Subject: Re: Coyote graphics and resizeable draw widgets
Posted by David Fanning on Fri, 23 Mar 2012 14:18:37 GMT
View Forum Message <> Reply to Message

Wayne Landsman writes:

> Coyote graphics allows one to display plots in a resizeable graphics window
> (http://www.idlcoyote.com/graphics_tips/cgwindow.php). While I
> don't often need a resizeable window for a single plot, I do often
> want to have a resizeable draw window for a plot in a widget
> application, where I can have buttons controlling the features of the
> plot. Creating a resizeable draw widget is currently quite
> tedious (http://www.idlcoyote.com/widget_tips/resize_draw.html)
> but it is not clear to me that the process can be simplified using
> Coyote graphics. Or can it?

Well, of course, it can. :-)

After fooling around with cgWindow for some time, I realized that I wanted much of that functionality in another widget program. Then I realized that most of the functionality is already written as an object-widget, which is how I normally write resizeable widget windows. So, about a month ago I separated cgWindow from the underlying object-widget, which I call cgCmdWindow.

If you look at the cgWindow code now, you will see that it is nothing but a shell or "skin" to the underlying cgCmdWindow code. The idea is that you can write any kind of "skin" to the functionality that you like. In other words, you can wrap that window up in its own graphical user interface.

I haven't found a compelling reason to do this myself yet, but I have no doubt that it is absolutely going to work. ;-)

All you will have to do to add this to your widget program is pass the cgCmdWindow the ID of its parent widget. Then it's, well, done! Keep it's object reference around so you can call its methods. It is really as simple as that. :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Coyote graphics and resizeable draw widgets
Posted by David Fanning on Fri, 23 Mar 2012 14:34:29 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> All you will have to do to add this to your widget
> program is pass the cgCmdWindow the ID of its parent
> widget. Then it's, well, done! Keep it's object reference
> around so you can call its methods. It is really as
> simple as that. :-)

I should have probably said a little more about this.
At the time I separated cgWindow from the underlying
object-widget functionality, I added a few methods
to make working with the object-widget easier. In
particular, the PackageCommand method will package
an IDL command up into a cgWindow_Command object.
(This was the essential functionality of the old
cgWindow code.) Once you have the command object,
you simply add it to the window with either the
ReplaceCommand or AddCommand methods.

```
commandObj = windowObj -> PackageCommand('cgPlot', data)
windowObj -> AddCommand, commandObj
```

If your interface is resized, you determine what
size your graphics window should be, and you call
the Resize method with the new sizes.

```
windowObj -> Resize, event.x, event.y
```

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Coyote graphics and resizeable draw widgets
Posted by wlandsman on Fri, 23 Mar 2012 15:52:34 GMT
View Forum Message <> Reply to Message

On Friday, March 23, 2012 10:34:29 AM UTC-4, David Fanning wrote:

> If your interface is resized, you determine what
> size your graphics window should be, and you call
> the Resize method with the new sizes.
>
> windowObj -> Resize, event.x, event.y

Enlightenment comes slowly, Sensei.

I assume that I still need to compute the new size of the graphics window, i.e. if there are other buttons on the widget, I need to get their sizes and subtract them from the new size reported by TLB_GET_SIZE keyword for the resizing event.

Where cgcmdwindow does save me (a lot) is having to redraw the plot (and keep track of the data needed to redraw the plot) when the window is resized.

So instead of my current widget_draw() call, I create a cgcmdwindow object with the id of the parent widget. I currently use a single giant event handler, that distributes to the method name and object reference stored in a UVALUE. Its not yet clear to me how to store a UVALUE (that can be fetched with a GET_UVALUE) or even how to get the widget draw ID with the cgcmdwindow object, though there are things I can try.

Another thing that cgcmdwindow might help with are graphics overlays. For example, I want a circle to appear (only) at the position where the user has clicked with his cursor. Currently, I keep a separate pixmap window of the plot which gets copied back (to erase any old circles), and then a circle drawn at the cursor position. (The pixmap window also needs to be resized when the widget is resized.) But now I could use a DeleteCommand method for the old circle, and an AddCommand method for the new circle. (This might be slower since the plot would be redrawn rather than copied.)

Thanks, --Wayne

Subject: Re: Coyote graphics and resizeable draw widgets
Posted by David Fanning on Fri, 23 Mar 2012 16:34:07 GMT
View Forum Message <> Reply to Message

Wayne Landsman writes:

> Enlightenment comes slowly, Sensei.

Of course, my son. Life is messy. But imagine
if you had to learn the iTools framework. It would

take more than one lifetime! ;-)

> I assume that I still need to compute the new size
> of the graphics window, i.e. if there are other buttons
> on the widget, I need to get their sizes and subtract
> them from the new size reported by TLB_GET_SIZE
> keyword for the resizing event.

Yes, you will have to do this.

> I currently use a single giant event handler, that
> distributes to the method name and object reference
> stored in a UVALUE. Its not yet clear to me how to
> store a UVALUE (that can be fetched with a GET_UVALUE)
> or even how to get the widget draw ID with the
> cgcmdwindow object, though there are things I can try.

Well, I cobbled this together a little bit, so there
isn't a UVALUE, per se. Although there is a "storage"
pointer that can be used in the same way. Access it with
the STORAGE keyword to the Get/Set property methods.

> Another thing that cgcmdwindow might help with are
> graphics overlays. For example, I want a circle to
> appear (only) at the position where the user has
> clicked with his cursor. Currently, I keep a separate
> pixmap window of the plot which gets copied back (to erase
> any old circles), and then a circle drawn at the cursor
> position. (The pixmap window also needs to be resized
> when the widget is resized.) But now I could use a
> DeleteCommand method for the old circle, and an AddCommand
> method for the new circle. (This might be slower since
> the plot would be redrawn rather than copied.)

I think I would try the cgPixmapWindow object first. This
is a subclassed cgCmdWindow object that can be used as
a pixmap window. Methods allow fast copying between windows
and so forth.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Coyote graphics and resizeable draw widgets
Posted by David Fanning on Fri, 23 Mar 2012 16:47:40 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> I think I would try the cgPixmapWindow object first. This
> is a subclassed cgCmdWindow object that can be used as
> a pixmap window. Methods allow fast copying between windows
> and so forth.

I should point out that one of the HUGE advantages
of a Pixmap object is that if need be (and, believe
me, when you are doing pixmap smoke and mirror tricks
there is OFTEN need!) you can actually *see* the pixmap
window, so you can figure out what the heck is going on!

Simply set the visible property to see it:

```
pixmap -> SetProperty, Visible=1
```

And turn it off, when you don't want to see it:

```
pixmap -> SetProperty, Visible=0
```

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")