Subject: IDLgrClipboard and IDLgrPrinter: wrong vector output order ?
Posted by Nicolas Decoster on Tue, 23 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

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 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:

--- begin of the little code ---

```
myView = obj_new('IDLgrView', dimensions = [5, 5], viewplane_rect = [0,
0, 5, 5], units = 2)
myModel = obj_new('IDLgrModel')
myView->Add, myModel

poly1 = obj_new('IDLgrPolygon', [[1, 1], [1, 3], [3, 3], [3, 1]], color
= [255, 0, 0])
myModel->Add, poly1

poly2 = obj_new('IDLgrPolygon', [[2, 2], [2, 4], [4, 4], [4, 2]], color
= [0, 255, 0])
myModel->Add, poly2

myWindow = obj_new('IDLgrWindow', dimensions = [5, 5], units = 2)
myWindow->Draw, myView

myClipboard = obj_new('IDLgrClipboard', dimensions = [5, 5], units = 2)
myClipboard->Draw, myView, filename = 'gah.eps', vector = 1, postscript
= 1
```

--- end of the little code ---

You see what I mean?

Well, the question is: bug or... "strange feature"?

hum ?

Later.

Nicolas.

--
Tï¿½l. : 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

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.

--
Tï¿½l. : 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