Subject: Re: CURSOR skips a few beats :-(
Posted by Paul Van Delst[1] on Fri, 29 Jan 2010 17:58:09 GMT
View Forum Message <> Reply to Message

Gianguido Cianci wrote:
> On Jan 27, 4:29 pm, David Fanning <n...@dfanning.com> wrote:
>> Well, you can do shapes easily enough just by connecting
>> the dots. For example, in AnnotateWindow you can choose
>> a pencil cursor and draw whatever shape you like (if you
>> have any drawing talent, of course). You just can't get
>> ALL the points the cursor crosses over. I don't think
>> object graphics will help in this case, either. You will
>> have learned them for nothing. :-)
>>
>
> Right, you can join the dots... so when IDL plots the line between the
> dots, it calculates which pixels need to turn white. How do I get the
> coords of all those pixels?

Maybe you can reverse engineer Brensenham's algorithm?

See: http://en.wikipedia.org/wiki/Bresenham's_line_algorithm

cheers,

paulv

p.s. BTW, David Fanning is correct - you should make this a widget program.

>
> I have come up with a three-step linear interpolation that I do
> between each pair of points and it seems to be working (with 2 probs).
> here is a snippet:
>
> pro dlines
>
> f_xsize = 300
> f_ysize = 300
>
> map = bytarr(f_xsize, f_ysize,2)*0b
>
> FOR line = 0, 1 DO BEGIN
> cursor, d1, d2, /down, /device
> device, cursor_standard = 32
> !mouse.button=0
> x1 = 0 > d1 < f_xsize-1
> y1 = 0 > d2 < f_ysize-1
> WHILE (!MOUSE.button NE 4) DO BEGIN

```
>       plots, x1, y1, /device, ps = 3, color = fsc_color((['green',
>  'red'])[line])
>       map[x1,y1, line] = 1b
>       oldx = x1
>       oldy = y1
>       CURSOR, X1, Y1, /device,1
>       x1 = 0 > x1 < f_xsize-1
>       y1 = 0 > y1 < f_ysize-1
>
>       dx = abs(x1-oldx)
>       dy = abs(y1-oldy)
>       l = sqrt((dx)^2+(dy)^2)
>       IF ~keyword_set(nointerpolation) AND l GT sqrt(2) THEN BEGIN
>         IF dx EQ 0 THEN BEGIN             ; if I need to interpolate
>  a vertical segment
>           xx = indgen(dy)+min([y1, oldy]) ;new x's
>           yy = round(interpol([x1, oldx], [y1, oldy], xx))
>           map[yy,xx, replicate(line, n_elements(xx)) ] = 1b
>         ENDIF ELSE BEGIN                  ; all other orientations
>           xx = indgen(dx)+min([x1, oldx]) ;new x's
>           yy = round(interpol([y1, oldy], [x1, oldx], xx))
>           map[xx,yy, replicate(line, n_elements(xx)) ] = 1b
>
>           ;;need to do this for certain diagonals
>           IF dy NE 0 THEN BEGIN    xx = indgen(dy)+min([y1,
>  oldy]) ;new x's
>             yy = round(interpol([x1, oldx], [y1, oldy], xx))
>             map[yy,xx, replicate(line, n_elements(xx)) ] = 1b
>           ENDIF
>         ENDELSE
>       ENDIF
>     ENDWHILE
>     w = where(map[*,*,line] EQ 1b)
>     a = array_indices(map[*,*,line], w)
>     device, /cursor_crosshair
>  ENDFOR
>
>  END
>
>  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
>
```

> Problem #1: it is not very pretty, but I could live with that I
> suppose. Though I feel there must be a better way.
>
> Problem #2: when you (slowly!) move the mouse out of the left edge of
> the window the program crashes because x1= 0>x1<f_xsize-1 sets x1 to
> -1!!! And I can't figure that one out :-(
>

>
>> P.S. Even using the pencil tool in something like Photoshop
>> you see that if you move your pencil fast you get a straight
>> line, while if you move it slowly you can get a nice even
>> bend in the line. I think this is a function of your
>> medium (a computer) and not a function of your art skills.
>>
>
> Hmm... I am not sure where you're going with the above. I hope the
> first part of this reply and the code clarify my issues...
>
>
> Thanks,
> Gianguido
