
Subject: Device Coordinates vs Normalized Coordinates

Posted by [Helder Marchetto](#) on Fri, 23 Nov 2012 15:43:37 GMT

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

Hi,
I'm dealing with something strange here, or at least it is strange for me.
I'm plotting a two perpendicular lines on a plot where the mouse is moving on a plot. This I do by managing with Motion_events.
It happens that on the day I wrote this, I used normalized coordinates. Notice that I don't want to have my line going beyond the plotted range, therefore I had to use the !X.Window and !Y.Window coordinates.

By chance, a few days later I tried plotting after a click the same. This time I used data coordinates (because I had the exact min and max from the plot).

Here is what I find strange:
The motion events where processed extremely quickly.
The button events took from half to over a second.
After placing a lot of SysTime(1) around in my code, I found that this happened exactly at the command PlotS.

Is this to be expected? I'm quite puzzled at the moment and can't imagine that the PlotS command takes so much time to convert the coordinates...

I don't see any other reason for this. It could be something in my code, but I can't pin point it (The plot command does not call any function whilst passing parameters). The code is almost 2000 lines and growing, therefore I don't append it, but here are the lines creating the problem:

```
TPostDraw = SYSTIME(1)
PlotS, [devcoord[0], devcoord[0]], [(*(self.yCommonPlotRange))[0],
(*(self.yCommonPlotRange))[1]], Color=GetColor, Thick=2.0, LineStyle=2, /data
PlotS, [(*(self.xCommonPlotRange))[0], (*(self.xCommonPlotRange))[1]], [DevCoord[1],
DevCoord[1]], Color=GetColor, Thick=2.0, LineStyle=2, /data
TPlotS = SYSTIME(1)
```

and the differences between TPlotS and TPostDraw that I found were:

```
0.32700014
0.46799994
0.70199990
1.0289998
1.5600002
```

I switched to norm coordinates, and everything is fast as I would expect it.

Cheers,
Helder
