Subject: Re: David Fanning's rubberband box
Posted by davidf on Thu, 12 Mar 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Joseph Scott Stuart (nospam@ll.mit.edu) writes:

> I've recently gotten David Fanning's book, and I like it quite a bit.
> Good job, David! I am having a problem though with the rubberband box
> code on page 117. I'm using IDL 5.0.2 on an SGI. When I am finished
> drawing the rubberband box and release the mouse button, the program
> does not exit the repeat loop. I have to push another button before it
> will register that I've released the first button.

I am skiing this week and I...well...forgot to bring a book
with me. An oversight I will not soon repeat, even if I have
to smuggle it up here in my dirty clothes bag. :-(

I presume you are working with the code that can be found
in the rubberbox.pro program. As it happens, this code is
designed to act *exactly* as it is acting. That is to say,
it is designed NOT to exit the loop until you click some
button OTHER than the left mouse button.

The point of the exercise is to show you that you can hold
the button down and drag (the way you are using it), or
you can simply click different locations in the window and
the box is drawn correctly, too. This is a benefit of the
Wait keyword to the Cursor command.

I am NOT saying this is the best user interface. In fact,
it is probably not. In my programming courses, we sometimes
try to make the rubberband box act as you prefer it to.
Some programmers can get it to work, but not very many.
It involves some pretty convoluted code. (I'm sorry I
don't have an example and I doubt I can code it up
in the time I have before my wife returns from an errand.
Perhaps some of the people who have made it work in my
course could offer us an example.)

I introduce the coding exercise in my courses not because
I want people to write code like this, but because I want
people to work really, really hard to do something that
is trivially easy to do with widget programs. After
struggling with the rubberband-thing-in-normal-IDL for
an hour, most people are highly motivated to learn how
to write a widget program. :-)

(For an example of how this is done in a widget program,

see the program ZIMAGE. The advantage of doing this in
a widget program is that it is also trivially easy to
make the program do something else if you use the right
or middle button instead of the left. You simply switch
event handlers appropriately. One of my associates is
currently working on some code that will do this in a
generalized way so that widget programmers don't have
to write similar code in each program they write.)

Whoops! I think that is my wife pulling up. I've got
to go out in this beautiful Colorado weather and ski. :-)

If you are still struggling next week, contact me and
I'll figure something out for you.

Cheers!

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: David Fanning's rubberband box
Posted by davidf on Fri, 13 Mar 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Joseph Scott Stuart (nospam@ll.mit.edu) writes:

> I've recently gotten David Fanning's book, and I like it quite a bit.
> Good job, David! I am having a problem though with the rubberband box
> code on page 117. I'm using IDL 5.0.2 on an SGI. When I am finished
> drawing the rubberband box and release the mouse button, the program
> does not exit the repeat loop. I have to push another button before it
> will register that I've released the first button. I added the
> following line just before the ENDREP:
>
> cursor, x, y, /NoWait
> ENDREP UNTIL !Mouse.Button NE 1
>
> That mostly fixed the problem, but it still seems to only work
> sporadically. That is, sometimes I'll draw a box, release the button,
> and it ends the loop as it is supposed to, other times it does not,
> and I have to either redraw the box or push another button. This is a
> problem because I was planning to use the other mouse buttons for
> other things (middle button to zoom out, right button to exit the

> program). This behavior persists when I run the program on the SGI
> with the display on the console or on a PC (NT) with X-Win32

Whoops, blush. On getting home and having a look at this
code, I realized that you could probably get what you want
just by changing the keyword on the CURSOR command that is
inside the loop to CHANGE from WAIT. In this state, the
box is drawn as long as you hold the left button down and
drag, but you exit the program when you release the button.
(Because !Mouse.Button is set to 0.)

Here is a poor man's box cursor that draws a rubberband
box in any window. (You can pass the window's ID as a
parameter.) Moreover, keywords specify the color index used
to draw the box (COLOR) and whether you want the output
box coordinates to be in data coordinates (DATA) or normalized
coordinates (NORMAL). The default is to report the box
coordinates in DEVICE coordinates. The box coordinates
are ordered like the are in the Position keyword
(i.e, [smallest_X, smallest_Y, largest_X, largest_Y]).
I've written this as a function because I like to get
the box coordinates back explicitly.

To see it work, try this:

```
  image = BytScl( Dist(200), Top=149)
  TVLCT, 255, 255, 0, 150
  Window, XSize=200, YSize=200
  TV, image
  box = DrawBox(Color=150)
```

Cheers,

David

```
 ************************************************************ ****
Function DrawBox, $
  wid, $        ; ID of window where box is drawn. (Default: !D.Window)
  Color=color, $ ; The color index of the box. (Default: !D.N_Colors-1)
  Data=data, $   ; Box coordinates returned as DATA coordinates.
  Normal=normal  ; Box coordinates returned as NORMAL coordinates.

  ; Check for parameters.

IF N_Params() EQ 0 THEN wid = !D.Window > 0
IF N_Elements(color) EQ 0 THEN color = (!D.N_Colors - 1) < 255

  ; Make requested window active. Get size of window.
```

```
WSet, wid
xsize = !D.X_VSize
ysize = !D.Y_VSize

   ; Create a pixmap for erasing the box. Copy window
   ; contents into it.

Window, /Pixmap, /Free, XSize=xsize, YSize=ysize
pixID = !D.Window
Device, Copy=[0, 0, xsize, ysize, 0, 0, wid]

   ; Get the first location in the window. This is the
   ; static corner of the box.

WSet, wid
Cursor, sx, sy, /Down, /Device

   ; Go into a loop.

REPEAT BEGIN

      ; Get the new cursor location (dynamic corner of box).

   Cursor, dx, dy, /Change, /Device

      ; Erase the old box.

   Device, Copy=[0, 0, xsize, ysize, 0, 0, pixID]

      ; Draw the new box.

   PlotS, [sx, sx, dx, dx, sx], [sy, dy, dy, sy, sy], $
      /Device, Color=color

ENDREP UNTIL !Mouse.Button NE 1

   ; Erase the final box.

Device, Copy=[ 0, 0, xsize, ysize, 0, 0, pixID]

   ; Delete the pixmap.

WDelete, pixID

   ; Order the box coordinates.

sx = Min([sx,dx], Max=dx)
```

```
sy = Min([sy,dy], Max=dy)

   ; Need coordinates in another coordinate system?

IF Keyword_Set(data) THEN BEGIN
   coords =  Convert_Coord([sx, dx], [sy, dy], /Device, /To_Data)
   RETURN, [coords[0,0], coords[1,0], coords[0,1], coords[1,1]]
ENDIF

IF Keyword_Set(normal) THEN BEGIN
   coords =  Convert_Coord([sx, dx], [sy, dy], /Device, /To_Normal)
   RETURN, [coords[0,0], coords[1,0], coords[0,1], coords[1,1]]
ENDIF

   ; Return device coordinates, otherwise.

RETURN, [sx, sy, dx, dy]
END
```

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: David Fanning's rubberband box
Posted by Martin Schultz on Fri, 13 Mar 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Joseph Scott Stuart wrote:
>
> I've recently gotten David Fanning's book, and I like it quite a bit.
> Good job, David! I am having a problem though with the rubberband box
> code on page 117. I'm using IDL 5.0.2 on an SGI. When I am finished
> drawing the rubberband box and release the mouse button, the program
> does not exit the repeat loop. I have to push another button before it
> will register that I've released the first button. I added the
> following line just before the ENDREP:
>
>   cursor, x, y, /NoWait
> ENDREP UNITL !Mouse.Button NE 1
>
> That mostly fixed the problem, but it still seems to only work
> sporadically. That is, sometimes I'll draw a box, release the button,
> and it ends the loop as it is supposed to, other times it does not,

Scott,

please find attached my version of David F.'s rubberband. It draws a rubberband box as long as you press the mouse button and exits if you release it (*and* if you move the mouse which is caused by the CHANGE flag). It's not perfect but works for me. If you come up with a better solution let me know.

Regards,
Martin.

---------- -------
Dr. Martin Schultz
Department for Earth&Planetary Sciences, Harvard University
186 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
IDL-homepage: http://www-as.harvard.edu/people/staff/mgs/idl/
---------- -------


```
pro trackcursor,x1,y1,x2,y2,first=first,pixwin=pixwin
; draws rubberband for data selection
; as of 23 DEC 1997: uses device,copy (idea D.Fanning)
; return pixwin to destroy window afterwards

common trackcommon,curwin,pixwinindex

  ; create pixmap and copy current image
  if (keyword_set(first)) then begin
     curwin = !d.window
     window,/free,xsize=!d.x_size,ysize=!d.y_size,/pixmap
     pixwin = !d.window
     pixwinindex = pixwin
     device,copy=[0,0,!d.x_size,!d.y_size,0,0,curwin]
     wset,curwin
  endif

  ; copy old image from pixwin
  device,copy=[0,0,!d.x_size,!d.y_size,0,0,pixwinindex]
```

```
   ; overlay red rectangle with new mouse coordinates
   plots,[x1,x2,x2,x1,x1],[y1,y1,y2,y2,y1],color=2,thick=1

return
end



pro use_cursor,x1=mx1,x2=mx2,y1=my1,y2=my2
; activates graphics cursor for rubberband selection

!err=0
   cursor,mx1,my1,/down
   trackcursor,mx1,my1,mx1,my1,/first,pixwin=pixwin

   while (!err ne 0) do begin
      cursor,mx2,my2,/change
      trackcursor,mx1,my1,mx2,my2
   endwhile

   wdelete,pixwin

;   print,'X1=',mx1,',Y1=',my1
;   print,'X2=',mx2,',Y2=',my2

end
```