People In Space Uno Demo – Adaptive UX

The demo can be found here <https://github.com/Appmilla/PeopleInSpace_Uno>.

The demo is now extended to explore the problem space of defining a UX which works across different from factors such as web, desktop and mobile phone.

The demo now presents a list of crew which when tapped presents a detail view so the master details UX pattern,

After some research there are a number of different ways this can be achieved using Uno Platform, essentially the adaptive layouts techniques from UWP can be used which should be the preferred initial approach.

Other approaches could include defining different views for the different platforms. I experimented with creating a Xamarin.Forms app and successfully connected a UX list view to the existing view model and queries defined in the shared library in the People In Space Uno demo. If using this approach creating composable view models of shared UX with specific interactions controlled by platform specific code should work. The Use Case approach of Android Clean Architecture would be a good pattern to investigate.

The use of Visual States with StateTriggers is described in this [Tutorial: Create adaptive layouts](https://docs.microsoft.com/en-us/windows/apps/design/basics/xaml-basics-adaptive-layout) and an overview of the [List/details pattern](https://docs.microsoft.com/en-us/windows/apps/design/controls/list-details).

For the demo I have used the Windows Community Toolkit’s [ListDetailsView](https://docs.microsoft.com/en-us/windows/communitytoolkit/controls/masterdetailsview) control. This I found to work well on web, desktop, iOS and Android. Of particular note is that rotating the phone to landscape changes the view correctly and resizing the web and desktop apps to a narrow window width works correctly and mimics the portrait phone view.

The article [Using SplitView with Uno Platform](https://platform.uno/blog/using-splitview-with-uno-platform/) presents an alternative control the SplitView which can be used if some specific behaviour is required as it provides manual control over the view switching.

There is a useful video course on Pluaralsight [Uno Platform Fundamentals](https://app.pluralsight.com/library/courses/uno-platform-fundamentals/table-of-contents) which has a section on Adapting to Different Screen Sizes.

When using the toolkit the Nuget packages used for the UWP project should be the original Microsoft versions, using the Uno versions for the other projects.

Add the nuget packages below to the UWP project:

Microsoft.Toolkit.Uwp.UI.Controls

Microsoft.Toolkit.Uwp.UI.Controls.Layout

And to the other projects add these:

Uno.Microsoft.Toolkit.Uwp.UI.Controls

Uno.Microsoft.Toolkit.Uwp.UI.Controls.Layout

Other useful links:- [How to use Windows Community Toolkit](https://platform.uno/docs/articles/uno-community-toolkit.html?tabs=tabid-vswin)

The ListDetailsView control is in the packages:- <https://www.nuget.org/packages/Microsoft.Toolkit.Uwp.UI.Controls.Layout> and <https://www.nuget.org/packages/Uno.Microsoft.Toolkit.Uwp.UI.Controls.Layout>