Tab Control in Qt and WP7

This article demonstrates how to create a "tab-based" user interface in Qt and WP7.

Introduction

A tabbed interface provides a set of tab buttons, which may be selected to change the view (the content displayed). This code example creates three tabs and each tab has its own content. When user clicks on the tab it displays its content - in this case one of three images.

Tab Control in Qt

Tab Control in WP7

For Qt we will use TabGroup and related elements.

For Windows phone we use a Pivot control. There is a template that can be used to create tab based applications, but we won't use it here - instead we will use it from the toolbox, so that we can also see how to add the `Pivot` control on any part of the page like other controls.

Implementation

Qt Project (main.qml)

In Qt, tabs are defined using the TabBarLayout and TabGroup:

- The `TabBarLayout` defines the position of the Tab bar, and contains TabButton elements which provide the button text for each tab and the id of the element containing their content.
- The `TabGroup` defines the pages/content to be displayed for each tab - these are referenced in the `TabButtons` by id.

```
// define a tab bar layout with three buttons and link them to the content
    TabBarLayout {
        id: tabBarLayout
        anchors { left: parent.left; right: parent.right; top: parent.top }
        TabButton { tab: tab1content; text: "Tab 1" }
        TabButton { tab: tab2content; text: "Tab 2" }
        TabButton { tab: tab3content; text: "Tab 3" }
    }
    // define a blank tab group so we can add the pages of content later
    TabGroup {
        id: tabGroup
        anchors { left: parent.left; right: parent.right; top: tabBarLayout.bottom; bottom: parent.bottom }
        // define the content for tab 1
        Page {
            id: tab1content
            Image {
                id: image1
                anchors.centerIn: parent
                source: "Thumbnails/1.jpeg"
            }
        }
        // define the content for tab 2
        Page {
            id: tab2content
            Image {
                id: image2
                anchors.centerIn: parent
```


(C) Copyright Nokia 2013. All rights reserved.

```
                source: "Thumbnails/2.jpeg"
            }
        }
        // define content for tab 3
        Page {
            id: tab3content
            Image {
                id: image3
                anchors.centerIn: parent
                source: "Thumbnails/3.jpeg"
            }
        }
    }
```

Windows Phone 7 (MainPage.xaml)

First we will add the **Pivot** control in the toolbox:

1. **Right Click** on ToolBox->**Choose Items**-> Select **Pivot**
2. Now drag-drop the **Pivot** control from the toolbox to the XAML page.

Once we add the **Pivot** on the XAML page, we will see the tabs on WP style. On each tab content we add image which defines as its content. Thus when user clicks on any of the tabs it displays a different image.

```
<controls:Pivot Height="574" HorizontalAlignment="Left" Margin="12,27,0,0" Name="pivot1" Title="" VerticalAlignment="Top" Width="444
                <controls:PivotItem Header="Tab 1">

                    <Grid>
                        <Image Height="150" HorizontalAlignment="Left" Margin="103,101,0,0" Name="image1" Stretch="Fill" VerticalAli
                    </Grid>
                </controls:PivotItem>
                <controls:PivotItem Header="Tab 2">
                    <Grid>
                        <Image Height="150" HorizontalAlignment="Left" Margin="103,101,0,0" Name="image2" Stretch="Fill" VerticalAli
                    </Grid>
                </controls:PivotItem>
                <controls:PivotItem Header="Tab 3">
                    <Grid>
                        <Image Height="150" HorizontalAlignment="Left" Margin="103,101,0,0" Name="image3" Stretch="Fill" VerticalAli
                    </Grid>
                </controls:PivotItem>
            </controls:Pivot>
```

The way this works is fairly self-explanatory; the `Pivot` tags contain `PivotItem` definitions which define the tab text, and which themselves contain the tab content.

Source Code

- The full source code of Qt example is available here: File:TabControlQt.zip
- The full source code of WP7 example is available here: File:TabControlWP7.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.