Subject: Re: shade_volume and quality change causes IDLDE to crash?
Posted by Karl Schultz on Wed, 10 Dec 2003 17:32:57 GMT
View Forum Message <> Reply to Message

"David Yip" <dcw_yip@yahoo.com> wrote in message
news:201431cc.0312091537.3fee6cbe@posting.google.com...
> It doesn't crash with software rendering or on other machines with
> different cards. I've already tried updating both XP and the drivers
> to no avail. My graphics card isn't bad per se. It works just fine
> with other apps. The driver must have a bad interation with IDL.

It is probable that IDL is just using a feature of OpenGL that is buggy in
your driver. Read on.

> Strangely this card is a professional level OpenGL card unlike most of
> the cards people buy to play games at home. It is one of the
> recommended cards for high end professional 3D graphics work.

I'm surprised that this card is giving you trouble, as nVidia usually does a
good job on their drivers.

>
> David
>
> David Fanning <david@dfanning.com> wrote in message
news:<MPG.1a3edbe2a2e3224a989775@news.frii.com>...
>> David Yip writes:
>>
>>> The code I'm working on uses shade_volume. Whenever, I switch the
>>> quality of the window from high to low or medium and back to high
>>> IDLDE crashes out.

The quality change is interesting here. Are you drawing the polygon mesh
filled or wireframe? Using hidden lines? If hidden and wireframe, then the
rendering method, and hence OpenGL calls, used to draw the mesh varies quite
a bit depending on the quality setting and the version of OpenGL. It is
possible that the switching back and forth between rendering methods is
exposing a flaw in the driver that is causing the crash.

If you can, please try running IDL under a debugger, make it crash, and then
examine the call stack. If the crash is in an nvidia component (like a DLL
that starts with nv), then it is likely a driver problem. If it looks like
it is in IDL, then you'd need to report the problem to RSI tech support.
Sometimes a graphics driver can hose itself so badly that there is no
recognizable stack trace, in which case, I'd still think of it as a driver
problem.

>>> It doesn't happen if I sub in isosurface for

>>> shade_volume.

This is odd. Shade_volume generates quad meshes that generally have fewer polygons and vertices than the tri meshes generated by isosurface.

One weakness of shade_volume is that it is susceptible to the famous marching cubes ambiguity problem. This problem can cause the code to generate a self-intersecting "bow-tie" quad in the mesh. A graphics driver and/or its hardware can really trip over this geometry when it tries to render it. Sometimes you can spot a bow-tie as a hole in your mesh, if you can get it draw on a different card or with the software renderer. Otherwise, they are pretty easy to find analytically. The isosurface function does not suffer from the ambiguity problem because it uses a different algorithm.

I really can't guess at much more without seeing the data or knowing more about how the mesh is rendered.

Karl

>>> This only happens on one machine but unfortunately it's
>>> my development machine. It's an XP Pro machine with a Nvidia Quadro
>>> FX 1000 graphics card. Has anyone else experienced this?
>>
>> Try software rendering. This will rule out the most
>> likely candidate: a bad graphics card. :-)
>>
>> Cheers,
>>
>> David
>>
>> P.S. You can try updating your driver if this is the
>> case, or--sometimes--software is even faster than
>> hardware.
