
Subject: Re: IDL vs Direct3D, OpenGL
Posted by [rkneusel](#) on Fri, 16 Jan 2004 23:05:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

dcw_yip@yahoo.com (David Yip) wrote in message
news:<201431cc.0401151807.29395496@posting.google.com>...

> 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.

I'd like to think of myself as a programmer, and perhaps I have the
credentials to back it up, and I certainly don't consider IDL to be a
programmer's worst nightmare. I guess you've never had to use COBOL.
:)

Seriously, it seems to me that you've been a tad unfair to IDL.
Consider what IDL was **designed** to be: an interactive language for
data exploration and analysis. And, consider **when** this was done:
back long, long ago when about the only other interactive system
around was Forth.

The point is, IDL wasn't designed as a replacement for C/C++/Java. It
was designed for interactive use and for programs that, for the most
part, are written and maintained by one person. Of course, there are
exceptions to this statement.

If you think that the folks who developed and continue to develop IDL
are somehow lacking in a solid foundation in computer science and
compiler design think again, it ain't true! They know full well what
sort of language they have and intend to keep it in that vein.

That said, yes, IDL isn't what you'd choose to write a really large
software project in. And, this is okay. It **is** something you'd
consider prototyping with, or developing algorithms that will
ultimately be implemented in another language.

IDL isn't speedy? I strongly suspect that it will perform on par with
or better than many existing interpreters (Perl, Java, Python, Matlab,
Mathematica, etc).

It might be fun to consider what IDL would need to be like to make it
a suitable choice for a large software development project, but to
fault it for not being that when it never was meant to be isn't
exactly fair.

Besides, there is a certain elegance to be found in not having
strongly-typed variables, it lets you do some fun things. Consider it

living on the edge.

Ron Kneusel
rkneusel@qwest.net
