Subject: Re: IDL vs Direct3D, OpenGL
Posted by Michael Wallace on Tue, 13 Jan 2004 20:41:27 GMT
View Forum Message <> Reply to Message

IDLUser wrote:
> Hello,
> I am just learning IDL and I am considering to write 3D visualization
> code with IDL. Before I start to do that, I want to know how IDL is
> good for my purpose. Does IDL have enough functionality as Direct3D or
> OpenGL has? What is the bottom line of IDL for object graphics? If it
> does, what is the good way to learn about object graphics in IDL?
> Thanks.

It depends on what you're trying to accomplish. OpenGL and IDL are completely different programs. IDL is built around display and analysis of data, while OpenGL is an API for creating 2D and 3D graphics. If I were going to create a flight simulator program, I'd use OpenGL. But if I was going to do analysis of stress on an airplane wing, I'd use IDL. OpenGL doesn't have the scientific analysis capabilities of IDL and IDL doesn't have the graphics constructs of OpenGL. So, it could be one or the other or both depending on what your visualization is.

Mike

Subject: Re: IDL vs Direct3D, OpenGL
Posted by Rick Towler on Wed, 14 Jan 2004 18:40:01 GMT
View Forum Message <> Reply to Message

"IDLUser" wrote in message ...

> I am just learning IDL and I am considering to write 3D visualization
> code with IDL. Before I start to do that, I want to know how IDL is
> good for my purpose. Does IDL have enough functionality as Direct3D or
> OpenGL has? What is the bottom line of IDL for object graphics? If it
> does, what is the good way to learn about object graphics in IDL?
> Thanks.

I second Michael's comments.

Further, consider the time you can justify investing in your viz application. IDL allows you to rapidly build applications by handling the more mundane aspects of application programming. IDL object graphics uses OpenGL as it's 3d API so given the appropriate hardware it performs quite well. But, RSI does limit what is exposed to the IDL programmer (for x-platform compatibility and to limit IDL's reliance on a single 3d API) so many of the better/cooler/would be nice to have features of OpenGL are not available in IDL. (Karl, you really need to consider exposing *at least* the

surface properties. The reasons to withhold them don't stand anymore.)

With C++ and OpenGL or DirectX you can be sure that you will spend much more time developing your application. In return for your investment in time, you will most likely see increased application performance and full access to the 3d API. Depending on your approach, your application may not be very portable which is in contrast to IDL.

It would be safe to say that you will be able to at least prototype your visualization application in IDL. If you find IDL too restrictive you can take what you have learned and continue development in another, more suited language.

FWIW, I have pushed a lot of polys thru IDL's object graphics system and I am still using it. A few tips to get the most out of IDL object graphics:

Use the Win32 version of IDL.

Buy the best "consumer grade" video adapter you can afford. nVidia historically has had the best openGL drivers. ATI is working hard to change this but I can't tell you how far they have come.

Buy the fastest x86 processor you can afford running on the fastest bus that processor architecture supports.

Consider Ronn Kling's "Power Graphics with IDL" a beginner's guide to object graphics. It is available from his website www.kilvarock.com. AFAIK it is the only IDL object graphics book that is currently available.

"Advanced animation and rendering techniques" by Watt and Watt is another general resource. It is a little old now but the basics still apply. There are many more. Search your library.

-Rick

Subject: Re: IDL vs Direct3D, OpenGL
Posted by Matt Feinstein on Wed, 14 Jan 2004 19:32:46 GMT
View Forum Message <> Reply to Message

On 13 Jan 2004 10:44:51 -0800, bbhyun2001@yahoo.com (IDLUser) wrote:

> Hello,
> I am just learning IDL and I am considering to write 3D visualization
> code with IDL. Before I start to do that, I want to know how IDL is
> good for my purpose. Does IDL have enough functionality as Direct3D or
> OpenGL has? What is the bottom line of IDL for object graphics? If it
> does, what is the good way to learn about object graphics in IDL?

> Thanks.

It depends on what you mean by 'visualization code'. If visualization
code means comparing and combining visualization of different kinds of
image, mapping, and 3D data with a not-too-complicated GUI, then IDL
is an excellent tool.

On the other hand, if the visualization code just takes in a specific
kind of 3D 'world coordinate' data and renders it at a high frame
rate, you would be better off working directly with a lower level API
such as OpenGL or Direct3D.

Matt Feinstein

--
There is no virtue in believing something that can be proved to be true.

Subject: Re: IDL vs Direct3D, OpenGL
Posted by Mark Hadfield on Wed, 14 Jan 2004 20:38:26 GMT
View Forum Message <> Reply to Message

Matt Feinstein wrote:
> On 13 Jan 2004 10:44:51 -0800, bbhyun2001@yahoo.com (IDLUser) wrote:
>
>
>> Hello,
>> I am just learning IDL and I am considering to write 3D visualization
>> code with IDL. Before I start to do that, I want to know how IDL is
>> good for my purpose. Does IDL have enough functionality as Direct3D or
>> OpenGL has? What is the bottom line of IDL for object graphics? If it
>> does, what is the good way to learn about object graphics in IDL?
>> Thanks.
>
> It depends on what you mean by 'visualization code'. If visualization
> code means comparing and combining visualization of different kinds of
> image, mapping, and 3D data with a not-too-complicated GUI, then IDL
> is an excellent tool.
>
> On the other hand, if the visualization code just takes in a specific
> kind of 3D 'world coordinate' data and renders it at a high frame
> rate, you would be better off working directly with a lower level API
> such as OpenGL or Direct3D.

In this latter case, you might consider looking at VTK:

http://www.kitware.com/vtk/index.html

VTK is an open-source, 3D computer graphics package, somewhat higher level than OpenGL or Direct3D, but not a scientific & numerical analysis system like IDL. It is a C++ class library, but can invoked via various interface layers in Java, Tcl/Tk & Python. As an example of VTK's capabilities, there's a very nice data visualiser, using the Python-VTK interface, called MayaVi:

http://mayavi.sourceforge.net/

PV-Wave, a cousin of IDL, uses VTK for its 3D graphics.

--
Mark Hadfield "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: IDL vs Direct3D, OpenGL
Posted by dcw_yip on Wed, 14 Jan 2004 22:49:18 GMT
View Forum Message <> Reply to Message

"Rick Towler" <rtowler@u.washington.edu> wrote in message
news:<bu42jc$bm4$1@nntp6.u.washington.edu>...
> Further, consider the time you can justify investing in your viz
> application. IDL allows you to rapidly build applications by handling the
>

I'll have to disagree with this. It takes me far longer, far longer, to develop an application in IDL than it would in straight C++ and OpenGL. A good C++ compiler catches a lot of errors that IDL's "compiler" doesn't. Many of these errors, such as undefined variables, only show up at runtime. In a complex application it takes forever to validate the code again after a fairly minor change. If performance is a concern at all. Don't use IDL. It's not exactly speedy.

Also, IDLDE's constant crashing doesn't help the development process.

> Buy the best "consumer grade" video adapter you can afford. nVidia
> historically has had the best openGL drivers. ATI is working hard to change
> this but I can't tell you how far they have come.
>

Ahh.. I would buy the best video adapter you can that will run with IDL. I have one of the best adapters available, it's an nVidia by the way, and the driver crashes under IDL. Just with IDL, it doesn't seem to crash with any of the the other apps I've tried. I'm forced to run

in software mode now. Make sure you can exchange the graphics card if
it's not compatible with IDL.

Subject: Re: IDL vs Direct3D, OpenGL
Posted by David Fanning on Wed, 14 Jan 2004 23:35:55 GMT
View Forum Message <> Reply to Message

David Yip writes:

> I'll have to disagree with this. It takes me far longer, far longer,
> to develop an application in IDL than it would in straight C++ and
> OpenGL. A good C++ compiler catches a lot of errors that IDL's
> "compiler" doesn't. Many of these errors, such as undefined
> variables, only show up at runtime.

Undefined variables. In software code!? Oh, dear. :-(
I would think a typing course might help more than
a better compiler. :-)

> In a complex application it takes
> forever to validate the code again after a fairly minor change.

Say what!?

> If performance is a concern at all. Don't use IDL. It's not exactly
> speedy.

Oh, come on.

> Also, IDLDE's constant crashing doesn't help the development process.

My goodness. Has your warranty run out? I really think you
should take this back. Whatever you got was NOT the IDL
I work with.

> I would buy the best video adapter you can that will run with
> IDL. I have one of the best adapters available, it's an nVidia by the
> way, and the driver crashes under IDL. Just with IDL, it doesn't seem
> to crash with any of the the other apps I've tried. I'm forced to run
> in software mode now. Make sure you can exchange the graphics card if
> it's not compatible with IDL.

I have to say, my nVidia GForce 4 graphics card has been terrific.
I don't think I have ever crashed the IDLDE for *any* reason, at
least not in memory.

Cheers,

David

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

Subject: Re: IDL vs Direct3D, OpenGL
Posted by Rick Towler on Thu, 15 Jan 2004 02:16:07 GMT
View Forum Message <> Reply to Message

"David Yip" wrote in message...
> "Rick Towler" wrote in message ...
>> Further, consider the time you can justify investing
>> in your viz application. IDL allows you to rapidly
>> build applications by handling the...
>>
>
> I'll have to disagree with this. It takes me far
> longer, far longer, to develop an application in IDL
> than it would in straight C++ and OpenGL. A good
> C++ compiler catches a lot of errors that IDL's
> "compiler" doesn't. Many of these errors, such as
> undefined variables, only show up at runtime.
> In a complex application it takes forever to validate
> the code again after a fairly minor change.

While I certainly respect your stance, I think that for the majority of people creating scientific 3d visualizations in IDL will be far easier than doing it in C++ and OpenGL.

It isn't as simple as throwing up some polygons on the screen. IDL has a relatively rich functional library that would be difficult to put together when building your C++ application. While simply decoupling the visualization from the data processing and analysis would mitigate this to a certain extent, you will lose the ability to interactively work with the data which is often an important step in understanding data.

> If performance is a concern at all. Don't use IDL.
> It's not exactly speedy.

Maybe I didn't make it clear in my original post but I didn't mean to present IDL as the performance king. I tried to stress that if performance

was a key concern that C++ and OpenGL would be the way to go. But performance is only one parameter in an often complicated model.

As far as performance is concerned I can render a ~140k triangle scene at ~11 frames/second on a 1.7 GHz machine with a Geforce 3 in IDL. While I can't do an exact comparison with openGL I do have an application that displays DEMs written in C++/openGL and it rendered an ~140k triangle scene at ~16 frames/second. A sizeable difference in speed for sure but is it enough to justify dumping IDL?

> Also, IDLDE's constant crashing doesn't help the
> development process.

True IDLDE for unix is unusable, IDLDE for windows has been nothing but rock solid for me on windows 98 thru windows 2000. The only time IDLDE has crashed in recent memory is when external code behaves badly. I would suspect that this would be the case for 99% of windows users.

I don't know what your issues are, but they are most likely not related to IDLDE.

>> Buy the best "consumer grade" video adapter you
>> can afford. nVidia historically has had the best
>> openGL drivers. ATI is working hard to change
>> this but I can't tell you how far they have come.
>
> Ahh.. I would buy the best video adapter you can
> that will run with IDL. I have one of the best
> adapters available, it's an nVidia by the way, and
> the driver crashes under IDL. Just with IDL, it
> doesn't seem to crash with any of the the other
> apps I've tried. I'm forced to run in software mode
> now. Make sure you can exchange the graphics card if
> it's not compatible with IDL.

To second Fanning's comments, I have run IDL on nVidia based adapters from the original TNT thru the Geforce4 and except for one issue with anti-aliasing support while using a beta driver I have had no problems. (O.K. I have to take that back a bit. I run into problems when my polygon count skyrockets to say 9,000,000+ triangles. nvsys.dll takes down the entire machine. I'm willing to cut it some slack in this case)

I haven't run any of their Quadro based cards which I believe is what you are running (from a previous post). While I can't comment on your specific case, I know that David F. has used a Quadro based adapter in the past and he certainly didn't note any issues.

Not to get too far off topic here but I am suspicious of your machine as you clearly have either an OS or hardware problem. This issue seems to be at the heart of your disdain of IDL. You will most likely reply that IDL is the only application that is causing problems and that your machine is rock solid otherwise. My only response would be that the IDL that you run and the IDL that I run are the exact same code (that is if you run 5.5 thru 6.0) and that I have no problems worthy of noting. Clearly your issue lies elsewhere and instead of disparaging IDL, you could consider how much time you can justify investing in repairing your PC so that IDL runs happily vs. the amount of time you can invest in doing all of your work in C++. But you probably have already considered this. And you are still running IDL...

;)

-Rick

Subject: Re: IDL vs Direct3D, OpenGL
Posted by Michael Wallace on Thu, 15 Jan 2004 17:06:39 GMT
View Forum Message <> Reply to Message

>> Also, IDLDE's constant crashing doesn't help the
>> development process.
>
>
> True IDLDE for unix is unusable, IDLDE for windows has been nothing but rock
> solid for me on windows 98 thru windows 2000. The only time IDLDE has
> crashed in recent memory is when external code behaves badly. I would
> suspect that this would be the case for 99% of windows users.

I don't mean to take things off-subject here, but does RSI have any plans to make a decent IDLDE for Linux/Unix? Or will they always leave it in that half-backed, unusable and ugly state? Anyway, there are many reasons while my primary development consists of gvim and command-line IDL.

Mike

Subject: Re: IDL vs Direct3D, OpenGL
Posted by dcw_yip on Fri, 16 Jan 2004 02:07:50 GMT
View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> wrote in message news:<MPG.1a6f8044e81d72cc9897a1@news.frii.com>...
> David Yip writes:

>
>> I would buy the best video adapter you can that will run with
>> IDL. I have one of the best adapters available, it's an nVidia by the
>> way, and the driver crashes under IDL. Just with IDL, it doesn't seem
>> to crash with any of the the other apps I've tried. I'm forced to run
>> in software mode now. Make sure you can exchange the graphics card if
>> it's not compatible with IDL.
>
> I have to say, my nVidia GForce 4 graphics card has been terrific.
> I don't think I have ever crashed the IDLDE for *any* reason, at
> least not in memory.
>

As I was saying. Get the best video adapter you can that will run with IDL. A Geforce 4 is hardly high end. It's old enough that IDL must have caught up with it. Try using Quadro FX boards and tell me about IDL's stablity.

IDL may be better for non programmers but if you are a programmer, it's your worse nightmare. In many ways, it's the incarnation of all the things you shouldn't do in a good programming language.
