

# BEÁGYAZOTT RENDSZEREK ALAPJAI

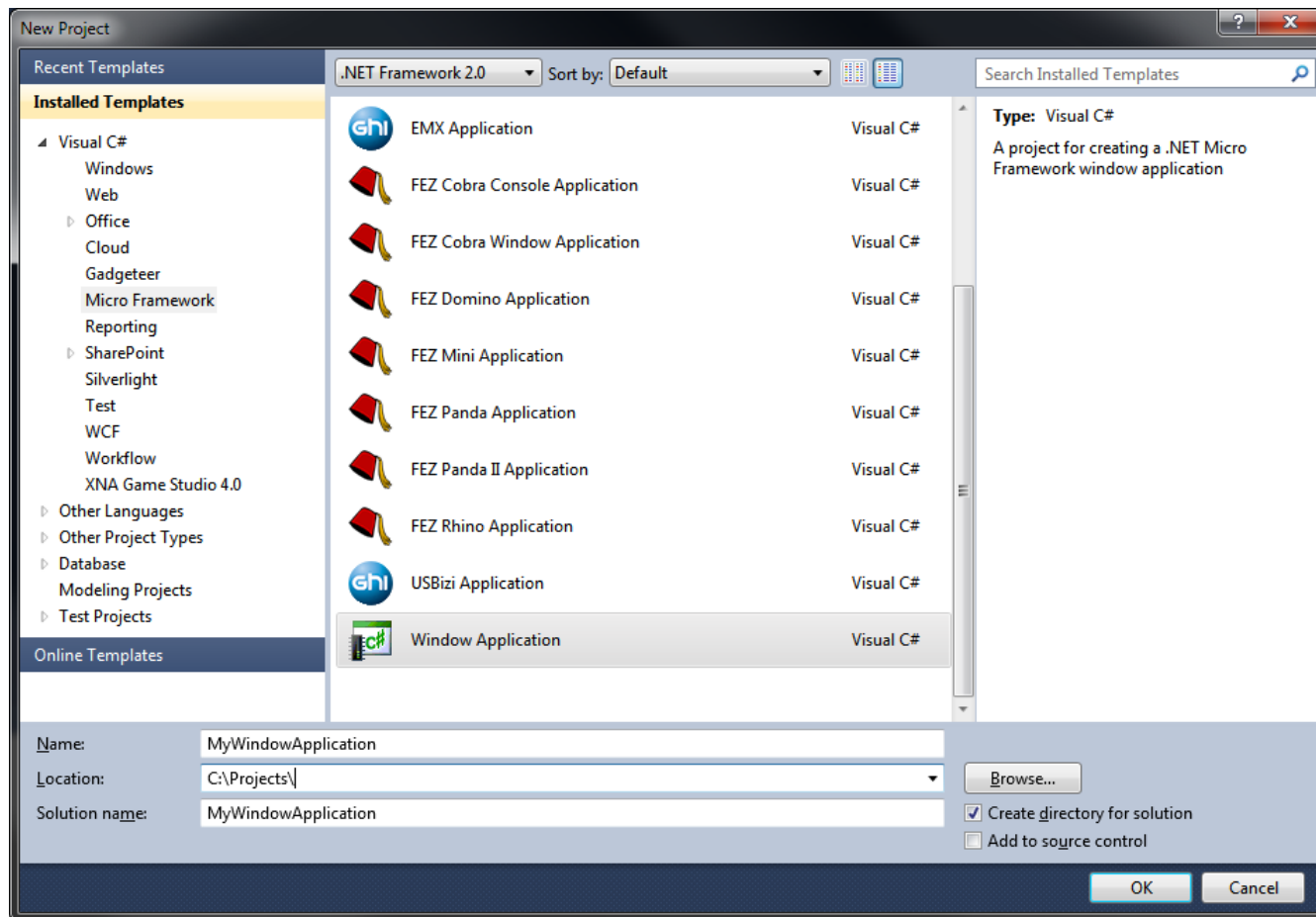
---

Grafika II.

# A laborfoglalkozás célja

- A foglalkozás célja a Window alapú megjelenítés alapvető bemutatása emulátor segítségével:
  - UIElement osztály és leszármazottai
  - Rajzolás Canvas-ra

# MicroFramework projekt létrehozása



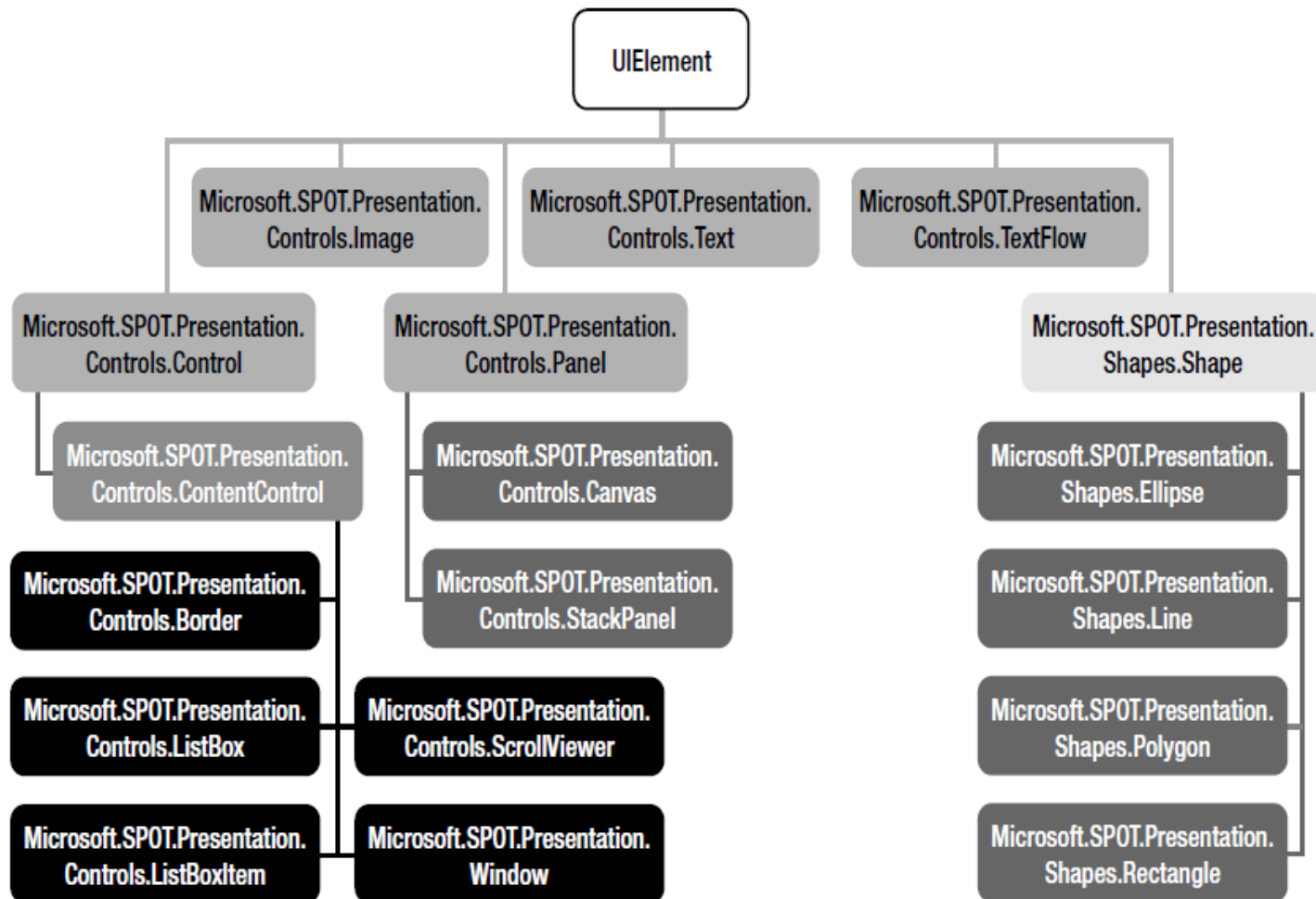
# MicroFramework projekt

- Egy keretprogramot kapunk, mely létrehoz egy ablakot, benne a klasszikus „Hello world” szöveggel, illetve egy egyszerű eseménykezelővel, valamint a szokásos Main() „belépési ponttal”
- Az eszközt ne csatlakoztassuk, csak Futtassuk a keretprogramot a szokásos módon (F5)!
- Az alapbeállítás miatt az emulátor fog elindulni

# Window

- A Window objektum egy grafikus ablakot reprezentál, mely lehetővé teszi különféle „vezérlők” elhelyezését és grafika megjelenítését
- Az ablakba UIElement, vagy azok leszármazottai helyezhetők el, valamint saját UIElementet is készíthetünk

# UIElement - Osztályhierarchia



# Control, ContentControl

```
public class Control: UIElement
{
    protected internal Brush _background;
    protected internal Font _font;
    protected internal Brush _foreground;

    public Control();

    public Brush Background { get; set; }
    public Font Font { get; set; }
    public Brush Foreground { get; set; }

    public override void OnRender(DrawingContext dc);
}

public abstract class ContentControl: Control
{
    protected ContentControl();

    public UIElement Child { get; set; }

    protected override void MeasureOverride(int availableWidth,
        int availableHeight, out int desiredWidth, out int desiredHeight);
}
```

# Canvas, Text

```
public Window CreateWindow()
{
    mainWindow = new Window();
    mainWindow.Height = SystemMetrics.ScreenHeight;
    mainWindow.Width = SystemMetrics.ScreenWidth;

    Text text = new Text();
    text.Font = Resources.GetFont(Resources.FontResources.small);
    text.TextContent = Resources.GetString(Resources.StringResources.String1);
    text.HorizontalAlignment = Microsoft.SPOT.Presentation.HorizontalAlignment.Center;
    text.VerticalAlignment = Microsoft.SPOT.Presentation.VerticalAlignment.Center;
    text.ForeColor = Colors.Red;

    Canvas canvas = new Canvas();
    mainWindow.Child = canvas;
    canvas.Children.Add(text);
    Canvas.SetLeft(text, 100);
    Canvas.SetTop(text, 50);

    mainWindow.AddHandler(Buttons.ButtonUpEvent, new RoutedEventHandler(OnButtonUp), false);
    mainWindow.Visibility = Visibility.Visible;
    Buttons.Focus(mainWindow);

    return mainWindow;
}
```



# Background, Border

```
public Window CreateWindow()
{
    mainWindow = new Window();
    mainWindow.Height = SystemMetrics.ScreenHeight;
    mainWindow.Width = SystemMetrics.ScreenWidth;
    mainWindow.Background = new LinearGradientBrush(Colors.White, Colors.Blue, 0, 0,
        SystemMetrics.ScreenWidth, SystemMetrics.ScreenHeight);

    Canvas canvas = new Canvas();
    mainWindow.Child = canvas;

    Text text = new Text();
    text.Font = Resources.GetFont(Resources.FontResources.small);
    text.TextContent = Resources.GetString(Resources.StringResources.String1);
    text.ForeColor = Colors.Red;
    text.SetMargin(5, 1, 1, 1);

    Border textBorder = new Border();
    textBorder.SetBorderThickness(5, 5, 5, 5);
    textBorder.BorderBrush = new SolidColorBrush(Colors.Yellow);
    Canvas.SetLeft(textBorder, 50);
    Canvas.SetTop(textBorder, 50);

    textBorder.Child = text;
    canvas.Children.Add(textBorder);

    mainWindow.AddHandler(Buttons.ButtonUpEvent, new RoutedEventHandler(OnButtonUp), false);
    mainWindow.Visibility = Visibility.Visible;
    Buttons.Focus(mainWindow);

    return mainWindow;
}
```

# TextFlow

```
public Window CreateWindow()
{
    mainWindow = new Window();
    mainWindow.Height = SystemMetrics.ScreenHeight;
    mainWindow.Width = SystemMetrics.ScreenWidth;
    Font font = Resources.GetFont(Resources.FontResources.small);
    TextFlow textFlow = new TextFlow();
    textFlow.ScrollingStyle = ScrollingStyle.LineByLine;

    for (int i = 0; i < 100; ++i)
    {
        textFlow.TextRuns.Add("Item " + i, font, Colors.Red);
        textFlow.TextRuns.Add(TextRun.EndOfLine);
    }

    mainWindow.Child = textFlow;
    mainWindow.Visibility = Visibility.Visible;

    Buttons.Focus(textFlow);
    return mainWindow;
}
```

# ListBox - Létrehozás

```
public Window CreateWindow()
{
    mainWindow = new Window();
    mainWindow.Height = SystemMetrics.ScreenHeight;
    mainWindow.Width = SystemMetrics.ScreenWidth;
    Font font = Resources.GetFont(Resources.FontResources.small);

    ListBox listBox = new ListBox();
    for (int i = 0; i < 10; ++i)
    {
        string str = "Item " + i.ToString();
        listBox.Items.Add(new Text(font, str));
    }

    mainWindow.Child = listBox;
    mainWindow.Visibility = Visibility.Visible;

    Buttons.Focus(listBox);
    listBox.SelectionChanged += new
        SelectionChangedEventHandler(listBox_SelectionChanged);
    listBox.AddHandler(Buttons.ButtonDownEvent,
        new RoutedEventHandler(listBox_ButtonDown), false);

    return mainWindow;
}
```

# ListBox - Eseménykezelés

```
private void listBox_SelectionChanged(object sender, SelectionChangedEventArgs args)
{
    Debug.Print("Item " + args.SelectedIndex + " was selected.");
}

private void listBox_ButtonDown(object sender, RoutedEventArgs args)
{
    ButtonEventArgs evt = (ButtonEventArgs) args;
    ListBox listBox = (ListBox) sender;
    if (evt.Button == Button.VK_SELECT)
        Debug.Print("Item " + listBox.SelectedIndex + " was pressed.");
}
```

# ListBox - Feladat

- A ListBox nem jeleníti meg a kijelölt elemet, ezért készítsünk olyan ListBoxItem-et, amely biztosítja ezt!
  - A ListBoxItem osztály bővíthető örökléssel
  - Definiáljuk felül a ListBoxItem osztály OnIsSelectedChanged metódusát, mely paraméterben adja át a kijelölt állapotot!
  - A kijelölt állapot legyen kékszínű háttérrel megjelenítve, egyébként használjuk az átlátszó háttérrel (ekkor a Background tulajdonság null)!
  - A ListBoxItem egy ContentControl, tehát elemek helyezhetők el benne, ha a Child property-nek értéket adunk...

# Rajzolás Canvas-ra

```
public class MyCanvas: Canvas
{
    private int halfWidth;
    private int halfHeight;

    public MyCanvas()
        : base()
    {
        halfWidth = SystemMetrics.ScreenWidth / 2;
        halfHeight = SystemMetrics.ScreenHeight / 2;
    }

    public override void OnRender(Microsoft.SPOT.Presentation.Media.DrawingContext dc)
    {
        base.OnRender(dc);
        Brush brush = new SolidColorBrush(Colors.Orange);
        Pen pen = new Pen(Colors.Green, 1);
        dc.Translate(halfWidth, halfHeight);
        dc.DrawEllipse(brush, pen, 0, 0, 50, 50);
    }
}
```

# Feladatok

1. Készítsen alkalmazást, mely egy pattogó labdát jelenít meg a kijelzőn!

VAGY

2. Készítse el az előzőekben bemutatott „Benchmark” alkalmazás Window alapú változatát!

# Irodalomjegyzék

- Jens Kühner: Expert .NET Micro Framework, Apress, April 28, 2008, ISBN-10: 159059973X , ISBN-13: 978-1590599730

