Subject: Re: object graphics: any similar tools already written?
Posted by Robbie on Fri, 25 May 2007 23:01:49 GMT
View Forum Message <> Reply to Message

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

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 fromhttp://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

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

Posted by Robbie on Sat, 26 May 2007 12:00:40 GMT
View Forum Message <> Reply to Message

> I've been trying to get 1 image and 3 line plots laid out on a single
> draw widget
Perhaps you could simplify things by laying out 4 separate views on a
scene and rendering the IDLgrScene on you draw widget.

It's interesting how the lack of automatic layout of object graphics
is initially perceived as a weakness. Some authors of IDL packages try
to
automate layout by making wrapper classes to manage the object
graphics. I did this at first, however, now I leave layout as an
exercise for the caller. This gives the caller more chance to mix and
match layouts.

Actually , Ihave a separate class for managing and laying out views on
a scene. I use the /GRID_LAYOUT option on my atom tree (http://
barnett.id.au/idl/nmtkatomtree.html). For some instances I don't find
atom tree useful, so I defer to managing the atoms by hand.

Anyway, my OO jargon is probably not going to help you just yet. Have
you used IDL revolution (http://ittvis.com/codebank/search.asp?
FID=473) at all? Was it useful as a learning experience?

Robbie

Subject: Re: object graphics: any similar tools already written?
Posted by jkj on Sun, 27 May 2007 09:51:36 GMT
View Forum Message <> Reply to Message

On May 26, 7:00 am, Robbie <ret...@iinet.net.au> wrote:
> Perhaps you could simplify things by laying out 4 separate views on a
> scene and rendering the IDLgrScene on you draw widget.

Yea, throwing a scene into the mix was one of the last things I had
done with it... it'll be slow, tedious and ugly but I'll work my way
out of this paperbag by eventually find the correct chainsaw! :-)

>
> It's interesting how the lack of automatic layout of object graphics
> is initially perceived as a weakness.

Uh-huh... at least it has the feeling of being left out on a desert
without a roadmap... (...mumbling to self: why are pixels the default
units?)

> Actually , I have a separate class for managing and laying out views on
> a scene. I use the /GRID_LAYOUT option on my atom tree (http://
> barnett.id.au/idl/nmtkatomtree.html). For some instances I don't find
> atom tree useful, so I defer to managing the atoms by hand.

> Anyway, my OO jargon is probably not going to help you just yet. Have
> you used IDL revolution (http://ittvis.com/codebank/search.asp?
> FID=473) at all? Was it useful as a learning experience?

This is a code generator? It seems to require something newer than IDL 5.5, so I'll have to try this from the Institute. The goal/ determination on my part is to avoid relying on a code generation tool or using someone else's code as the basic building block (though that tool sounds like it could be instructive). I will possibly never fully "think OO" or embrace OO but I do see how your earlier references to control/layout classes is pretty cool - my code will end up being something that uses OO without looking OO (oooh-well!).

Thanks,
-Kevin