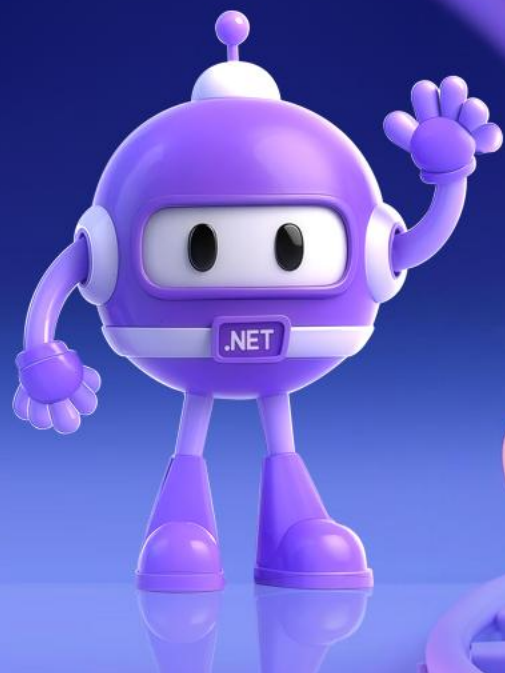


.NET Conf China 2025

改变世界 改变自己

2025 年 11 月 30 日 | 中国 上海



.NET Conf China 2025

改变世界 改变自己

使用 Avalonia 将 .NET MAUI 引入 Linux 和 WebAssembly 平台

Bringing .NET MAUI to Linux and WebAssembly
Platforms with Avalonia

Tim Miller
Avalonia 核心开发者
30/11/2025



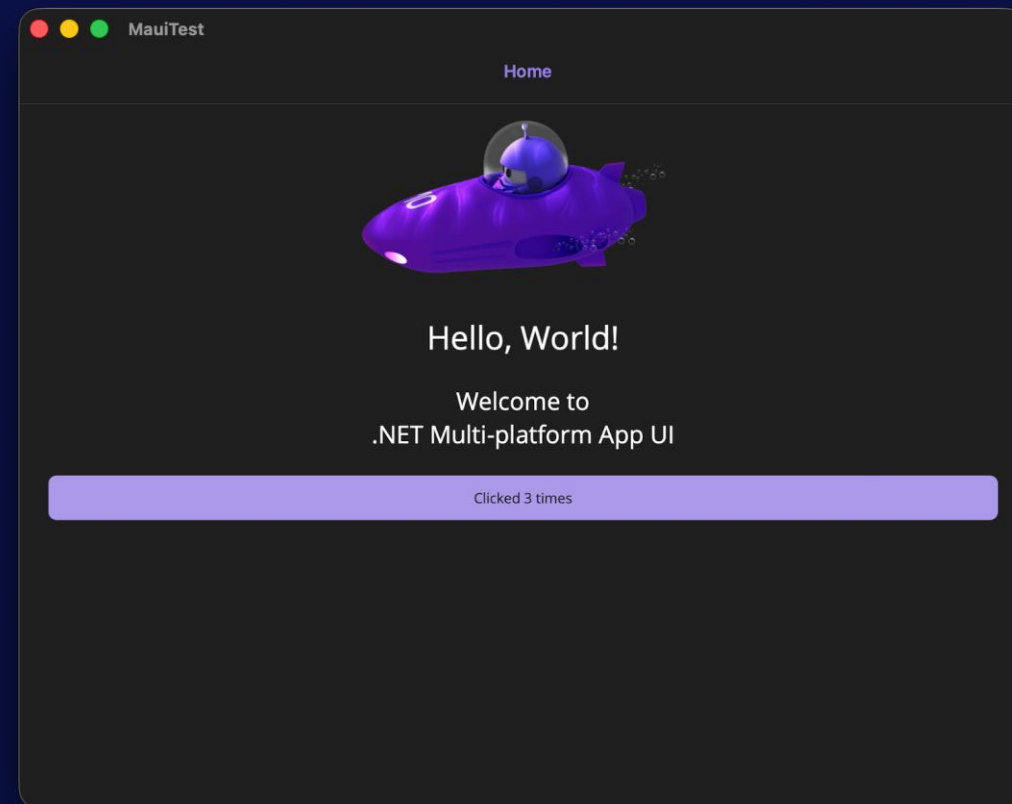


Outline

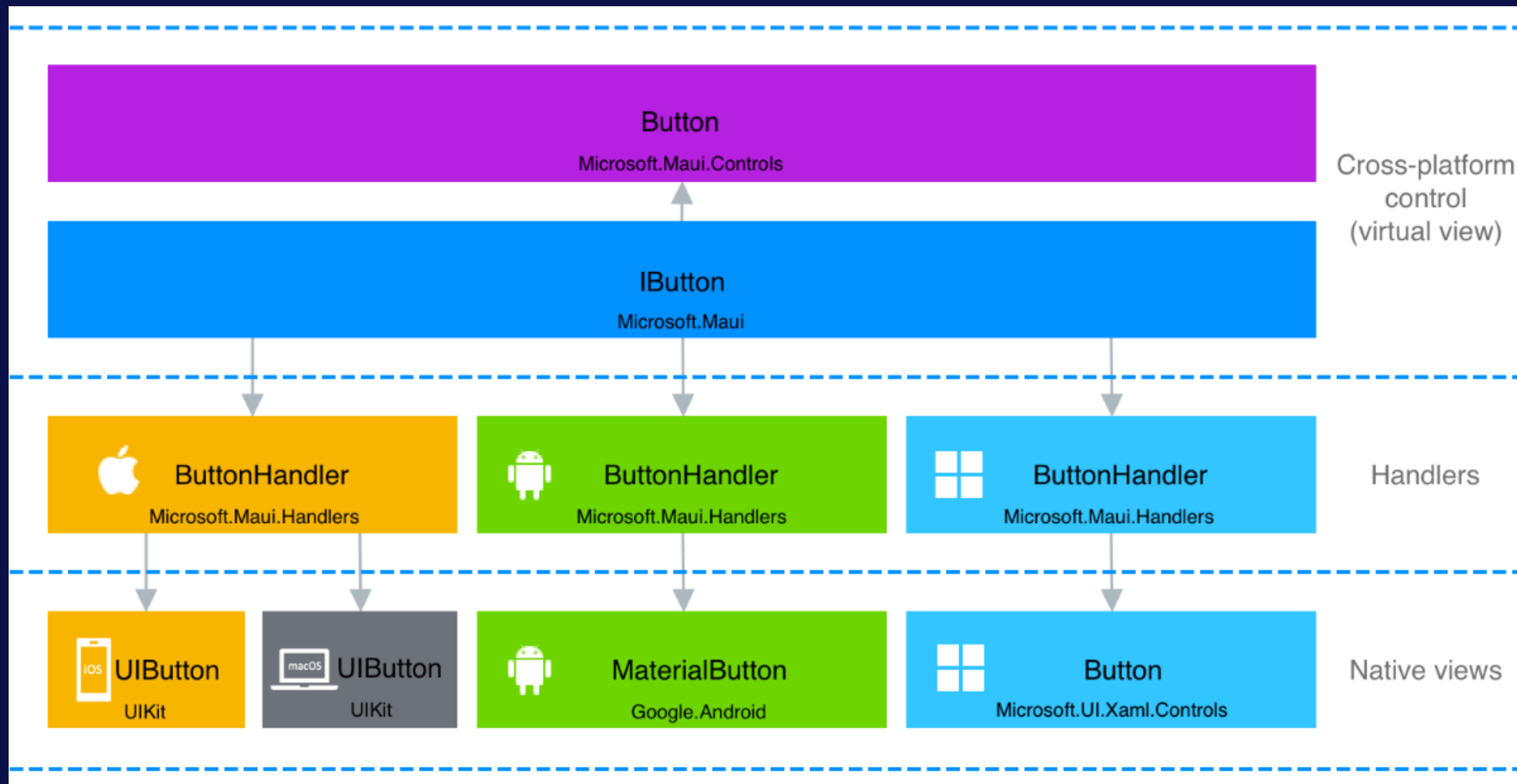
- Overview of .NET MAUI
 - Handlers/Layouts
- What Avalonia Brings...
 - Demo
- Current Progress/Issues
- When does it come out

.NET MAUI

- .NET Multi-platform App UI
- Cross-platform framework for creating native mobile and desktop apps with C# and XAML.
- Evolution of Xamarin.Forms
- iOS, macOS (Mac Catalyst), Android, WinUI 3



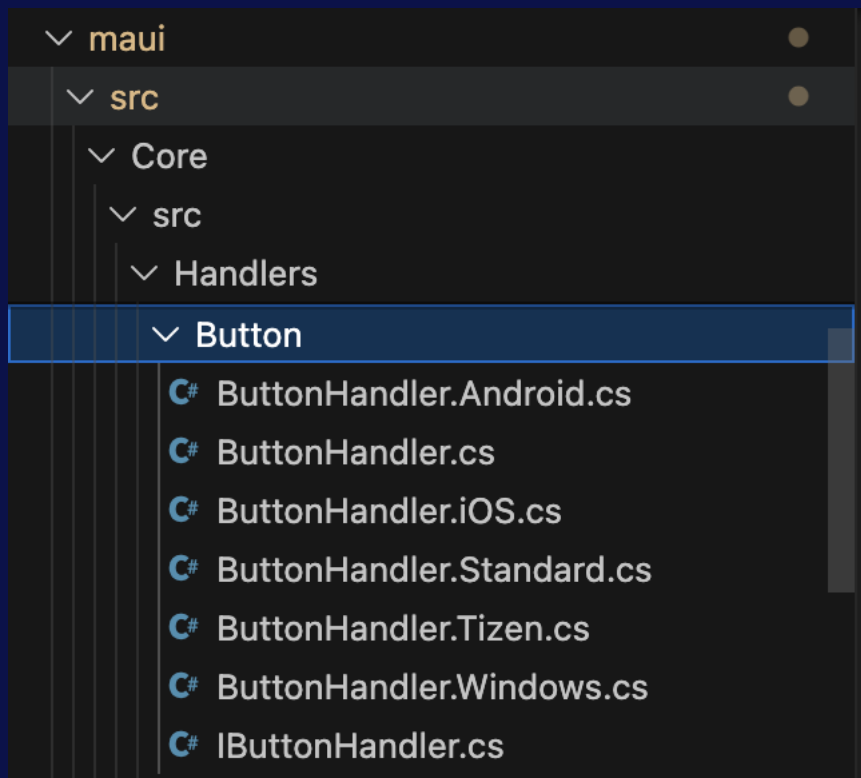
.NET MAUI - Handlers



.NET MAUI - Handlers

.NET Conf China 2025

改变世界 改变自己

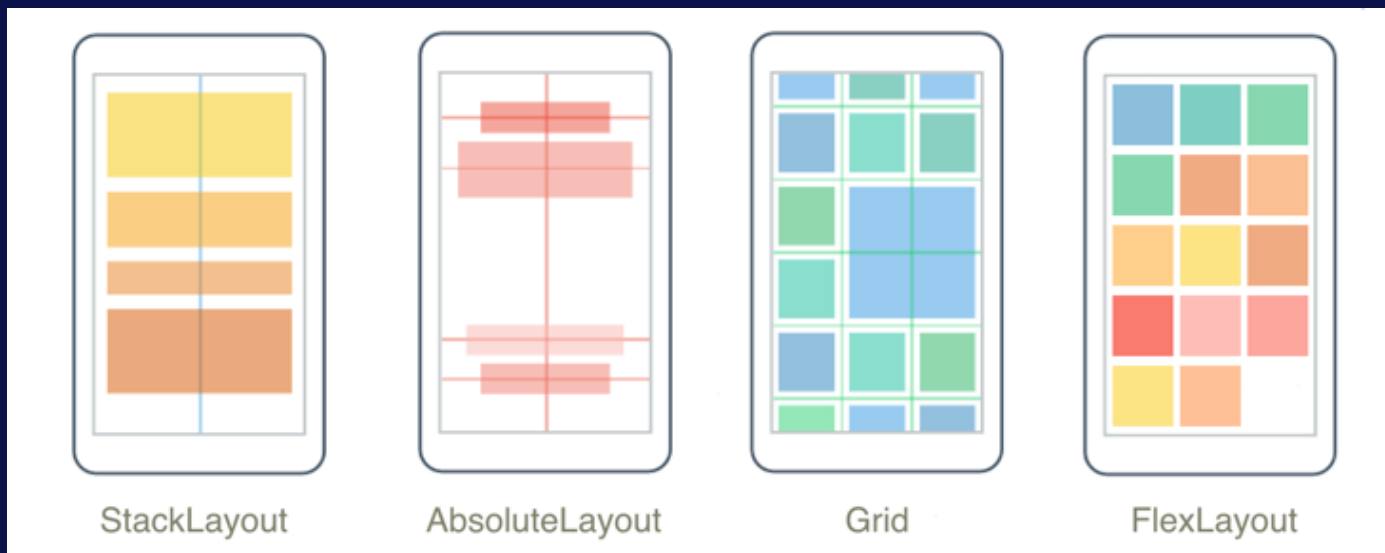


```
#if __IOS__ || MACCATALYST
using PlatformView = UIKit.UIButton;
#elif MONOANDROID
using PlatformView = Google.Android.Material.Button.MaterialButton;
#elif WINDOWS
using PlatformView = Microsoft.UI.Xaml.Controls.Button;
#elif TIZEN
using PlatformView = Tizen.UIExtensions.NUI.Button;
#elif (NETSTANDARD || !PLATFORM) || (NET6_0_OR_GREATER && !IOS && !ANDROID && !TIZEN)
using PlatformView = System.Object;
#endif
```

```
namespace Microsoft.Maui.Handlers
{
    0 個の参照
    public partial interface IButtonHandler : IViewHandler
    {
        0 個の参照
        new IButton VirtualView { get; }
        0 個の参照
        new PlatformView PlatformView { get; }
        0 個の参照
        ImageSourcePartLoader ImageSourceLoader { get; }
    }
}
```

.NET MAUI - Layouts

- ILayoutHandler
- Based on platform primitives
 - iOS/Catalyst – UIView
 - Android – View
 - WinUI – FrameworkElement
- Placed by MAUI directly onto platform view.
 - Ex. Maui StackPanel does not use WinUI's StackPanel





Avalonia + .NET MAUI

- Avalonia-based Handlers and Controls
 - Replacements for standard .NET MAUI Native Controls
 - Your existing XAML / C# will “just work!”*
 - Ways for you to create your own handlers and controls!
- Supported Operating Systems
 - Linux
 - WebAssembly
 - Should work wherever Avalonia can run*
 - Windows
 - macOS
 - Etc...



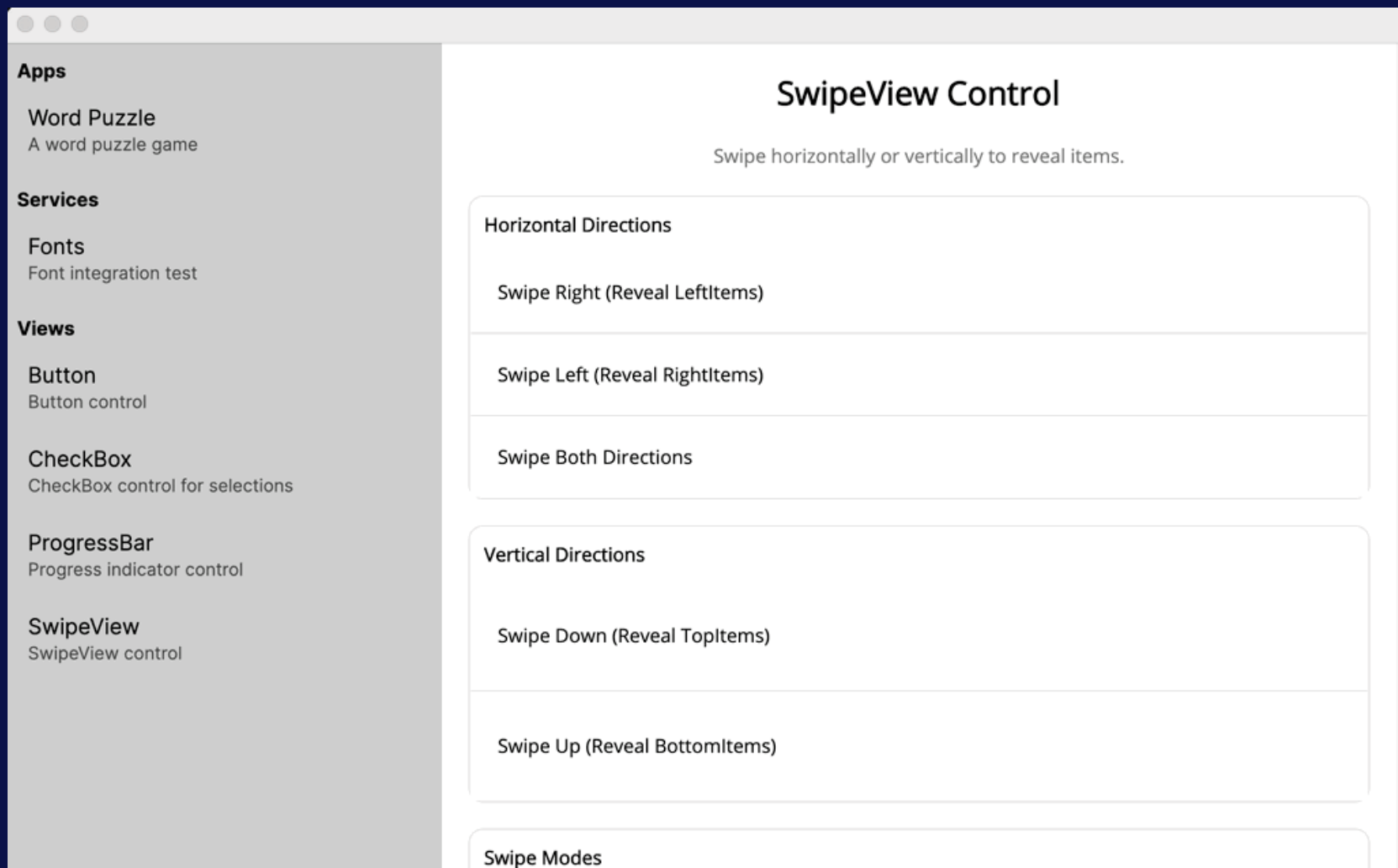
Avalonia + .NET MAUI

- Why?
- For .NET MAUI
 - Bring support to new platforms people want
 - Linux is frequently requested, hard to deliver
 - GTK? QT?
 - Binding and maintaining every control to a new platform
- For Avalonia
 - Bring new controls to the platform
 - .NET MAUI has rich mobile-based controls Avalonia lacks
 - Ex. NavigationPage, SwipeView, etc.
 - New users to Avalonia
 - Create Avalonia-based controls that can run in Avalonia and .NET MAUI

SwipeView

.NET Conf China 2025

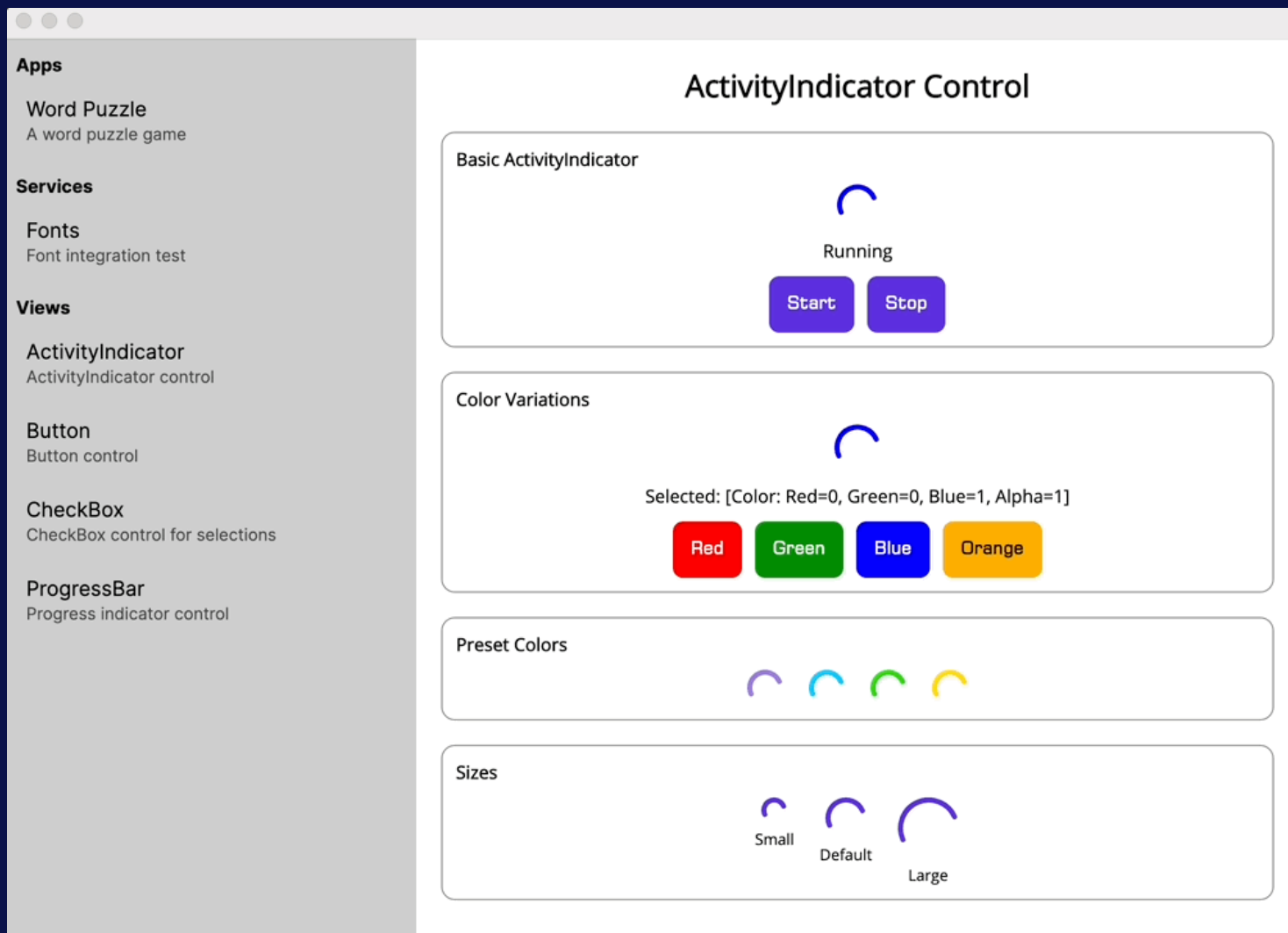
改变世界 改变自己



ActivityIndicator

.NET Conf China 2025

改变世界 改变自己

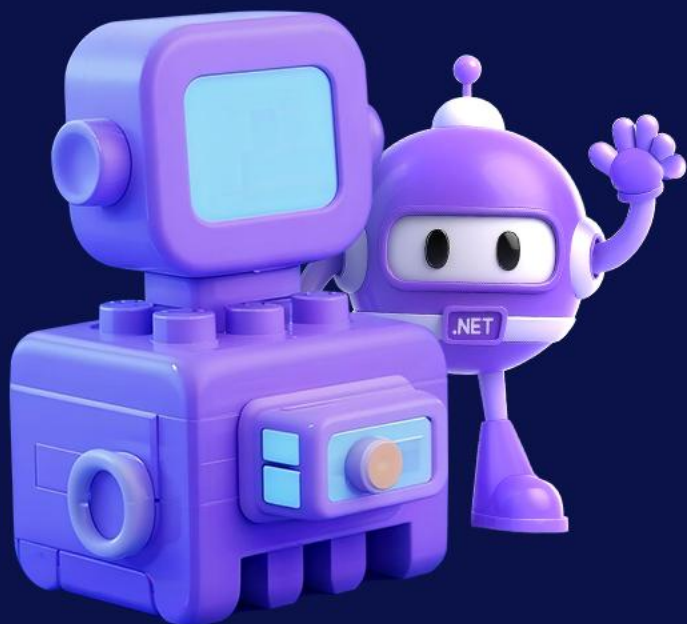


.NET Conf China 2025

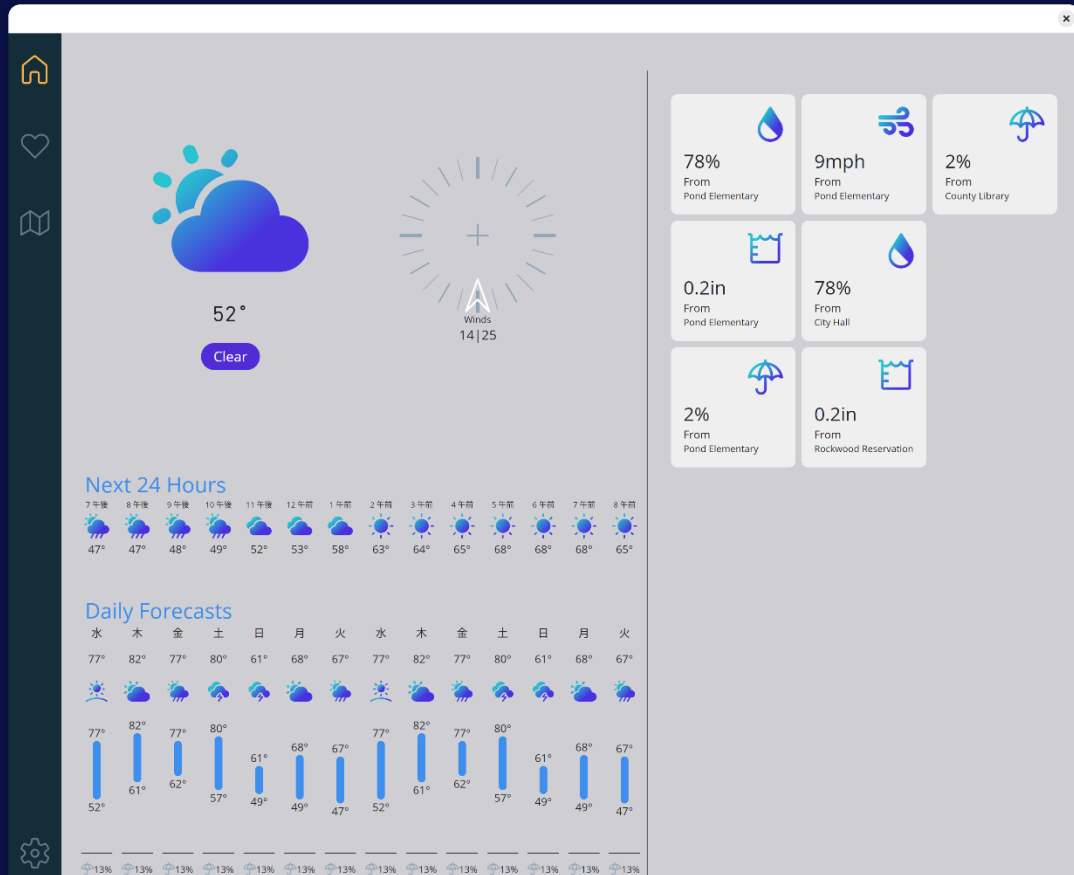
改变世界 改变自己



Demo

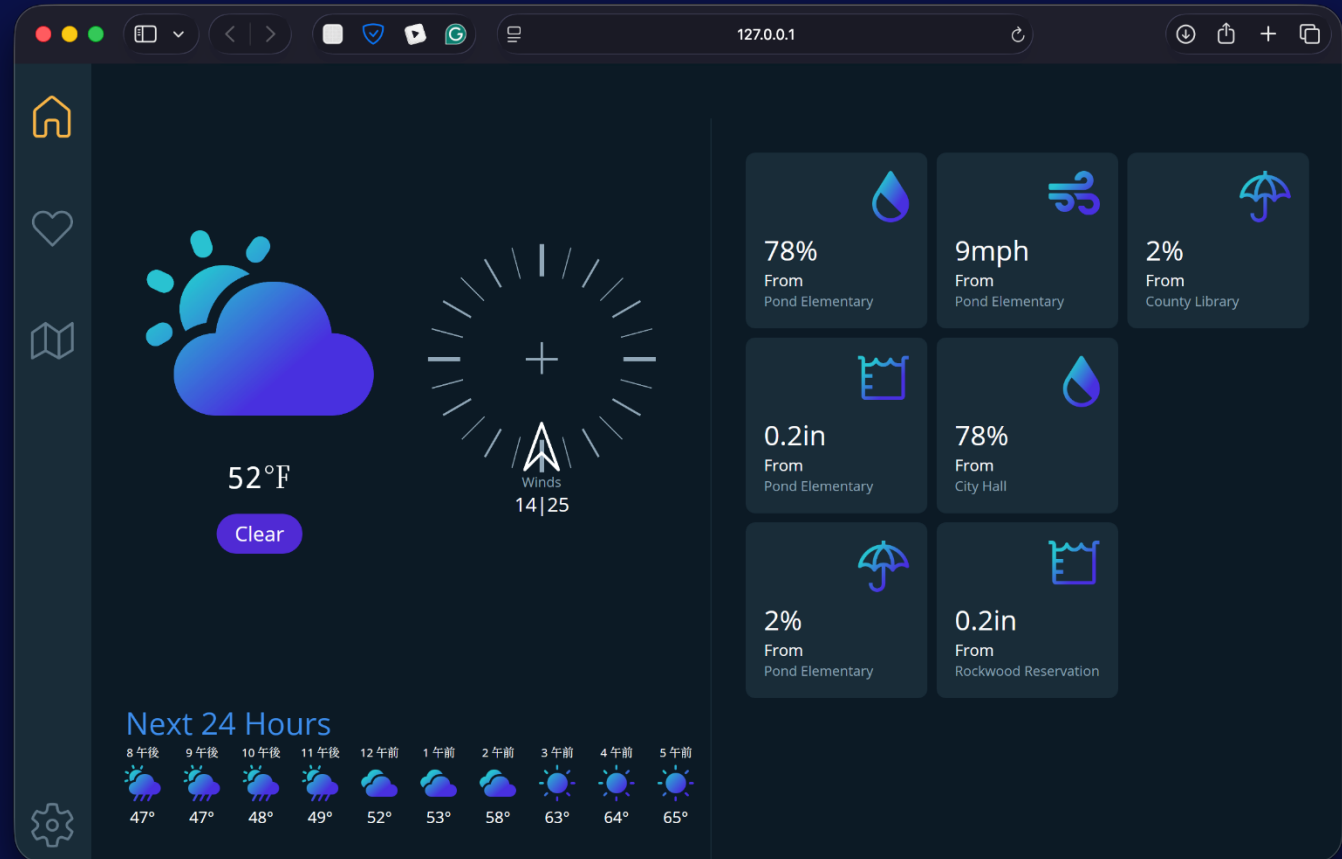


Weather 21' - .NET MAUI



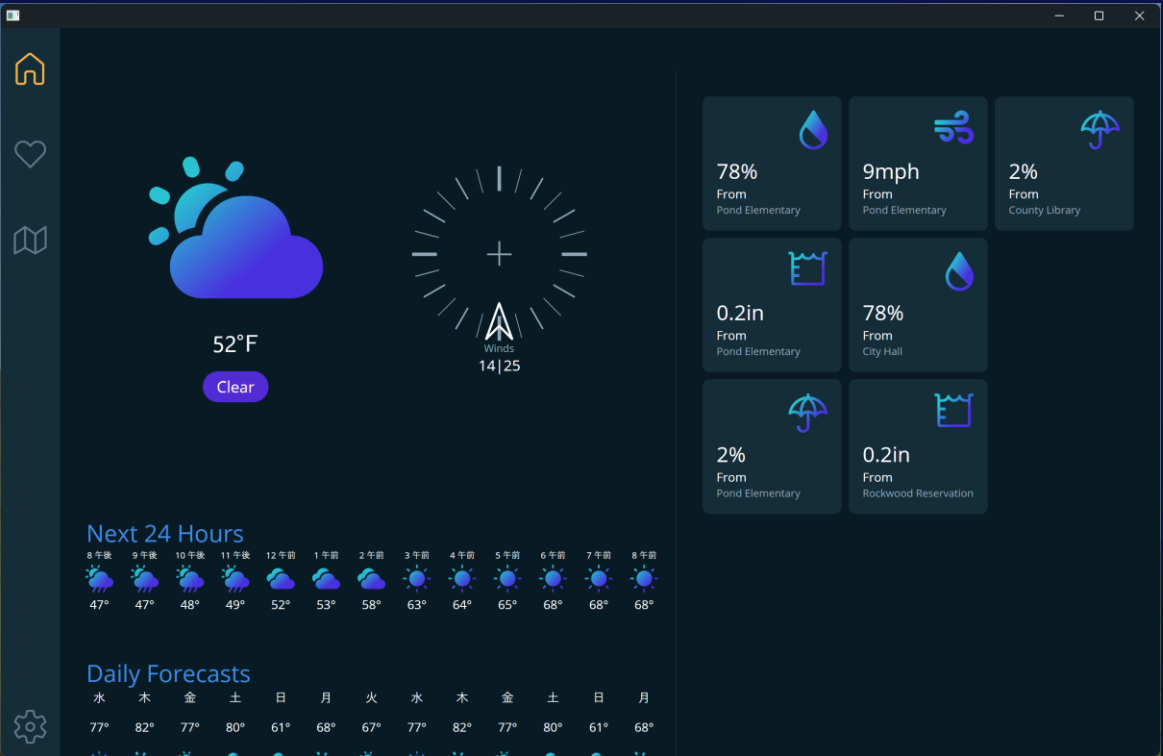
Linux

WebAssembly

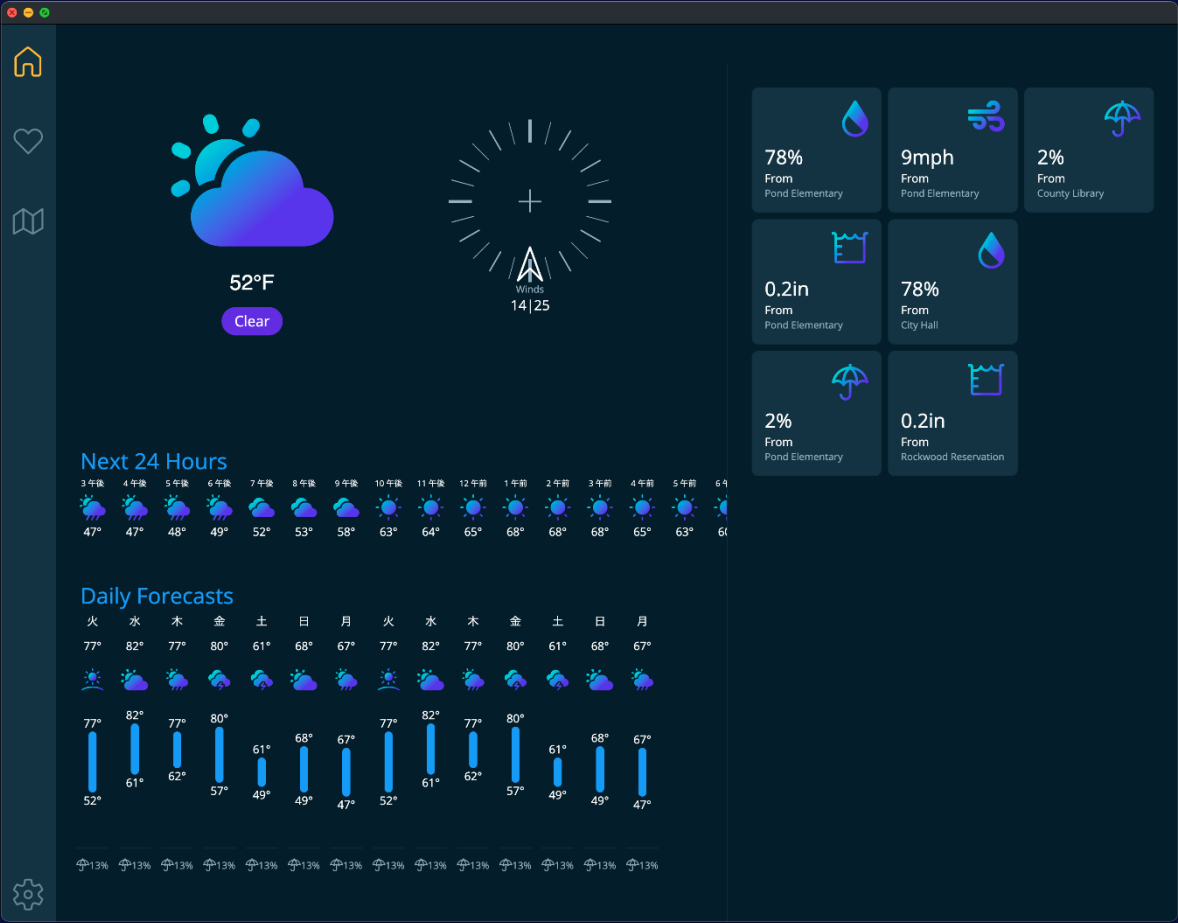


Weather 21' - .NET MAUI

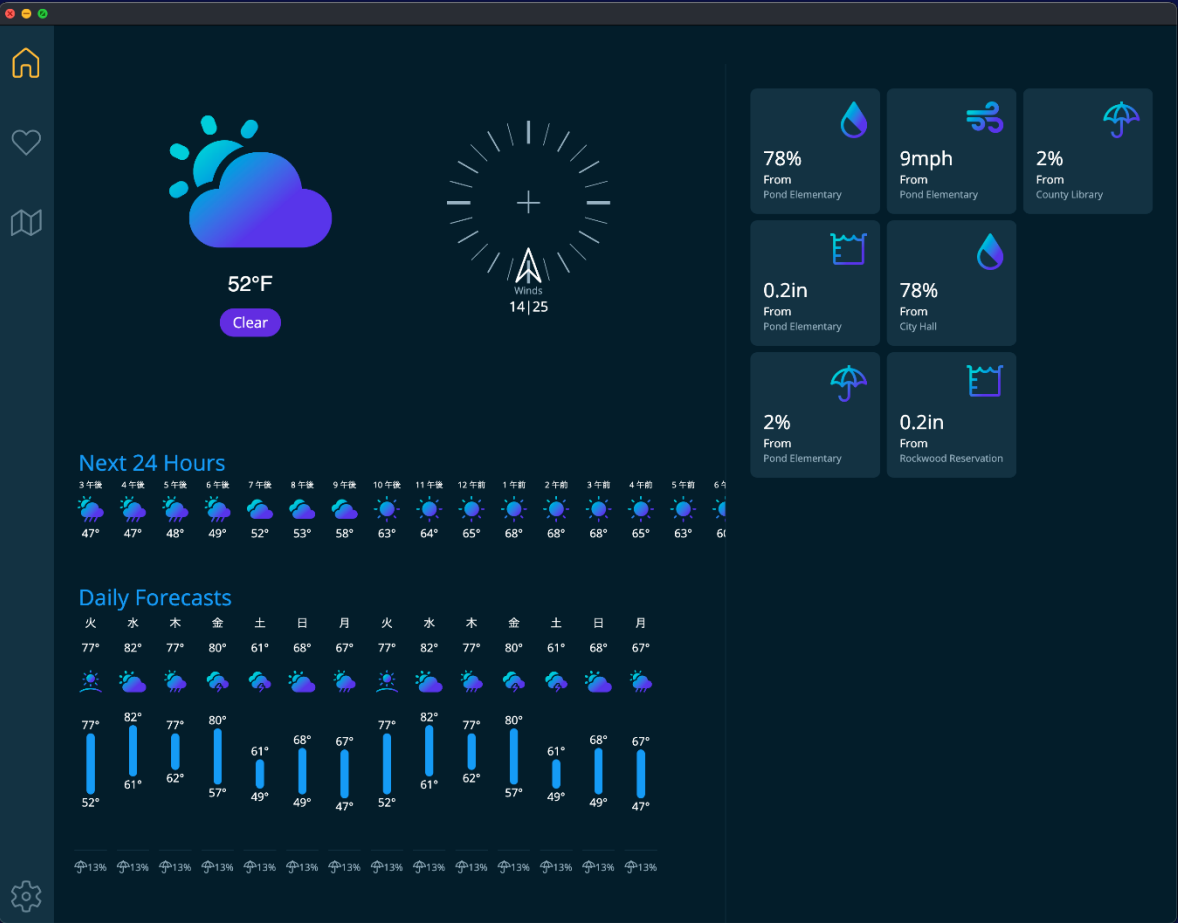
macOS



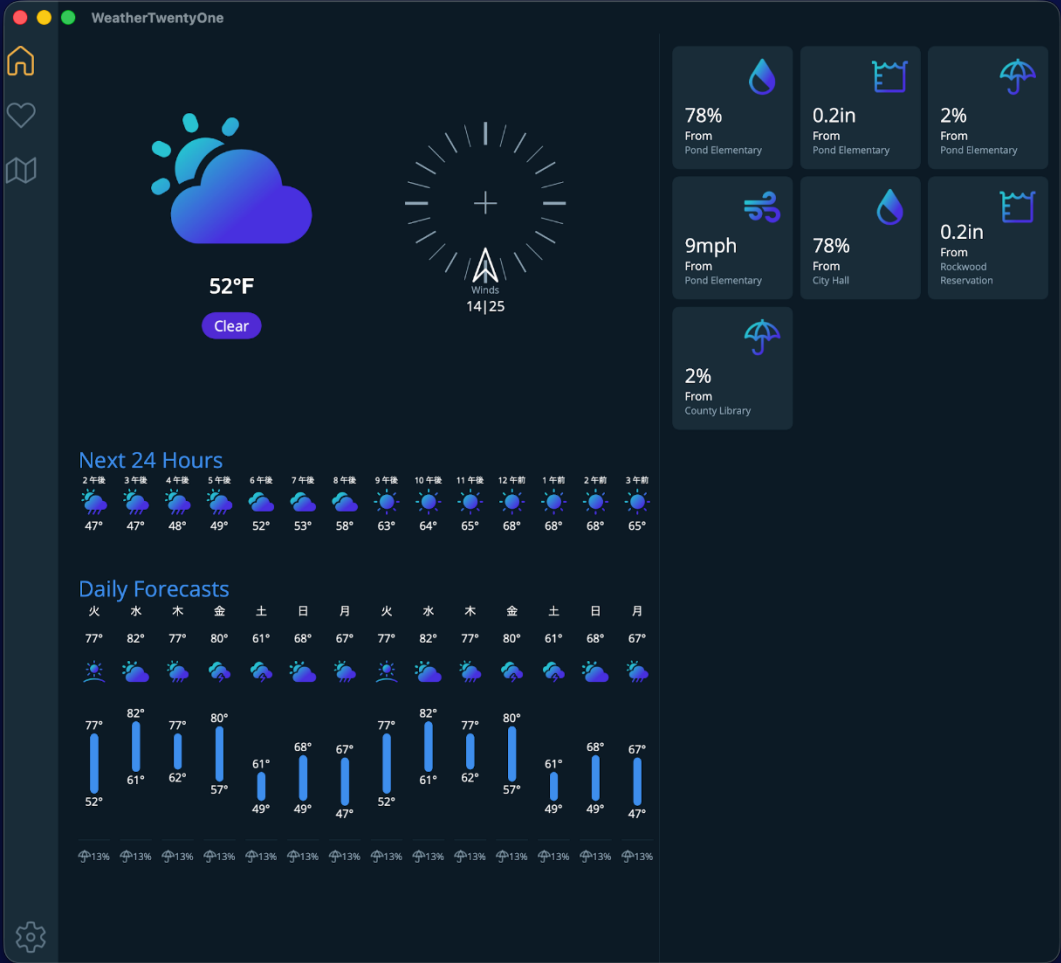
Windows



Weather 21' - .NET MAUI Mac



Avalonia



Mac Catalyst

Current Issues

- .NET MAUI Internal Infrastructure...
 - Ex. App Lifecycle Support is internal.
 - Some common interfaces need to be opened up.
- .NET MAUI Target Frameworks...
 - Native UI Controls bound to handlers (ex. net10-ios expects UIButton, and nothing else...)
 - Impossible to target existing native .NET MAUI Platforms like iOS/Android without hacks.
 - Hence, Linux, WASM, etc. (macOS, Windows work because generic TFMs)

```
using System;
#if __IOS__ || MACCATALYST
using PlatformView = UIKit.UIButton;
#elif MONOANDROID
using PlatformView = Google.Android.Material.Button.MaterialButton;
#elif WINDOWS
using PlatformView = Microsoft.UI.Xaml.Controls.Button;
#elif TIZEN
using PlatformView = Tizen.UIExtensions.NUI.Button;
#elif (NETSTANDARD || !PLATFORM) || (NET6_0_OR_GREATER && !IOS && !ANDROID && !TIZEN)
using PlatformView = System.Object;
#endif

namespace Microsoft.Maui.Handlers
```



Current Issues/Status

- Working on standard .NET MAUI Controls and Handlers
- SkiaSharp, Maui.Graphics, Avalonia, Impeller...
- Adding proper third-party integration support to .NET MAUI
 - Allow any platform to integrate with .NET MAUI from outside of their repo.
 - Our intention to drive it forward for the community.
 - No Avalonia-specific code goes into .NET MAUI.
- Identifying areas within Avalonia that need to be enhanced or written.
 - Navigation, Collection Views, Mobile-first controls, etc.
- Research third-party controls and how to maintain proper ABI support.



When?

- Intentions to target .NET 11
 - Open Source repo at or around .NET 11 Preview 1
 - or whenever .NET MAUI publishes their first preview.
 - Nightly builds
 - New preview releases to follow .NET MAUIs releases.
 - Stable for .NET 11...
- A lot can happen though...
 - No internal hacks in .NET MAUI for stable release, must be clean
 - Making sure controls are top-notch and stable
 - Making sure existing apps can feasibly be ported
 - Avalonia 12 and binding work back into repo

.NET Conf China 2025

改变世界 改变自己



THANK YOU