Subject: Re: Object graphics under Linux: are they supposed to be that slow?
Posted by Rick Towler on Thu, 31 Jan 2002 17:55:54 GMT
View Forum Message <> Reply to Message

These threads always seem to lead to the same place. Benchmarking is tricky business. There are many factors that play into these results most of which aren't included in the posts. Processor speed. Bus speed. Amount of RAM, type, and speed. Video driver revision. Video driver settings (vsync, texture depth, anti-aliasing). OS. OpenGL revision...

The one thing that we can take away from this is that modern video cards are implementing more openGL features on the silicon and are quickly rendering the software renderer obsolete. From the posts, Mark and Karl have video chipsets that are about 3.5 and 2.5 years old respectively and they perform quite poorly in many of the benchmarks. David's card is newer and does significantly better vs. the software renderer. My card is one generation newer still and it does even better.

I think a solid OG benchmark is long overdue. Karl touched on the key issue, how do you write a general benchmark that produces results useful to the wide range of IDL OG users? I push polygons all day and want a card that can drive huge polygon numbers and handle many lights. Someone else may be doing image processing and could care less about triangle setup and lighting. What are people doing with OG and what information is key in helping them make that next hardware purchase?

-Rick

"Karl Schultz" <karl_schultz@yahoo.com> wrote in message
news:e415b359.0201301023.731ceffa@posting.google.com...
> David Fanning <david@dfanning.com> wrote in message
news:<MPG.16c12aa8f85252b59897e1@news.frii.com>...
>> Mark Hadfield (m.hadfield@niwa.co.nz) writes:
>>
>>> The configurations are:
>>>
>>>   * IDL 5.5 on Windows 2000 using RENDERER=0 (hardware)
>>>   * IDL 5.5 on Windows 2000 using RENDERER=1 (software)
>>>   * IDL 5.5 on Linux. This uses RENDERER=0 but, as is obvious from
>>>   the DeviceInfo string, the rendering is carried out by the Mesa
>>>   software library and does not access any hardware acceleration
>>>
>>> The geometric-mean elapsed time figure provides a rough ranking of the
>>> configurations:
>>>
>>>   Windows RENDERER=0   4.58 s

>>>   Windows RENDERER=1   3.11 s
>>>   Linux   5.55 s
>>
>> Just to give you something to chew over, Mark. Here are
>> my results with IDL 5.5 on Windows 2000, with a 32MB
>> NVIDIA GeForce 2GTS graphics card. Screen resolution
>> is 1280 by 1024 at 32 bits True-Color.
>>
>>   Windows RENDERER=0   0.71 s
>>   Windows RENDERER=1   1.34 s
>>
>> That graphics card was a couple of hundred bucks, as
>> I recall. :-)
>>
>> Cheers,
>>
>> David
>
> Here's another data point:
>
> PIII 750Mhz Windows NT 4.
> nVidia RIVA TNT2 AGP SSE. 32 bits/pixel
>
> I'd characterize this graphics card as a medium-low range card today.
> The driver uses the AGP port, which is good, and apparently leverages
> the Intel SSE instructions, which is also good.
>
> Windows RENDERER=0   9.41 s
> Windows RENDERER=1   2.72 s
>
> The hardware was MUCH slower at image operations. (GL is generally
> not a good image processor) In fact, I had to disable the image
> STRETCH test because it was taking way too long. The hardware really
> only beat out the software at texture-mapped polygons. I suppose that
> this hardware/driver package was tuned for the Quake-like games :-).
> (I should check for a driver update)
>
> While this is a great benchmark, it may not be representative of
> typical IDL application usage of graphics. For example, I might think
> that this test is a little heavy on images. This program was written
> to monitor the object graphics performance during development and
> later modification and so hits most aspects of object graphics.
> Therefore this program probably isn't the best means to select a card
> or even in deciding between hardware and software rendering. Looking
> at the individual test results can help a bit more if you know what
> sort of things you are drawing a lot.
>
> For example, in this case, removing the image tests would probably

> bring the hardware and software numbers closer together. And that
> would be important to me if my programs didn't use IDLgrImage very
> much.
>
> Karl
> RSI