
Subject: : AmigaOS4

Topic: : SDL1 open issues

Re: SDL1 open issues

Author: : kas1e

Date: : 2019/8/25 12:43:44

URL:

@Capehill

Quote:

I don't understand why, because if I'm reading the correct code, Irrlicht SW renderer is hard-coded to 16-bit: <https://github.com/okuoku/irrlight-gen> ... t/CSoftwareDriver.cpp#L33

I think its done for simplify things and for speed. Software rendering mostly used for some test probably and for comparing things, it probabaly just have no needs to be 32bit aware.

Quote:

It would be interesting to know would Irrlicht work nicely if this hard-coded format was changed to 32-bit. Maybe it's slower to render 32-bit but it should look nicer too.

Its already not that fast on amigaos4 as it should be, because of all kind of reassons (i will post benchmarks later in some other thread), so make it be even more slower even in software rendering imho no go. Its just on level of old 1.6ghz amd :) (Probably because of missing DMA in graphics.library as well as anything else). But probabaly for theoretical reassons and for checks, it can be intersting to play with..

Quote:

Second option is to modify texture format to 16-bit (same as Irrlicht's). But not even SDL software renderer support this exact format with 1-bit alpha so it means (as far as I know) that SDL will do colour conversion step during texture update.

Third option is to manually convert colour format but I don't like this one.

What i mean, is that we do call `SDL_CreateTexture()` with some image format. So, all we need there, is to check what image format we have, and put it to the `SDL_CreateTexture()` instead of `SDL_PIXELFORMAT_ARGB8888`.

I think about something like this:

```
void ClrrDeviceSDL::resizeWindow(u32 x, u32 y)
{
    if (ScreenWindow)
    {
        SDL_SetWindowSize(ScreenWindow, x, y);
    }
    if (ScreenTexture)
    {
        SDL_DestroyTexture(ScreenTexture);
        ScreenTexture = NULL;
    }

    if ((surface->getColorFormat()==video::ECF_R8G8B8) || (surface->getColorFormat()==video::
ECF_A8R8G8B8))
    {
        //printf("video::ECF_R8G8B8) || (surface->getColorFormat()==video::ECF_A8R8G8B8n");
        if (surface->getColorFormat()==video::ECF_R8G8B8)
        {
            //printf("video::ECF_R8G8B8n");
            ScreenTexture = SDL_CreateTexture(ScreenRenderer, SDL_PIXELFORMAT_RGB888,
SDL_TEXTUREACCESS_STREAMING, Width, Height);
        }
        else
        {
        }
    }
    else if (surface->getColorFormat()==video::ECF_R5G6B5)
    {
        //printf("video::ECF_R5G6B5n");
        ScreenTexture = SDL_CreateTexture(ScreenRenderer, SDL_PIXELFORMAT_RGB565,
SDL_TEXTUREACCESS_STREAMING, Width, Height);
    }
    else if (surface->getColorFormat()==video::ECF_A1R5G5B5)
    {
        //printf("video::ECF_A1R5G5B5n");
        ScreenTexture = SDL_CreateTexture(ScreenRenderer, SDL_PIXELFORMAT_ARGB1555,
SDL_TEXTUREACCESS_STREAMING, Width, Height);
    }
}
```

}
But that didn't work, as at moment when this `resizeWindow()` is called , it seems that image's surface didn't contain necessary format, and it return `ECF_R8G8B8` sadly.. Maybe just code need to be restuctured a little, so when we will call `SDL_CreateTexture()` format of image's surface will be known, and so all will works.