Custom splash screen page for Windows Phone applications

This article explains how to create a custom splash screen page for your Windows Phone application.

Introduction

In most cases during the launch of an application, you need to do some background operations such as making a web request, parsing some data or something else and while doing this in the background it is necessary to keep the user engaged by showing a splash screen.
This article helps you to create a splash screen from a XAML page and display it as soon as your application launches, and show it until your background activity finishes. The advantage of this approach rather than the popup approach is that you can dynamically set start up or the next page depending on the result of your background activity.

Necessary steps

1. First create a Windows phone Project in visual studio.
2. Delete the default splash screen image from project.
3. Your project already contains a MainPage.xaml file, keep it as it is. Add a new Page named SplashPage.xaml in the project, replace the existing xaml file code with the following code.

```
<phone:PhoneApplicationPage
    x:Class="WindowsPhoneSample5.SplashPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="Portrait" Orientation="Portrait"
    mc:Ignorable="d"
    shell:SystemTray.IsVisible="False">

    <!--LayoutRoot is the root grid where all page content is placed-->
    <Grid x:Name="LayoutRoot" Background="Transparent">
        <Image Source="/splash.jpg" Stretch="Fill"/>
    </Grid>

</phone:PhoneApplicationPage>
```

Make sure to set `shell:SystemTray.IsVisible` to False to use the full screen to display the image.

Modifying SplashPage.cs

Now, edit the SplashPage.cs file

1. Handle the `Loaded` Event of SplashPage - `SplashPage_Loaded`, as shown below,

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
```


(C) Copyright Nokia 2014. All rights reserved.

```
using System.Windows.Navigation;
using Microsoft.Phone.Controls;
using Microsoft.Phone.Shell;
using System.IO;

namespace WindowsPhoneSample5
{
    public partial class SplashPage : PhoneApplicationPage
    {
        private HttpWebRequest request;

        public SplashPage()
        {
            InitializeComponent();
            Loaded += SplashPage_Loaded;
        }

        void SplashPage_Loaded(object sender, RoutedEventArgs e)
        {
            request = (HttpWebRequest)HttpWebRequest.Create(new
Uri("http://www.developer.nokia.com/Community/Wiki/Portal:Windows_Phone_UI_Articles"));
            request.BeginGetResponse(new AsyncCallback(ReceiveResponseCallBack), null);
        }

        private void ReceiveResponseCallBack(IAsyncResult result)
        {
            HttpWebResponse response =
(HttpWebResponse)this.request.EndGetResponse(result);
            using (var stream = response.GetResponseStream())
            {
                using (StreamReader reader = new StreamReader(stream))
                {
                    string str =  reader.ReadToEnd();
                    Deployment.Current.Dispatcher.BeginInvoke(() =>
                    {
                        NavigationService.Navigate(new Uri("/MainPage.xaml",
UriKind.Relative));
                    });
                }
            }
        }
    }
}
```

Please note that the `SplashPage_Loaded` function merely makes an asynchronous HTTP request, and that the `ReceiveResponseCallBack` is called when the HTTP request returns a response. Upon receiving a response, the user is navigated to MainPage.xaml. deployment.Current.Dispatcher.BeginInvoke is used to access the UI thread. Thus in the function, you may add code to perform any necessary background tasks while the splash screen is shown to the user.

Modifying MainPage.xaml.cs

From the SplashPage the user is navigated to the MainPage. Once the MainPage is loaded we should remove the entry of the SplashPage from the back stack. This is done so that the SplashScreen isn't considered as a page and the user doesn't navigate back to it by using the back key. To do this we need to modify the file MainPage.xaml.cs as follows.

```
namespace WindowsPhoneSample5
{
```


(C) Copyright Nokia 2014. All rights reserved.

```
    public partial class MainPage : PhoneApplicationPage
    {
        // Constructor
        public MainPage()
        {
            InitializeComponent();
            Loaded += MainPage_Loaded;
        }

        void MainPage_Loaded(object sender, RoutedEventArgs e)
        {
            NavigationService.RemoveBackEntry();
        }

    }
}
```

ModifyingWMAppManifest

The last step is to add SplashPage.xaml in the Navigation Page tab of WMAppManifest.xml file. This is done so that the user is shown the splash screen when the application is launched.

Sample Code

Sample project demonstrated in this article is available for download here: Media:SplashDemoWinPhone.zip

Summary

This article illustrated how you can develop a custom splash screen page for your Windows Phone application. There are other approaches to create a custom splash screen too, please refer to the see also section for related articles.

(C) Copyright Nokia 2014. All rights reserved.