Subject: Re: object -> direct (colors)
Posted by davidf on Wed, 04 Nov 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Struan Gray (struan.gray@sljus.lu.se) writes:

> There has to be a better way....
>
> Take a look at the IDLgrVRML object. This lets you save a scene
> as a VRML file which can be used by other 3D drawing programs (and, I
> think, web browsers) to display the 3D graphics and, more importantly,
> print them without rasterisation.
^^

Now, Struan, do you know this to be the case or are you speculating?
It's not nice to get our hopes up, you know.

I've been spending too damn much time avoiding my real job,
so I'm going to leave it up to you to make sure this is
right. On the surface of it, I find it hard to believe.

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438, Toll-Free Book Orders: 1-888-461-0155
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Note: A copy of this article was e-mailed to the original poster.

Subject: Re: object -> direct (colors)
Posted by Struan Gray on Wed, 04 Nov 1998 08:00:00 GMT
View Forum Message <> Reply to Message

seanosea@my-dejanews.com writes:
> So, I've decided to go with the IDLgrBuffer idea--rendering
> to the buffer, then getting the rastered image with IMAGE_DATA
> keyword and tv'ing the image to PostScript by set_plot, 'ps'.

There has to be a better way....

Take a look at the IDLgrVRML object. This lets you save a scene
as a VRML file which can be used by other 3D drawing programs (and, I

think, web browsers) to display the 3D graphics and, more importantly, print them without rasterisation.

Not perfect, but good enough for a large proportion of government work.

Struan

Subject: Re: object -> direct (colors)
Posted by davidf on Wed, 04 Nov 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Sean O'Sea (seanosea@my-dejanews.com) is on the Path when he writes:

> Dia daoibh!
>
> So, I've decided to go with the IDLgrBuffer idea--rendering to the buffer,
> then getting the rastered image with IMAGE_DATA keyword and tv'ing the image
> to PostScript by set_plot, 'ps'.
>
> I have a new question about colors. IMAGE_DATA returns an array of dimension
> (3,xsize,ysize). The "3" is the color issue: if I make a black plot object
> and rasterize, all three color slots have the line information. If I make a
> red plot object, then (perhaps a little surprisingly) only (1,*,*) and
> (2,*,*) show a line, (0,*,*) having just background.
>
> So, I guess I need an i-j loop that considers each (*,i,j), assigning a number
> to a new array (i,j) based upon the three color slots? Then tv will have all
> the information in a single array, as it appears to like it.
>
> Am I right about this? or do I misunderstand?

Ah, I think you misunderstand, but only slightly. :-)

What you want to do is take that 3D image you get and write
it to your PostScript file like this:

```
Set_Plot, 'PS'
Device, Bits_Per_Pixel=8 ; (Plus whatever else. See web page.)
TV, niceObject24bitImage, True=1
```

> metaphysics is the job of cutting the beast of reality by the joints.

Damn straight! :-)

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438, Toll-Free Book Orders: 1-888-461-0155
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: object -> direct (colors)
Posted by thompson on Thu, 05 Nov 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Struan Gray <struan.gray@sljus.lu.se> writes:

> David Fanning, davidf@dfanning.com writes:

>>> print them without rasterisation.
>> ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
>> Now, Struan, do you know this to be the case or
>> are you speculating?

> PS to my previous post:

> I realise of course that all postscript is eventually rastered.
> What I - and I assume everyone else - wants is for that rasterisation
> to take place at the dot pitch of the printer and not at the dot pitch
> of the screen used to build the model.

I would most heartily agree with that statement, and since a single PostScript output could be printed on a wide variety of PostScript printers, that rasterization should take place within the printer (or printer driver). PostScript output also needs to be scalable to allow it to be incorporated into other documents without loss of resolution. As a secondary concern, I would also want to see PostScript output written in a diskspace-efficient way, because it's quite common for us to send PostScript files through the mail and serve them up on the internet.

William Thompson

Subject: Re: object -> direct (colors)
Posted by Struan Gray on Thu, 05 Nov 1998 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning, davidf@dfanning.com writes:

>> print them without rasterisation.
> ^^
> Now, Struan, do you know this to be the case or
> are you speculating?

PS to my previous post:

I realise of course that all postscript is eventually rastered.
What I - and I assume everyone else - wants is for that rasterisation
to take place at the dot pitch of the printer and not at the dot pitch
of the screen used to build the model.

Struan

Subject: Re: object -> direct (colors)
Posted by Struan Gray on Thu, 05 Nov 1998 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning, davidf@dfanning.com writes:
> Struan Gray (struan.gray@sljus.lu.se) writes:
>
>>   There has to be a better way....
>>
>>   Take a look at the IDLgrVRML object. This lets you save a scene
>> as a VRML file which can be used by other 3D drawing programs (and, I
>> think, web browsers) to display the 3D graphics and, more importantly,
>> print them without rasterisation.
> ^^
>
> Now, Struan, do you know this to be the case or are you speculating?

I'm semi-speculating. A combination of doctoral students (those
well-known founts of wisdom) with no known reason to want to see me
crash and burn in public assure me that this is possible. Mind you,
one of them has recently fled to Japan.

There are of course specialist CAD and VRML authoring packages
that read and write VRML files. Of the progams in regular use round
here, the Coreldraw 8 package includes a 3D modelling program that
reads and writes VRML files, and we currently get very nice colour
postscript output from our dye sub printer using 3D models constructed
in CorelDraw 7, so once we upgrade (and if the promises made in the
IDL and CorelDraw manuals are mostly correct) then what I outlined
should be possible.

That said, I haven't actually tested this, so caveat browseur. I am investing a decent amount of time in writing a crystal structure modelling program for a course that starts in December so you can expect keening ewails or aloof told-you-so's sometime before then.

> It's not nice to get our hopes up, you know.

All progress depends upon the irrational man.

Struan

Subject: Re: object -> direct (colors)
Posted by steinhh on Thu, 05 Nov 1998 08:00:00 GMT
View Forum Message <> Reply to Message

In article <71qe72$q5o$1@nnrp1.dejanews.com> seanosea@my-dejanews.com writes:
[..snip..]
> ruminative PostScript :)
> I don't know much about the relative utility of Object and Direct
> Gx--but Object certainly seems to me more conceptually elegant.
> (This won't convince anyone who just wants to get things *done*, of
> course.) Dividing more complex things into simpler things along their
> natural lines is a very fun thing to ponder, and to implement, and
> to use. Reminds me of an old saying by Plato: metaphysics is the job
> of cutting the beast of reality by the joints.

I do agree that for 3D graphics, OG seems more conceptually elegant - at least at first glance.

However, OG vs DG is really only about how things are *drawn*, not how things are grouped/stored. There's nothing's stopping you from doing DG with identical concepts. You could do DG using objects and object hierarchies, even using the OG object classes themselves.

The only thing that would be different is in how things get drawn - instead of using a destination object's draw method, you extract the idlgrview properties like the viewplane rectangle, projection and eye ("perspective" in DG) to build the initial T3D transformation. Then, for each object in the hierarchy (traverse it), find the 3D data points, find the CTM (current transformation matrix), for that object and plot it using a combination of the CTM and the initial T3D "viewing" transformation. Ok, there is some tinkering to be done, but that's what I'm hoping that RSI could provide as a compensation for not having *line* drawings at all in PostScript for OG.

Another thing is that object orientation is not an attribute of a programming language, but an attribute of the concepts that guide the programmer! An array of points representing an arc is just as much an "object" as an idlgrpolyline object describing the same object. That said, the syntactic features of a language can be *very* powerful in guiding the concepts of a programmer into a good OO style!

So, OG gives you really sharp knives for "cutting the beast of reality by the joints". Sharp knives, but unfortunately not so sharp *rendering* :-)

Regards,

Stein Vidar

Subject: Re: object -> direct (colors)
Posted by seanosea on Thu, 05 Nov 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Thanks, everybody. Fumbling forward...

Btw: I thought a bit about VRML earlier. I didn't know anything about it, so I browsed around without too much success. Even posted to the VRML newsgroup, asking if there was a way to raster images easily... but got no response.

Thank goodness for the Friendly IDL-pvwave People!
(What *is* pvwave, anyway? :>)

Sean O'Sea
