Adding Vibration on UI elements in WP7

This article shows how to use VibrateController to vibrate the phone on user interaction in WP7.

Introduction

Vibration can be used to provide feedback from user action, for example when user clicks on a Button, to notify the user that it has been clicked.

This article demonstrates adding a VibrateController to some UI controls.

Vibration Demo

Implementation

First create an empty WP7 project. We add few UI elements in the XAML file like Button, CheckBox and Toggle Switch.

```
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
            <Button Content="Short Vibration" Height="143" HorizontalAlignment="Left" Margin="29,276,0,0" Name="button1" VerticalAli
            <Button Content="Long Vibration" Height="143" HorizontalAlignment="Left" Margin="29,430,0,0" Name="button2" VerticalAlig
            <CheckBox Content="CheckBox is checked " Height="72" HorizontalAlignment="Left" Margin="29,160,0,0" Name="checkBox1" Ver
            <toolkit:ToggleSwitch x:Name="ToggleSwitch" Header="Toggle Switch" IsChecked="false" Margin="29,48,0,0" Content="Switch
</Grid>
```

We add the *Checked* and *Unchecked* event handler for both CheckBox and Toggle Switch , and *Click* for the Button . Also we change the *Content* property for both CheckBox and Toggle Switch when *Checked* or *Unchecked* events are called to make sure that the UI elements are working as expected.

```
private void switch_Checked(object sender, RoutedEventArgs e)
        {
            ToggleSwitch.Content = "Switch On";
        }
 private void switch_Unchecked(object sender, RoutedEventArgs e)
        {
            ToggleSwitch.Content = "Switch Off";
        }
private void checkBox1_Checked(object sender, RoutedEventArgs e)
        {
            checkBox1.Content = "CheckBox is Checked";
        }
private void checkBox1_Unchecked(object sender, RoutedEventArgs e)
        {
            checkBox1.Content = "CheckBox is Unchecked";
        }
```

Now to add Vibration to the UI elements first we add the Namespace *Microsoft.Devices* to our *XAML.CS* file.

```
using Microsoft.Devices;
```

Next we create an instance of Vibration.

```
VibrateController iVibrateController = VibrateController.Default;
```

We create two methods `shotrVibration()` and `longVibration()`, which defines the time span of Vibration.

```
void shortVibration()
        {
```


(C) Copyright Nokia 2013. All rights reserved.

```
            iVibrateController.Start(TimeSpan.FromMilliseconds(100));
        }
void longVibration()
        {
            iVibrateController.Start(TimeSpan.FromSeconds(1));
        }
```

To generate a vibration we call these methods from UI elements event handlers.

```
private void switch_Checked(object sender, RoutedEventArgs e)
        {
            ToggleSwitch.Content = "Switch On";
            shortVibration();
        }
 private void switch_Unchecked(object sender, RoutedEventArgs e)
        {
            ToggleSwitch.Content = "Switch Off";
            shortVibration();
        }
private void checkBox1_Checked(object sender, RoutedEventArgs e)
        {
            checkBox1.Content = "CheckBox is Checked";
            shortVibration();
        }
 private void checkBox1_Unchecked(object sender, RoutedEventArgs e)
        {
            checkBox1.Content = "CheckBox is Unchecked";
            shortVibration();
        }
```

Now when user clicks on any of the UI elements, the event handlers are called and thus a short VibrateController gets generated to notify user input. To show the long vibration we can call the `longVibration()` method from one of the Button controls. One can call the respective short or long vibration function as and when required.

Summary

This article is a very basic implementation of VibrateController. Vibration duration can have a different meaning to the users. If the device vibrates for a long time, the user can be notified that there is something wrong with the device, likewise if there is any wrong input from user's end device can be vibrated for a long time.

Source Code

- The full source code of the example is available here: File:VibrationControlWP7.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.