

---

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 12:04:16

URL:

@thellier

You misread / skipped stuff. And maybe overread the latest post.

Quote:

The only thing that helps is to create a simple 1 array VBO in the first place

This one is outdated. See post above your post. If the unused arrays are "disabled" in the way (and only this way) I outlined there then it works with VBOs with arbitrary numbers of arrays.

Quote:

Do you mean a simple array of floats declared as `W3DNEF_NONE + uint8` ?

I dont understand what you mean by that.

Quote:

Because later on your text you say it dont works but say "we still have a potential up-to-factor-4

As I stated: 7 fps vs 30 fps (~ factor 4) was what I measured:

a) 30 fps if I used a 1-array index VBO with the no-endian-conv-trick.

b) 7 fps if I instead made it a 2-array index VBO with the 2nd array disabled via `W3DNEF_NONE`, which weirdly enough results in Novas slow (standard) copy-conv being triggered.

What did not work at this time was that I could not have (b) to be as fast as (a) because I didn't find a way how to trick Nova into the raw-byte-copy-mode if the VBO had more than 1 array.

Hans `W3DNEF_NONE` info turned out to simply not work at all. And his `W3DNEF_UINT8` hint lacked proper usage info and Nova has a logic bug when it comes to mode selection, which is why I did not try it in a way that would suite Nova.

NOW I found such a way. So now get (b) with the speed of (a), which means that if I apply that wisdom to my internal interleaved vertex data VBOs, then the upload of those should speed up accordingly.

Important: note that this expected speedup is usually **not** going to result in a 4x higher framerate!! What's being sped up will be the VBO upload only! If the respective app uses its own VBOs then there most likely won't be too many uploads... Also, ogles2 does a lot of tricks to avoid uploads at all costs.

So you can expect biggest improvements for situations where a) the ogles2 client uses client-RAM instead of his own VBOs and b) that vertex data is frequently changing.