Subject: Re: IDL and OpenGL
Posted by Steve Houston on Wed, 07 Nov 2007 14:52:25 GMT
View Forum Message <> Reply to Message

Robbie wrote:
>> exactly what tasks it does using OpenGL is not transparent
> I've also had some difficulties optimising IDL's image tiling. I
> wanted to place tiles off-screen so that they don't need to be fully
> redrawn when they become visible. I kept finding that the off-screen
> tiles were not being stored in the GPU and I had to re-render them on
> demand.

Could you explain in more detail what you are doing?

When you call SetTileData for a tile that is offscreen the data is loaded into a texture map and stored in video RAM. However, any tiles that are offscreen will not be rendered, OpenGL clips everything to the view frustum, so anything offscreen is essentially ignored.

When your tile comes onscreen, it will need to be rendered, but this should be extremely fast as the texture data is already loaded into video RAM, ready to go.

> I think that these kind of tricks are really beyond IDL at the moment.
> IDL provides a level of abstraction that is easy to program but
> difficult to optimise.

IDL is designed to hide the details of the underlying rendering API, to make it easier to program and so a different renderer can be used if necessary. It does this while adding as little overhead to the underlying rendering API as possible.

The IDL object graphics API is designed to be as flexible as possible, but it's not as low level as OpenGL, so there will be some things you can do in an application calling OpenGL directly that you can't in IDL. If this is the case I recommend you contact ITTVIS or post in this newsgroup so we can evaluate the functionality you require.

> If I wanted to render using openGL then I would
> write a C++ application using Coin3D. I would probably call that
> application as a DLM from IDL or insert it as an ActiveX component.
>
> Robbie
>

Steve.