I should also add that on subsequent clicks, the new item is still plotted in the 2nd window even
though the over-plotting is breaking in the first window.

On Wednesday, January 9, 2013 8:20:08 PM UTC-5, Brian Cherinka wrote:
> I have 2 widget draw windows and I want to click and highlight in one and plot something in the
other. I've managed to get the mouse coords to convert properly upon switching back and forth
between windows, but when I overplot lines for highlighting, the lines, after the first call, either
become shifted or do not plot at all. So something is still not quite right with the window coord
system.
>
>
>
> I've set the 1 window to be the active one and I load in that window's !x, and !y so I'm not sure
what else I need to do to get it to consistently overplot.
>
>
>
> My code is something like
>
>
>
>   (*pState).wDrawFlux:begin
>
>   widget_control, (*pState).wDrawFlux, get_value=value, get_uvalue=uvalue
>
>   case event.type of
>
>   ;button press
>
>   0:begin
>
>   ;left mouse button press
>
>   if event.press eq 1 then begin
>
>   print, 'left mouse button pressed!'
>
>   wset, value
>
>   !x=(*pState).fluxd.x
>
>   !y=(*pState).fluxd.y
>
>   dataxy = convert_coord(event.x, event.y, /device, /to_data)

```
>
>               fiberid = floor(dataxy[1])
>
>               print, 'Fiber id: '+strtrim(fiberid,2)
>
>
>
>               ;highlight
>
>               oplot, [-5000,5000], fltarr(2)+fiberid, color=!red
>
>               oplot, [-5000,5000], fltarr(2)+fiberid+1, color=!red
>
>
>
>               ;plot in other window
>
>               widget_control, (*pState).wDrawSpec, get_value=specwin
>
>               wset, specwin
>
>               plot, flat[*,fiberid]
>
>
>
> dot dot dot
>
>
>
> Thanks for any help.
```

Subject: Re: convert_coord & overplotting across multiple widget draw windows
Posted by David Fanning on Thu, 10 Jan 2013 01:36:47 GMT
View Forum Message <> Reply to Message

Brian Cherinka writes:

> I have 2 widget draw windows and I want to click and highlight in one and plot something in the
other. I've managed to get the mouse coords to convert properly upon switching back and forth
between windows, but when I overplot lines for highlighting, the lines, after the first call, either
become shifted or do not plot at all. So something is still not quite right with the window coord
system.

I would say you need to at least restore the !P system variable, too.
Otherwise, the coordinates won't know where the plot is in the window.
(The windows are presumably the same size, right?)

I always position graphics with the POSITION keyword, so I
find the cgCoord object perfect for keeping track of this
kind of thing. Just call the Draw method before you do any
drawing of any sort.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: convert_coord & overplotting across multiple widget draw windows
Posted by David Fanning on Thu, 10 Jan 2013 01:40:07 GMT

Brian Cherinka writes:

> I should also add that on subsequent clicks, the new item is still plotted in the 2nd window even
though the over-plotting is breaking in the first window.

The first rule of widget programming is to know which window you are
drawing your graphics into. WSET is your friend here, and you should use
it in any event handler that includes graphics commands. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: convert_coord & overplotting across multiple widget draw windows
Posted by havok2063 on Thu, 10 Jan 2013 02:48:18 GMT

Thanks. Saving the !p state did the trick. I have two related questions.

1.) Is there an easy way to erase the previous over-plotted lines without re-plotting the whole window?

2.) How would you suggest combining a single left-click to "click and plot something" and a left-click drag a rubber-band box to create a window to zoom in on the plot?   I was thinking of putting the "click+plot" on a left button release event and have the box drawing on the motion, with a condition on release that if the new mouse coords are different than the old, then it's not a single click..so do the box thing and zoom.

On Wednesday, January 9, 2013 8:40:07 PM UTC-5, David Fanning wrote:
> Brian Cherinka writes:
>
>
>
>> I should also add that on subsequent clicks, the new item is still plotted in the 2nd window even though the over-plotting is breaking in the first window.
>
>
>
> The first rule of widget programming is to know which window you are
>
> drawing your graphics into. WSET is your friend here, and you should use
>
> it in any event handler that includes graphics commands. :-)
>
>
>
> Cheers,
>
>
>
> David
>
>
>
>
>
>
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/

>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: convert_coord & overplotting across multiple widget draw windows
Posted by David Fanning on Thu, 10 Jan 2013 03:07:20 GMT

View Forum Message <> Reply to Message

Brian Cherinka writes:

> 1.) Is there an easy way to erase the previous over-plotted lines without re-plotting the whole window?

Put the basic plot in a pixmap window and use the ol' smoke and mirrors trick of Device Copy to "erase" the overplotted lines. It happens so fast no one will suspect what you are doing. It is shocking how many fabulous things can be done with such a simple technique. See the AnnotateWindow program, for example:

   http://www.idlcoyote.com/catalyst/annotate.html

> 2.) How would you suggest combining a single left-click to "click and plot something" and a left-click drag a rubber-band box to create a window to zoom in on the plot?   I was thinking of putting the "click+plot" on a left button release event and have the box drawing on the motion, with a condition on release that if the new mouse coords are different than the old, then it's not a single click..so do the box thing and zoom.

It is VERY easy for draw widget event handling code to get completely out of control if you try to be too clever. It's sometimes a lot easier to put the draw window into "modes": draw plot mode, zoom mode, make a box mode, etc. That way, you can just have different event handlers that get selected for the draw widget, depending on what mode it is in. This is a LOT easier to handle. The example above is a good example of this technique, too.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: convert_coord & overplotting across multiple widget draw windows
Posted by havok2063 on Sat, 12 Jan 2013 01:08:05 GMT

Thanks a lot. This helps a lot. This pixmap is neat and a little confusing at the same time, especially when you're juggling them over 3 different windows back and forth. I've got it working now though, both for my over-plotting and for zooming with the rubberband zoom box. My next step is to try to add image panning.

On Wednesday, January 9, 2013 10:07:20 PM UTC-5, David Fanning wrote:
> Brian Cherinka writes:
>
>
>
>> 1.) Is there an easy way to erase the previous over-plotted lines without re-plotting the whole window?
>
>
>
> Put the basic plot in a pixmap window and use the ol' smoke and mirrors
>
> trick of Device Copy to "erase" the overplotted lines. It happens so
>
> fast no one will suspect what you are doing. It is shocking how many
>
> fabulous things can be done with such a simple technique. See the
>
> AnnotateWindow program, for example:
>
>
>
>   http://www.idlcoyote.com/catalyst/annotate.html
>
>
>
>> 2.) How would you suggest combining a single left-click to "click and plot something" and a left-click drag a rubber-band box to create a window to zoom in on the plot?   I was thinking of putting the "click+plot" on a left button release event and have the box drawing on the motion, with a condition on release that if the new mouse coords are different than the old, then it's not a single click..so do the box thing and zoom.
>
>
>
> It is VERY easy for draw widget event handling code to get completely
>
> out of control if you try to be too clever. It's sometimes a lot easier
>
> to put the draw window into "modes": draw plot mode, zoom mode, make a
>

> box mode, etc. That way, you can just have different event handlers that
>
> get selected for the draw widget, depending on what mode it is in. This
>
> is a LOT easier to handle. The example above is a good example of this
>
> technique, too.
>
>
>
> Cheers,
>
>
>
> David
>
>
>
>
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")