
Subject: trouble with image(), /data, /norm

Posted by [Helder Marchetto](#) on Wed, 24 Jun 2015 11:01:42 GMT

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

Hi,

I'm fighting since a while with this problem and can't seem to find a good solution, so I'm going to ask for some help.

I have a widget_window in which I draw images with annotations in the form of /data or /norm. I often have to change the image and I have to redefine the dimensions of the image. Below is some example code to show/explain the questions/problems. Sorry if the code is a bit long, but if you copy 'n paste you should have it running in no time.

1) According to the image() documentation, image_dimensions cannot be changed. However, if one changes the image using the setData method, the new dimensions are used. To test this, move the mouse cursor to the top-right of the image. You will see that this matches with the given dimensions (label on the right). If you click on the 'change image dimensions' button, the dimensions are changed using the setData method. The dimensions (seen with the mouse) are correct. Good.

Furthermore the objects defined in data space stay put to the data-space coordinates (and move). The objects defined using normal coordinates don't move.

2) Now comes the tricky part. If you move the image with the middle mouse button or zoom in with the mouse wheel, the data-space defined objects move along and the norm-space defined objects stay where they were. If the image dimensions are now changed with the 'change image dimensions' button (setData!) the following occurs (against expectations):

- the data-space defined objects don't move!
- the displayed image size changes! (to see this zoom out and then change use the change image dimensions button...)

3) Zoom out and click on 'change image pix size'. Now the image will slightly change in size on the screen. What is causing this?

This pretty much sums up my problems. Doesn't seem like much, but I can't get a consistent method for keeping objects in data space.

Is there a better way to change image (pixel size and data-dimensions) than the one used here?

Thanks for any help,
Helder

Here is the code

```
function ww_mouse_motion_event, win, x, y, keymods
info = win.uvalue
MousePos = win->ConvertCoord(x, y, /device, /to_data)
widget_control, (*info).wBase, tlb_set_title=string(MousePos[0:1],format='("Pos=(",f0.2,"
```

```

",f0.2,"")')
return, 1
end

pro testObjWidgets_event, event
;help, event
widget_control, event.top, get_uvalue=info
case event.id of
(*info).wButtonImage:begin
    print, 'change image completely'
    imgSize = long(randomu(s, 1)*300.0)+200l
    newDims = randomu(s, 1)*20.0 + 10.0
    (*info).dims = [newDims[0],newDims[0]]
    (*info).slmg = [imgSize[0],imgSize[0]]
    img = dist((*info).slmg[0])
    xArr = (*info).dims[0]*dindgen((*info).slmg[0])/((*info).slmg[0]-1d )
    yArr = (*info).dims[0]*dindgen((*info).slmg[0])/((*info).slmg[0]-1d )
    (*info).iObj->setData, img, xArr, yArr
    widget_control, (*info).wStatus, set_value=string([( (*info).slmg[0],(*info).slmg[0],(*info).di
ms[0],(*info).dims[0]), format='(i4,"x",i4," pix, ",f0.1,"x",f0.1," mm")')
end
(*info).wButtonDimensions:begin
    print, 'change image dimensions'
    newDims = round(randomu(s, 1)*20d + 10d )
    (*info).dims = [newDims[0],newDims[0]]
    img = dist((*info).slmg[0])
    xArr = (*info).dims[0]*dindgen((*info).slmg[0])/((*info).slmg[0]-1d )
    yArr = (*info).dims[0]*dindgen((*info).slmg[0])/((*info).slmg[0]-1d )
    (*info).iObj->setData, img, xArr, yArr
    widget_control, (*info).wStatus, set_value=string([( (*info).slmg[0],(*info).slmg[0],(*info).di
ms[0],(*info).dims[0]), format='(i4,"x",i4," pix, ",f0.1,"x",f0.1," mm")')
end
(*info).wButtonPixCount:begin
    print, 'change image pix size'
    imgSize = long(randomu(s, 1)*300.0)+200l
    (*info).slmg = [imgSize[0],imgSize[0]]
    img = dist((*info).slmg[0])
    xArr = (*info).dims[0]*dindgen((*info).slmg[0])/((*info).slmg[0]-1d )
    yArr = (*info).dims[0]*dindgen((*info).slmg[0])/((*info).slmg[0]-1d )
    (*info).iObj->setData, img, xArr, yArr
    widget_control, (*info).wStatus, set_value=string([( (*info).slmg[0],(*info).slmg[0],(*info).di
ms[0],(*info).dims[0]), format='(i4,"x",i4," pix, ",f0.1,"x",f0.1," mm")')
end
(*info).wBase:begin
    widget_control, (*info).wDraw, xSize=event.y, ySize=event.y
end
else:print, 'dunno what to do with id ', event.id
endcase

```

```

end

pro testObjWidgets_cleanup, event
print, 'cleanup'
widget_control, event, get_uvalue=info
ptr_free, info
end

pro testObjWidgets
;create starting widgets
wBase = widget_base(/row, title='Pos=( 0, 0)', /tlb_size_events)
wDraw = widget_window(wBase, xsize=600, ysize=600,
mouse_motion_handler='ww_mouse_motion_event')
wButtonCol = widget_base(wBase, /col)
wButtonImage = widget_button(wButtonCol, /dynamic_resize, value='change image
completely')
wButtonDimensions = widget_button(wButtonCol, /dynamic_resize, value='change image
dimensions')
wButtonPixCount = widget_button(wButtonCol, /dynamic_resize, value='change image pix size')
wStatus = widget_label(wButtonCol, /dynamic_resize, value='600x600 pix, 10x10 mm')
widget_control, wBase, /realize
; Retrieve the newly-created Window object.
widget_control, wDraw, get_value=oWin
iObj = image(dist(400), current=oWin, image_dimensions=[10,10], dimensions=[400,400],
margin=0)
tdd = text(3.0,0.3,'data defined', 'r', /data, target=iObj)
tnd = text(0.5,0.5,'norm defined', 'b', /norm, target=iObj)
pdd = polygon([3.0,4.0,4.0,3.0],[7.0,7.0,8.0,8.0],'r2', /data, target=iObj, /fill_background,
fill_color="red")
pdd = polygon([0.3,0.4,0.4,0.3],[0.3,0.3,0.4,0.4],'b2', /norm, target=iObj, /fill_background,
fill_color="light steel blue")

info = ptr_new({wBase:wBase,$
wDraw:wDraw,$
wButtonImage:wButtonImage,$
wButtonDimensions:wButtonDimensions,$
wButtonPixCount:wButtonPixCount,$
wStatus:wStatus,$
dims:[10.0,10.0],$
slmg:[400,400],$
oWin:oWin,$
iObj:iObj})
oWin.uvalue=info
widget_control, wBase, set_uvalue=info
xmanager, 'testObjWidgets', wBase, cleanup = 'testObjWidgets_Cleanup', event_handler =
'testObjWidgets_event', /no_block
end

```