Subject: : AmigaOS4
Topic: : SDL2
Re: SDL2
Author: : kas1e
Date: : 2018/10/30 6:38:23
URL:

@Capehill
Quote:

> Regarding Q3, If I understand correctly, there should be a Q3 makefile-level switch for choosing the GL. Or, code could try to look for CVA extension with or without 's'.

Yeah, seems we both misunderstood :)

What i trying to say, is that if anyone will try to compile from your repo SDL2 version of Q3 for amigaos4, but not for minigl then he will have issues, because your minigl fix, are done as "#ifdef __amigaos4__", which mean ANY amigaos4 build will remove "s" and end.

For example , if one will compile it over any other GL realisation than MiniGL where no extra "s" for CVA. In all those cases, hardcore "#ifdef __amigaos4__" as you have now are non-go, as will make things works wrong. It is only fix for minigl, so there should be something like "#ifdef __amiaos4__ && __minigl__" , or something, instead of just "#ifdef __amigaos4__" :)

Quote:

> Regarding SDL2 renderer hint, I'm not sure which problem you want to solve by using that hint?

Just thinking about way how to build SDL2 with all renderers + gl4es, and how to choice it later instead of minigl.

Quote:

> What benefit gl4es will bring when used as a (2D) renderer, over pure GLES2? As I understand it, gl4es is meant to wrap OpenGL 1.x (MiniGL) calls with OGLES2 ones, but why not use directly OGLES2?

GL4ES its not only wrap of OpenGL1.x calls, it is also wrap OpenGL2.0, OpenGL2.1 and some of OpenGL3.x, and translate it to OGLES2.

I.e. many SDL1/SDL2 apps use OpenGL, but not just OpenGL1.x, but OpenGL 2.x, 3.x, etc. And GL4ES for us, it is that OpenGL 2.x, etc which we always want.

We can't "just use OGLES2", because almost none real live and opensouce apps/games use it, they use instead OpenGL2.x and more. And , yes, of course better was to port just MESA over warpdnova, so to have full OpenGL driver working over Warp3dnova, and to not make such combinations like "gl4es->ogles2->warp3dnova", but it was too big task, and instead they choice OGLES2, which, is limited subset of OpenGL2.x/etc for mobile devices.

In other words, we have OPENGLES2 not because its better, than full reall OpenGL, it is smaller and worse, just was real to do in our limited resources and time. And so, as result we still need full OpenGL2.x and more, which, gl4es do for us over ogles2.

And if we have in SDL1/SDL2 minigl support only , we limited to OpenGL1.5, if we use GL4ES we can build apps which use OpenGL1.x, OpenGL2.x, and some of OpenGL3.x, with all that shaders support and stuff.

Quote:

> If you want to force opening the correct GL library for OpenGL context instead of, it has to happen by some other means. Please refer to readme document how to select between MiniGL and OGLES2 context. At the moment I don't know what is OpenGL version that gles4 supports - that's one thing that might be used.

Probabaly you miss the point of GL4ES.. It is OpenGL2.x and more for us, which can works only over OGLES2/Warp3DNOVA.

What i ask, is how we can integrate GL4ES together with other renderers, so when one will have needs, he can just choice it.

Quote:

> Is gl4es a static library, shared or both?

At moment it is static link library, and it can be .so as well, just i do not worry about .so at moment.

Quote:

> Have you added configure support for gl4es?

At moment its pure list of objects to compile, like a small makefile.

Quote:

> In other words, how to recognize SDL_os4gl4es.c should be compiled?

just compile it in any case without needs to recognize ?

Quote:

> It sounds difficult, because if SDL2 library starts to require symbols from gl4es, then there has to be 3-4 different SDL2 binaries (nightmare). Versions for sdl.a, sdl.so, sdl.a with gl4es.a, sdl.so with gl4es.so.

For build gl4es version of SDL1/2 i just add include of GL4ES includes, and that all. Only on linking stage of final apps, i add -lgl4es to link with linker lib.

I mean, for me it looks like this: We add 2 new files to compilation process , they compiles, injects inside of libsdl2.a , and that all. When programmer want gl4es he will choice it (need to think how, that i why i think about SDL_HINT), and on linking add -lgl4es together with -lsdl, and that all.

Or you fear about conflicts of the same GL functions in libsdl .a from both minigl and gl4es compiled at the same time ? But imho that no problems, those ones will be choice, which are connected to choicen renderer from sdl app.