
Subject: : AmigaOS4

Topic: : SDL2

Re: SDL2

Author: : kas1e

Date: : 2018/11/6 10:41:57

URL:

@Capehill

Quote:

Still regarding Q3, I was trying to say that Q3 code can be fixed so that it tries to look for either functions, with or without 's'. If it finds one, then good. If none, then there is no CVA implementation and Q3 has to live with that.

Seeing your change in q3-sdl2 repo, i understand now what you mean. It works that way too, thanks!

Quote:

If you want to be able to select both MiniGL or gl4es for SDL_Renderer, you need to provide a symbol that describes it, like [https://github.com/AmigaPorts/SDL/blob ... c/render/SDL_renderer.c#L78](https://github.com/AmigaPorts/SDL/blob...c/render/SDL_renderer.c#L78) . Maybe one of the existing OpenGL renderers can be extended, or duplicated. But I still don't see the value over unwrapped OGLES2 renderer. This is not about application code, this is about SDL internal 2D renderer implementation. When renderer is used, all drawing happens using SDL calls: <https://wiki.libsdl.org/CategoryRender> . If you are porting a Quake, you simply don't use the SDL_Renderer. You create OpenGL context and command that.

Ok, got it. When i all the time mean "renderers" i just mean contexts then. I of course do not need any new rendereres, all i want, is to add support of gl4es (i.e. creating of opengl context via gl4es instead of minigl and instead of directly ogles2 which sdl2 have now). Just to have it in SDL2 repo, so it will always be there, works, and code can be checked not only by me.

Quote:

Again, SDL_Renderer is something different. It's an abstraction higher that OpenGL: application doesn't make glCalls. You seem to be talking about OpenGL context creation instead of. SDL renderer hint is

about choosing a desired renderer backend: it could be a Direct3D one, or compositing.

Yeah, i of course mean opengl context creation, and usage of it with glCalls from SDL app.

Quote:

Can gl4es match the compositing speed in 2D?

Dunno. Its the same as MiniGL in all aspects, just works through ogles2/warp3dnova, and provide us with opengl2.1 instead of opengl1.5 as in minigl (+ give ability to use shaders and all that stuff).

Quote:

Choosing the OpenGL context happens currently by setting the appropriate version numbers. It could happen via hint system, but not by https://wiki.libsdl.org/SDL_HINT_RENDER_DRIVER , it's for 2D.

So, that just mean, that we can add 2 files SDL_os4gl4es.c and SDL_os4gl4es.h, and later from the SDL app, via setting appropriate version number to change gl4es instead of minigl ?

Quote:

I have to worry about people building SDL. That's why configure script should sniff around whether gl4es exist or not, similarly than for OGLES2. Not all people have OGLES2 headers but some want to build SDL.

Sure that mean that gl4es headers should be inside of SDL repo as well. By that way, support of it will be compiled in, and anyone who will have needs for will use it, anyone who don't will not.

You already have includes for OGLES2 anyway (which not eveyone have), so having gl4es ones will be kind of the same (right ?)

I mean it will make libSDL.a or libSDL2.a be fatter for about some bytes only, and will have no impact on anyone who don't have ogles2/warpdnova, all will be the same for them. Just when developers will want to use gl4es instead of minigl, they will add in their sdl app that call to change version number or how it can be, and on linking stage will add libgl4es.a

It's all will like the same ogles2 support you have in, just with some modifications, and there will be needs to choice it from SDL app the same as one choice ogles2 context creating instead of minigl.

I of course can just create separate builds of sdl1 and sdl2, with compiles in support of gl4es instead of minigl ,
so SDL apps will have no needs to be changed at all (by calling that call which will do choice which opengl
context to create), but that mean that code will be not in repo , and bugs which i keep will keeps forever)