
Subject: : AmigaOS4

Topic: : GL4ES: another OpenGL over OpenGL ES2 emulation - some tech. info and porting progress

Re: GL4ES: another OpenGL over OpenGL ES2 emulation - some tech. info and porting progress

Author: : Daytona675x

Date: : 2019/9/18 9:26:31

URL:

@thellier

Quote:

But will it avoid the "write from buffer to GPU vram" part ?

In theory it could. I mean, we explicitly tell Nova "look, what's coming is just a plain byte buffer, no conversion whatsoever required and write-only" even *before* locking the VBO! So in theory it *could* give us a pointer into VRAM at this point in such a scenario. But I bet it doesn't. Actually, I *know* it doesn't because Hans just told us:

Quote:

Hans:

Huh? If a VBO contains only uint8 data, then it should be using a straight copy routine (one that uses doubles if possible).

If we had direct VRAM access then no such extra copy at his end would be necessary at all 😊

Quote:

I mean perhaps Nova will proceed as usual : copy reordered data from a buffer to vram but only skip all reordering for each items

Yes, that's the case. But while we don't have direct VRAM access we still have a potential up-to-factor-4 (7 vs 30 fps in my test) performance gain around the corner with the plain-bytes trick nevertheless - if VBOSetArray with W3DNEF_NONE would work as promised.

But yes, completely getting rid of the Nova-copy in such a scenerio would be great. This here is meant to be a hack using what we already got (well, with what I *thought* we already got). And for sth. like that the potential gain is unbelievable big already.

Quote:

>RGBA8 data converted to RGBA float

Interesting too

Yes, but that's not everything ogles2 does under the hood to workaround some Nova slowdowns through type-conversions. E.g. it also silently converts 16bit index data to 32bit because native 16bit indices are slow (and 8bit indices because they aren't supported at all). ogles2 even does so if you supply your indices in your own VBO - then not your VBO is being used but an internal 32bit version of it 😊

Quote:

Anyway we just need a new VBO mode (let it call W3DN_RAW_ACCESS)

Yes, that won't hurt, the potential speedup is gigantic. Although for now I'd be happy already if `VBOSetLArray(W3DNEF_NONE)` would do what it's supposed to do.