
Subject: Re: Combining function graphics pan and zoom changes

Posted by [Helder Marchetto](#) on Thu, 19 Dec 2013 14:13:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Thursday, December 19, 2013 3:01:19 PM UTC+1, alx wrote:

> Le jeudi 19 décembre 2013 14:14:09 UTC+1, Helder a écrit :

>

>> Hi,

>

>>

>

>> is there a way to combine function graphics pan/zoom changes.

>

>>

>

>> If you have two images in two windows and with the mouse wheel change the first image by zooming in, is there an easy way to do the same to the other image?

>

>>

>

>> I'm considering using the event_handler object again, but if there is a more straight forward way... please let me know.

>

>>

>

>>

>

>>

>

>> Cheers,

>

>>

>

>> Helder

>

>

>

> Please look at the various WINDOW handler functions in the documentation.

>

> alx.

Yeah, I'm just doing that. Not that easy, but I'm getting around it. Here is what I put in the method handling wheel changes:

```
FUNCTION AvoidMovingObj::MouseWheel, Win, x, y, Delta, KeyMods
```

```
o = Win.GetSelect()
```

```
IF ISA(Win.GetSelect(), 'POLYLINE') && (o.NAME EQ self.Name) THEN RETURN, 0 $
```

```
ELSE BEGIN
```

```
xy = (*self.oWin).ConvertCoord(x, y, 0, /DEVICE, /TO_NORMAL)
*self.oWin->Refresh, /DISABLE
ShiftVal = [xy[0]-0.5, xy[1]-0.5]
zoomFactor = (Delta LT 0) ? 1/1.25d : 1.25d
*self.oWin->Translate, ShiftVal[0], ShiftVal[1], /NORM
*self.oWin->Scale, zoomFactor, zoomFactor
*self.oWin->Translate, -ShiftVal[0], -ShiftVal[1], /NORM
*self.oWin->Refresh
RETURN, 1
ENDELSE
END
```

The property self.oWin is the other object (image in my case) that is changed.
To get the zoom factor right I had to do some digging in IDL function graphic procedures... a bit tricky, but I use the same line that function graphic is using...

Cheers,
Helder
