Extending the Windows Phone 8 Camera App with Lenses

This tutorial explains how to support the new lens feature in the Windows Phone 8 Camera App.

Introduction

The Camera app in Windows Phone 8 allows you to extend the app with lenses. Creating a “lens” in Windows Phone 8 will make it easier for your users to modify or share an image with your app directly from the Camera app. For instance, the Microsoft Research Face Touch app (which allows users to modify the facial expressions in photos) could create a lens so that users can quickly and more efficiently modify their photo.

First Things First

Required Icons

Lens apps require extra icons for the lens picker. You must add three different icons in your Assets folder: **Lens.Screen-WVGA.png** should be a 173x173 pixel image, **Lens.Screen-720p.png** should be a 259x259 pixel image, and **Lens.Screen-WXGA.png** should be a 277x277 pixel image.

Registering as a Lens

You’ll also need to register your app as lens-enabled. You can do this by editing the **WMAppManifest.xml** file located in the Properties folder of your app. Add the following lines to the XML file after the closing `</token>` tag (via MSDN):

```
<Extensions>
 <Extension ExtensionName="Camera_Capture_App"
  ConsumerID="{5B04B775-356B-4AA0-AAF8-6491FFEA5631}"
  TaskID="_default" />
</Extensions>
```

Handling a Lens Request

When your app is launched as lens, it will launch the default page unless you specify otherwise. If you’d like to have a separate page for your lens viewfinder, create a new page which does contain a viewfinder. After you have a viewfinder, you will need to forward the app to that page if the app is opened from the Camera app. Open your **App.xaml.cs** file, then add the following class to it (via MSDN):

```
class LensExampleUriMapper : UriMapperBase
{
    private string tempUri;
    public override Uri MapUri(Uri uri)
    {
        tempUri = uri.ToString();
        // Look for a URI from the lens picker.
        if (tempUri.Contains("ViewfinderLaunch"))
        {
            // Launch as a lens, launch viewfinder screen.
            return new Uri("/viewfinderExperience.xaml", UriKind.Relative);
        }
        // Otherwise perform normal launch.
        return uri;
    }
}
```

Finally, add the following to your `InitializePhoneApplication` function (you might need to show the "Phone application initialization" region) in **App.xaml.cs**:

```
//Assign the lens example URI-mapper class to the application frame.
RootFrame.UriMapper = new LensExampleUriMapper();
```

The above example will send your app to the **viewfinderExperience.xaml** page if it’s being launched as a lens, and to your app’s default start page otherwise.

Developing the Viewfinder Page

Getting the MediaViewer

The easiest way to go about getting the viewfinder page set up would be to download Microsoft's `MediaView` control and use it in your app. Download the Basic Lens Sample on MSDN, and then import the "MediaView" project into your own application. Build the solution (Build -> Build solution), and add a reference in your own project (right-click references in the Solution Explorer and click Add) to `MediaView`, then go back to (or create) your **viewfinderExperience.xaml** page and add the following line to the `<phone:PhoneApplicationPage>` element:

(C) Copyright Nokia 2013. All rights reserved.

```
xmlns:controls="clr-namespace:Microsoft.Phone.Controls;assembly=MediaViewer"
```

Frontend/XAML

At the very least, you must add a

```
<controls:MediaViewer>
```

element with

```
Items="{Binding CameraRoll}"
```

in the root grid. You can find a complete `MediaViewer` example as part of the Basic Lens Sample download on MSDN.

Backend/C#

You should download the code from the sdkBasicLensWP8CS project (again, part of the Basic Lens Sample download on MSDN), and then copy the ViewModels folder into your project (changing sdkBasicLensWP8CS to the name of your project). You can then modify the code in the **MainPage.xaml.cs** file found in the sdkBasicLensWP8CS project of the Basic Lens Sample download on MSDN to fit your needs.

How to Test Your Lens

The Windows Phone 8 Emulator has a functioning camera which will display a multi-coloured block moving around the screen. While this is satisfactory to ensure that you've set up the lens correctly, it's highly recommended that you test your app on an actual Windows Phone 8 device before completion.

Credit

Most of the code samples on this page were taken from the Basic Lens Sample on MSDN and Lens Extensibility on MSDN.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.