

---

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: : 2019/9/17 18:15:09

URL:

@kas1e

Quote:

Interestingly also, that in the documentation about BufferUnlock of warp3dnova, there is no mention about any big->little endian conversion

That's right, that documentation is not in the Unlock function's desc but in the Lock functions's 😊

In VBOLock's description it is mentioned that you must call VBOSetArray first to tell Nova the data types and other info so that it knows for which parts of the buffer's data it has to perform endian conversion. Although it's not explicitly stated, you may safely assume that the actual conversion is then done inside BufferUnlock (and then back again in VBOLock if a read is requested, unless the driver keeps a copy of the original data, no idea if it does).

If the conversion would take place later, internally, then this info in the docs wouldn't make sense because then you could tell Nova the data layout later (prior to first use) as well.

@kas1e

@thellier

The fun part with that VBOSetArray convention:

you can (ab)use it to trick Nova into not doing its slow endian conversion. Simply tell it beforehand that the VBO is just a package full of plain bytes 😊

But keep your seat, yes, it works, but unfortunately another Nova-slowdown-area kills the potential gain again: you may remember that ogles2 contains a workaround for plain byte stuff like RGBA8 data. I found upload of such endian-free-simple-data to be so extremely dead-slow for unknown reasons, so the lib converts those to RGBAfloat32 data... You'd expect it to be much slower than (the 4x byte-to-float conversion, 4x as much data to transfer) but it's muuuuuch faster than letting Nova do the simple job on the plain bytes.

So, unfortunately to avoid Nova's endian-conversion also means to switch to that slow byte-data layout, so we end up with a netto fps loss.

Damn 😊 I will experiment a bit more, but this one looks like a dead end. So we'll have to wait for Hans to improve speed and eventually also implement that requested manual-endian-conv-conversion.