

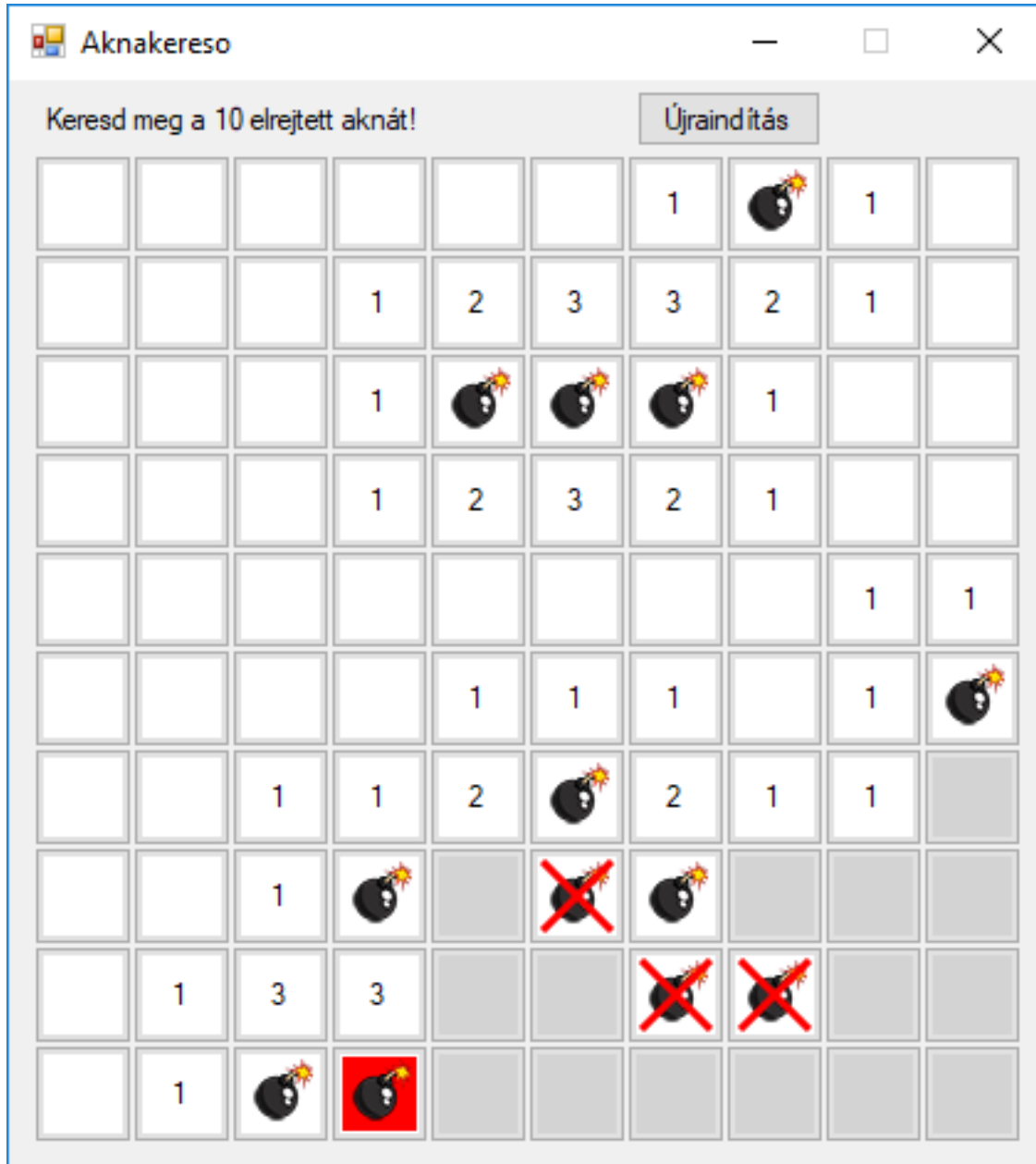
# Programozás C# nyelven

7. előadás



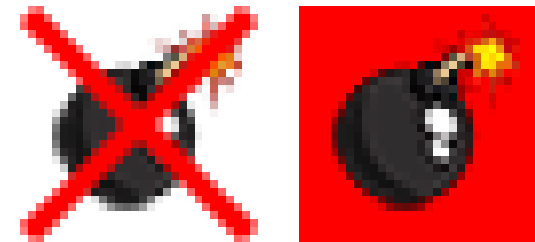
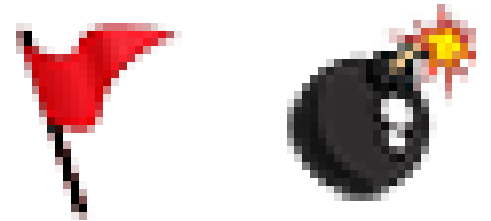
<http://e-learning.ujs.sk/>

# Feladat: Készítsünk aknakereső játékot!



Mire lesz szükségünk?

- **Képekre:**



- **Nyomógombokra:**  
**100 darab !!!**  
(futási időben létrehozva)

# I. Nyomógomb létrehozása futási időben

027 OOP Button

```
public partial class Form1 : Form
```

```
{
```

```
    public Form1()
```

```
    {
```

```
        InitializeComponent();
```

```
    }
```

```
    private void Form1_Load(object sender, EventArgs e)
```

```
    {
```

```
        // új nyomógomb létrehozása
```

```
        Button b = new Button();
```

```
        // a gomb tulajdonságainak beállítása
```

```
        b.Text = "Katt ide!";
```

```
        b.Location = new Point(50, 50);
```

```
        b.Size = new Size(80, 30);
```

```
        // a függvény hozzárendelése a 'MouseDown' eseményhez
```

```
        b.MouseDown += new MouseEventHandler(MouseDownEvent);
```

```
        // a gomb hozzáadása a Form-hoz
```

```
        this.Controls.Add(b);
```

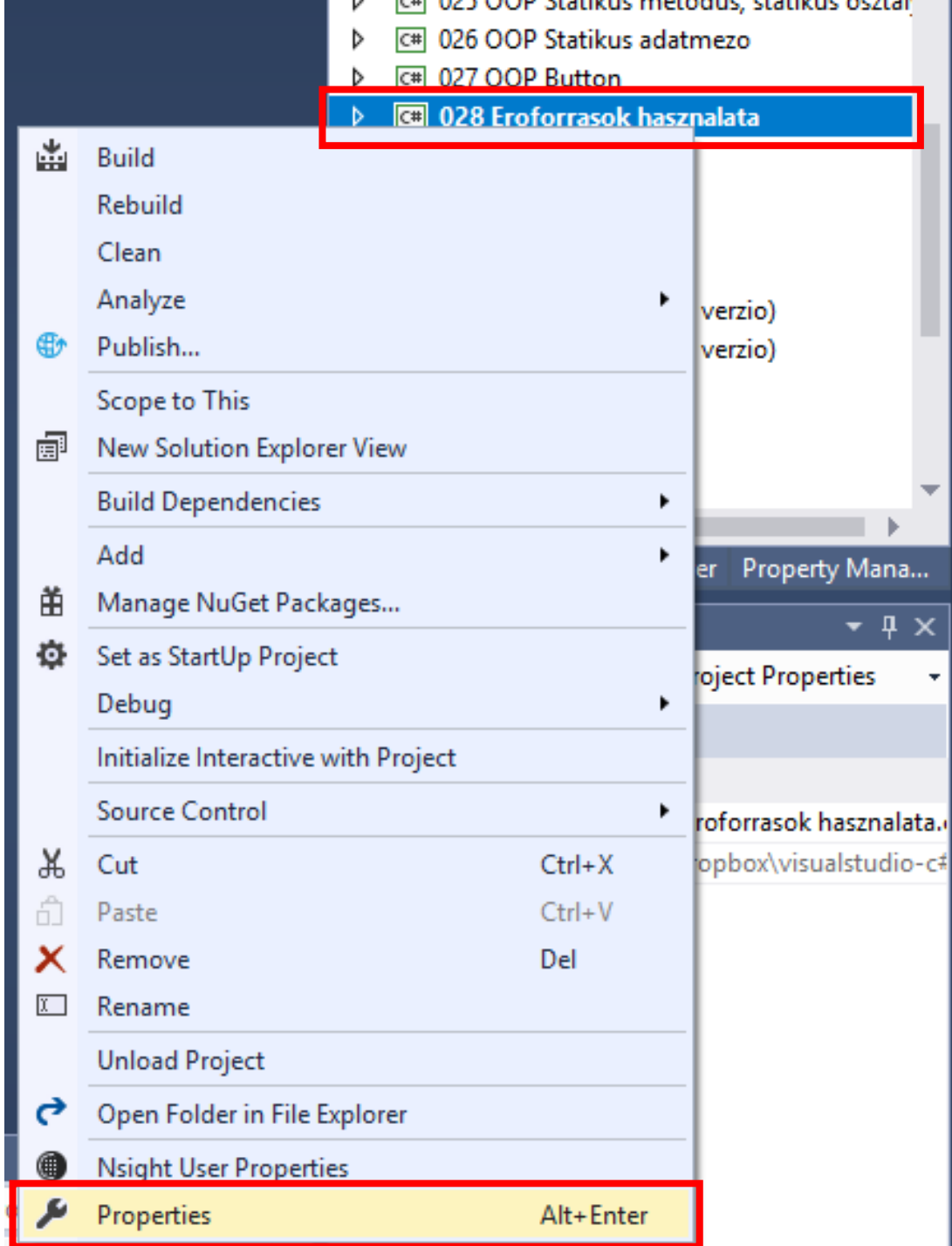
```
    }
```

```
private void MouseDownEvent(Object sender, MouseEventArgs e)
{
    if (e.Button == MouseButton.Right)
    {
        MessageBox.Show("A gombon JOBB egérgomb kattintás történt!",
            "Kattintás");
    }
    if (e.Button == MouseButton.Left)
    {
        MessageBox.Show("A gombon BAL egérgomb kattintás történt!",
            "Kattintás");
    }
}
```

```
}
```

## II. Erőforrások (resources) használata

**Erőforrás (Resource):** a forráskódon és adatokon túli összetevők, például az alkalmazásban használt képek, hangok, stb.



- Application
- Build
- Build Events
- Debug
- Resources**
- Services
- Settings
- Reference Paths
- Signing
- Security
- Publish
- Code Analysis

Strings Add Resource Remove Resource Access Modifier: Internal

Add Existing File...  
Add New String  
New Image  
Add New Icon  
Add New Text File

	Name	Value
▶▶	Strin	

Add existing file to resources

< > << 028 Eroforrasok h... > ! kepek > Prehľadávateľ: ! kepek

Usporiadať Nový priečinok

> python dc

> python-n

> python-pr

> python-pr

> python-sp

> visualstud

> eclipse

> Egyetem

> Egyetem ny

> csd

bomba.png

felrobbant\_bomb  
a.png

nincs\_bomba.png

zaszlo.png

Názov súboru: "bomba.png" "felrobbant\_bor" All Files (\*.\*)

Qtvorit' Zrušit'

```
private int bombaSzamlalo = 0;
```

```
private void button2_Click(object sender, EventArgs e)
{
    bombaSzamlalo++;
    switch (bombaSzamlalo)
    {
        case 1:
            button2.Image = Properties.Resources.bomba;
            break;
        case 2:
            button2.Image = Properties.Resources.nincs_bomba;
            break;
        case 3:
            button2.Image = Properties.Resources.felrobbant_bomba;
            break;
        case 4:
            button2.Image = null;
            bombaSzamlalo = 0;
            break;
    }
}
```

Keress meg a 10 elrejtett aknát!

Aknakereso

Keress meg a 10 elrejtett aknát!

Újraindítás

Button

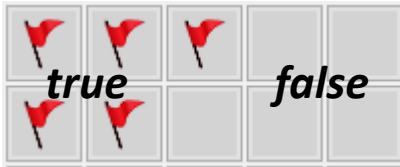
Hely : Button

....  
....  
....

Milyen új adatmezői,  
tulajdonságai és metódusai  
legyenek a Hely osztálynak?



„**private bool zaszlo**” – meg van-e jelölve zászlóval?



„**public void SetZaszlo(bool x)**”

„**public bool GetZaszlo()**”

„**private bool akna**” – van-e ezen a helyen akna?



„**public void SetAkna(bool x)**”

„**public bool GetAkna()**”

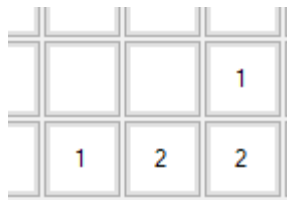
„**private bool felfedve**” – fel van-e fedve a hely?



„**public void SetFelfedve(bool x)**”

„**public bool GetFelfedve()**”

„**public int AknakMellette { set; get; }**” – a hely körüli aknák száma



0, 1, 2, ...

**„public Hely(int x, int y)”** – konstruktor

**„public void Alaphelyzet()”** – a hely beállítása alaphelyzetbe



**„private void JelolesJobbEgerkattintassal(object sender, MouseEventArgs e)”** – a „MouseDown” eseményhez

hozzárendelt függvény (konstruktóban rendeljük hozzá)

--> jobb egérgombbal kattintáskor a hely megjelölése zászlóval, vagy a zászló eltüntetése (de csak ha még nincs felfedve a hely és nincs vége a játéknak)



**„public static bool vege”** – vége van-e a játéknak (true/false)

```
using System.Drawing;
using System.Windows.Forms;

namespace _029_Aknakereso
{
    public class Hely : Button
    {
        private bool zaszlo = false;    // meg van-e jelolve zaszloval
        private bool akna = false;       // van-e ezen a helyen akna
        private bool felfedve = false;    // fel van-e fedve a hely

        public int AknakMellette { set; get; } = 0;    // a hely koruli aknak szama

        public static bool vege = false;              // vege-e a jateknank

        // konstruktor
        public Hely(int x, int y)
        {
            this.Size = new Size(40, 40);
            this.Location = new Point(x, y);
            this.SetStyle(ControlStyles.Selectable, false);
            this.Alaphelyzet();
            this.MouseDown += new MouseEventHandler(JelolesJobbEgerkattintassal);
        }
    }
}
```

```
// a hely beallitasa alaphelyzetbe
```

```
public void Alaphelyzet()
```

```
{
```

```
    this.Text = "";
```

```
    this.SetZaszlo(false);
```

```
    this.SetAkna(false);
```

```
    this.SetFelfedve(false);
```

```
    this.AknakMellette = 0;
```

```
}
```

```
// "MouseDown" esemenyhez hozzarendelt fuggveny
```

```
// - a hely megjelolese zaszloval / jeloles visszavonasa
```

```
private void JelolesJobbEgerkattintassal(Object sender, MouseEventArgs e)
```

```
{
```

```
    if (e.Button == MouseButton.Right && !Hely.vege && !this.felfedve)
```

```
    {
```

```
        this.SetZaszlo(!this.zaszlo);
```

```
    }
```

```
}
```

```
// akna beallitasa igaz/hamis-ra
```

```
public void SetAkna(bool x)
```

```
{
```

```
    this.akna = x;
```

```
}
```

```
// van-e akna a helyen
```

```
public bool GetAkna()
```

```
{
```

```
    return this.akna;
```

```
}
```

```
// zaszlo beallitasa igaz/hamis-ra
```

```
public void SetZaszlo(bool x)
```

```
{
```

```
    this.zaszlo = x;
```

```
    if (this.zaszlo)
```

```
    {
```

```
        this.Image = Properties.Resources.zaszlo;
```

```
    }
```

```
    else
```

```
    {
```

```
        this.Image = null;
```

```
    }
```

```
}
```

```
// meg van-e jelolve a hely zaszloval
```

```
public bool GetZaszlo()
```

```
{
```

```
    return this.zaszlo;
```

```
}
```

// a hely fedese - beallitasa igaz/hamis-ra

```
public void SetFelfedve(bool x)
```

```
{
```

```
    this.felfedve = x;
```

```
    if (this.felfedve)
```

```
    {
```

```
        this.BackColor = Color.White;
```

```
        if (this.akna)
```

```
        {
```

```
            if (Hely.vege)
```

```
            {
```

```
                this.Image = Properties.Resources.bomba;
```

```
            } else
```

```
            {
```

```
                this.Image = Properties.Resources.felrobbant_bomba;
```

```
            }
```

```
        } else
```

```
        {
```

```
            if (Hely.vege && this.zaszlo)
```

```
            {
```

```
                this.Image = Properties.Resources.nincs_bomba;
```

```
            } else
```

```
            {
```

```
                this.Image = null;
```

```
                if (this.AknakMellette > 0)
```

```
                {
```

```
                    this.Text = this.AknakMellette.ToString();
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
    else
    {
        this.BackColor = Color.LightGray;
    }
}
```

```
// fel van-e fedve a hely
public bool GetFelfedve()
{
    return this.felfedve;
}
```

```
}
```

```
}
```

## A „Hely” osztály használata a Form1 osztályban:

- 10x10 drb. objektum létrehozása, ezekre hivatkozás egy 10x10-es mátrix segítségével:

```
private Hely[,] aknamezo = new Hely[10, 10];
```

```
for (int i = 0; i < 10; i++)  
{  
    for (int j = 0; j < 10; j++)  
    {  
        aknamezo[i, j] = new Hely(i * 40 + 10, j * 40 + 30);  
        aknamezo[i, j].Click += new EventHandler(BalEgerKattintas);  
        Controls.Add(aknamezo[i, j]);  
    }  
}
```

- 10 drb. akna elhelyezése: „**SetAkna(true);**”.
- Annak megszámlálása, hogy mindegyik hely körül mennyi akna van, majd az „**AknakMellette**” tulajdonság beállítása.



- „private void BalEgerKattintas(object sender, EventArgs e)“:

- Ha **nincs vége** a játéknak:
  - **Felfedni a helyet**: „SetFelfedve(true);“.
  - Ha **akna van** ezen a helyen,  
akkor **vége a játéknak** 😞.
  - Ha **nincs akna** ezen a helyen ...
    - Ha **üres hely** (nincs körül sehol sem akna),  
akkor **rekurzívan felfedni nagyobb részt**.
    - Ha **már az összes hely fel van fedve** (ahol  
**nincs akna**), akkor **vége a játéknak** 😊.

- A játék végén az aknák helyeinek megjelenítése.
- Új játék indítása:
  - a helyek alaphelyzetbe állítása: „Alaphelyzet()”,
  - 10 akna véletlen elhelyezése, majd mindegyik hely  
körül az aknák megszámlálása.

```
private Random randgen = new Random();
private Hely[,] aknamezo = new Hely[10, 10];

private void Form1_Load(object sender, EventArgs e)
{
    // Form beallitasa
    ClientSize = new Size(420, 440);
    FormBorderStyle = FormBorderStyle.FixedSingle;
    MaximizeBox = false;
    // Hely objektumok letrehozasa es hozzaadasa a Form-hoz
    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            aknamezo[i, j] = new Hely(i * 40 + 10, j * 40 + 30);
            aknamezo[i, j].Click += new EventHandler(BalEgerKattintas);
            Controls.Add(aknamezo[i, j]);
        }
    }
    // jatek inditasa
    JatekInditasa();
}
```

```
private void button2_Click(object sender, EventArgs e)
{
    // a játék újraindítása
    JatekInditasa();
}
```

// "Click" esemény - kattintás bal egergombbal egy adott helyre

```
private void BalEgerKattintas(object sender, EventArgs e)
{
    if (!Hely.vege)
    {
        Hely h = (Hely)sender;
        h.SetFelfedve(true);
        if (h.GetAkna())
        {
            JatekVege();
        }
        else {
            if (h.AknakMellette == 0)
            {

```

```

// melyik helyre kattintottunk?
int x = 0, y = 0;
for (int i = 0; i < 10; i++)
{
    for (int j = 0; j < 10; j++)
    {
        if (aknamezo[i, j] == h)
        {
            x = i;
            y = j;
        }
    }
}

// rekurzivan felfedjuk a kozeli helyeket is
RekurzivFelfedes(x, y);
}

// osszes hely fel van-e fedve, kiveve az aknakat?
int drb = 0;
for (int i = 0; i < 10; i++)
{
    for (int j = 0; j < 10; j++)
    {
        if (!aknamezo[i, j].GetFelfedve())
        {
            drb++;
        }
    }
}
}

```

```
    if (drb == 10)
    {
        JatekVege();
        MessageBox.Show("Sikerült az összes aknát megtalálnod!",
            "Gratulálunk!");
    }
}
}
```

```

private void RekurzivFelfedes(int x, int y)
{
    aknamezo[x, y].SetFelfedve(true);
    if (aknamezo[x, y].AknakMellette == 0)
    {
        int[] xd = { -1, 0, +1, -1, +1, -1, 0, +1 };
        int[] yd = { -1, -1, -1, 0, 0, +1, +1, +1 };
        for (int i = 0; i < 8; i++)
        {
            if (x + xd[i] >= 0 && x + xd[i] < 10
                && y + yd[i] >= 0 && y + yd[i] < 10)
            {
                if (!aknamezo[x + xd[i], y + yd[i]].GetFelfedve())
                {
                    RekurzivFelfedes(x + xd[i], y + yd[i]);
                }
            }
        }
    }
}

```

```
private void JatekInditasa()
{
    // helyek alapertelmezett helyzetbe allitasa
    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            aknamezo[i, j].Alaphelyzet();
        }
    }
    // 10 drb. akna elrejtése veletlen helyekre
    int n = 0;
    do
    {
        int x = randgen.Next(10);
        int y = randgen.Next(10);
        if (!aknamezo[x, y].GetAkna())
        {
            aknamezo[x, y].SetAkna(true);
            n++;
        }
    } while (n < 10);
}
```

```
// a helyek koruli aknak szamanak hozzarendelese az egyes helyekhez
for (int i = 0; i < 10; i++)
{
    for (int j = 0; j < 10; j++)
    {
        aknamezo[i, j].AknakMellette = AknakAHelyKorul(i, j);
    }
}
// a jatek aktivra allitasa
Hely.vege = false;
}
```



```
private int AknakAHelyKorul(int x, int y)
{
    int[] xd = { -1, 0, +1, -1, +1, -1, 0, +1 };
    int[] yd = { -1, -1, -1, 0, 0, +1, +1, +1 };
    int drb = 0;
    for (int i = 0; i < 8; i++)
    {
        if (x + xd[i] >= 0 && x + xd[i] < 10
            && y + yd[i] >= 0 && y + yd[i] < 10)
        {
            if (aknamezo[x + xd[i], y + yd[i]].GetAkna())
            {
                drb++;
            }
        }
    }
    return drb;
}
```

```
private void JatekVege()
{
    // a jatek inaktivra allitasa
    Hely.vege = true;
    // az aknak helyeinek es a bejelolt helyeknek a felfedese
    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            if (!aknamezo[i, j].GetFelfedve()
                && (aknamezo[i, j].GetAkna() || aknamezo[i, j].GetZaszlo()))
            {
                aknamezo[i, j].SetFelfedve(true);
            }
        }
    }
}
```

## Összefoglalás:

- Nyomógomb (Button) létrehozása futási időben
- Erőforrások (Resources)
- Aknakereső játék!