

---

Subject: : AmigaOS4

Topic: : The OpenGL ES 2.0 thread

Re: The OpenGL ES 2.0 thread

Author: : Daytona675x

Date: : 2019/7/23 14:26:05

URL:

## OpenGL ES 2 version 2.9 for Warp3D Nova / AmigaOS4 is on my FTP for testers to test now!

- Fix: if an enabled vertex-attribute-array was being disabled then the corresponding generic vertex-attribute wasn't applied eventually, resulting in semi-random vertex-attribute data. This will rarely happen in a real-world app, which is why this bug wasn't found earlier. Thanks to Capehill for reporting!

- extension to the `glGetString` **GL\_VERSION** query: it will now not only return the ogles2 lib's version but also the Nova version! Programs should check both versions because especially GLSL functionality depends on Nova, not so much on ogles2. The returned string now looks like this:

"OpenGL ES 2 2.9 on top of Warp3D Nova 1.65"

Thanks to Raziell for indirectly inspiring me to do this one 😊

- and for convenience (or if you find it too hard to parse the 2nd version number (cough, Entwickler-X-Frank, cough 😊) which may appear at different positions depending on the 1st one) the Nova version number is also appended to the **GL\_RENDERER** string which now reads like:

"Warp3D Nova 1.65"

- Implemented the **OES\_get\_program\_binary** extension as requested by kas1e. Because Nova doesn't provide any support here, this implementation works with the internal SPIR-V code of all shaders that make up the respective program. So the binaries here are no true machine-code but intermediate code, packed into a proprietary format. Providing such binaries essentially skips the vertex- and fragment-shader's GLSL compilation steps. Anyway, this means:

- new function **glGetProgramBinaryOES** to extract such a binary representation of the respective successfully linked shader program.

- new function **glProgramBinaryOES** to supply the GL with such a cached binary.

- support for the `glGet` parameter **GL\_NUM\_PROGRAM\_BINARY\_FORMATS\_OES** - which returns 1 now.

- support for the `glGet` parameter **GL\_PROGRAM\_BINARY\_FORMATS\_OES**, which returns one single format ID identifying my binary format (0xC00DA675)

- support for the `glGetProgramiv` parameter **GL\_PROGRAM\_BINARY\_LENGTH\_OES** which gives you the buffer size required to store the binary for the respective program.

- consequently added **GL\_OES\_get\_program\_binary** to the extension string.

- added glProgramBinaryOES and glGetProgramBinaryOES to the stubs-library for direct use.

- Fix: aglGetProcAddress returned function pointers in the Amiga interface style, with the first parameter being a pointer to the interface. Now the compatible "stub"-style signature is being returned which matches the GL function pointer prototypes.

Apparently nobody used aglGetProcAddress before, so this went unnoticed. On our ogles2.lib even the extension functions are all simply directly accesible just like all other functions, so if you code for Amiga only you don't need aglGetProcAddress.

gl4es however uses it - and this was where it all exploded now 😊 Thanks to kas1e for reporting..

- !don't forget to download the **new include-folder** too!

- version set to 2.9 (23.7.2019)

Thanks to kas1e for testing!

Cheers,  
Daniel