

---

Subject: : AmigaOS4

Topic: : SDL2

Re: SDL2

Author: : kas1e

Date: : 2018/11/28 19:49:22

URL:

@Capehill

I have some VRAM leaks in all SDL2 stuff (that include opengl ones as well), but all of them do not do any call to `SDL_DestroyRenderer()`. So probably linux/windows take care about unfreed vram, and doing it for them.

Question is : are `SDL_DestroyRenderer` is necessary to use and when opengl in no use, and when in use ? For example, in one game, i found that kind of code:

```
if ( screen )
{
    SDL_DestroyWindow(screen);
    screen = NULL;
}
if ( renderer )
{
#ifdef APPLE
    SDL_DestroyRenderer(renderer);
#else
    SDL_GL_DeleteContext(renderer);
#endif
    renderer = NULL;
}
if ( mainsurface )
{
    SDL_FreeSurface(mainsurface);
    mainsurface = NULL;
}
TTF_Quit();
SDL_Quit();
```

So, for APPLE they do call `SDL_DestroyRenderer(renderer);`, but `_dont_` call `SDL_GL_DeleteContext(renderer);`. For others , instead only `SDL_GL_DeleteContext(renderer);` is called, but not `SDL_DestroyRenderer(renderer)`.

What is correct way for us ? And should we allways in any case use `SDL_DestroyRenderer()` and does not matter if it opengl or not ?

As i understand that, if one create renderer with:

Quote:

```
if ((renderer = SDL_CreateRenderer(screen, -1, 0)) == NULL)
```

Then there should be used `SDL_DestroyRenderer`. But if one create renderer like this:

Quote:

```
if ((renderer = SDL_GL_CreateContext(screen)) == NULL)
```

Then there should be no needs to call `SDL_DestroyRenderer`, but just `SDL_GL_DeleteContext(renderer)`;

In our case (i mean on os4), i still have VRAM leaks even if context created with `SDL_GL_CreateContext` and removed via `SDL_GL_DeleteContext()`.