
Subject: Re: IDLgrClipboard and IDLgrPrinter: wrong vector output order ?

Posted by [davidf](#) on Tue, 23 May 2000 07:00:00 GMT

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

Nicolas Decoster (Nicolas.Decoster@Noveltis.fr) writes:

> The problem is that the IDLgrWindow::Draw method and the IDLgrClipboard
> one don't seem to use the same atomic object order to render the view.

Here is a slightly edited response from someone at RSI with respect to this issue. I thought I would see if anyone else has some thoughts on this. I'll collect them, if you like, and make sure they get to the right person at RSI. Or, you can send your comments directly to RSI. Remember to put how MUCH you like IDL in the first line of your e-mail. Response is greatly improved that way. :-)

Cheers,

David

> I think what we have here is a Z-buffer "tie" issue. The primitives
> are being drawn in the same order.
>
> In your example, the red poly is drawn first, then the green one. On
> the Window device, the red one overlaps the green one, so the Z buffer
> rule is that the pixels are not modified when there is a tie. This
> corresponds to the default OpenGL rule of GL_LESS, which means draw
> only if the fragment's Z is less (closer) than the Z currently in the
> frame buffer.
>
> The IDL vector-mode support essentially does not have any notion of a
> Z-buffer. However, it makes a reasonable effort to sort primitives by
> their Z values. You can get into problems with this, particularly
> when polygons overlap each other, or intersect, etc.. In other words,
> we just do a real simple Z-buffer sort to try to approximate things.
> I think that the 5.3 manuals discuss the fact that you won't get
> really good Z-ordering in complex scenes.
>
> Anyway, apparently, the Z-sort is implemented so that if there is a
> tie, the order that the primitives are drawn is retained. So, in this
> case, since both polys have the same Z value, the relative drawing
> order of these two polys is maintained and the green one is drawn
> second. Since there is no Z-buffer in vector mode, the green one ends
> up overlapping the red one.
>
> So, what is the RIGHT way to do this? I'm not sure it is clear. The
> argument to try to be consistent with raster Z buffer rules is a good

> one. But with vector output, people are used to thinking in terms of
> the "painter's algorithm" of drawing the thing you want on top last,
> so the intuitive notion of maintaining the order of prims that are at
> the same Z is also useful.
>
> Filing the bug with us is a good idea. But I'm not sure what the
> right approach is, so I'm interested in your thoughts on this as well,
> now that you've read the above discussion. There's still time to get
> a fix into 5.4. :-)

--

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