

Bug fixing the MESA 3D drivers

The Mesa 3D Graphics Library

Most of our projects are around performance optimisation, but we're cleaning up bugs too. This is because you can only speed up software when certain types of bugs are cleared out. A few months ago, we got a different type of request. If we could solve bugs in MESA 3D that appear in games.

Yes, we wanted to try that and got a list of bugs to solve. And as you can read, we were successful.

Below you found a detailed description of one of the 5 bugs we solved by digging deep into the different games and the MESA 3D drivers. At the end of the blog post you'll find the full list with links to issues in MESA's bugtracker.

Fixing the Render Error in Wasteland 2

Here's a peek in our kitchen. The text below is taken from our issue-tracker, but does not include the logs of the tools like ApiTrace.

Problem Description

There are black rectangles instead of blood splashes. They are rendered in call 1570549, for example.



Fragments' color values are NaNs. The fragment shader (1337, call 1080782) receives NaN values for two inputs: `xlv_TEXCOORD2` and `xlv_TEXCOORD2_1`.

The vertex shader (1336, call 1080779) produces these values by calling `normalize` on `TANGENT` attribute.

`TANGENT` is mapped to 1 generic vertex attribute. There is 1570545 @1 `glDisableVertexArrayARB(index = 1)` before 1570549 @1 `glDrawElements`. I.e. the attribute uses the current value, but this value is never set directly (using `glVertexAttrib4f`, for example), so its value is (0, 0, 0, 1) according to the OpenGL specification.

Problem Analysis

In the OpenGL specifications normalize is not defined for zero-vectors:

Returns a vector in the same direction as x but with a length of 1.

The behavior of all drivers is correct:

- Mesa returns NaNs
- NVIDIA proprietary driver returns NaNs
- AMD proprietary driver returns zeros

Why it works on NVIDIA: although normalize returns NaNs for zero vectors, for some reason `GL_CURRENT_VERTEX_ATTRIB` is (1, 0, 0, 1), this behavior violates the specification!

Why it works on AMD: normalize returns zeros for zero vectors, so other values are not "damaged".

Catalyst uses `v_mul_legacy` here which treats NaNs as zeros (Mesa uses `v_mul`).

Work-around

Here is a workaround that makes Mesa return zeros for zeros vectors (as AMD on Catalyst does) :

```
diff --git a/src/compiler/gslsl/builtin_functions.cpp b/src/compiler/gslsl/builtin_functions.cpp
index d902a91a77..2b0e17e193 100644
--- a/src/compiler/gslsl/builtin_functions.cpp
+++ b/src/compiler/gslsl/builtin_functions.cpp
@@ -4355,7 +4355,10 @@ builtin_builder::_normalize(builtin_available_predicate avail, const gslsl_type *
     if (type->vector_elements == 1) {
         body.emit(ret(sign(x)));
     } else {
-        body.emit(ret(mul(x, rsq(dot(x, x))));
+        ir_variable *sq_length = body.make_temp(type->get_base_type(), "sq_length");
+        body.emit(assign(sq_length, dot(x, x)));
+        ir_constant *zero = ir_constant::zero(mem_ctx, type->get_base_type());
+        body.emit(ret(mul(x, csel(nequal(sq_length, zero), rsq(sq_length), zero)))));
     }

     return sig;
```

How the screen looks like with this workaround:



This workaround also fixes the bug in "Megamaggots' Trails".

In my opinion the bug is in the game engine, not Mesa. `glVertexAttrib4(1, ...)` should be added after `glDisableVertexAttribArray(1)`

to set correct values for TANGENT attribute. It's a coincidence that it works well on proprietary NVIDIA (because of its incorrect behavior) and AMD (because of its normalize implementation).

MESA 3D bugs we fixed

We fixed the following bugs:

Render error in Wasteland 2

Textures are blocky in Crusader Kings 2

Improper rendering in Europa Universalis IV

rendering artifacts with WebGL zombies

Stellaris - colored overlay of sectors doesn't render

At your service! We like it that more of our work is really visible out there.