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
