
Subject: Re: Ongoing Object Graphics Quest
Posted by [Martin Downing](#) on Mon, 19 Nov 2001 16:48:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.1662c9a6c36c44e6989782@news.frii.com...
> Martin Downing (martin.downing@ntlworld.com) writes:
>
>> Oh I see - its actually the other way round. The drawn rubber band,
defined
>> as "theBox", does not draw round the top-right corner of the top right
>> pixel.
>> You could round this up like for instance:
>>
>> box_x0 = min([info.xs, info.xd])
>> box_x1 = max([info.xs, info.xd])+1
>> box_y0 = min([info.ys, info.yd])
>> box_y1 = max([info.ys, info.yd])+1
>>
>> box[0,*] = [box_x0,box_x1,box_x1,box_x0,box_x0]
>> box[1,*] = [box_y0,box_y0,box_y1,box_y1,box_y0]
>>
>> info.theBox->SetProperty, Data=box
>
> Actually, I think it is a little more pernicious
> than this. :-)
>
> The problem is that I get a location in the window,
> but what I want is an image subscript. If I have two
> locations in the window (e.g. a line), those endpoints
> are one pixel longer than the subscripts that I need.
>
> But here is the dilemma: which end of the line should
> I subtract one from (or add one to)? It depends on
> how you have drawn the line (e.g. right to left, or
> left to right) and whether one end of the line is
> on the image boundary. I think there must be at
> least eight possibilities you have to check to be
> able to get your box boundaries correct.
>

Hi David,

The 8 possibilities was why I used max() and min(). This modification was
only intended for the bounding box polygon of the selected area, not for
indexing the sub image.

I tried it in your code and the only problem was that sometimes the
top/right side was just outside the drawable view, other times it worked

fine.

Ok, I'll stop messing now!

Martin
