Como usar Pop-Ups em Windows Phone

Este artigo explicar como usar PopUp.

Introdução

A pop-up é uma interface gráfica do utilizdor (GUI) área de exposição, geralmente uma pequena janela, que aparece de repente ("pops-up") no plano da interface visual. Ele pode ser usado para exibir mensagens personalizadas, pegue as credenciais de login e outros insumos.

Este artigo mostra como definir um pop-up, tanto em XAML como em C #, como exibir e ocultar o pop-up no código e os valores selecionados de usuários de retorno, e como fechar o pop-up quando a tecla *volta* for pressionada (além para quando ele é fechado ou reconhecido usando botões na tela).

A aplicação exemplo mostra uma mensagem customizada.

Aplicação exemplo

Pop-up no caso " Pop-up em Xaml"

Pop-up no caso " Pop-up em.cs"

Começando

Primeiro crie um aplicativo para Windows Phone

- Abra o Visual Studio e selecione *'Windows Phone Application'* dos modelos instalados
- Seleccione Windows Phone 8.0 como versão "target".
- Clique direito sobre o projeto,"References", e clicar "Add Reference...". De seguida, procurar por *Microsoft.Phone.Controls* e selecionar e adicionar ao projeto.

Definindo o Pop-Up através de XAML

```
  <Popup x:Name="my_popup_xaml" Grid.Row="2">
    <Border BorderThickness="2" Margin="10" BorderBrush="{StaticResource
PhoneAccentBrush}">
      <StackPanel Background="LightBlue">
        <Image Source="/Images/disclaimer.png" HorizontalAlignment="Center"
Stretch="Fill" Margin="0,15,0,5"/>
        <TextBlock Text="Disclaimer" TextAlignment="Center" FontSize="40" Margin="10,0"
/>
        <TextBlock Text="This is a pop-up window to display disclaimer" FontSize="21"
Margin="10,0" />
        <StackPanel Orientation="Horizontal" Margin="0,10">
          <Button x:Name="btn_continue"  Content="continue" Width="215"
```


(C) Copyright Nokia 2014. All rights reserved.

```
Click="btn_continue_Click"/>
          <Button x:Name="btn_cancel"    Content="cancel"   Width="215"
Click="btn_cancel_Click"/>
        </StackPanel>
      </StackPanel>
    </Border>
  </Popup>
```

Nesta aplicação de exemplo adicionamos um "StackPanel" dentro do pop-up. Todos os outros controles exibidos no pop-up - TextBlocks, botões, estarão contido no StackPanel.

Definindo Pop-up usando CSharp

- primeiro criamos uma instância do pop-up:

```
  Popup my_popup_cs = new Popup();
```

- Adicionando controles para o pop-up:

```
 public void display_cspopup()
{
            Border border = new Border();
// to create green color border
            border.BorderBrush = new SolidColorBrush(Colors.Green);
            border.BorderThickness = new Thickness(2);
            border.Margin = new Thickness(10, 10, 10, 10);

            StackPanel skt_pnl_outter = new StackPanel();                          //
stack panel
            skt_pnl_outter.Background = new SolidColorBrush(Colors.LightGray);
            skt_pnl_outter.Orientation = System.Windows.Controls.Orientation.Vertical;

            Image img_disclaimer = new Image();                                    //
Image
            img_disclaimer.HorizontalAlignment =
System.Windows.HorizontalAlignment.Center;
            img_disclaimer.Stretch = Stretch.Fill;
            img_disclaimer.Margin = new Thickness(0, 15, 0, 5);
            Uri uriR = new Uri("Images/disclaimer.png", UriKind.Relative);
            BitmapImage imgSourceR = new BitmapImage(uriR);
            img_disclaimer.Source = imgSourceR;

            TextBlock txt_blk1 = new TextBlock();
// Textblock 1
            txt_blk1.Text = "Disclaimer";
            txt_blk1.TextAlignment = TextAlignment.Center;
            txt_blk1.FontSize = 40;
            txt_blk1.Margin = new Thickness(10, 0, 10, 0);
            txt_blk1.Foreground = new SolidColorBrush(Colors.White);

            TextBlock txt_blk2 = new TextBlock();
// Textblock 2
            txt_blk2.Text = "This is a pop-up window to display disclaimer";
            txt_blk2.TextAlignment = TextAlignment.Center;
            txt_blk2.FontSize = 21;
```


(C) Copyright Nokia 2014. All rights reserved.

```
            txt_blk2.Margin = new Thickness(10, 0, 10, 0);
            txt_blk2.Foreground = new SolidColorBrush(Colors.White);


            //Adding control to stack panel
            skt_pnl_outter.Children.Add(img_disclaimer);
            skt_pnl_outter.Children.Add(txt_blk1);
            skt_pnl_outter.Children.Add(txt_blk2);

            StackPanel skt_pnl_inner = new StackPanel();
            skt_pnl_inner.Orientation = System.Windows.Controls.Orientation.Horizontal;

            Button btn_continue = new Button();
// Button continue
            btn_continue.Content = "continue";
            btn_continue.Width = 215;
            btn_continue.Click += new RoutedEventHandler(btn_continue_Click);

            Button btn_cancel = new Button();
// Button cancel
            btn_cancel.Content = "cancel";
            btn_cancel.Width = 215;
            btn_cancel.Click += new RoutedEventHandler(btn_cancel_Click);

            skt_pnl_inner.Children.Add(btn_continue);
            skt_pnl_inner.Children.Add(btn_cancel);


            skt_pnl_outter.Children.Add(skt_pnl_inner);

            // Adding stackpanel to border
            border.Child = skt_pnl_outter;

           // Adding border to pup-up
            my_popup_cs.Child = border;

            my_popup_cs.VerticalOffset = 400;
            my_popup_cs.HorizontalOffset = 10;

            my_popup_cs.IsOpen = true;
}
```

Exibindo o pop-up

Para mostrar o pop-up :

```
    popup.IsOpen = true;
```

Para esconder ou cancelar o pop-up :

```
   popup.IsOpen = false;
```

Para exibir o pop-up pressione **Mostrar Pop-Up em Xaml** :

(C) Copyright Nokia 2014. All rights reserved.

```
 private void btn_displayfrom_xaml_Click(object sender, RoutedEventArgs e)
{
            if (my_popup_cs.IsOpen && !my_popup_xaml.IsOpen)
            {
                my_popup_cs.IsOpen = false;
                my_popup_xaml.IsOpen = true;
            }
            else
            {
                my_popup_xaml.IsOpen = true;
            }
}
```

Para exibir o pop-up pressione **Mostrar Pop-Up em .cs**:

```
 private void btn_displayfrom_cs_Click(object sender, RoutedEventArgs e)
 {
            if (my_popup_xaml.IsOpen && !my_popup_cs.IsOpen)
            {
                my_popup_xaml.IsOpen = false;
                display_cspopup();
            }
            else
            {
                display_cspopup();
            }
 }
```

Escondendo o pop-up e retorno de valores

Neste exemplo de aplicativo que está capturando dois eventos no menu pop-up

- on **continue** botão pressione a tecla:

```
private void btn_continue_Click(object sender, RoutedEventArgs e)
{
             if (my_popup_xaml.IsOpen)
            {
                my_popup_xaml.IsOpen = false;
            }
            else if (my_popup_cs.IsOpen)
            {
                my_popup_cs.IsOpen = false;
            }
}
```

aqui estamos simplesmente esconder o pop-up, enquanto pode-se usar esta função para executar o módulo desejado, como "navegar para outra página"

- on *cancel* 'botão pressione a tecla:

```
 private void btn_cancel_Click(object sender, RoutedEventArgs e)
```


(C) Copyright Nokia 2014. All rights reserved.

```
{
        if (my_popup_xaml.IsOpen)
        {
              my_popup_xaml.IsOpen = false;
        }
        else if (my_popup_cs.IsOpen)
        {
              my_popup_cs.IsOpen = false;
        }
}
```

Escondendo o pop-up quando '*backkey*' é pressionado

Este código mostra como esconder o pop-up quando o *'backkey'* é pressionado

```
protected override void OnBackKeyPress(CancelEventArgs e)
{
      if (this.my_popup_cs.IsOpen )
      {
              my_popup_cs.IsOpen = false;
              e.Cancel = true;
      }
      else if (this.my_popup_xaml.IsOpen)
      {
              my_popup_xaml.IsOpen = false;
              e.Cancel = true;
      }
}
```

Código Fonte

O código fonte pode ser obtido em File:Updated-Pop-Up Control.zip.

(C) Copyright Nokia 2014. All rights reserved.