Introduction to multiple-resolution support on Windows Phone 8 apps

This article discusses how to handle the issues that may arise due to the multiple screen resolutions that are supported in Windows Phone 8.

Available resolutions in WP8

Windows Phone 8 phones can have either of the two aspect ratios – 15:9 or 16:9, hence it is important to take care of screen resolutions during the design of the application. If a developer design and hardcode the UI components specific to one aspect ratio, the UI might get messed up in the other aspect ratio. To avoid this, a resolution free app design is very critical.

Windows Phone 8 supports phones following three resolutions:

- WVGA, also used in Windows Phone 7
- WXGA, essentially an HD version of WVGA
- 720P, HD

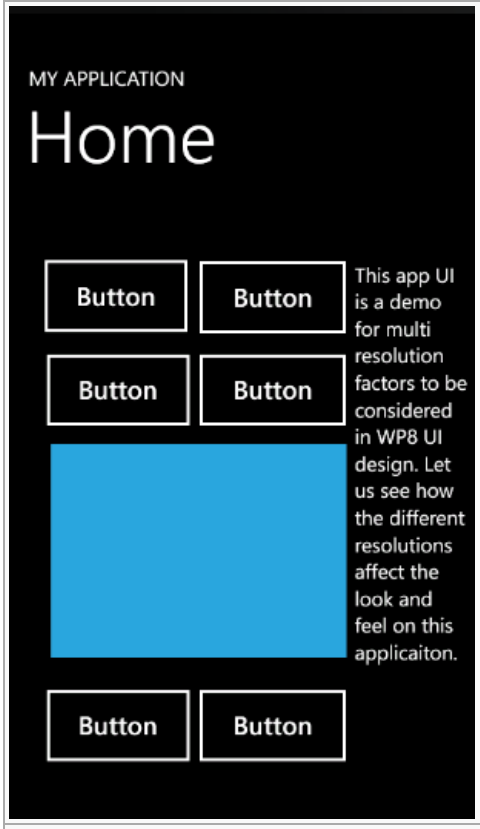
Resolution	Aspect Ratio	Scaled Resolution
WVGA (480x800 pixels)	15:9	480 × 800
WXGA (768x1280 pixels)	15:9	480 × 800
720P (720x1280 pixels)	16:9	480 × 853

Problems caused by unhandled multi-resolution

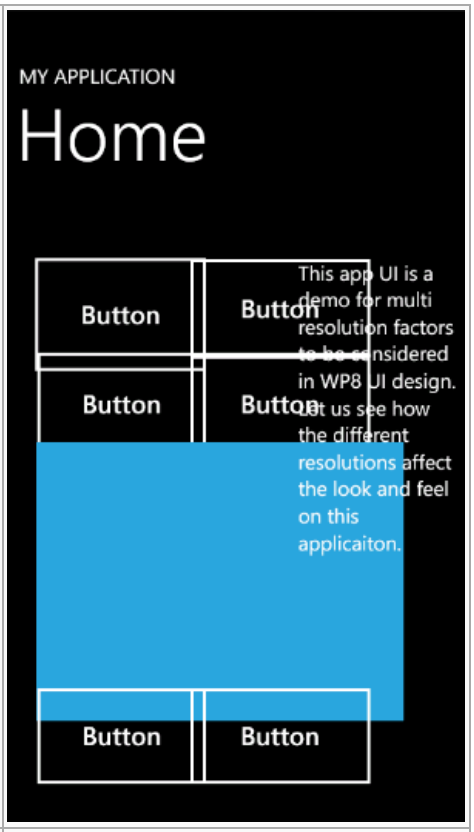
- **Overlap of UI controls and cluttered look**

If you designed the UI of app for a higher resolution and the UI is not having enough white space in between the control then the controls will overlap. This will not only make the UI look cluttered but also difficult for the user to interact with your app.

Below is a sample of how the app UI might get affected:



UI design as per expectation (Higher Resolution)

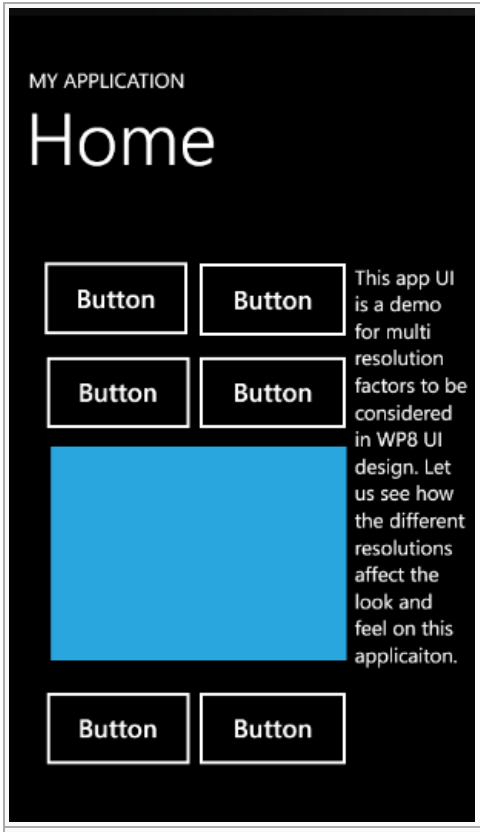
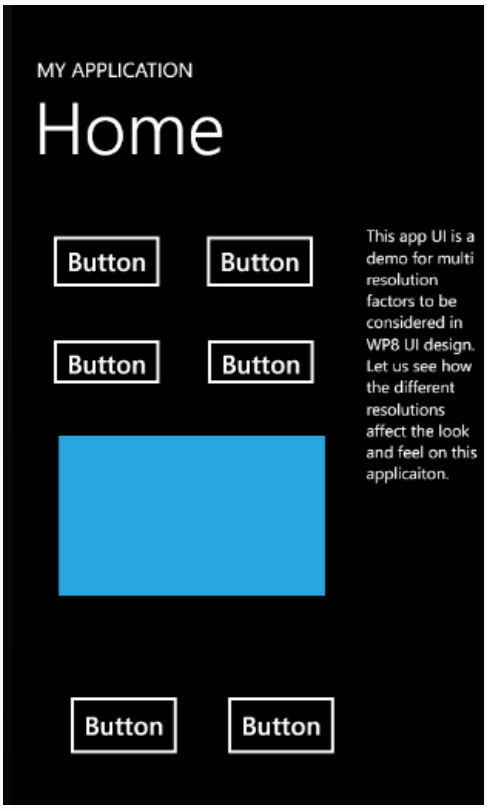


Overlapping UI controls (Low Resolution)

- **Too much white space**

(C) Copyright Nokia 2014. All rights reserved.

If you keep more than sufficient amount of white spacing for low resolution then the same app when viewed in a higher resolution WP8 device, will show much more white spacing in between the fixed height/width controls of the app. And too much of white space is not good for aesthetic design of the apps. Below is a sample of how the app UI might get affected:

	
UI design as per expectation (Lower Resolution)	Much empty space in between (Higher Resolution)

How to create a ‘resolution friendly’ App

Multi-resolution might seem like a bit of overhead to the WP developers as we are used to the one standard UI resolution design. But imagine the situation of ‘some’ other phone developers where the resolution set is not even defined and can differ vastly. We are still in a better situation and to handle this design aspect is quiet simple.

- **Develop for 480x800 pixels (lowest resolution)**

  Designing for the lowest resolution guarantees that the above mentioned overlapping of controls won’t happen. On higher resolution the app will have sufficient space and hence the controls won’t clutter.

- **Decide the UI container & controls as per the App Type**

  Some apps are content driven (e.g. tweeter display) while others are Fixed UI format driven (e.g. chess game).

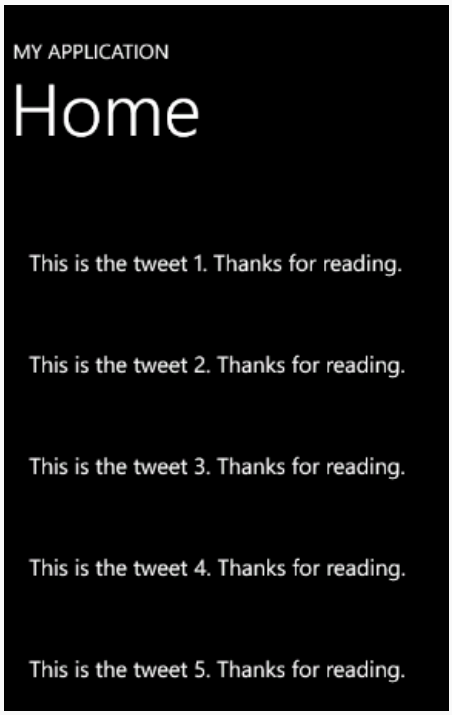
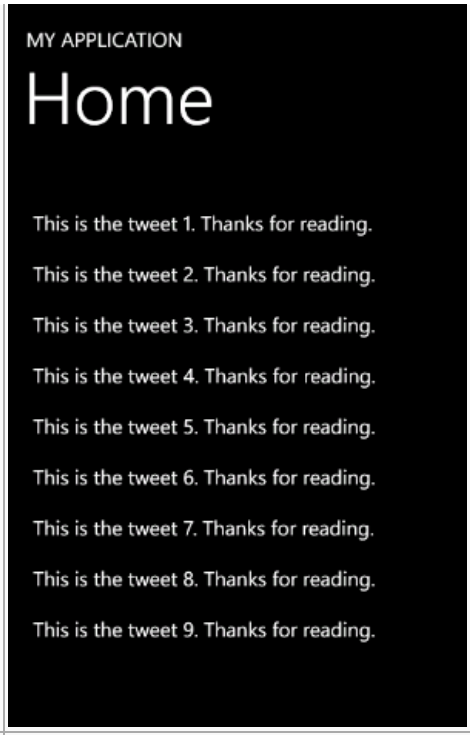
Content driven Apps

For a content driven app, you can display more content on WXGA/ WVGA resolution and the developer must make use of this additional space. This can be done by using appropriate containers and controls.

- Controls like ListBox or LongListSelector (new & recommended in WP8) which will expand as per available resolution.
- Use a container like the Grid control. Put the controls in a grid and do not hard code the height and width of its rows and columns. This way, the app adjusts itself to the screen resolution without need to do any special coding technique to handle the resolution factor.

Below is a sample of how the app UI might get affected under resolution changes:

(C) Copyright Nokia 2014. All rights reserved.

	
UI design in Lower Resolution	UI design adapted correctly for Higher Resolution

Fixed UI format driven Apps

As mentioned above consider an example of Chess game. In this app, you might consider using a grid container and the images of chess coins. The chess board layout will be taken care here but what about the coin images? This images need to be pixel perfect for the correct feel of the UI of this app which is very important. For this we can detect the resolution and load the appropriate images. And to do this we need to have different images for each resolution and load the appropriate one based on the phone's resolution.

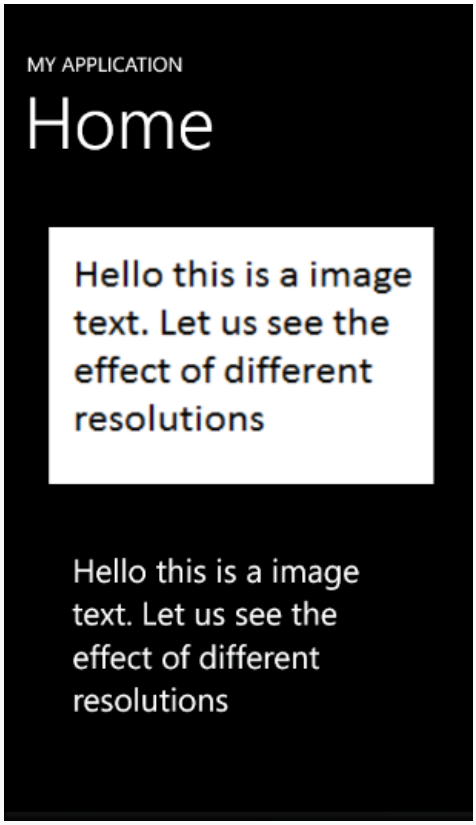
Current phone resolution can be got from `App.Current.Host.Content.ScaleFactor` and then load the appropriate image that is resolution specific.

For this types of apps avoid using self-adaptable controls like listbox as they might change the look of the application depending on the resolution, and this is something we are not expecting.

Use Fonts instead of Images where ever possible

For simple text, use fonts instead of images. Fonts automatically handle the resolution factor; **just make sure you don't use a too small font size**.
In the below scenario you can see that the left image is in a low resolution and the text looks perfect inside image control and the 'text control'. But when the same app is used in higher resolution(right), the controls resize i.e. get scaled as per resolution change the text inside image gets distorted but the text inside the text control still looks good, only a bit small in size.

 (C) Copyright Nokia 2014. All rights reserved.

	
Text in image & text control looks good (Lower Resolution)	Text in Image gets distorted, but text control renders it clearly (Higher Resolution)

Tiles,App Icon & Splash Screen

- Tiles and App Icon are not affected by multi-resolution as the phone takes care of scaling the images. Simply use images of WXGA resolution.
- The same applies for Splash screen as well, but if you want pixel perfect resolution then you can add three resolution specific splash screen images in the root folder.

For further information please refer : How to create a splash screen for Windows Phone

Tip: Developers can choose the required resolution in the Emulator while development. Make sure you do the resolution testing in the Emulator before publishing the app.

Summary

Creating a visual appeal is highly important for success of any application. Make sure to give equal amount of importance to the design aspect of WP8 apps. With the proper design practices in place and amazing features of WP8, your application is going to look great and work great!!!

This article didn't go into coding aspects of multi-resolution as those are simple and already explained in the below article: http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj206974%28v=vs.105%29.aspx

 (C) Copyright Nokia 2014. All rights reserved.

 (C) Copyright Nokia 2014. All rights reserved.