formulier weer de Randomize opdracht gegeven te worden. De lege plaats krijgt bij aanvang de waarde 15.

```
Private Sub Form_Load()
Randomize
legeplaats = 15
End Sub
```

Omdat de lege plaats door heel het programma heen wordt gebruikt dient deze bij de General declaraties te worden opgegeven.

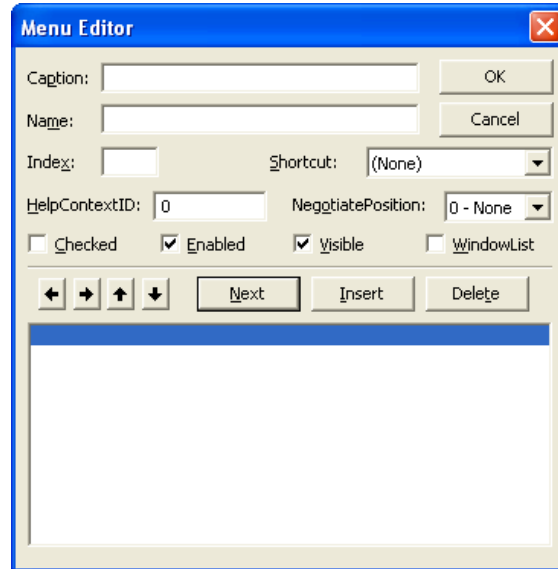
```
Private legeplaats As Integer
```

9.2 Menu-items toevoegen aan de schuifpuzzel.

De bovenstaande uitleg van de programmacode is bedoeld om je te laten begrijpen hoe het programma schuifpuzzel werkt. Waar het in deze brief echter om gaat is: hoe kan je aan een Visual Basic programma menu-items toevoegen?

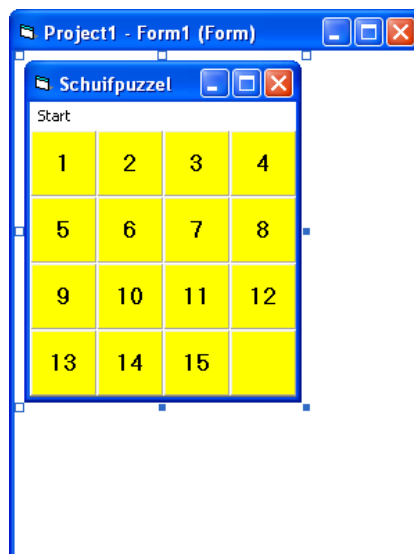
- ☞ We gaan de startknop vervangen door een startmenu.
Verwijder de startknop van het formulier.
Kies vervolgens voor “Tools” / “Menu-editor”.

In het beeld verschijnt de menu-editor van Visual Basic.



- ☞ Zet zowel in het vak “Caption” als in het vak “Name” de tekst “Start” en druk op “OK”.

Op je Visual Basic formulier is nu het menu-item “Start” toegevoegd.



- ☞ Dubbelklik op het menu-item “Start”.
Knip de code die bij de (inmiddels verwijderde) startknop staat en plak deze bij het startmenu.
Verwijder de startknop en de resterende code. Kijk of het menu werkt.

☞ Voeg met de menu-editor met de optie “Next” een menu-item “Stop” toe.

De code, die met een dubbelklik aan de stopknop moet worden toegevoegd, is

```
Private Sub Stop_Click()  
    End  
End Sub
```

Het resultaat is dat de schuifpuzzel er een stuk fraaier uit ziet met een heus menu:



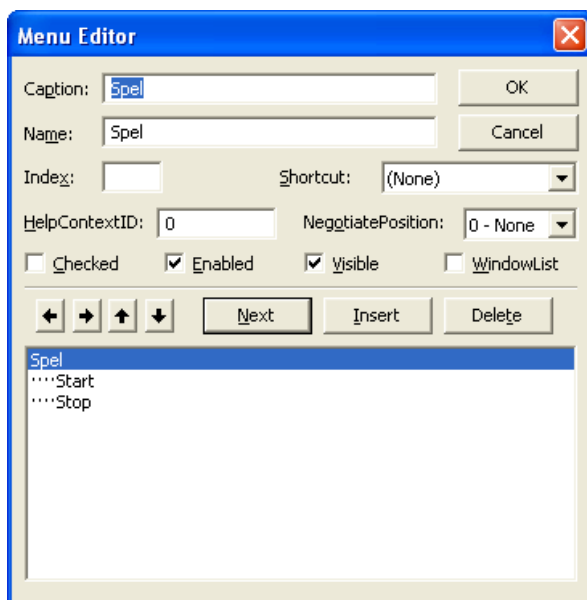
☞ Sla het project op onder de naam “De Schuifpuzzel met menu”.

9.3 Submenu's toevoegen aan een menu-item

Ook submenu's kunnen in Visual Basic eenvoudig worden aangemaakt:



Plaats met "Insert" het (hoofd)menu-item "Spel" boven de menu-items "Start" en "Stop".
Schuif de menu-items "Start" en "Stop" met de pijl naar rechts zodat het submenu-items worden.



De programmacode, die aan de submenu-items "Start" en "Stop" is gekoppeld, hoeft niet te worden veranderd. Het is immers nog steeds zo dat er code hoort bij een klik op de startknop en de stopknop. Alleen de positie van deze knoppen is veranderd van menu-item naar submenu-item.

9.4 Memory



Maak een project “Memory”, waarin je in elke beurt steeds 2 van de 16 kaartjes mag omdraaien. In het begin van het spel worden de kaartjes, met daarop de getallen 1 tot en met 8, willekeurig neergelegd. Draait de speler in één beurt twee kaartjes met daarop hetzelfde nummer, dan worden deze uit het spel genomen. Het spel is afgelopen als de speler alle tweetallen heeft gevonden. Voeg een menu toe met daarin de opties “nieuw spel” en “stop spel”.

