Subject: direct graphics or object graphics ?
Posted by natha on Tue, 06 May 2008 19:59:28 GMT
View Forum Message <> Reply to Message

Hi people,

I'm a student programming a software for analysis and visualisation of
some kinds of data. In my interface I want to show some plots.
I have an object for do OPLOTs using object graphics but I dont know
if is better to use object graphics when I'll try to plot more than
1000 plots.

I think programming in object graphics is better than direct graphics
but Im not sure.

Any opinions ????

Thanks,

Bernat

Subject: Re: direct graphics or object graphics ?
Posted by David Fanning on Tue, 06 May 2008 20:16:37 GMT
View Forum Message <> Reply to Message

Brian Larsen writes:

> that isn't exactly a resounding endorsement of object graphics :)

Really!? I guess I was remembering back to the time when I
was an object graphics newbie. :-)

Cheers,

David

P.S. I can be a *lot* more excited about object graphics
if we are talking about doing something in 3D space!

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming (www.dfanning.com)
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: direct graphics or object graphics ?
Posted by natha on Tue, 06 May 2008 20:24:24 GMT
View Forum Message <> Reply to Message

But,

There are any problems with virtual memory when I've more than 1000
IDLgrPlots and I want to draw my View?

Subject: Re: direct graphics or object graphics ?
Posted by David Fanning on Tue, 06 May 2008 20:32:32 GMT
View Forum Message <> Reply to Message

nata writes:

> There are any problems with virtual memory when I've more than 1000
> IDLgrPlots and I want to draw my View?

You want to draw 1000 line plots at the same time!
How big is your display? Are you displaying this
on the side of the MGM Grand in Las Vegas?

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming (www.dfanning.com)
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: direct graphics or object graphics ?
Posted by Jean H. on Tue, 06 May 2008 20:39:43 GMT
View Forum Message <> Reply to Message

nata wrote:
> But,
>
> There are any problems with virtual memory when I've more than 1000
> IDLgrPlots and I want to draw my View?
>

don't you want to draw one after another, and to save your plots as
image file? ... or at least to close them after viewing them? ... I
doubt anyone has a brain large enough to capture, remember, analyze and
correlate to your original data that many plots at once!

Jean

Subject: Re: direct graphics or object graphics ?
Posted by natha on Tue, 06 May 2008 20:40:47 GMT
View Forum Message <> Reply to Message

On May 6, 4:32 pm, David Fanning <n...@dfanning.com> wrote:
> nata writes:
>> There are any problems with virtual memory when I've more than 1000
>> IDLgrPlots and I want to draw my View?
>
> You want to draw 1000 line plots at the same time!
> How big is your display? Are you displaying this
> on the side of the MGM Grand in Las Vegas?
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming (www.dfanning.com)
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

jajajaj, no, no

In Direct Graphics you can do that:

plot, x, y, xrange=....
for i=0, 1000 do oplot, datax[i,*], datay[i,*]

I want to the same in object graphics and I dont know what is the best
way. I'll try to create 1000 IDLgrPlots for each 'oplot' and add this
in the oModel object. Is this a good idea?

Subject: Re: direct graphics or object graphics ?
Posted by David Fanning on Tue, 06 May 2008 20:47:06 GMT
View Forum Message <> Reply to Message

nata writes:

> In Direct Graphics you can do that:
>

> plot, x, y, xrange=....
> for i=0, 1000 do oplot, datax[i,*], datay[i,*]

Well, you *can* do that. The real question is,
Do you *want* to!?

Have you tried it? What does it look like (other
than a mess, of course)?

> I want to the same in object graphics and I dont know what is the best
> way. I'll try to create 1000 IDLgrPlots for each 'oplot' and add this
> in the oModel object. Is this a good idea?

I'm not sure this will look a whole lot better in
object graphics, even if it worked. Why do you want
to do this in object graphics?

Creating thousands of objects in often NOT a good idea
in object graphics. But until I get my head around why in
the world someone would want to do this, I can't really
thing about how I would implement the mess. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming (www.dfanning.com)
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: direct graphics or object graphics ?
Posted by natha on Tue, 06 May 2008 20:53:11 GMT
View Forum Message <> Reply to Message

On May 6, 4:39 pm, Jean H <jghas...@DELTHIS.ucalgary.ANDTHIS.ca>
wrote:
> nata wrote:
>> But,
>
>> There are any problems with virtual memory when I've more than 1000
>> IDLgrPlots and I want to draw my View?
>
> don't you want to draw one after another, and to save your plots as
> image file? ... or at least to close them after viewing them? ... I
> doubt anyone has a brain large enough to capture, remember, analyze and

> correlate to your original data that many plots at once!
>
> Jean

This is not the problem,

Drawing all the plots is a good idea for analyse the data.
The problem is the virtual memory used for draw all of these in the
same IDLgrWindow...

In Direct Graphics the memory used for OPLOT data is "almost nothing".
If I try the same using object graphics the time for draw the View
increases very much.

Subject: Re: direct graphics or object graphics ?
Posted by David Fanning on Tue, 06 May 2008 20:59:19 GMT
View Forum Message <> Reply to Message

nata writes:

> This is not the problem,
>
> Drawing all the plots is a good idea for analyse the data.
> The problem is the virtual memory used for draw all of these in the
> same IDLgrWindow...
>
> In Direct Graphics the memory used for OPLOT data is "almost nothing".
> If I try the same using object graphics the time for draw the View
> increases very much.

I think you choice is becoming clearer...

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming (www.dfanning.com)
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: direct graphics or object graphics ?
Posted by natha on Tue, 06 May 2008 21:08:02 GMT
View Forum Message <> Reply to Message

I used to create my interfaces with Object Graphics.

For this example I combined Direct Graphics and Object Graphics for draw all the plots..
Then I realized my software doesn't work in Windows (widget draws are not visibles). The reason: is not a good idea to combine Direct Graphics and Object Graphics.

Then I thought that will be a good idea representate all the analisys in Object Graphics but well, the time for draw 1000 IDLgrPlots is too much.

Representate all the data in Direct Graphics will be the best way.
Thanks for the discussion !

Cheers,

Bernat

Subject: Re: direct graphics or object graphics ?
Posted by R.G. Stockwell on Tue, 06 May 2008 21:10:16 GMT
View Forum Message <> Reply to Message

"nata" <bernat.puigdomenech@gmail.com> wrote in message
news:9fdc4a77-c6a7-421d-9bc0-092386840790@b64g2000hsa.google groups.com...
> Hi people,
...
>
> I think programming in object graphics is better than direct graphics
> but Im not sure.
>
> Any opinions ????

My two cents:

If you want an output file for publications - do DG and postscript.
(amazingly this is what I almost exclusively do)

If you want interactive plots - with a user changing the view etc,
(especially with surface plots) then use OG.

It sounds like you want to just look at a batch of data up on the screen,
I'd go with a quick dirty DG.

Cheers,
bob

Subject: Re: direct graphics or object graphics ?
Posted by Spon on Wed, 07 May 2008 10:53:34 GMT
View Forum Message <> Reply to Message

Bernat wrote:
> for i=0, 1000 do oplot, datax[i,*], datay[i,*]

David Fanning wrote:
> until I get my head around why in
> the world someone would want to do this, I can't really
> thing about how I would implement the mess. :-)

Jean H wrote:
>> ... I
>> doubt anyone has a brain large enough to capture, remember, analyze and
>> correlate to your original data that many plots at once!
>> Jean

Bernat writes:
> This is not the problem,
> Drawing all the plots is a good idea for analyse the data.

Something tells me there are several people you've yet to convince of this ;-) (myself included). Just out of curiosity, what are you trying to visualise in this way? I'd be interested to know how sparsely scattered a dataset would need to be for this method to give you something other than a headache. :-)

Regards,
Chris

Subject: Re: direct graphics or object graphics ?
Posted by Mark[1] on Wed, 07 May 2008 22:41:49 GMT
View Forum Message <> Reply to Message

On May 7, 10:53 pm, Spon <christoph.b...@gmail.com> wrote:
> Bernat wrote:
>> for i=0, 1000 do oplot, datax[i,*], datay[i,*]
>> Drawing all the plots is a good idea for analyse the data.
>
> Something tells me there are several people you've yet to convince of
> this ;-) (myself included). Just out of curiosity, what are you trying
> to visualise in this way? I'd be interested to know how sparsely
> scattered a dataset would need to be for this method to give you
> something other than a headache. :-)

I'm with Bernat: even with large datasets, drawing all the data as

line plots can be useful as a quick way of distinguishing sparsely
populated regions of the parameter space from densely populated
regions, and for detecting outliers.

I recall that back in 1994 or so I used IDL to plot several time
series of air pollution data, with ~ 10^6 points per time series, and
saw stuff in there that others had not seen, because they were stuck
with puny spreadsheet-based plotting tools. ... That would have been
Direct Graphics, BTW :-)

Subject: Re: direct graphics or object graphics ?
Posted by David Fanning on Wed, 07 May 2008 22:48:45 GMT
View Forum Message <> Reply to Message

Mark writes:

> I'm with Bernat: even with large datasets, drawing all the data as
> line plots can be useful as a quick way of distinguishing sparsely
> populated regions of the parameter space from densely populated
> regions, and for detecting outliers.
>
> I recall that back in 1994 or so I used IDL to plot several time
> series of air pollution data, with ~ 10^6 points per time series, and
> saw stuff in there that others had not seen, because they were stuck
> with puny spreadsheet-based plotting tools. ... That would have been
> Direct Graphics, BTW :-)

I suppose with object graphics you could look at this
traffic accident in 3D space, too, and notice how high
the stack was growing as you piled one thing on top of
another. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming (www.dfanning.com)
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: direct graphics or object graphics ?
Posted by wtt on Sat, 14 Jun 2008 12:32:54 GMT
View Forum Message <> Reply to Message

In article <MPG.228a6a87bee296919896f9@news.frii.com>,
 David Fanning <news@dfanning.com> wrote:

> Brian Larsen writes:
>
>> that isn't exactly a resounding endorsement of object graphics :)
>
> Really!? I guess I was remembering back to the time when I
> was an object graphics newbie. :-)

I am sort of an object graphics newbie. I think it works really well, and I would recommend it (in a newbieish sort of way). One thing that I've noticed is that if I'm rendering a very large view, the IDL process's memory size can grow to be enormous (upwards of 2-3 GB sometimes).

On the machine I'm rendering on, this wouldn't be a problem except that IDL does not seem to relinquish that memory after the view has been rendered and the objects destroyed. Its only after the IDL process quits that the memory is recovered by the OS. This seems contrary to the way IDL manages its heap.

So I was thinking that this memory is overflow from the videocard's ram and is allocated by the operating system independent of IDL, and so it is not under IDL's direct control. Does that seem reasonable?

Thanks,
Bill

>
> Cheers,
>
> David
>
> P.S. I can be a *lot* more excited about object graphics
> if we are talking about doing something in 3D space!

Subject: Re: direct graphics or object graphics ?
Posted by David Fanning on Sat, 14 Jun 2008 12:43:23 GMT
View Forum Message <> Reply to Message

Bill Triplett writes:

> On the machine I'm rendering on, this wouldn't be a problem except that
> IDL does not seem to relinquish that memory after the view has been
> rendered and the objects destroyed. Its only after the IDL process quits

> that the memory is recovered by the OS. This seems contrary to the way
> IDL manages its heap.

It is contrary to the way IDL manages its heap, but the
heap is already in process memory anyway. But not
relinquishing memory is standard operating procedure, as
far as I know, for any program written in C and using
malloc and free to allocate and free memory.

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: direct graphics or object graphics ?
Posted by Craig Markwardt on Mon, 16 Jun 2008 14:06:59 GMT
View Forum Message <> Reply to Message

David Fanning <news@dfanning.com> writes:

> Bill Triplett writes:
>
>> On the machine I'm rendering on, this wouldn't be a problem except that
>> IDL does not seem to relinquish that memory after the view has been
>> rendered and the objects destroyed. Its only after the IDL process quits
>> that the memory is recovered by the OS. This seems contrary to the way
>> IDL manages its heap.
>
> It is contrary to the way IDL manages its heap, but the
> heap is already in process memory anyway. But not
> relinquishing memory is standard operating procedure, as
> far as I know, for any program written in C and using
> malloc and free to allocate and free memory.

David, I believe that most modern malloc()/free() implementations
can relinquish memory back to the system -- at least under Unix --
but it may not always be obvious when it can and when it can't.

The most obvious example would be a function with no side effects,
that temporarily uses a lot of memory. During the function call, some
additional memory is allocated from the top of the heap, and after the
function is finished, the heap can be reset to its original size.

However, such behavior is highly dependent on your system and your call pattern, so it will not *always* hold. (i.e. it's the memory fragmentation problem)

Craig
