

Tili-toli elkészítése VB 2005 Express-ben

Cél: A program minél egyszerűbb megvalósítása; ne legyen benne tömb; segítse az objektumok megértését.

A felület beállítása

Űrlap elnevezése: Tili-toli

Űrlapra egy **toolbar**

Name: toolbar 1

Előnye: A VB 2005-ben a toolbarra tett gombok és egyéb vezérlők önállóan kezelhetők. Mivel fix helyen fix méretben van és viszonylag elegáns megoldás, kis programoknál kényelmesebb mint a vezérlőket az ablakra tenni.

Űrlapra egy **panel**

Name: panel1

Size: 400×400

középre húzni

Előnye: a rajta lévő objektumok hozzá vannak igazítva. (Az ablak paraméterei függenek a Windowban kiválasztott skin-től!, így pozícionálásra kényelmetlen használni!)

Panelra **nyomógombok**

Ezek valósítják meg a tologatható négyzeteket. Amelyikre kattintunk, az megy be az üres helyére. Lehet rá képet is tenni, de most feliratozzuk őket 1-15-ig. Most ez a felirat fogja azonosítani őket, de ha képeket tennénk rájuk, akkor a nevüket kellene használni.

Egyenként helyezzük fel őket.

(Kódból lehet objektumvektort is létrehozni, de ehhez már több előképzettség szükséges.)

Button1

Size: 100;100

Location: 0;0

Másolás egy sorba 100;0 --- 200;0 --- 300;0

Egy sor kijelölése

Másolás, igazítás: Format>Align>Bottoms

Egyenként átírjuk a *Text* tulajdonságot: 1, 2, 3, ... 15-re

Egybe kijelöljük és átírjuk a *Font* tulajdonságot: betűméret 36 pont

Gombok mozgatása

Fontos tudni, hogy hol vannak a gombok tologatás közben. Ehhez hagyományosan egy mátrix vagy egy vektor kellene, de ezt most nem szeretnénk kihasználni.

Megkerülése: *Tag* tulajdonság, amit mi adhatunk meg (ez egy biankó tulajdonság).

Induláskor megegyezik a sorszámmal, kézzel végigírjuk.

(Írható lenne hozzá egy ciklus, de kézzel sem tart hosszabb ideig.)

Hol van az üres hely? Erre bevezetünk egy változót *üres* néven, kezdeti értéke 16:

```
Public Class Form1
    Dim üres As Integer = 16
```

A gombok kijelölése után a *Click* eseményt választjuk, majd az eljárást átnevezzük. Fontos tudnunk, hogy eltérően pl. a VB6-tól, nem a neve mutatja meg, hogy mit kezel, hanem a sor végén lévő lista. *Sender*: az eseményt küldő objektum, amelyet *melyik* néven tipizálunk. Fontos megérteni, hogy ez csak másik neve egy létező objektumnak.

```
Private Sub ButtonX_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) ...

    Dim melyik As Button = sender
    Dim hol As Integer = melyik.Tag

    MsgBox(hol) 'csak tesztelésre
```

Attól függően, hogy a kiválasztott gombhoz képest merre van az üres hely, négy eset van:

```
If üres = hol - 4 Then
    melyik.Top -= 100
    melyik.Tag = üres
    üres = hol
End If

If üres = hol + 4 Then
    melyik.Top += 100
    melyik.Tag = üres
    üres = hol
End If

If üres = hol - 1 And üres Mod 4 <> 0 Then
    melyik.Left -= 100
    melyik.Tag = üres
    üres = hol
End If

If üres = hol + 1 And hol Mod 4 <> 0 Then
    melyik.Left += 100
    melyik.Tag = üres
    üres = hol
End If
```

Keverés

Matematika: A $16! = 20\,922\,789\,888\,000$ féle lehetséges elrendezésnek csak a fele rakható ki. A kirakás menete gráfelméleti módszerekkel modellezhető, akár algoritmus is készíthető hozzá. Érdeklődőbb osztályokban bepillantást adhat a programozás elméletébe is.

Irodalom: Csákány-Vajda: Játékok számítógéppel, Műszaki Könyvkiadó, Budapest, 1985

Célszerű keverés: pl. 1.000 db véletlenszerű kattintás, amely meghívja az előző rutint.

A toolstriphez hozzáadunk egy **Button**-t

Name: Keverés

DisplayStyle: Text (de lehetne hozzárendelni egy képet is)

Text: Keverés

majd két kattintással jön az eljárás.

Most visszaköszön, hogy nem tömbként hoztuk létre a nyomógombokat, meg kell keresnünk, hogy melyik van a véletlenszerűen kiválasztott helyen. A *b* változó végigszalad a *Panel 1* gombjain, és megnézi, hogy melyiknek a *Tag*-je a kiválasztott szám. Nem kell tudatosítani a tanulóknak, hogy ez egy nevezetes eljárás is lehetne ☺. Az adott gombra a program rákattint a *ButtonX_Click* eljárásunk meghívásával.

```
y = Int(Rnd() * 16) + 1
For Each b As Button In Panel1.Controls
    If b.Tag = y Then ButtonX_Click(b, e)
Next
```

A teljes eljárás pedig:

```
Private Sub Keverés_Click(ByVal sender As System.Object, ByVal e As...
    Dim y As Integer
    Randomize()

    For i As Integer = 1 To 1000
        y = Int(Rnd() * 16) + 1
        For Each b As Button In Panel1.Controls
            If b.Tag = y Then ButtonX_Click(b, e)
        Next
    Next
End Sub
```

Elkészültség ellenőrzése

Célszerű automatikusan felismerni, ha kész a tábla, és gratulálni. Ez az előzőek alapján már egyszerű lesz: akkor van kirakva, ha minden gomb *Tag* és *Text* tulajdonsága megegyezik. Lényegében az előző eljárást használjuk:

```
Sub Készvan()
    For Each b As Button In Panel1.Controls
        If b.Tag <> b.Text Then Exit Sub
    Next
    MsgBox("Gratulálunk, Ön nyert!")
End Sub
```

Az eljárást a kattintást kezelő eljárásból kell meghívni.

Fontos, hogy ez a helyzet már keveréskor is előállhat, így azt egy logikai változóval levédjük:

A *ButtonX_Click* vége:

```
    If Not kever Then Készvan()  
End Sub
```

Egy modulváltozó a keverés ellenőrzésére a modul elején:

```
Public Class Form1  
    Dim üres As Integer = 16  
    Dim kever As Boolean = False
```

A *Keverés* eljárás kiegészítése az elején és a végén:

```
Private Sub Keverés_Click(ByVal sender As System.Object, ByVal e As...  
    Dim x As Integer  
    Randomize()  
  
    kever = True  
    ...  
    kever = False  
End Sub
```

Továbbfejlesztési lehetőség: a lépések számolása. A *toolbar*-et kell kiegészíteni még egy címkével, amelybe folyamatosan írogatjuk a lépések számát. Az ehhez szükséges *lépés* nevű modulváltozót a *Keverés* eljárásban nullázzuk.

Farkas Csaba
farkascs@jedlik.hu