Subject: IDL Graphics Objects & Heap Variables
Posted by Mike[3] on Wed, 15 Feb 2006 19:34:27 GMT
View Forum Message <> Reply to Message

I cluged together the following code in an attempt to render 3D obj files (similar to CAD models) in an interactive IDL window. By "cluged", I mean that I modified the existing XPLOT3D code to create PolyPlot3D. PolyPlot3D renders individual facets into an interactive window. Using the OVERPLOT keyword, I can place many facets on the same window, ultimately building a 3D rendering of an object.

While the code appears to work, it slows down tremendously for obj files with more than 2,000 facets. I believe it slows down b/c the code generates a heap variable to keep track of every rendered facet. My opinion here: the heap variable is necessary to maintain z-buffer heirarchy information.

A few questions: (1) Can anyone confirm that the heap variables are causing my code to slow down? (2) Does anyone have any suggestions for how I might be able to speed up my renderings?

If anyone is inclined to check out my code, they are attached on my wiki page at the following URL:

http://wiki.cis.rit.edu/bin/view/People/MikeFosterGUIFinalProject?CGISESSID=01bde2a2d02fcd405336d0b5fe490658

To actually run the code, please download the following files from the wiki site:

(1) vertex.dat
(2) faces2.dat
(3) polyplot3d.pro
(4) Renderedpickup.obj.pro

Compile the polyplot3d.pro function before running Renderedpickup.obj.pro. Vertex.dat and faces2.dat are ASCII files with all of the vertex / connectivity information associated with the pickup model.

Thanks for reading this far,
Mike

Subject: Re: IDL Graphics Objects & Heap Variables
Posted by Rick Towler on Tue, 21 Feb 2006 17:21:22 GMT

Hi Mike,

Few of us have time to wade thru someone else's code but I do have a question or two and a suggestion.

First, what idl primitive are you using to display your facets and how are you doing it? One primitive for each polygon or are you adding the polygons to a single primitive (most likely IDLgrPolygon)?

What exactly is the issue? Does it take a long time to create the object but once created it is easily manipulated? Or is it painfully slow to manipulate the object?

Are you using hardware, or software rendering? What hardware are you running this on? What platform?

My guess is that you are creating a bunch of IDLgrPolygon objects, one for each facet. This really will not work for large #'s of polygons. The correct approach would be to add the polygons to a single (or few) instance(s) of IDLgrPolygon.

My suggestion is to use the PROFILER procedure to look at where your code is spending its time. You'll want to set the /SYSTEM keyword to make sure system routines are returned in the report. Check out help on the PROFILER procedure for more info.

-Rick

Mike wrote:
> I cluged together the following code in an attempt to render 3D obj
> files (similar to CAD models) in an interactive IDL window. By
> "cluged", I mean that I modified the existing XPLOT3D code to create
> PolyPlot3D. PolyPlot3D renders individual facets into an interactive
> window. Using the OVERPLOT keyword, I can place many facets on the
> same window, ultimately building a 3D rendering of an object.
>
> While the code appears to work, it slows down tremendously for obj
> files with more than 2,000 facets. I believe it slows down b/c the
> code generates a heap variable to keep track of every rendered facet.
> My opinion here: the heap variable is necessary to maintain z-buffer
> heirarchy information.
>
> A few questions: (1) Can anyone confirm that the heap variables are

> causing my code to slow down? (2) Does anyone have any suggestions for
> how I might be able to speed up my renderings?
>
> If anyone is inclined to check out my code, they are attached on my
> wiki page at the following URL:
>
> http://wiki.cis.rit.edu/bin/view/People/MikeFosterGUIFinalPr
oject?CGISESSID=01bde2a2d02fcd405336d0b5fe490658
>
> To actually run the code, please download the following files from the
> wiki site:
>
> (1) vertex.dat
> (2) faces2.dat
> (3) polyplot3d.pro
> (4) Renderedpickup.obj.pro
>
> Compile the polyplot3d.pro function before running
> Renderedpickup.obj.pro. Vertex.dat and faces2.dat are ASCII files with
> all of the vertex / connectivity information associated with the pickup
> model.
>
>
> Thanks for reading this far,
> Mike
>
