XDC 2017 Piglit Discussion

Brian Paul, VMware

How many of you use (or care about) Piglit?

How many of you use Piglit for CI / automated testing?

At VMware we run Piglit (along with several other test suite workloads) many times per day in various VM configurations. Piglit's long runtime limits how frequently we can run it (on Windows we have to run in serial / non-parallel mode for stability).

Why is piglit-run.py start-up so slow?

- tests/all.py scans the whole tree for files named *.shader_test, *.vert, *.frag, etc. to build a list of shader tests to run.
- About 50,000 shader files!
- This is especially slow in a Windows VM, taking several minutes. 30 seconds in a Linux VM.
- Can we construct this list of tests at Piglit build time instead of run time?

Why is Piglit runtime slow?

- We’re needlessly running many tests for unsupported extensions.
- For example, for GL_ARB_vertex_attrib_64bit there’s about 10,000 generated shader tests.
- This extension isn’t widely supported in our drivers.   We’re going to run 10,000 instances of shader_running only to report PIGLIT_SKIP for all of them! That’s a lot of wasted time.
- Note: Marek’s enhancement of shader_runner to run multiple tests is not used by default, AFAIK.
- Can we skip entire subdirectories like generated_tests/arb_vertex_attrib_64bit?
- Maybe do something like this in the Piglit framework:

```
if is_extension_supported(“GL_ARB_vertex_attrib_64bit”):
        run tests under “tests/spec/arb_vertex_attrib_64bit/”
```

and,

```
if is_glversion_supported(“4.5”):
        run tests under “tests/spec/gl-4.5/”
```

- Note the difference in “GL_ARB_vertex_attrib_64bit” vs. “arb_vertex_attrib_64bit” and “4.5” vs “gl-4.5”.
- Why did we use directory names like “tests/spec/arb_vertex_attrib_64bit” instead of “tests/spec/GL_ARB_vertex_attrib_64bit”??

A great renaming?

- Rename directories to use precise extension names. For example arb_vertex_attrib_64bit → GL_ARB_vertex_attrib_64bit
- Recategorize existing tests. Move/rename all the tests under tests/general/, tests/bugs/ tests/fbo/ tests/shaders/, etc into tests/spec/GL_FOO_bar/ or tests/spec/gl-X.Y/
- A lot of mundane work
- Disruptive to CI / automated testing
- Thoughts?

Test classification

- AFAICT, about 10,000 Piglit tests do no rendering. They test API error detection, GLSL compilation tests, etc. No need to run those tests on multiple drivers.
- It would be nice to have any easy way to partition the Piglit tests into rendering and non-rendering buckets to further optimize run times (no need to run compiler tests in multiple configurations).

End.