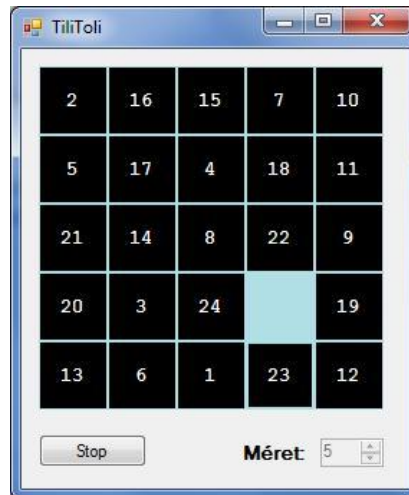
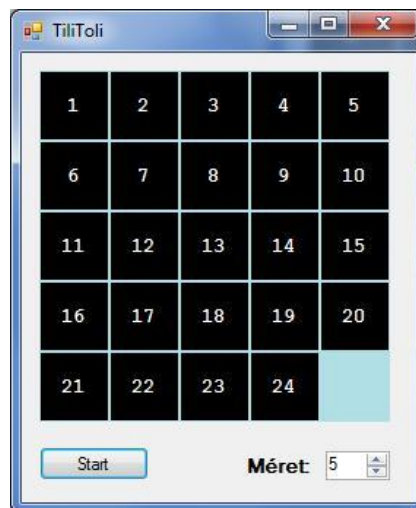


A TiliToli játék egyszerű változatában egy  $N \times N$ -es táblán összekevert számokat kell a helyükre tologatnunk! (A tologatáshoz egy négyzetet természetesen üresen kell hagynunk.)



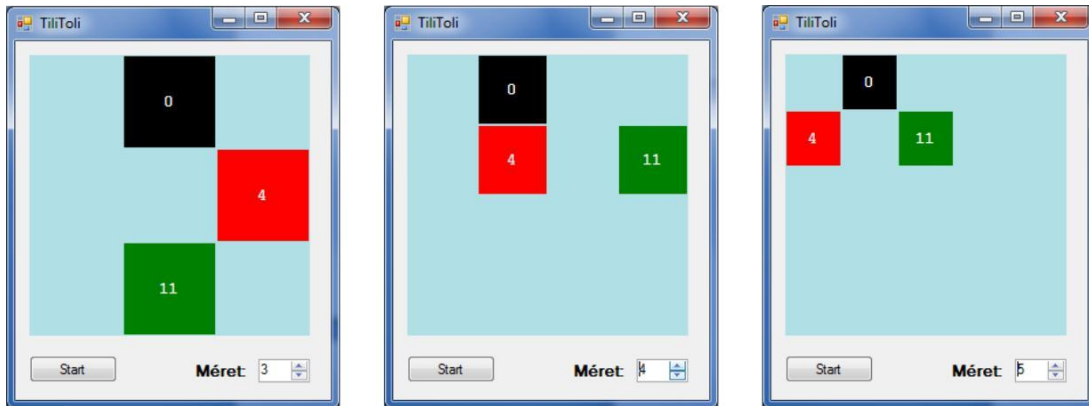
**Kevert állapot**



**Kezdő, illetve kirakott állapot**

1. Teszteld a kiadott TiliToli.exe programot  $3 \times 3$ -as táblán!
2. Töltsd be a Visual Studio-ban a megvalósítás alapját képező TiliToli\_A projektet. A program e kezdetleges verziója elhelyez 3 gombot a játéktérületen, különböző helyeken, számfelírással, és színnel.

a. Próbálj ki minden lehetséges méretbeállítást!



- Mitől függ, hogy melyik sorba és oszlopba kerülnek a gombok?

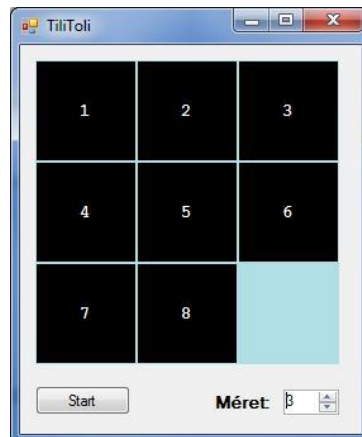
b. Értelmezd a FrmTiliToli osztály kódját!

- Tegyé fel még két gombot: az egyik a bal felső sarokba kerüljön, 7-es felirattal, a másikat tetszőlegesen válaszd meg!

c. Értelmezd a JatekAdatok és Gomb osztályok kódját!

- Mi a szerepe a JatekAdatok osztályban tárolt adatoknak?  
 - Mit jelent a Gomb osztály sorszám és érték adattagja? Melyik látható a gombon?  
 - Mit és hogyan számít ki a Gomb osztály SzamitTop() és SzamitLeft() metódusa?  
 Értelmezd a számítások képletét!

d. **Feladat:** módosítsd a Form1 osztály GombFelrak metódusát úgy, hogy a program indulását követően, illetve méretbeállítás után is a kezdőállapot legyen látható!



**3\*3-as méret beállítása után**

Minta: TiliToli\_B.exe

Segítség a kódoláshoz a következő oldalon:

```

public void Beallit()
{
    JatekAdatok.N = (int)nudMeret.Value; // Sorok száma
    JatekAdatok.Meret = pnlJatekTerulet.Width / JatekAdatok.N; // Egy gomb mérete pixelekben
    /* Új sor */
    JatekAdatok.Gombokszama = (int)Math.Pow(JatekAdatok.N,2)-1; // A gombok darabszáma
}

private void GombFelrak()
{
    pnlJatekTerulet.Controls.Clear();
    /* Új, illetve módosított sorok */
    for (int i = 0; i < JatekAdatok.Gombokszama; i++)
    {
        pnlJatekTerulet.Controls.Add(new Gomb(i, i + 1, Color.Black));
    }
}

```

### 3. Teszteld a TiliToli\_C.exe verziót!

Ebben a verzióban, ha olyan gombra kattintunk, amely mellett üres hely található, akkor a gomb az üres helyre mozog, és az elmozgatott gomb korábbi helye lesz üres.

#### a. Bővítsd a projekt kódját a következőkkel!

```

namespace TiliToli
{
    class JatekAdatok
    {
        private static int n;
        private static int meret;
        private static int gombokszama;
        /* Új sor */
        private static int ureshely;

        public static int N { get { return n; } set { n = value; } }
        public static int Meret { get { return meret; } set { meret = value; } }
        public static int Gombokszama { get { return gombokszama; } set { gombokszama=value; } }
        /* Új sor */
        public static int Ureshely { get { return ureshely; } set { ureshely = value; } }
    }
}

```

JatekAdatok.cs

```

public void Beallit()
{
    JatekAdatok.N = (int)nudMeret.Value; // Sorok száma
    JatekAdatok.Meret = pnlJatekTerulet.Width / JatekAdatok.N; // Egy gomb mérete pixelekben
    JatekAdatok.Gombokszama = (int)Math.Pow(JatekAdatok.N,2)-1; // A gombok darabszáma
    /* Új sor */
    JatekAdatok.Ureshely = JatekAdatok.Gombokszama; // Az üres hely kezdőpozíciója
}

```

FrmTiliToli.cs

b. Bővítsd a Gomb osztályt a következő metódusokkal!

```
/* Új metódus */
private int UresHelyIrany()
{
    if ((sorszam + 1 == JatekAdatok.Ureshely))
        return 0; // Jobbra
    else if ((sorszam - 1 == JatekAdatok.Ureshely))
        return 1; // Balra
    else if ((sorszam + JatekAdatok.N == JatekAdatok.Ureshely))
        return 2; // Le
    else if ((sorszam - JatekAdatok.N == JatekAdatok.Ureshely))
        return 3; // Fel
    else return -1;
}
```

```
/* Új metódus */
public void Eltol(int irany)
{
    if (irany > -1)
    {
        JatekAdatok.Ureshely = sorszam;
        switch (irany)
        {
            case 0: sorszam++;
                    break;
            case 1: sorszam--;
                    break;
            case 2: sorszam += JatekAdatok.N;
                    break;
            case 3: sorszam -= JatekAdatok.N;
                    break;
        }
        Top = SzamitTop(); Left = SzamitLeft();
    }
}

/* Új metódus */
private void Gomb_Click(object sender, EventArgs e)
{
    Eltol(UresHelyIrany());
}
```

c. A Gomb osztály konstruktorát bővítsd a következő sorral!

```
/* Új sor */
this.Click += new EventHandler(Gomb_Click);
```

- Értelmezd a beírt metódusokat, kódokat!
- Teszteld a programot! Vedd észre, hogy némely esetben nem megengedett mozgás is történhet!
- Javítsd a kódot úgy, hogy ez ne következzen be!

#### 4. Önállóan megoldandó feladatok

##### Egyszerűbb feladatok:

- a. A gombok mozgatása csak a Start gombra kattintás után legyen lehetséges! Ekkor a gomb felirata váltson Stop-ra! Ha a Stop gombra kattintunk, akkor ne lehessen a gombokat tovább mozgatni!
- b. Ha olyan gombra állunk az egérkurzorral, amely mellett nincs üres hely (azaz nem mozgatható el), a színe változzon pirosra! Egyénként pedig változzon zöldre!
- c. A gombokon látható számok mérete igazodjon a gombok méretéhez!

##### Összetettebb feladatok:

- d. A Start gombra kattintva a gombok keveredjenek össze!
  - 1. megoldási javaslat: már a gombok felrakása előtt keverjük össze egy listában, vagy tömbben a gombokra tehető számokat, és a felrakás során a kevert sorrendben helyezzük azokat a gombokra! (Ez a megoldás az esetek felében úgy kever, hogy nem lesz kirakható!)
  - 2. megoldási lehetőség: a Start gombra kattintva automatikusan kezdje el a program tologatni a gombokat! Ha így keverünk, az biztos kirakható lesz!
- e. Ha a gombok a helyükre kerültek, akkor a program jelezze ezt (Pl. "Gratulálok, kirakta!"), és ne lehessen tovább tologatni! Lehessen új játékot kezdeni!
- f. Ha játék közben be akarjuk zárni a programot, akkor a program kérjen megerősítést!

#### 5. Teszteld a Kepkivag programot!

Írd át a Tili-Toli programot úgy, hogy számok helyett egy kép részleteit kelljen a helyére tologatni!