Subject: object graphics: any similar tools already written?
Posted by jkj on Fri, 25 May 2007 21:54:55 GMT
View Forum Message <> Reply to Message

All right, IDL object graphics is driving my dyslexia into orbit.
What I've done for now is put some code together that allows me to extend some of the view properties to a separate GUI so that I can dynamically see the effect changing viewplane_rect, location, and dimensions have on my image. I'll post a copy here:

safaripass.com/IDL/extend_view_controls.pro

in case anyone wants to try this crude but functional contribution.
I'm going to extend the idlgrplot properties as well but figured if I posted this someone might say "hey, this sort of thing has already been done - look 'here'" or "dude, your buttons are so crude - why not use 'this'" or whatever (my coding style is not changing, tough luck!).

Basically, to call this thing, create an idlgrwindow, a view of some sort which is drawn by that window and a view in which your image is displayed. My test code for this is:

safaripass.com/IDL/widget_object_test.pro

where I'm manipulating an object graphic within a widget_draw canvas.
This bit of code would work just as well:

```
pro small

  ydata  = indgen(520, /float)
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

  lplot  = obj_new('idlgrplot', ydata)
  view   = obj_new('idlgrview', viewplane_rect=[-200,-50,800,800])
  model  = obj_new('idlgrmodel')
  scene  = obj_new('idlgrscene')
  win    = obj_new('idlgrwindow', retain=2)
  model->add, lplot
  view->add, model
  scene->add, view
  win->draw, scene
  extend_view_controls, view, win, scene, $
                      title='testing only'

end ; end pro small
```

The ultimate goal here, for me, is to learn some keys to "safe

passage" through IDL's object graphics and interactive widgets. Once I learn a few safe, reliable paths from which complicated widget/ graphics can be built I won't care about the esoteric stuff. For now, however, I find myself unable to predict the visual effect when stringing one or more object graphic commands together. It does seem to me that although pixel units pops up as default for object graphics, IDL really, really intends for everything to be defined in terms of normal units. The referenced code uses pixel units, but I think I'm about to conclude that pixel units are not one of the keys to "safe passage".

Cheers,
-Kevin
