
Subject: Re: Gravity ?

Posted by [thompson](#) on Sat, 14 May 1994 17:42:24 GMT

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

LINDSTROM@acs.harding.edu (Greg Lindstrom) writes:

> Greeting All-

> I am running PV-Wave4.2CL on a Sun SPARC IPC (SunOS 4.1.3) and
> X11R5. What I would really like to do is set "gravity" on the
> cursor in my display window so that the crosshair lines will
> extend to the edges of the display window. I have seen it done
> in "X", but not in WAVE. Can it be done? Can you tell me how?

> Thanks,

> Greg Lindstrom
> Harding University

> BTW- My grant runs out this summer. If you are looking for a
> programmer/administrator.....

I think this may be another entry for the FAQ. I've come across two different routines that do this, which I include below. I believe the first is available as part of the JHU/APL IDL library, via anonymous ftp to [fermi.jhuapl.edu](ftp://fermi.jhuapl.edu) [128.244.147.18] in /pub/idl. There may be a more up-to-date version there; this one is pretty old.

Bill Thompson

```
=====
=====
;----- --
;+
; NAME:
;   MOVCROSS
; PURPOSE:
;   Interactive cross-hair on a plot.
; CATEGORY:
; CALLING SEQUENCE:
;   movcross, [x, y]
; INPUTS:
; KEYWORD PARAMETERS:
;   Keywords:
;     COLOR=c set color of line.
;     LINESTYLE=s line style.
; OUTPUTS:
;   x = data X coordinate of cross-hair.    out
```

```

;   y = data Y coordinate of cross-hair.    out
; COMMON BLOCKS:
; NOTES:
;   Note: Works in data coordinates so must make a plot first.
; MODIFICATION HISTORY:
;   R. Sterner, 1 Feb, 1991
;
; Copyright (C) 1991, Johns Hopkins University/Applied Physics Laboratory
; This software may be used, copied, or redistributed as long as it is not
; sold and this copyright notice is reproduced on each copy made. This
; routine is provided as is without any express or implied warranties
; whatsoever. Other limitations apply as described in the file disclaimer.txt.
;-
;-----

```

```

pro movcross, xdt, ydt, color=color, linestyle=linestyle, $
  help=help

if keyword_set(help) then begin
  print, ' Interactive cross-hair on a plot.'
  print, ' movcross, [x, y]'
  print, '  x = data X coordinate of cross-hair.    out'
  print, '  y = data Y coordinate of cross-hair.    out'
  print, ' Keywords:'
  print, '  COLOR=c set color of line.'
  print, '  LINESSTYLE=s line style.'
  print, ' Note: Works in data coordinates so must make a plot first.'
  return
endif

  if (total(abs(!x.crangle)) eq 0.0) then begin
    print, ' Error in movcross: data coordinates not yet established.'
    print, '  Must make a plot before calling movcross.'
    return
  endif

  print, ' '
  print, ' Interactive cross-hair.'
  print, ' Move with mouse.'
  print, ' Left button:  List X,Y  position.'
  print, ' Middle button: Quit, erase cross-hair.'
  print, ' Right button: Quit, keep cross-hair.'
  print, ' '

flag = 0
yl = 0
xl = 0
if n_elements(color) eq 0 then color = 255

```

```

if n_elements(linestyle) eq 0 then linestyle = 0
    tmp = [!y.range, !y.crange]
    yy = [min(tmp), max(tmp)]
    tmp = convert_coord([0,0],yy,/to_dev)
    yydv = transpose(tmp(1,0:1))
    tmp = [!x.range, !x.crange]
    xx = [min(tmp), max(tmp)]
    tmp = convert_coord( xx, [0,0], /to_dev)
    xxdv = transpose(tmp(0,0:1))

loop: tvrdc, xdt, ydt, 0, /data ; Read data coordinates of cursor.
xdt = xdt > xx(0) < xx(1)
ydt = ydt > yy(0) < yy(1)
if !err ne 0 then begin ; Button pressed?
    if !err eq 1 then begin
        print, ' X,Y = ',xdt,ydt
        wait, .1
    endif
    if !err eq 4 then return
    if !err eq 2 then begin
        tv, tx, xl, 0 ; Erase X line on exit.
        tv, ty, 0, yl ; Erase Y line on exit.
        return
    endif
endif
wait, 0 ; Pause.
tmp = convert_coord(xdt, ydt, /to_dev) ; Convert data to device coord.
xdv = tmp(0) & ydv = tmp(1)
if flag eq 0 then ty = tvrd(0,ydv,!d.x_size,1) ; row at ydv 1st time.
    if flag eq 0 then tx = tvrd(xdv,0,1,!d.y_size) ; col at xdv 1st time.
if flag eq 1 then begin ; Not first time?
    if (ydv ne yl) or (xdv ne xl) then begin ; Cursor moved?
        tv, tx, xl, 0 ; Replace old column at xl.
        tv, ty, 0, yl ; Replace old row at yl.
        tx = tvrd(xdv,0,1,!d.y_size) ; Read new column at xdv.
        ty = tvrd(0, ydv,!d.x_size,1) ; Read new row at ydv.
    endif
endif
flag = 1 ; Set first time flag.
xl = xdv ; Remember where column read.
yl = ydv ; Remember where row read.
;--- Plot vertical at new = xdv. ----
if (xdt ge xx(0)) and (xdt le xx(1)) then begin
    plots, [xdv, xdv], yydv, color=color, $
    linestyle=linestyle, /dev
endif
;--- Plot horizontal at new = ydv. ----
if (ydt ge yy(0)) and (ydt le yy(1)) then begin

```

```
plots, xxdv, [ydv, ydv], color=color, $
linestyle=linestyle, /dev
endif
goto, loop
end
```

```
=====
=====
```

Newsgroups: comp.lang.idl-pvwave
Path: nsisrv!ames!sun-barr!cs.utexas.edu!zaphod.mps.ohio-state.edu!
!news.acns.nwu.edu!ossenu.astro.nwu.edu!pendleton
From: pendleton@ossenu.astro.nwu.edu (Jim Pendleton)
Subject: Re: Giant crosshairs (with arrows)
Message-ID: <1992Aug29.174136.1@ossenu.astro.nwu.edu>
Lines: 123
Sender: usenet@news.acns.nwu.edu (Usenet on news.acns)
Organization: GRO/OSSE Northwestern U.
References: <laa-270892173708@berserk.c3.lanl.gov> <Btpv88.4ss@csn.org>
Date: Sat, 29 Aug 1992 22:41:36 GMT

In article <Btpv88.4ss@csn.org>, james@teal.csn.org (Jim Phillips) writes:
> In article <laa-270892173708@berserk.c3.lanl.gov> laa@lanl.gov (Lee Ankeny) writes:

```
>> Hello Wave gurus,
>>
>> I'd like to implement a "giant crosshair", like the old tek terminals used
>> to have. It would extend from top edge to bottom edge, and left edge to
>> right edge of the window, intersecting at the current mouse pointer
>> location.
```

```
>>
>
```

```
> Hi Lee. You might give this a try. Not guaranteed to be bullet proof :-)
> But it should get you going in the right direction. Courtesy of
> Mike 4 Mayer, PVI field Technical Sales Engineer.
```

```
>
> James K. Phillips      Sprocket Scientist    pvi!james@boulder.colorado.edu
> Precision Visuals, Inc.          (303) 530-9000
> 6260 Lookout Rd      Boulder - A quaint little town nestled between
> Boulder, CO 80301      the ROCKIES and REALITY.
```

I've modified BIGHAIR to optionally accept arrow keys from the keyboard (your escape sequences may vary...I'm using ESC[A, B, C, D). Also the blips from the continuous overplotting have been eliminated by selectively plotting only when the cursor position changes.

```
> ----8<-----CUT HERE ----8<-----CUT HERE-----
PRO BIGHAIR, x, y, keyboard=keyboard, accelerate=accelerate
;
; Procedure to get cursor location in current window,
```

```

; using cursor lines across the entire window.
; The X and Y cursor locations (in pixels) are returned.
; Further modification could allow Normalized or
; Data coordinates to be returned.
; /Keyboard accepts arrow keys input from keyboard, rather than mouse
; Accelerate = number of pixels to jump in "/keyboard" mode.
;
; Mike Mayer, 8/28/92
; Modified:
; Jim Pendleton, Dept. Physics & Astronomy
; Northwestern U.
; 8/29/92
;
; Usage:
; WAVE> BIGHAIR, x, y [, /Keyboard [, Accelerate=# pixels]]
;
DEVICE, Get_Graphics = oldmode
DEVICE, Set_Graphics = 6 ; XOR write mode.
xmax = !D.X_Vsize
ymax = !D.Y_Vsize
!Err = 0 ; Reset.
oldx = 0
oldy = 0
If (not Keyword_Set(Accelerate)) then Begin
  Accelerate = 1.
EndIf
If (not Keyword_Set(KeyBoard)) then Begin
  PRINT, 'Press left mouse button to quit...'
  CURSOR, oldx, oldy, /Device, /Change
EndIf Else Begin
  PRINT, 'Press space bar to quit...'
  oldx = xmax/2
  oldy = ymax/2
  x = oldx
  y = oldy
EndElse
PLOTS, [oldx, oldx], [0, ymax], /Device ; Draw first crosshairs.
PLOTS, [0, xmax], [oldy, oldy], /Device ; Draw first crosshairs.
EMPTY
Up = String(27B) + '[A'
Down = String(27B) + '[B'
Right = String(27B) + '[C'
Left = String(27B) + '[D'
Clear = Get_Kbrd(0)
Null = ""
WHILE !Err NE 1 DO BEGIN
  If (Keyword_Set(Keyboard)) then Begin
    A = String(Byte(Get_Kbrd(0)))

```

```

While ((A ne String(27B)) and (StrUpCase(A) ne ' ')) Do Begin
  A = String(Byte(Get_Kbrd(0)))
EndWhile
B = String(Byte(Get_Kbrd(0)))
C = String(Byte(Get_Kbrd(0)))
Key = A + B + C
EndIf Else Begin
  CURSOR, x, y, /Device, /Change
EndElse
PLOTS, [oldx, oldx], [0, ymax], /Device ; Erase last crosshairs.
PLOTS, [0, xmax], [oldy, oldy], /Device ; Erase last crosshairs.
If (Keyword_Set(Keyboard)) then Begin
  !Err = 0
  Case Key Of
Up : y = oldy + Accelerate
Down : y = oldy - Accelerate
Left : x = oldx - Accelerate
Right : x = oldx + Accelerate
Null : x = x
Else : !Err = 1
  EndCase
EndIf
If ((x ne oldx) or (y ne oldy)) then Begin
  PLOTS, [x, x], [0, ymax], /Device ; Draw new crosshairs.
  PLOTS, [0, xmax], [y, y], /Device ; Draw new crosshairs.
  oldx = x
  oldy = y
  EMPTY
EndIf
ENDWHILE
If (not Keyword_Set(Keyboard)) then Begin
;
;
; Already erased in Keyboard mode.
;
PLOTS, [oldx, oldx], [0, ymax], /Device ; Erase last crosshairs.
PLOTS, [0, xmax], [oldy, oldy], /Device ; Erase last crosshairs.
EndIf
Empty
DEVICE, Set_Graphics = oldmode
PRINT, 'X = ' + STRTRIM(x, 2), ' Y = ' + STRTRIM(y, 2)
END
> ----8<-----CUT HERE ----8<-----CUT HERE-----

```

Jim Pendleton, SysMgr, etc.
 GRO/OSSE, Dept. Physics & Astronomy
 Northwestern U.
 pendleton@ossenu.astro.nwu.edu
