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

Posted by [Nicolas Decoster](#) on Wed, 24 May 2000 07:00:00 GMT

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

Hi.

kschultz@rsinc.com wrote:

>

> 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.

In fact, for postscript output it is more than "people are used to",
postscript `_do_` draw the last object on top. So, if I understand well
the Z-buffer thing, I think that to be as much as possible near this
Z-buffer rendering, you must do the following in vector mode for
postscript (and the like) output: if there are several objects at a same
Z value you output them in the postscript in the reverse order of the
graphic object tree. This way I think it will work for my very simple
square example. For more general situation, I am not enough experienced

with IDL.

Any comments ?

Anyway, I think that in a near future I will use the Mark Hadfield's suggestion: "code-control" of the order. While (for the moment) I only work with 2D graphics, I will use the Z value of my objects to implement a layered organisation of them.

Of course a fix into 5.4 will be nice too.

Later.

Nicolas Decoster.

--

Tél. : 00 (33) 5 62 88 11 16

Fax : 00 (33) 5 62 88 11 12

Nicolas.Decoster@Noveltis.fr

Noveltis

Parc Technologique du Canal

2, avenue de l'Europe

31520 Ramonville Saint Agne - France
