raval.chintan@gmail.com wrote:
> Karl Schultz wrote:
>> On Thu, 02 Feb 2006 00:01:57 -0800, raval.chintan wrote:
>>
>>> Dear All,
>>>
>>> I am using NVidia Quadro 4 980 XGL. One can visualize stereo images
>>> using Quad Buffer , available on this card. Proper interface
>>> (functions) for rendering images using this buffer are given in
>>> OpenGL.. Quad Buffer includes Left and Right Buffer ,which will be
>>> rendered one by one.For Visulization of images one needs, shutter
>>> glasses with infrared emitter to be connected with graphics card.
>>>
>>>
>>> Will it be possible to render an image with use of this buffer on IDL
>>> Draw widget? or in other way ,Will it be possible call this
>>> interface(functions) of OpenGL through IDL?
>> This is not supported in IDL 6.2, nor will it be in 6.3. We're getting
>> enough requests for it that it may be in a release soon.
>>
>> Although it is not documented or supported, you can make OpenGL calls from
>> C-language DLM's if you do things carefully. To do what you want, you'll
>> probably have to subclass IDLgrView with IDL code where you would override
>> the Draw method. You would also write a C DLM that calls glDrawBuffer
>> with GL_BACK_RIGHT.
>>
>> By the time IDLgrView::Draw is called, the correct GL context is in
>> place and IDL has already called glDrawBuffer with GL_BACK (implies
>> GL_LEFT as well). Your new Draw method would call the C DLM code to call
>> the superclass Draw method, call the C DLM code to set GL_BACK_RIGHT, and
>> then call the superclass Draw method again.
>
>
> If i am correct with my limited knowledge of IDL, IDLgrView class does
> not have the draw
> method. IDLgrWindow , IDLgrModel,IDLgrBuffer have draw method. Now the
> question comes that which class's draw method i should override.
> IDLgrWindow , IDLgrModel or IDLgrBuffer? Where will C dlm render the
> Left and right image? Will it be on IDL's window or on C's window?

All IDLgr* atoms have a draw method. When you call IDLgrWindow::Draw,
it in turn calls IDLgrView::Draw which in turn calls IDLgrModel::Draw
which in turn calls IDLgrPolygon::Draw and so on.

See for yourself. Create a subclass of IDLgrView and override the Draw
method like so:

```
pro MyView::Draw, destObject

    print, 'Well I know my name is Simon, ' $
       'and I like to do drawings...'
    self -> IDLgrView::Draw, destObject

end
```

Now every time you draw this view, it will print out some text.

> It would be helpful if you can provide sample code or detailed
> explaination of your suggested technique.
>

I *strongly* suggest that you pursue the method I suggested. The method
Karl outlined isn't for the faint of heart. Especially when there is an
alternative method that will be easier to implement. There *may* be
some performance benefits to using the Stereo API but the work involved
doesn't justify pursuit of that method without at least trying my
method. IMO, of course.

But if you really want to do this the hard way, Karl posted code that
demonstrates his technique of making openGL calls from a DLM on the user
contrib site. It is in the advanced visualization section, the file is
IDL_VGL.zip.

-Rick