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

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:

> I am trying to build home tools to draw some kind of personnal graphics.
> Once I have my pretty drawing on a IDLgrView displaying on a
> IDLgrWindow, I want to print it or include it in document. I decided to
> use IDLgrClipboard with the vector keywords of the Draw method for
> various reasons.
>
> 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 little code to illustrate my words:
>
> You see what I mean?
>
> Well, the question is: bug or... "strange feature"?

Oh, that is interesting. :-)

I vote for a bug, for sure. Will you report this to the
good folks at RSI?

Cheers,

David

--
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

Subject: Re: IDLgrClipboard and IDLgrPrinter: wrong vector output order ?
Posted by Mark Hadfield on Wed, 24 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

"Nicolas Decoster" <Nicolas.Decoster@Noveltis.fr> wrote in message
news:392AA2DF.521D1AEE@Noveltis.fr...
> Hi.
>
> First of all:
> IDL> print, !version
> { sparc sunos unix 5.3 Nov 11 1999}
>
> I am trying to build home tools to draw some kind of personal graphics.
> Once I have my pretty drawing on a IDLgrView displaying on a
> IDLgrWindow, I want to print it or include it in document. I decided to
> use IDLgrClipboard with the vector keywords of the Draw method for
> various reasons.
>
> 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 little code to illustrate my words:
>
> --- begin of the little code ---
> ...
> --- end of the little code ---
>
> You see what I mean?

Well, no. On my system (x86 Win32 Windows 5.3.1 Feb 23 2000) the IDL window
& the Postscript file both look the same (red in front of green). At least
they do when the IDLgrWindow uses the hardware renderer (RENDERER=0), but if
I tell it to use the software renderer (RENDERER=1), the overlap is reversed
(green in front of red). I presume the latter is what you saw.

In my experience the only way to get reliable Z-ordering with Object
Graphics is to actually position atoms in the Z dimension, not to rely on
drawing order. The minimum offset required to ensure reliable overlap
control is approximately 1/65536 times the distance between the front and
rear clipping planes, e.g. see the following fragment from the IDLgrLegend

code:

```
oView->IDLgrView::GetProperty, ZCLIP = zClip
depthOffset = (double(zClip[0]) - double(zClip[1]))/65536.d
```

(I guess this means that OpenGL uses 16-bit integers internally?)

Actually I suggest increasing the depth offset for overlap control to twice
the above value, because this seems to be required with some combinations of
atoms on some output devices.

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield/
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand