
Subject: : AmigaOS4

Topic: : GL4ES: another OpenGL over OpenGL ES2 emulation - some tech. info and porting progress

Re: GL4ES: another OpenGL over OpenGL ES2 emulation - some tech. info and porting progress

Author: : Hans

Date: : 2019/7/27 2:26:56

URL:

@Daytona675x

Quote:

Your approach "I got issues with boost when optimizer turned on, so I globally disable the optimizer" is certainly not a good way of doing things...

Actually, my approach was to spend days trying to figure out what was going on before finally considering that maybe our rather old version of boost** plus a much newer compiler might be to blame.

Quote:

Is there a reason why you didn't do this replacement a long time ago, especially if you believe that the problem is because of a boost-pointer-casting issue? Should be very quick and unproblematic operation.

Because I have a huge to-do list and limited time. I did start changing it at one point, but it quickly became clear that the slight differences between boost and std are just enough to be a pain, and I couldn't spare the time. So, I reverted the code and shelved that for sometime in future.

Quote:

Out out curiosity: how does this code snippet look like?

No idea; it's been years since I looked at it. Besides, if I knew exactly what code triggered it, then I'd have fixed/worked-around the problem. The crash does **NOT** occur where things go wrong, but is delayed to wherever the counter ends up being 0.

Quote:

And what means "screwing up ref counting"? One too high? One too low? Two independent ref-counters all of a sudden? Random value?

Either one (or more) too low, or two separate ref counters. Otherwise the object wouldn't be deleted while it's

still in use...

I consider ending up with two separate ref counters to be unlikely. The only way to do that is to do something stupid like: `shared_ptr<Obj> newPtr = oldPtr.get()`. That's guaranteed to crash even with the optimization level at 0. Copying/casting one shared pointer to another should **NOT** split the ref count (or mess up the ref count, for that matter).

Quote:

It's an every-day serialization problem, it's certainly much less harder than you think at first glance.

Of course it isn't that hard. However, it's a whole lot more work than just dumping the binary to disk.

Hans

** Our current boost port is from 2006.