Capturing Gestures with TouchPanel

This article demonstrates how to capture gesture events.

Introduction

This article shows how to capture a gesture in our Silverlight application using the XNA assembly **Microsoft.Xna.Framework.Input.Touch**.

The example app first declares what type of gesture we want to capture (just simple user touch input), and once we capture that particular gesture we perform some task. This task could be anything - in this case we just display what type of gesture we have captured.

Capturing Gestures

Implementation

First create a project with **Windows Phone Application** Template. Once the project is being created we add **Microsoft.Xna.Framework.Input.Touch** assembly reference in our project.

In XNA the touch surface is involved with the TouchPanel. With the help of this TouchPanel we can access the Gesture API and can enable the type of gesture we are interested in. In constructor we enable the type of gesture we want in this application.

```
TouchPanel.EnabledGestures = GestureType.Hold | GestureType.Tap | GestureType.DoubleTap |
                GestureType.Flick | GestureType.Pinch|GestureType.DragComplete|GestureType.FreeDrag|
                GestureType.HorizontalDrag|GestureType.VerticalDrag;
```

EnabledGestures gets or sets the gestures that are enabled for the touch panel with a single or a combination of gestures that are listed in the GestureType enumeration. It needs the **Namespace:** *Microsoft.Xna.Framework.Input.Touch*. Till now we have enabled the gesture that we are looking for, now we need to recognize them. As in this example we are planing to capture the gesture that happens anywhere in the screen, so we call ManipulationCompleted event with `LayoutRoot` control to occupy the complete screen.

```
LayoutRoot.ManipulationCompleted += new EventHandler<ManipulationCompletedEventArgs>(LayoutRoot_ManipulationCompleted);
```

In the event handler `LayoutRoot_ManipulationCompleted()` we first check is gesture available or not .

```
while (TouchPanel.IsGestureAvailable)
```

Then we read gesture on the touch panel and compare it in a `switch()` statement. Once the gesture matches we display the name of the gesture in the screen.

(C) Copyright Nokia 2013. All rights reserved.

```
void LayoutRoot_ManipulationCompleted(object sender, ManipulationCompletedEventArgs e)
        {
            while (TouchPanel.IsGestureAvailable)
            {
                GestureSample gesture = TouchPanel.ReadGesture();

                switch (gesture.GestureType)
                {

                    case GestureType.FreeDrag:
                        gestureText.Text = "FreeDrag";
                        break;

                    case GestureType.HorizontalDrag:
                        gestureText.Text = "HorizontalDrag";
                        break;
                    case GestureType.Pinch:
                        gestureText.Text = "Pinch";
                        break;

                    case GestureType.VerticalDrag:
                        gestureText.Text = "VerticalDrag";
                        break;
                    case GestureType.Tap:
                        gestureText.Text = "Tap";
                        break;

                    case GestureType.DoubleTap:
                        gestureText.Text = "Double Tap";
                        break;
                    case GestureType.Hold:
                        gestureText.Text = "Hold";
                        break;
                    case GestureType.Flick:
                        gestureText.Text = "Flick";
                        break;

                }
            }
        }
```

Here GestureSample is the data that we get from a multitouch gesture after a time period.

Note: The Pinch gesture could not be tested in emulator.

Source Code

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

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.