Subject: Horizontal Mouse Wheel support request
Posted by Tom1 on Tue, 02 Jan 2024 14:07:26 GMT
View Forum Message <> Reply to Message

Hi,

Can we get support for horizontal mouse wheel in CtrlCore?

A prototype implementation in CtrlCore for Windows is attached in the zip file.

The horizontal scrolling is needed to support two finger panning of documents (or maps in my case) on laptop touch pads as a pair for two finger zoom which already automatically maps to Ctrl + MouseWheel.

Best regards,

Tom

File Attachments
1) WindowsHorizontalMouseWheelInCtrlCore.zip, downloaded 274 times

Subject: Re: Horizontal Mouse Wheel support request
Posted by Tom1 on Wed, 24 Jan 2024 12:51:02 GMT
View Forum Message <> Reply to Message

Hi Again,

Due to the fact that with a regular mouse the horizontal wheel scrolling is commonly simulated through 'Shift + Mouse Wheel' I decided to introduce a simpler lightweight implementation for Horizontal Mouse Wheel support in Windows. It requires only minimal changes in CtrlCore/Win32Proc.cpp, which you can find attached. (No changes at all in CtrlCore.h or CtrlMouse.cpp, like in the previous solution.)

So, there is no added MouseWheelHor() method, but instead the horizontal wheel comes in through regular MouseWheel() with keyflags |= K_SHIFT, which is the common horizontal wheel simulation solution anyway. Therefore, both simulated and real horizontal wheel events become available in MouseWheel().

NOTE: The attached CtrlCore/Win32Proc.cpp also attempts to fix a touch screen issue with Pen aware apps by slightly adjusting the Pen detection mask. This is based on pen/touch differentiation as per:

https://learn.microsoft.com/en-us/windows/win32/tablet/system-events-and-mouse-messages?redirectedfrom=MSDN

Best regards,

Tom

File Attachments
1) Win32Proc.cpp, downloaded 1447 times

Subject: Re: Horizontal Mouse Wheel support request
Posted by Tom1 on Wed, 24 Jan 2024 15:03:38 GMT
View Forum Message <> Reply to Message

Hi,

Please find attached the code for GTK in CtrlCore/GtkEvent.cpp. The implementation generally works the same as in Windows, but the granularity is fixed to 120 as it was before. (In Windows we can actually get smaller values too when using two finger gestures on touch pad, making the scrolling proportional to movement. I could not find a way to do that on GTK yet -- partly because I do not have Linux installed on a laptop with touch pad.)

Please note that these implementations use Windows native scrolling directions (i.e. Positive values are received with Up/Forward and Right wheel movement):

https://learn.microsoft.com/en-us/windows/win32/inputdev/wm- mousewheel#parameters
https://learn.microsoft.com/en-us/windows/win32/inputdev/wm- mousehwheel#parameters

I soon found out that horizontal scrolling via 'Shift + MouseWheel' is already supported within some U++ controls, but unfortunately the horizontal direction is reversed having positive going left. I think this should be corrected in those U++ Ctrls that support horizontal scrolling via 'Shift + MouseWheel'.

Best regards,

Tom

EDIT: Did small cleanup for GtkEvent.cpp.

File Attachments
1) GtkEvent.cpp, downloaded 1419 times
