15,871,273 members Sign in



articles

quick answers

discussions features Search for articles, questions, tip:

community

help

Articles / Desktop Programming / WPF











WPF simple zoom and drag support in a ScrollViewer

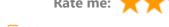
Kevin Stumpf













6 Nov 2013 CPOL

A sample describing a way to zoom with the mouse wheel or a slider and drag limited content which is hosted by a ScrollViewer.



Download demo - 24.3 KB

Introduction

I've been looking for a while for Open Source solutions that show a simple way of zooming and dragging arbitrary content which is hosted and managed by a ScrollViewer.

As I did not find a free one, I decided to write my own one and share it with you. It supports zooming by a slider as well as by the mouse wheel.

Using the code

If you have any questions regarding the usage, please feel free to post and I'll try to get back to you ASAP.

The main view is defined by the XAML below. The content that is to be zoomed and dragged is part of the "grid" control.

```
Shrink A
XML
<Window x:Class="ZoomExample.MainWindow"</pre>
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="500" Width="500">
    <Window.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <ResourceDictionary Source="Resources.xaml"/>
            </ResourceDictionary.MergedDictionaries>
        </ResourceDictionary>
    </Window.Resources>
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="Auto"/>
            <ColumnDefinition Width="*"/>
        </Grid.ColumnDefinitions>
        <Slider Grid.Column="0" Orientation="Vertical"</pre>
           HorizontalAlignment="Left" Minimum="1" x:Name="slider"/>
        <ScrollViewer Name="scrollViewer" Grid.Column="1"</pre>
              VerticalScrollBarVisibility="Visible"
              HorizontalScrollBarVisibility="Visible">
            <Grid Name="grid" Width="400"
              Height="400" RenderTransformOrigin="0.5,0.5">
                <Grid.LayoutTransform>
                    <TransformGroup>
                        <ScaleTransform x:Name="scaleTransform"/>
                    </TransformGroup>
                </Grid.LayoutTransform>
                <Viewbox Grid.Column="0" Grid.Row="0">
                    <ContentPresenter Content="{StaticResource Kompass}"/>
                </Viewbox>
            </Grid>
        </ScrollViewer>
    </Grid>
</Window>
```

Zooming and dragging is managed by the code-behind:

```
C#

public partial class MainWindow : Window
{
    Point? lastCenterPositionOnTarget;
    Point? lastMousePositionOnTarget;
    Point? lastDragPoint;

    public MainWindow()
    {
```

```
InitializeComponent();
    scrollViewer.ScrollChanged += OnScrollViewerScrollChanged;
    scrollViewer.MouseLeftButtonUp += OnMouseLeftButtonUp;
    scrollViewer.PreviewMouseLeftButtonUp += OnMouseLeftButtonUp;
    scrollViewer.PreviewMouseWheel += OnPreviewMouseWheel;
    scrollViewer.PreviewMouseLeftButtonDown += OnMouseLeftButtonDown;
    scrollViewer.MouseMove += OnMouseMove;
    slider.ValueChanged += OnSliderValueChanged;
}
void OnMouseMove(object sender, MouseEventArgs e)
    if (lastDragPoint.HasValue)
        Point posNow = e.GetPosition(scrollViewer);
        double dX = posNow.X - lastDragPoint.Value.X;
        double dY = posNow.Y - lastDragPoint.Value.Y;
        lastDragPoint = posNow;
        scrollViewer.ScrollToHorizontalOffset(scrollViewer.HorizontalOffset - dX);
        scrollViewer.ScrollToVerticalOffset(scrollViewer.VerticalOffset - dY);
    }
}
void OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)
    var mousePos = e.GetPosition(scrollViewer);
    if (mousePos.X <= scrollViewer.ViewportWidth && mousePos.Y <</pre>
        scrollViewer.ViewportHeight) //make sure we still can use the scrollbars
        scrollViewer.Cursor = Cursors.SizeAll;
        lastDragPoint = mousePos;
        Mouse.Capture(scrollViewer);
    }
void OnPreviewMouseWheel(object sender, MouseWheelEventArgs e)
    lastMousePositionOnTarget = Mouse.GetPosition(grid);
    if (e.Delta > 0)
        slider.Value += 1;
    if (e.Delta < 0)</pre>
        slider.Value -= 1;
    }
    e.Handled = true;
}
void OnMouseLeftButtonUp(object sender, MouseButtonEventArgs e)
    scrollViewer.Cursor = Cursors.Arrow;
    scrollViewer.ReleaseMouseCapture();
```

```
lastDragPoint = null;
}
void OnSliderValueChanged(object sender,
     RoutedPropertyChangedEventArgs<double> e)
{
    scaleTransform.ScaleX = e.NewValue;
    scaleTransform.ScaleY = e.NewValue;
    var centerOfViewport = new Point(scrollViewer.ViewportWidth/2,
                                     scrollViewer.ViewportHeight/2);
    lastCenterPositionOnTarget = scrollViewer.TranslatePoint(centerOfViewport, grid);
}
void OnScrollViewerScrollChanged(object sender, ScrollChangedEventArgs e)
{
    if (e.ExtentHeightChange != 0 || e.ExtentWidthChange != 0)
        Point? targetBefore = null;
        Point? targetNow = null;
        if (!lastMousePositionOnTarget.HasValue)
        {
            if (lastCenterPositionOnTarget.HasValue)
            {
                var centerOfViewport = new Point(scrollViewer.ViewportWidth/2,
                                                  scrollViewer.ViewportHeight/2);
                Point centerOfTargetNow =
                      scrollViewer.TranslatePoint(centerOfViewport, grid);
                targetBefore = lastCenterPositionOnTarget;
                targetNow = centerOfTargetNow;
        }
        else
            targetBefore = lastMousePositionOnTarget;
            targetNow = Mouse.GetPosition(grid);
            lastMousePositionOnTarget = null;
        }
        if (targetBefore.HasValue)
            double dXInTargetPixels = targetNow.Value.X - targetBefore.Value.X;
            double dYInTargetPixels = targetNow.Value.Y - targetBefore.Value.Y;
            double multiplicatorX = e.ExtentWidth/grid.Width;
            double multiplicatorY = e.ExtentHeight/grid.Height;
            double newOffsetX = scrollViewer.HorizontalOffset -
                                dXInTargetPixels*multiplicatorX;
            double newOffsetY = scrollViewer.VerticalOffset -
                                dYInTargetPixels*multiplicatorY;
            if (double.IsNaN(newOffsetX) || double.IsNaN(newOffsetY))
            {
                return;
            scrollViewer.ScrollToHorizontalOffset(newOffsetX);
```

```
scrollViewer.ScrollToVerticalOffset(newOffsetY);
}
}
}
```

License

This article, along with any associated source code and files, is licensed under The Code Project Open License (CPOL)

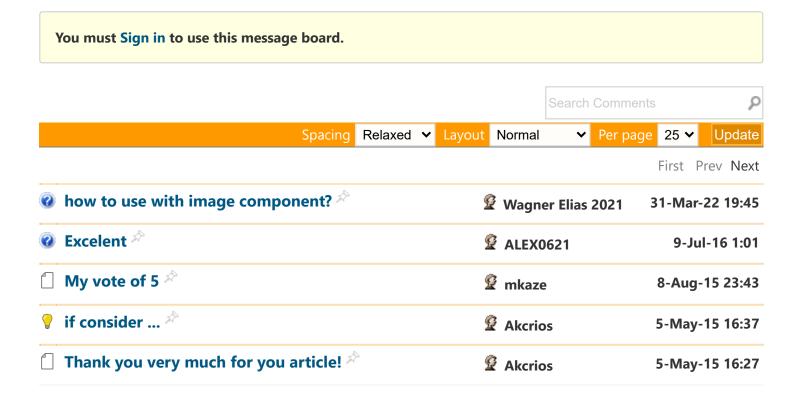
Written By

Kevin Stumpf

Germany Germany

This member has not yet provided a Biography. Assume it's interesting and varied, and probably something to do with programming.

Comments and Discussions



② Zoom on a click [∞]	Shyamala_123	16-Feb-15 18:52
☐ My vote of 5 Å	🙎 amir_pro	3-Feb-15 8:45
☐ My vote of 5 Å	🗣 amir_pro	3-Feb-15 8:43
② Great work [♠]	⊈ GPans	11-Oct-13 9:53
☐ My vote of 5 Å	🙎 Bijay Kant Salotry	19-Sep-13 6:52
☐ My vote of 5 Å	🗣 John Bracey	20-Jun-13 21:50
■ My vote of 5 🖈	Shahin Khorshidnia	11-Jun-13 21:19
☐ My vote of 5 Å		6-Jun-13 4:27
	🗣 redjzuzzj	4-Jun-13 15:22
② Splendid! [♠]	🗣 Porgram Lover	24-Feb-13 11:34
☐ My vote of 5 Å	🗣 arturo1346	31-Oct-12 5:37
☐ My vote of 5 🌣	🙎 Vivek Sharma 2	14-Jun-12 23:21
☐ My vote of 5 Å	🗣 njdnjdnjdnjd	16-May-12 6:27
☐ My vote of 5 Å	⊈ Dezfoul	18-Apr-12 19:29
② Created some helper classes [⋄]	🙎 kgoulding	18-Jan-12 10:03
Re: Created some helper classes A	⊈ Member 9442681	1-Jul-13 22:24
☐ My vote of 5 [★]	⊈ Ubloobok	22-Oct-11 21:46
☐ My vote of 5 🌣	⊈ Koss87	12-Jun-11 14:22
☐ Thanks for the Demo! 🌣	🙎 (noor)	31-Mar-11 3:49
Re: Thanks for the Demo!	Akcrios	5-May-15 16:17
Last Visit: 31-Dec-99 18:00 Last Update: 3-Apr-24 10:22	Refresh	1 2 Next ▷
☐ General ■ News Suggestion ② Question → Bug ☑ Admin	nswer 墜 Joke 🖒 Praise	e 嶐 Rant 🐠

 $Use\ Ctrl+Left/Right\ to\ switch\ messages,\ Ctrl+Up/Down\ to\ switch\ threads,\ Ctrl+Shift+Left/Right\ to\ switch\ pages.$

Permalink Advertise Privacy Cookies Terms of Use Layout: fixed | fluid

Posted 29 Jul 2010

Article Copyright 2010 by Kevin Stumpf Everything else Copyright © CodeProject, 1999-2024

Web01 2.8:2024-04-02:1