
Subject: Re: Object Graphics

Posted by [Michael Wallace](#) on Wed, 12 May 2004 21:14:59 GMT

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

>> I've done limited work with OpenGL and some with Java3D (which is Java
>> wrappers above OpenGL), so I'm familiar with the concepts used in the
>> graphics. My problem is that I'm having quite a time trying to figure
>> out how IDL has "wrapped" the OpenGL underneath. The IDL -> OpenGL
>> mapping is what I meant by wanting to understand "what's going on inside
>> the object graphics system."

>

>

> I'm not sure I see the value in understanding the IDL -> OpenGL mapping in a
> general sense. If you have an understanding of openGL you can usually make
> the connections back to IDL fairly easily. If you don't have the experience
> with openGL then why bother making the connections? I'm certainly not
> saying it's worthless to do so, I just don't see the value.

In a general sense, there is no value. What I'm trying to understand is where IDL has hidden things. For example, earlier today I was looking for something that'd write the image in the current window or buffer to a file. After hunting and hunting without turning up anything, I started reading through all of the properties and methods one by one until I happened upon an IDLgrWindow method named Read() which returned and IDLgrImage. Then after reading through the documentation on IDLgrImage, I didn't find any methods to write the image to a file. I finally discovered a property called simply, DATA. I had read past this property several times since I had no clue what 'DATA' meant. What kind of data is it? What is it doing there? It turns out DATA contains the rgb values of the image. Obviously! It's so simple! Everyone should just automagically know that's what DATA is there for.

When it was all said and done, the code to write the image in the current window turned out to be pretty simple.

```
grImg = grWin -> Read()  
grImg -> GetProperty, DATA = data  
write_png, 'image.png', data
```

I see this kind of thing happening again and again. I say "I know how to do such-and-such in OpenGL. Now how to I do it in IDL?" Search, search, search, dig, dig, dig. I then say, "Eureka! The information I need is stored in a very well named property called DATA!"

It's really all the syntax and naming of things that's causing me problems.

> RSI intentionally blurs this relationship as they don't want to associate
> object graphics with a single graphics API. Ronn's book doesn't spend any

> time on this either. I think that if you want to know, you'll have to go
> from the OpenGL side back to IDL.

Yes, and I completely understand the approach. IDL should be simpler to work with and do a lot for you already. Otherwise there'd be no point.

I guess the problem is that I don't yet know what all the classes do, what all the properties are for and how the methods are used. What'd be great would be a cheat sheet of the how to do the basics such as save an image, create standard plots, multiplots, etc. And then a cheat sheet to get you started with the advanced visualization stuff.... I can't wait until I can draw that first 3-D mesh model of an asteroid! (I assume this is possible in IDL -- you can do it in OpenGL. ;-))

-Mike
