Subject: Re: X11 0.2.1,OpenGL and IDL object Graphics
Posted by Karl Schultz on Wed, 19 Feb 2003 16:12:01 GMT
View Forum Message <> Reply to Message

"boccio" <boccio@swarthmore.edu> wrote in message
news:boccio-7D1372.12163216022003@cnews.newsguy.com...
> In article <4a097d6a.0302160143.27bbcb4c@posting.google.com>,
> MKatz843@onebox.com (M. Katz) wrote:
>
> No luck with these ideas. I already have such file and the look the same
> as your suggestions.
>
> I removed and reinstalled everything on the 733 MHz system and on a dual
> 1.25 GHz system.
>
> No luck.
>
> Same problem.
>
> Setting Hardware OpenGL in preferences crashes demo in Object-World etc.
>
> Any other ideas appreciated.

I dug a bit more and found that the IDLDE is crashing with the hw accel
OpenGL. I have to admit that I didn't suspect this, thinking that if
command line IDL worked with hw accel, then the IDLDE should too.

I'll submit a bug report to Apple. Hopefully they'll do another beta
release and this fix will be part of it. It turns out that this particular
problem is very similar to the first problem (in beta 0.1) that they fixed
that made command line IDL work. So, I have a feeling that they have a bit
more work to do in this area. All I can suggest is using software rendering
if you must use the IDLDE.

If anyone is interested, here is the stack trace from the crash log. The
line between frames 10 and 11 is the line between the IDL and the Apple
code. The DRI (Direct Rendering Infrastructure) is a mechanism for OpenGL
client programs that allows fast rendering on a local X server. In a "pure"
sense, the client program would encode OpenGL commands into GLX protocol,
send them to the server, which then decodes them and draws. This is
horribly inefficient, so the DRI exists to provide a direct path to the 3D
drawing code that runs in cooperation with the X server process.

So, if your IDLDE crashes and your log (in /Library/Logs/CrashReporter)
looks like this, then you know what is going on.

Thread 0 Crashed:
#0 0x936fb4bc in CGSLockShmemLockWithTimeout

#1 0x93701d7c in CGSShmemRWLockLockForReading
#2 0x937a127c in CGSGetOnlineDisplayList
#3 0x90a9558c in cglTrimDeviceMask
#4 0x90a966ec in cglConvertAttribs
#5 0x90a96d50 in cglChoosePixelFormat
#6 0x90a97348 in CGLChoosePixelFormat
#7 0x960a1058 in allocate_context
#8 0x960a10bc in XAppleDRIGetIndirectContext
#9 0x960a3cb8 in XAppleDRIUseIndirectDispatch
#10 0x96087afc in glXMakeCurrent
---------- ----------
#11 0x00122854 in XGLMakeCurrent
#12 0x0085fe68 in igSrcDestAcquireReleaseDC
#13 0x0085fdb0 in igSrcDestBeginDraw
#14 0x008603f4 in IDL_GrSrcDestDraw
#15 0x00a02bb0 in interpreter
#16 0x00b3b1f0 in IDL_Executive
#17 0x00a060f0 in IDL_InterpCallFromSysproc
#18 0x00a91960 in call_user_cb
#19 0x00a98bf4 in internal_widget_event
#20 0x00a98dd4 in IDL_widget_process_events
#21 0x00a02bb0 in interpreter
#22 0x00b3b1f0 in IDL_Executive
#23 0x0075d864 in IDL_Main
#24 0x00002814 in main
#25 0x000023ac in _start
#26 0x000021dc in start