Subject: Re: IDL and OpenGL
Posted by Steve Houston on Wed, 07 Nov 2007 14:28:42 GMT
View Forum Message <> Reply to Message

Karl Schultz wrote:
> It would probably be wise to profile your application to determine
> where the time is being spent. It would be unwise to go through
> a lot of effort to replace the display subsystem when the problem
> might be elsewhere.
>
> You can use IDL's profiler or use SYSTIME to check suspected hotspots.

I would like to second Karl's suggestion to use the profiling tools in
IDL to determine where the bottlenecks are.

>
> If the Window Draw method ends up being the one using all the time, then
> perhaps you are sending too much geometry or have too complex of a scene.
> You might consider adjusting the fineness of your meshes according
> to the view.
>
> If tiling is the bottleneck, consider using a tiling server running
> in another process. This will at least let the main app run smoother
> without stalling while waiting for tiles.
>
> Object Graphics is a fairly thin layer over OpenGL and so there are
> not many places where IDL slows things down for OpenGL. But you might
> be able to do better by accessing some of OpenGL's features that IDL
> doesn't use. Compiled vertex lists (VBO's) are an example, but I
> do not know if IDL uses them or not.

IDL 6.4 and later does use OpenGL vertex buffer objects (VBOs) to store
vertices and indices in video RAM, under most circumstances. This
greatly improves rendering performance as long as your vertices are
mainly static (you aren't providing new vertex data every time you
render the object).

>
> Finally, and this is NOT recommended, you can make OpenGL calls using
> the same IDLgrWindow and OpenGL context via C code. You start by making
> a subclass of IDLgrModel and override the Draw method. Your Draw method
> can then call C code that can make OpenGL drawing calls. The dicey part
> of this is that the IDL internals can change from release to release
> and something like this that worked in one release may not in another.
> But it is a good way to prototype something to see if you are on the
> right path.
>
> Karl

>
> Gaurav <selfishgaurav@gmail.com> wrote:
>> Hi all,
>> I have been using IDL to develop a Google Earth type application for
>> the past one year and have been pretty successful in doing so. But the
>> application is not as fast as I would like it to be. I believe I have
>> removed all the programming bottlenecks that might slow my program
>> down(Avoiding loops, using small tiles of images etc..ad infinitum). I
>> was wondering if leaning more on OpenGL might be of some help.
>>
>> Now, I am aware that IDL already uses OpenGL for many of its tasks-but
>> exactly what tasks it does using OpenGL is not transparent (at least
>> to me). Can anyone suggest me ways to get on if I want to acess the
>> OpenGL library of functions using IDL-the way they do it using C, C++,
>> VB or VB.NET.
>>
>> I would like to keep my discussion open ended and would welcome any
>> advice related to the matter-be it OpenGL+IDL or makeing my program
>> faster otherwise.
>>
>> Cheers!
>> Gaurav
>>
>