
Subject: Re: object graphics: any similar tools already written?

Posted by [jkj](#) on Sat, 26 May 2007 07:00:11 GMT

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

On May 25, 6:01 pm, Robbie <ret...@iinet.net.au> wrote:

- > Generally I don't feel comfortable with event loops which are as long
- > as yours. An idea I came up with was to write a "control" object for
- > each type of movement operation on the object graphic. For example,
- > you can add any model or scalable object graphic the 'zoom control'.
- > The control class takes care of capturing and sending events to
- > the ::event method. It also takes care of redrawing for you and will
- > do delayed redraws reduce the calls to the ::draw method.
- >
- > You could combine many controls to get the same effect as yours. The
- > advantage is that the same controls can be used for all relevant
- > object graphics.
- >
- > Source code for my control class is available from <http://barnett.id.au/idl/>
- > I have a personal library of my own controls, which are available on
- > request.
- >
- > I'm curious to know how someone else would use the control class. I
- > can't help but feel that my personal library of controls is biased to
- > my set of problems. I'm not sure that it's as generic as I'd like it
- > to be and someone else might have better ideas of how to implement the
- > same thing.
- >
- > Robbie

It sounds like you're saying there might be something useful to setting up each one of these as a separate control with the goal of incorporating various pieces into an application - sounds useful. In the end, I'm trying to avoid taking someone else's [i.e. Fanning's] code and building my application around it - nothing wrong with that, just personal determination to write something I know I understand completely - plus, I have a very clear idea of what I want in the end and am determined to figure it out.

I suspect I'll need to take a careful mathematical approach to this, but one gets used to the direct graphics way of plotting and only having to specify layouts in rather straightforward ways; with object graphics there is a lot more going on and usually I'm stuck with no understanding of why a line plot does or does not show up in the view. So I suspect some mathematics is the real solution, but exporting view properties is one more little exercise involving widgets and GUIs from scratch, so it feels like a productive learning experience.

I've been trying to get 1 image and 3 line plots laid out on a single draw widget which itself is contained inside of a widget_base with some buttons and stuff outside of the draw widget, but getting the lineplots to show in this context is still baffling to me (mostly using the "walking away from it" method of troubleshooting right now). It might be that I should use a separate draw widget for each and every image/plot rather than try to space several of them on one draw widget... something about placement of the plots within the draw widget and placement of the data within the plots that is escaping me for now - placing the plots is no problem, it's figuring out where the data is at within the view that baffles me - and some mathematics is probably going to be the cure.

Your page looks like a useful resource - thanks Robbie.
-Kevin
