

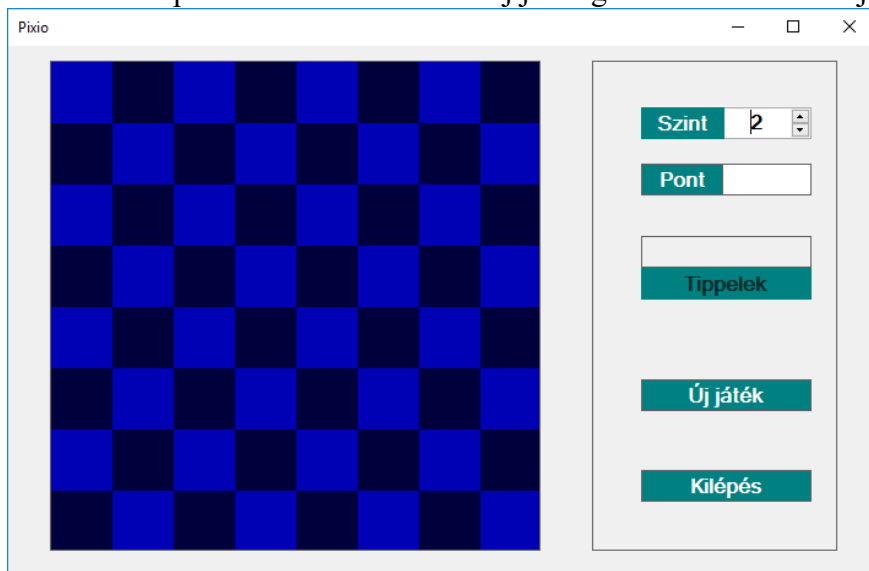
Programozási nyelvek gyakorlat
14P - 2017.12.11
Pixio (képkitalálás) játék megvalósítása

1. A játék leírása

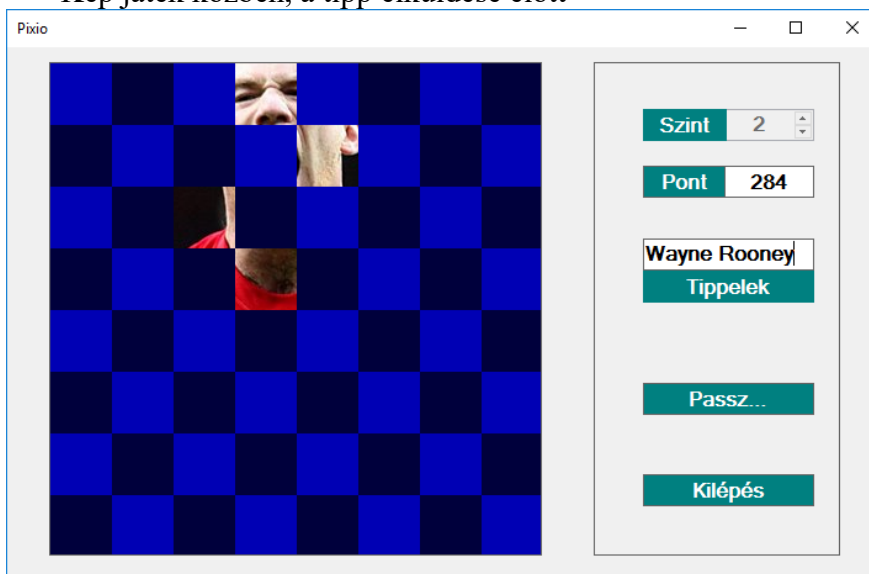
- A Pixio egy egyszerű képkitalálás játék.
- Lényege, hogy a program véletlenszerűen kisorsol egy képet, amelyeket a játék elején négyzet alakú lapokkal teljesen lefed, így az nem látható. A képen egy közismert ember látható, cél a nevének a kitalálása.
- A játékos egérekattintásokkal leszedhet általa kiválasztott lapokat, és amikor a már látható képrészletek alapján tippelni tud, akkor beírhatja a tipp mezőbe, hogy szerinte kit ábrázol a kép.
- A tipp után a program értékeli a választ, helyes válasz esetén kiszámít egy pontértéket, amely a leszedett lapok számától függ, minél kevesebb lapot kellett a játékosnak leszedni, annál több pontot kap a játékos.
- Ezután a játék új kép sorsolásával folytatható.

2. A játék egy lehetséges megvalósítása

- Indulókép: szintbeállítás után az Új játék gombbal kezdődik a játék



- Kép játék közben, a tipp elküldése előtt



- 1-es szinten 4*4, 2-es szinten 8*8, 3-as szinten 16*16 lap fedí a képet
- Egy rögzített pontszámról, pl. 300-ról indulunk, minden lap levétele valamennyivel csökkenti a pontszámot.

3. Segítség a fejlesztéshez

- Tanulmányozd a kiadott Pixio projektet, ez a játék egy nagyon leegyszerűsített változatát tartalmazza!
- A projekt a képeket egy **resource** (erőforrás) fájlban tartalmazza, amelynek tartalma a Solution explorer/ ResPixio.resx névre kattintva megtekinthető, illetve szerkeszthető, azaz bármely kép törölhető, illetve új kép hozzáadható.
- Egy link a resource fájlok kezeléséről:
[https://msdn.microsoft.com/en-us/library/7k989cfy\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/7k989cfy(v=vs.90).aspx)
- A resource fájlok képeket, ikonokat, hangokat stb. tartalmazhatnak, amelyek bekerülnek a lefordított állományba, így nagyon könnyen hordozható a program.
- A resource fájl-t egy ResourceManager objektum segítségével kezeljük, ennek létrehozása:

```
public FrmPixio()
{
    InitializeComponent();
    // A resource fájl betöltése, amelyben a képek vannak
    resourceManager = new ResourceManager("Pixio.ResPixio", GetType().Assembly);
    // A resource fájl tartalmának megtekintése és módosítása: Solution Explorer/ResPixio.resc
}
```

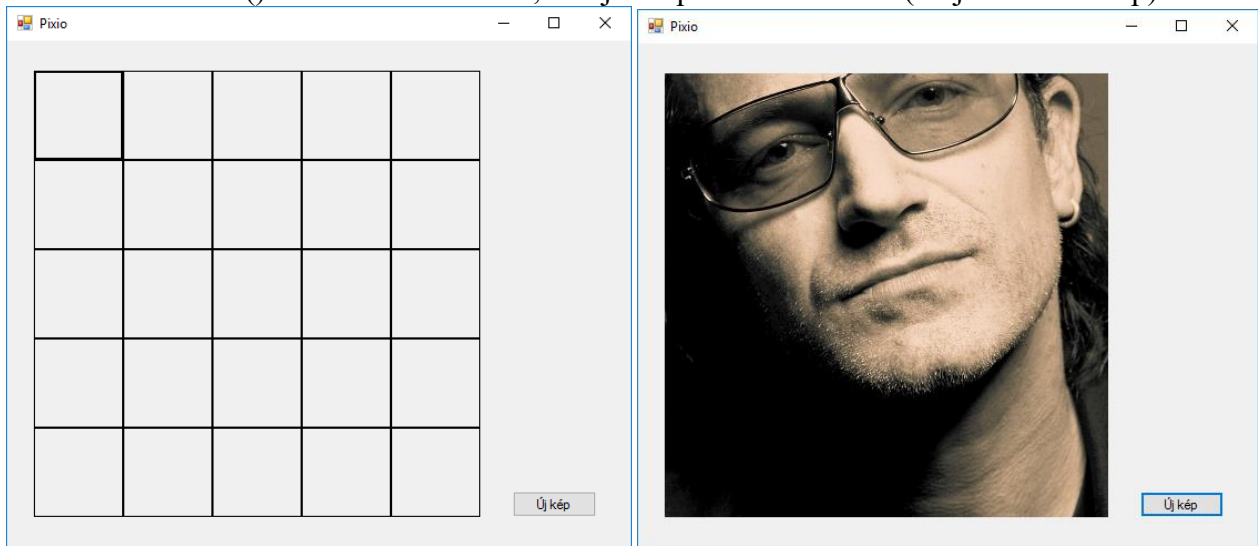
- A program egy **picturebox** komponenssel jeleníti meg a képeket, egy nevek tömb alapján, amelyben a képek resource fájl-beli elnevezései vannak.

```
private string[] nevek = { "Bono", "Kurt_Cobain", "John_Lennon", "Johnny_Depp", "Wayne_Rooney", "Julia_Roberts",
    "David_Beckham", "Angelina_Jolie", "Rihanna", "Roger_Federer", "Tiger_Woods" };
```

- A form betöltődésekor a 0. névnek megfelelő képet jelenítjük meg, de rögtön le is fedjük
- Megjegyzés: a 0. név helyett lehet

```
private void FrmPixio_Load(object sender, EventArgs e)
{
    pbKep.Image = (Bitmap)resourceManager.GetObject(nevek[0]); // Kép megjelenítése a 0. név alapján
    pbKep.SizeMode = PictureBoxSizeMode.CenterImage;
    Lefed();
}
```

- Ha a Lefed() metódust kivennénk, a teljes kép láthatóvá válna (ld. jobb oldali kép)



- A kép lefedését egy külön metódusba tettük. A metódus a pbKep picturebox Controls tulajdonságát kezeli, amely egy lista, és az ahhoz adott komponensek a képen jelennek meg.
- A lefedéshez **Button** objektumokat használtunk, de alkalmazhattunk volna **PictureBox**-okat is.
- Egy dupla ciklussal, mátrix-szerűen helyezzük el a button objektumokat, előtte töröljük a Controls esetleges tartalmát.
- A példában 5*5 button-t használunk a lefedéshez, a button méretét a picturebox méretéből számoljuk leosztással.

```
// A kép lefedése Button objektumokkal
private void Lefed()
{
    pbKep.Controls.Clear();
    int sor = 5;
    for (int i = 0; i < sor; i++)
        for (int j = 0; j < sor; j++)
        {
            Button b = new Button();
            b.Width = pbKep.Width / sor; // A gomb szélessége a kép szélességének az ötöde (5 sor esetén)
            b.Height = b.Width;
            b.Top = i * b.Width;
            b.Left = j * b.Height;
            b.FlatStyle = FlatStyle.Flat;
            b.Click += button_Click; // Click eseménykezelő hozzárendelése
            pbKep.Controls.Add(b); // Gomb felrakása a képre
        }
    btnUj.Focus(); // Fókusz az új kép gombra
}
```

- A button-ok (közös) eseménykezelője láthatatlanná teszi azt a gombot, amelyre éppen kattintottunk.

```
// Egy gomb "leszedése"
private void button_Click(object sender, EventArgs e)
{
    (sender as Button).Visible = false;
    btnUj.Focus();
}
```

- Az új kép gombra kattintva véletlenszerűen választunk egy képet a resource fájlból, és megjelenítjük, illetve rögtön le is fedjük.

```
// Új kép kiválasztása a resource fájlból
private void btnUj_Click(object sender, EventArgs e)
{
    pbKep.Image = (Bitmap)resourceManager.GetObject(nevek[r.Next(nevek.Length)]);
    Lefed();
}
```

4. Feladatok

- Válogass a projekthez saját kitalálendő képeket!
- Valósítsd meg a szintbeállítást, a leírás szerint, esetleg önálló verzió szerint. Minél magasabb a szint sorszáma, annál több lap fedje a képet!
- Valósítsd meg a pontérték folyamatos megjelenítését: minél több lapot levett a játékos, annál kevesebb legyen a pont!
- Valósítsd meg a tippelést! A játékos írhasson be egy nevet, és a tippelés gombra kattintva küldhesse el, a program pedig állapítsa meg, hogy helyes-e a válasz, illetve mutassa meg a képet!
- Valósítsd meg a passz funkciót, ekkor a program fedje fel a képet, és írja ki a nevet!
- Legyen lehetőség új játék indítására is!
- **Készíts egyedi design-t a programhoz!**
- Egészítsd ki legalább 2 önállóan tervezett funkcióval a programot!