Subject: Re: drawing lines in IDL
Posted by davidf on Tue, 31 Mar 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Steve Hartmann (morph@vuse.vanderbilt.edu) writes:

> Does anyone know how to use the mouse to draw free hand lines over a
> displayed image?

Here is an example that involved some simple modifications of
the example program to draw rubberband boxes that appears on my
web page.

Click and draw with the LEFT mouse button to draw free-hand
lines on the image. Click and drag with the RIGHT mouse
button to draw straight lines on the image. Among other
things this example illustrates is how you might go about
collecting points for creating a free-hand region of interest
using pointers.

Cheers,

David

```
************************************************************
PRO Example_Button_Events, event

; All program button events handled here.

Widget_Control, event.top, Get_UValue=info, /No_Copy

   ; Which button caused this event?

Widget_Control, event.id, Get_Value=thisButtonValue
CASE thisButtonValue OF

   'Quit': BEGIN
      Widget_Control, event.top, /Destroy
      RETURN
      ENDCASE

   'Erase Lines': BEGIN
      WSet, info.wid
      TV, BytScl(info.image, Top=info.drawColor-1)
      ENDCASE

ENDCASE
Widget_Control, event.top, Set_UValue=info, /No_Copy
```

```
END
 ;------------------------------------------------------------ -------



PRO Example_Draw_Events, event

; All program draw widget events handled here.

   ; Deal only with DOWN, UP, and MOTION events.

IF event.type GT 2 THEN RETURN

   ; Get the info structure.

Widget_Control, event.top, Get_UValue=info, /No_Copy

   ; What kind of event is this?

eventTypes = ['DOWN', 'UP', 'MOTION']
thisEvent = eventTypes[event.type]

whichButton = ['NONE', 'LEFT', 'MIDDLE', 'NONE', 'RIGHT']

CASE thisEvent OF

   'DOWN': BEGIN

      ; Which button was used? LEFT or RIGHT?

      info.buttonUsed = whichButton(event.press)

      ; Turn motion events on for the draw widget.

      Widget_Control, info.drawID, Draw_Motion_Events=1

      ; Create a pixmap. Store its ID. Copy window contents into it.

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

      ; Initialize the starting coordinates of the line.

      IF info.buttonUsed EQ 'RIGHT' THEN BEGIN
         info.xstart = event.x
         info.ystart = event.y
      ENDIF ELSE BEGIN
```

```
      info.xvalues = Ptr_New([event.x])
      info.yvalues = Ptr_New([event.y])
    ENDELSE

    ENDCASE

  'UP': BEGIN

      ; Erase the last line drawn. Destroy the pixmap.

    WSet, info.wid
    Device, Copy=[0, 0, info.xsize, info.ysize, 0, 0, info.pixID]
    WDelete, info.pixID

      ; Turn draw motion events off. Clear events queued for widget.

    Widget_Control, info.drawID, Draw_Motion_Events=0, Clear_Events=1

      ; Draw the final line.

    IF info.buttonUsed EQ 'RIGHT' THEN BEGIN

      PlotS, [info.xstart, event.x], [info.ystart, event.y], $
        /Device, Color=info.drawColor

      ; Reinitialize the line starting coordinates.

      info.xstart = -1
      info.ystart = -1

    ENDIF ELSE BEGIN

      PlotS, *info.xvalues, *info.yvalues, /Device, $
         Color=info.drawColor

      ; Reinitialize the pointers.

      Ptr_Free, info.xvalues
      Ptr_Free, info.yvalues
      info.xvalues = Ptr_New()
      info.yvalues = Ptr_New()

    ENDELSE
    ENDCASE

  'MOTION': BEGIN

      ; Here is where the actual line is drawn and erased.
```

```
      ; First, erase the last line.

    WSet, info.wid
    Device, Copy=[0, 0, info.xsize, info.ysize, 0, 0, info.pixID]

    IF info.buttonUsed EQ 'RIGHT' THEN BEGIN

      ; Draw a straight line.

    PlotS, [info.xstart, event.x], [info.ystart, event.y], /Device, $
        Color=info.drawColor

    ENDIF ELSE BEGIN

      ; Get the points of the new free-hand line and draw it.

    *info.xvalues = [*info.xvalues, event.x]
    *info.yvalues = [*info.yvalues, event.y]
    PlotS, *info.xvalues, *info.yvalues, /Device, Color=info.drawColor

    ENDELSE

    ENDCASE

ENDCASE

  ; Store the info structure.

Widget_Control, event.top, Set_UValue=info, /No_Copy
END
 ;------------------------------------------------------------ -------



PRO Example

  ; Open an image data set.

file = Filepath(SubDirectory=['examples','data'], 'ctscan.dat')
OpenR, lun, file, /Get_Lun
image = BytArr(256, 256)
ReadU, lun, image
Free_Lun, lun

xsize = (Size(image))[1]
ysize = (Size(image))[2]

  ; Create the TLB.
```

```
tlb = Widget_Base(Title='Free-Hand Lines in a Widget Program', $
   MBar=menubase)

   ; Create some menu bar buttons.

fileID = Widget_Button(menubase, Value='File', $
   Event_Pro='Example_Button_Events')
eraseID = Widget_Button(fileID, Value='Erase Lines')
quitID = Widget_Button(fileID, Value='Quit')

   ; Create the draw widget graphics window. Turn button events ON.

drawID = Widget_Draw(tlb, XSize=xsize, YSize=ysize, Button_Events=1, $
   Event_Pro='Example_Draw_Events')

   ; Realize widgets and make draw widget the current window.

Widget_Control, tlb, /Realize
Widget_Control, drawID, Get_Value=wid
WSet, wid

   ; Load drawing color and display the image.

drawColor = !D.N_Colors-1
TVLCT, 255, 255, 0, drawColor
TV, BytScl(Image, Top=drawColor-1)

   ; Create an "info" structure with information to run the program.

info = { image:image, $        ; The image data.
         wid:wid, $            ; The window index number.
         drawID:drawID, $      ; The draw widget identifier.
         pixID:-1, $           ; The pixmap identifier.
         xsize:xsize, $        ; The X size of the graphics window.
         ysize:ysize, $        ; The Y size of the graphics window.
         xstart:-1, $          ; The starting X coordinate of the line.
         ystart:-1, $          ; The starting Y coordinate of the line.
         xvalues:Ptr_New(), $  ; The X coordinates of the free-hand line.
         yvalues:Ptr_New(), $  ; The Y coordinates of the free-hand line.
         buttonUsed:'NONE', $  ; A flag to indicate which button is used.
         drawColor:drawColor } ; The rubberband box color.

   ; Store the info structure.

Widget_Control, tlb, Set_UValue=info, /No_Copy

   ; Start the program going.
```

```
XManager, 'example', tlb, /No_Block
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/
