Draggable markers with Windows Phone maps API

Introduction

23 Dec 2012

The article illustrates draggable markers with Windows Phone maps API on WP8.

Implementation

In general there is no option for disabling the map panning during mouse events with WP8, thus normal approach on doing so (f.ex. it did work with WP7 Bing maps) while dragging markers would not work.

For this specific reason, if you want to implement draggable markers, you would need to do a bit of working around the problem. Luckily there is simple solution which can be used.

The main idea is to catch touch events with `System.Windows.Input.Touch.FrameReported`, instead of utilizing the maps event handlers. This will effectively allow us to disable the Map (thus disabling the panning), while still receiving the touch events needed for the dragging.

So first you need to add event handled for the touch events, i.e. add following code into your map page's constructor:

```
System.Windows.Input.Touch.FrameReported += Touch_FrameReported;
```

Then we could assume that we have variable `draggingNow` defined in the class, and it's basically telling the internal logic whether we are currently doing dragging or not. Then of course the `draggable` marker needs to be also accessible, so we can set its position while dragging. With this example we also have the marker stored as a variable in the class, and it is called `oneMarker`. The `Touch_FrameReported` event handler could be implemented as follows:

```
void Touch_FrameReported(object sender, TouchFrameEventArgs e)
{
    if (draggingNow == true)
    {
        TouchPoint tp = e.GetPrimaryTouchPoint(map1);
 if (tp.Action == TouchAction.Move)
        {
     if (oneMarker != null)
            {
                oneMarker.GeoCoordinate = map1.ConvertViewportPointToGeoCoordinate(tp.Position);
            }
 }
        else if (tp.Action == TouchAction.Up)
        {
     draggingNow = false;
            map1.IsEnabled = true;
        }
    }
}
```

Effectively, the code first checks whether we are dragging, and if we are then with any `TouchAction.Move` event we would be moving the marker (i.e. dragging it), until we would be getting `TouchAction.Up` indicating that the user has lifted the finger up. This event would then end the dragging, thus we mark this to the internal variable by setting the `draggingNow` to false, and also re-enable the map, so user can continue interacting with it normally.

Then to be able to get to the dragging mode, we would need to set the internal variable `draggingNow` to true. With this example we have assigned `left_key_down` event handler to the marker we want to drag, and inside it, we simply set the `draggingNow` to true, and disable the map.

```
void marker_MouseLeftButtonDown(object sender, MouseButtonEventArgs e))
{
    if (oneMarker != null)
    {
        draggingNow = true;
        map1.IsEnabled = false;
    }
}
```

And any touch events after this one is handled in the function, are then handled inside the `Touch_FrameReported` and as the `draggingNow` is true they will be used for dragging the marker and eventually for ending the dragging and re-enabling the map.

Example code

The full example for this code can be found from Windows Phone 8 Maps examples project, the code used here is implemented for example in DraggableMarker and MapMarkers examples.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.