



# Creating User Interfaces

+

# Covered Topics

- UI Frameworks in .NET
- Windows Forms
  - When to use
  - Basic concepts
  - **Practical example**
  - Useful (more advanced) features for your semestral projects
- WPF
  - When to use (instead of WF)
  - Basic concepts
  - **Practical example**
  - Examples and concepts useful for your semestral projects

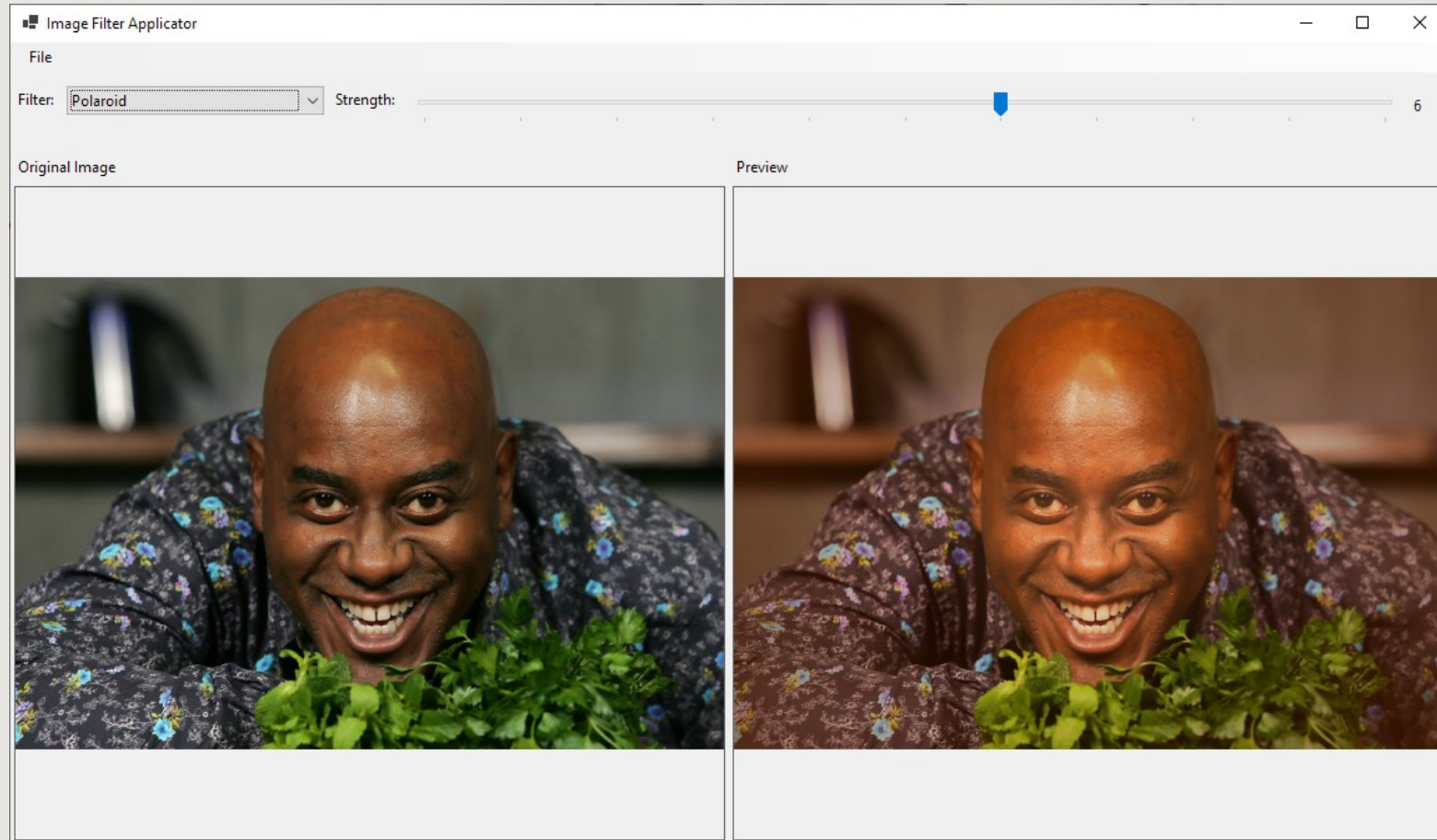


# Available UI Frameworks

+

(A non-exhaustive list)

# Windows Forms



# WPF

SHOW DATE FOR

15.03.2019

401 TWIN DELUXE	402 STANDARD	403 SUITE	404 PLATINUM SUIT
Available \$550	Available \$300	Disabled \$750	Available \$1200
405 SINGLE DELUXE	406 TWIN DELUXE	407 SUITE	408 TWIN DELUXE
Cleaning \$350	Problematic \$550	Available \$600	Cleaning \$400
301 TWIN DELUXE	302 STANDARD	303 SUITE	304 PLATINUM SUIT
Available \$550	Available \$300	Disabled \$750	Available \$1200
305 SINGLE DELUXE	306 TWIN DELUXE	307 SUITE	308 TWIN DELUXE
Cleaning \$350	Problematic \$550	Occupied \$600	Cleaning \$400

SUMMARY

ROOMS	AVAILABLE	CHECK IN/OUT
TOTAL	16	TWING DELUXE 2
OCCUPIED	1	STANDARD 2
AVAILABLE	7	SUITE 1
HAS PROBLEMS	2	PLATINUM SUIT 2
CLEANING	4	SINGLE DELUXE 0
OUT OF ORDER	2	

ROOM NUMBER: 401 NAME: Ocean View

TYPE: Twin Deluxe RATE (\$): 550

DESCRIPTION: View the magestic ocean with luxery and comfort

CAPACITY: 2 FACILITIES:

Available  Cleaning  Has Problems

CUSTOMER: CHECK OUT DATE:

ISSUES:

DISABLE SALE
ALLOCATE

CANCEL
SAVE CHANGES

GO HOME

# Avalonia

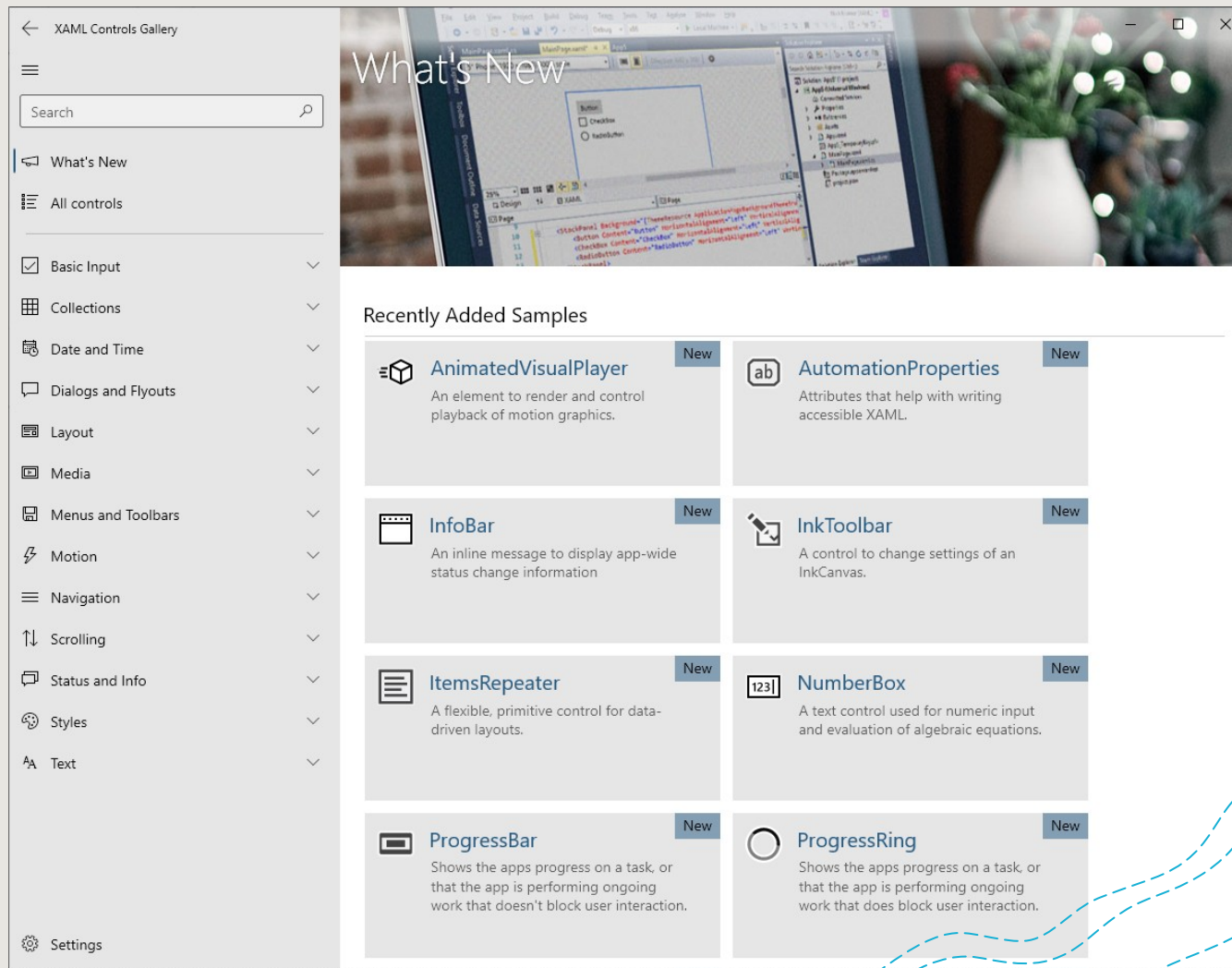
The screenshot displays the 'Overview' page of the Avalonia UI library. It features a dark-themed interface with a search bar at the top left. The main content is organized into several columns, each showcasing a different type of UI control. On the left, there is a sidebar with navigation links for 'Overview', 'Date and Time', and 'Layout'. The 'Overview' section is currently selected.

The controls shown include:

- Buttons:** Enabled button, Disabled button, and Toggle button.
- Date and Time:** Date picker (with day, month, year fields), Time picker (with hour, minute fields), and a Calendar for 'duben 2021'.
- Form Elements:** Textbox (with placeholder text), Numeric up down (with value 0,00 K€ and arrow buttons), and a Slider.
- Selection:** Combo box and List box (showing options like Linux, Mac, Windows).
- States:** Checkbox (with Unchecked, Checked, Third state, and Disabled options) and Radio button (with Unchecked, Checked, and Disabled options).
- Other:** Toggle switch (with Off, On, Disabled Off, and Disabled On states) and Progress bar.

po	út	st	čt	pá	so	ne
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

# UWP



XAML Controls Gallery

Search

What's New

All controls

- Basic Input
- Collections
- Date and Time
- Dialogs and Flyouts
- Layout
- Media
- Menus and Toolbars
- Motion
- Navigation
- Scrolling
- Status and Info
- Styles
- Text

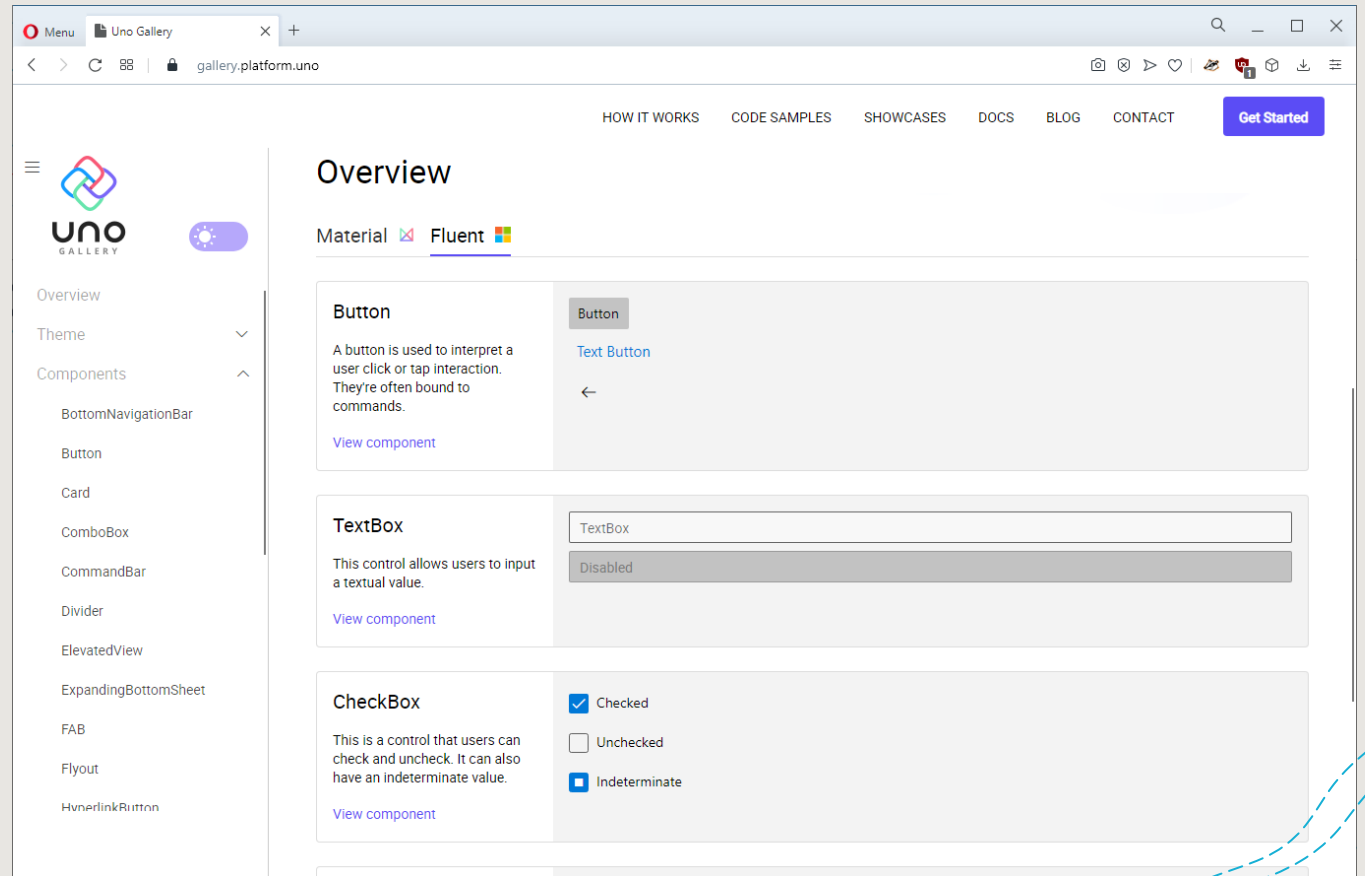
Settings

## What's New

### Recently Added Samples

- AnimatedVisualPlayer** (New)  
An element to render and control playback of motion graphics.
- AutomationProperties** (New)  
Attributes that help with writing accessible XAML.
- InfoBar** (New)  
An inline message to display app-wide status change information.
- InkToolBar** (New)  
A control to change settings of an InkCanvas.
- ItemsRepeater** (New)  
A flexible, primitive control for data-driven layouts.
- NumberBox** (New)  
A text control used for numeric input and evaluation of algebraic equations.
- ProgressBar** (New)  
Shows the apps progress on a task, or that the app is performing ongoing work that doesn't block user interaction.
- ProgressRing** (New)  
Shows the apps progress on a task, or that the app is performing ongoing work that does block user interaction.

# Uno Platform (Xamarin.Native)



The screenshot shows a web browser displaying the Uno Platform website. The browser's address bar shows the URL `gallery.platform.uno`. The website's navigation menu includes links for `HOW IT WORKS`, `CODE SAMPLES`, `SHOWCASES`, `DOCS`, `BLOG`, and `CONTACT`, along with a `Get Started` button. The main content area is titled `Overview` and features a theme selector with `Material` and `Fluent` options. The `Fluent` theme is currently selected. Below the theme selector, there are three sections, each with a title, a description, and a `View component` link. The `Button` section includes a description: "A button is used to interpret a user click or tap interaction. They're often bound to commands." and visual examples of a standard button, a text button, and a back arrow. The `TextBox` section includes a description: "This control allows users to input a textual value." and visual examples of a standard text box and a disabled text box. The `CheckBox` section includes a description: "This is a control that users can check and uncheck. It can also have an indeterminate value." and visual examples of checked, unchecked, and indeterminate states.

Menu Uno Gallery

gallery.platform.uno

HOW IT WORKS CODE SAMPLES SHOWCASES DOCS BLOG CONTACT [Get Started](#)

Overview

Theme

Components

- BottomNavigationBar
- Button
- Card
- ComboBox
- CommandBar
- Divider
- ElevatedView
- ExpandingBottomSheet
- FAB
- Flyout
- HyperlinkButton

## Overview

Material [Fluent](#)

### Button

A button is used to interpret a user click or tap interaction. They're often bound to commands.

[View component](#)

Button

Text Button

←

### TextBox

This control allows users to input a textual value.

[View component](#)

TextBox

Disabled

### CheckBox

This is a control that users can check and uncheck. It can also have an indeterminate value.

[View component](#)

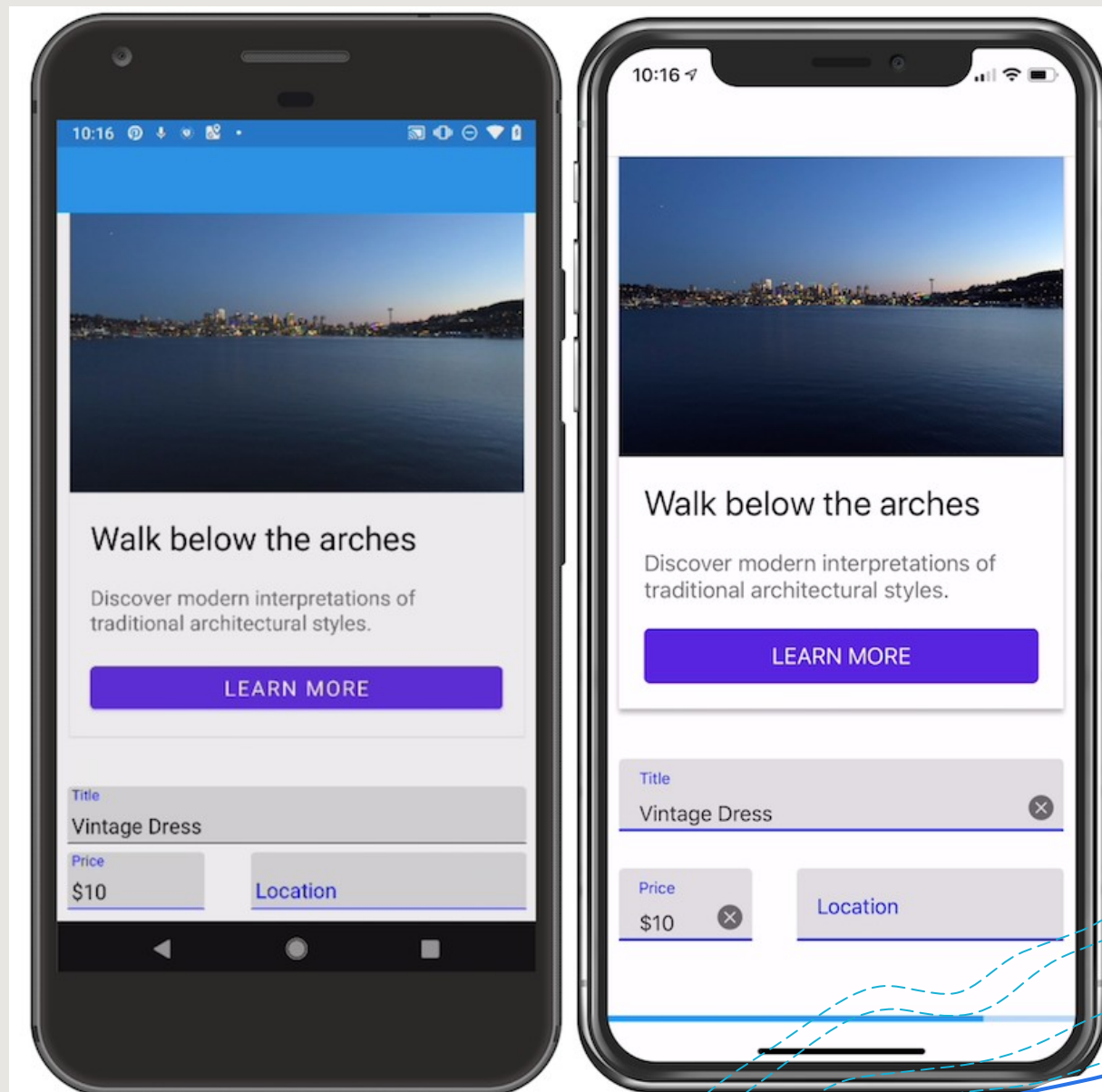
Checked

Unchecked

Indeterminate



# MAUI (Xamarin.Forms)



# Game Engines – Unity, Godot



The background features a light gray gradient with several wavy, dashed blue lines that create a sense of movement and depth. There are also two white circles: one in the top-left corner and another in the bottom-right corner.

# Windows Forms

+

(The *supposedly simple* drag&drop one)

# Required Visual Studio Packages

Upravuje se – Visual Studio Community 2019 – 16.8.6

Sady funkcí    Jednotlivé komponenty    Jazykové sady    Umístění instalací

Web a cloud (4)

- Vývoj pro ASP.NET a web**  
Umožňuje sestavovat webové aplikace pomocí technologií ASP.NET Core, ASP.NET, HTML/JavaScript a Containers v...
- Azure – vývoj**  
Sady Azure SDK, nástroje a projekty pro vývoj cloudových aplikací a vytváření prostředků pomocí .NET Core a .NET...
- Vývoj v jazyce Python**  
Úpravy, ladění, interaktivní vývoj a správa zdrojového kódu pro Python
- Vývoj aplikací Node.js**  
Můžete sestavovat škálovatelné síťové aplikace využívající Node.js, což je asynchronní javascriptový runtime založe...

Desktopové a mobilní platformy (5)

- Vývoj desktopových aplikací pomocí .NET**  
Umožňuje sestavovat aplikace WPF, Windows Forms a konzolové aplikace pomocí jazyků C#, Visual Basic a F# s...  
**.NET Framework**
- Vývoj desktopových aplikací pomocí C++**  
Umožní vám sestavovat moderní aplikace v C++ pro Windows pomocí nástrojů, které zvolíte, včetně těchto...
- Vývoj pro Univerzální platformu Windows**  
Umožňuje vytvářet aplikace pro Univerzální platformu Windows pomocí C#, VB nebo případně C++.
- Vývoj mobilních aplikací pomocí .NET**  
Umožňuje sestavovat multiplatformní aplikace pro iOS, Android nebo Windows pomocí Xamarinu.
- Vývoj mobilních aplikací pomocí C++**  
Umožňuje sestavovat multiplatformní aplikace pro iOS, Android nebo Windows pomocí C++.

Herní platformy (2)

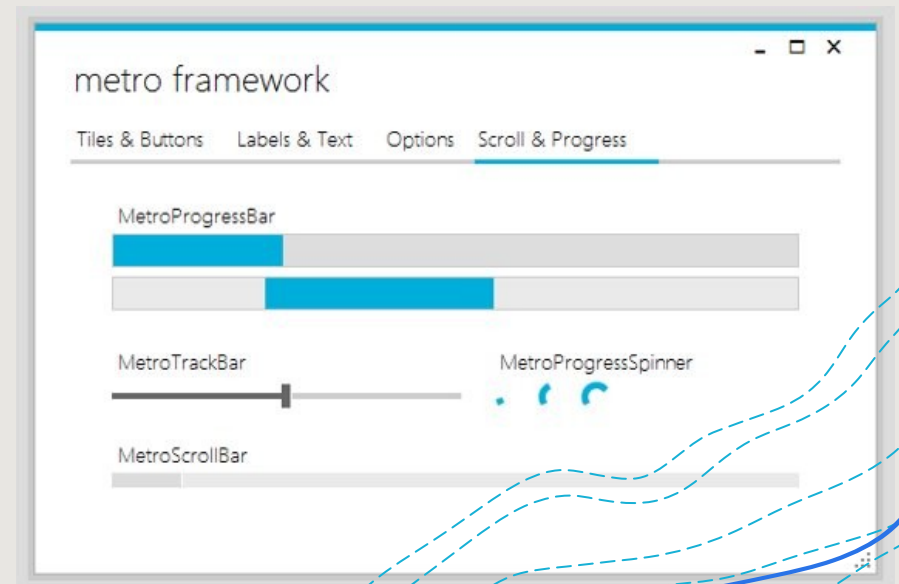
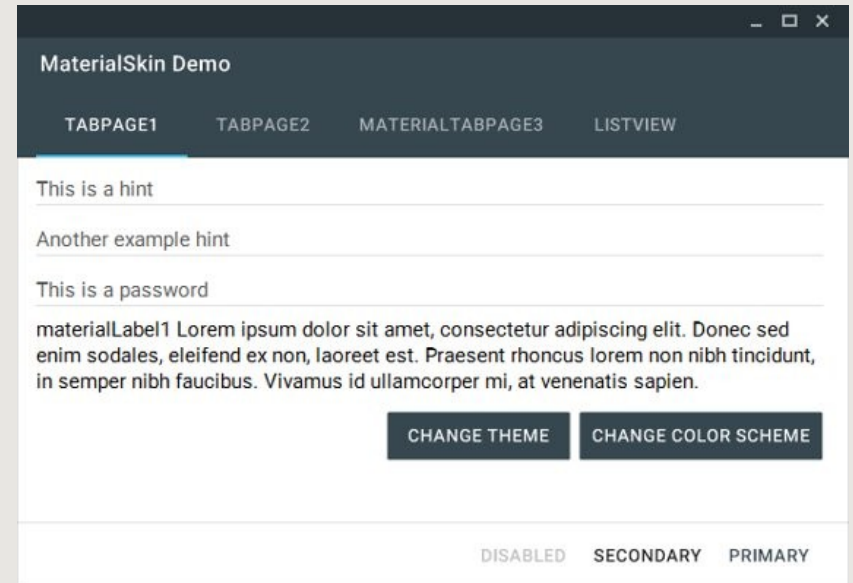
- Vývoj her pomocí Unity**  
Umožňuje vytvářet 2D i 3D hry v Unity, což je výkonné multiplatformní vývojové prostředí.
- Vývoj her pomocí C++**  
Sestavujte profesionální hry s využitím všech možností C++ na platformách DirectX, Unreal nebo Cocos2d.

Další sady nástrojů (6)

- Ukládání a zpracování dat**  
Umožňuje připojit, vyvíjet a testovat datová řešení pomocí SQL Serveru, služby Azure Data Lake nebo platformy Ha...
- Aplikace pro datovou vědu a analýzu**  
Jazyky a nástroje pro vytváření aplikací pro datové vědy, včetně podpory Pythonu a F#
- Vývoj rozšíření sady Visual Studio**  
Umožňuje vytvářet doplňky a rozšíření pro Visual Studio, včetně nových příkazů, analyzátorů kódu a panelů nástro...
- Vývoj pro Office/SharePoint**  
Vytvářejte doplňky pro Office a SharePoint, řešení pro SharePoint a doplňky pro VSTO pomocí C#, VB a JavaScri...
- Vývoj pro Linux pomocí C++**  
Umožňuje vytvářet a ladit aplikace pro linuxové prostředí.
- Vývoj multiplatformních aplikací pomocí rozhraní .NET...**  
Umožňuje sestavovat multiplatformní aplikace pomocí .NET Core, ASP.NET Core, HTML/JavaScript a kon...  
**.NET Core 3.0+**

# Windows Forms

- + Released in 2002
- + Mostly used in legacy software:
  - + Windows 95+ (.NET 2.0)
  - + Windows 2000+ (.NET 3.5)
  - + Windows XP+ (.NET 4.0)
- + Or in prototypes:
  - + Windows 7+ (.NET Core 3.0)
- + Event-driven
  - + Other paradigms can be used, but not natively
- + Dated API
- + **Really easy to use**
- + **Really hard to customize and maintain**



# Basic Windows Forms Concepts

## Three types of UI entities:

- Form – The window itself
- Control – The visible (groups of) controls in a window
- Component – The invisible "controls" in a window

## Almost all actions by the user or the framework emit events

- We can react to them or even emit our own
- Make sure to not make a cycle in the process

# Good To Know

- Writing to UI Components outside of the UI loop throws an exception – use `Invoke()`
- Windows Forms does not (most of the time) provide generics – LINQ `.OfType<T>()` solves this
- [.NET Core] Use the preview version of Visual Studio to get more mileage out of the designer
- You can use async event handlers – but beware of exceptions!



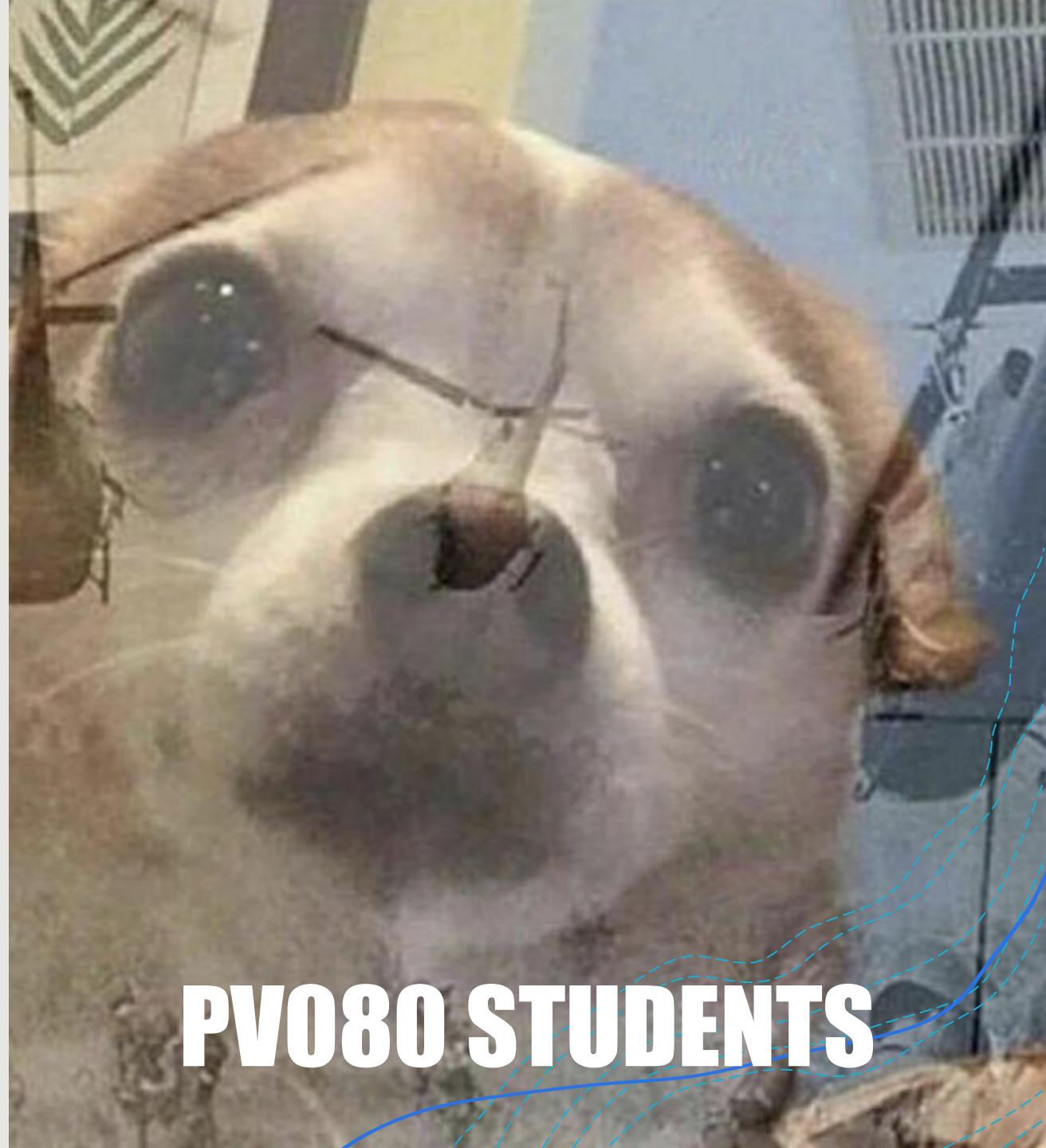
```
IList<DataGridViewColumn> DataGridView.SelectedColumns { get; }  
Gets the collection of columns selected by the user.
```

```
DataGridViewSelectedColumnCollection DataGridView.SelectedColumns { get; }  
Gets the collection of columns selected by the user.
```

# Practical Example: **MD5 Hasher**

Demonstrating:

- Basic controls
- Multiple windows
- IO
- Dialog windows
- Async code



**PV080 STUDENTS**





# WPF

It's newer and thereby more in tune with current standards (may be newer, but 2006 is not that new)

# About

- + Free and open source graphical subsystem.
- + Previously known as "Avalon"
- + Uses DirectX and attempts to provide a consistent programming model for building applications
- + Separates the user interface from business logic, and resembles similar XML-oriented object models
- + Supports a broad set of application development features, including an application model, resources, controls, graphics, layout, data binding, documents, and security.

# Basic concept knowledge

- + Markup and code behind
- + Layout, Data binding
- + Events/Commands
- + Resources (Static, Dynamic) – styles, triggers
- + Controls
- + Converters
- + PropertyChanged

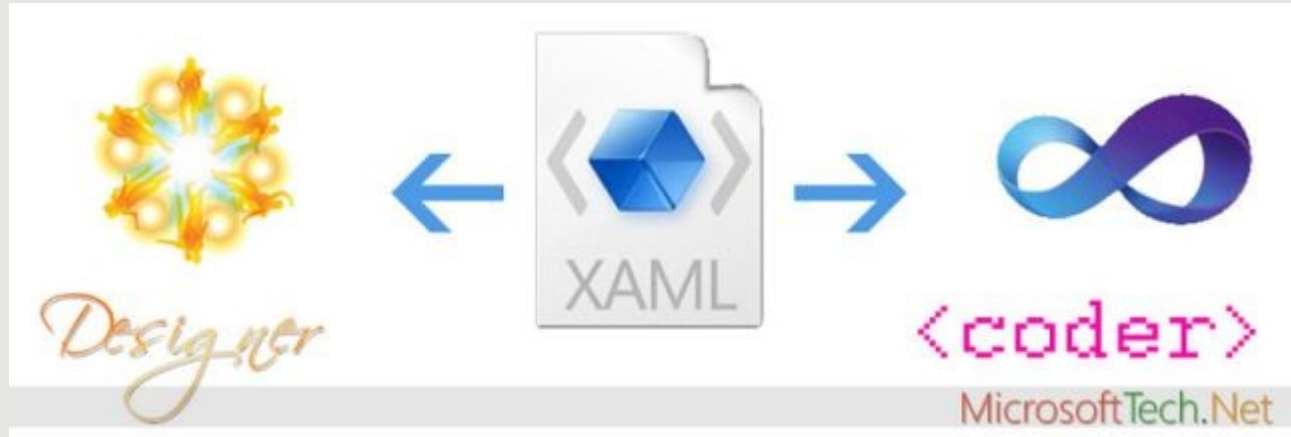
# XAML

eXtensible Application  
Markup Language

Much like with HTML,  
you are able to easily  
write and edit your GUI

<Button/>

<Button> </Button>



# XAML

[Download & run this example](#)

```
Button btn = new Button();
btn.FontWeight = FontWeights.Bold;

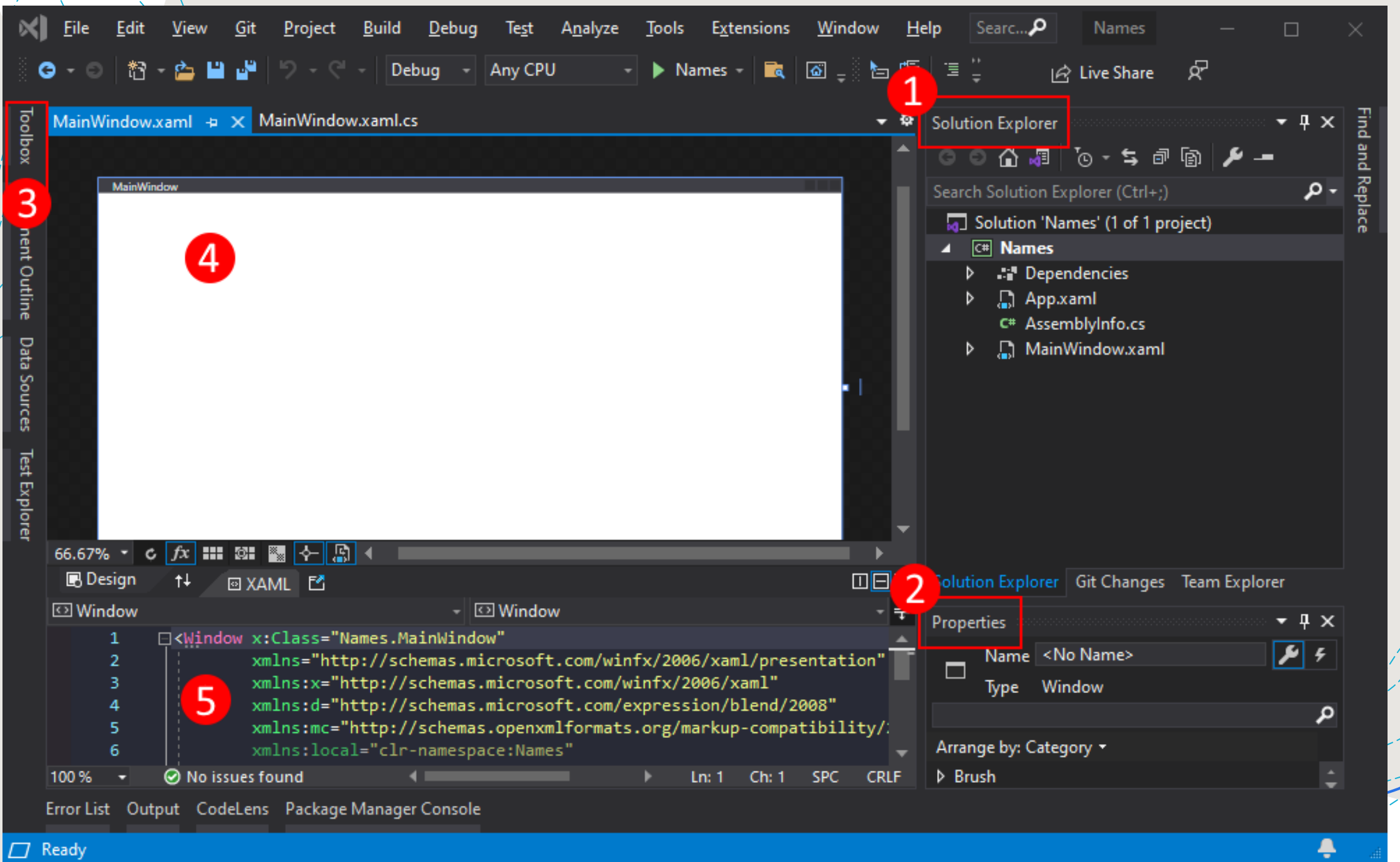
WrapPanel pnl = new WrapPanel();

TextBlock txt = new TextBlock();
txt.Text = "Multi";
txt.Foreground = Brushes.Blue;
pnl.Children.Add(txt);

txt = new TextBlock();
txt.Text = "Color";
txt.Foreground = Brushes.Red;
pnl.Children.Add(txt);

txt = new TextBlock();
txt.Text = "Button";
pnl.Children.Add(txt);

btn.Content = pnl;
pnlMain.Children.Add(btn);
```





# Basic example

- + Windows
- + Controls
- + UserControls
- + IO
- + Many more

# Advanced examples

