

Apitrace OpenGL profiling view

GSoC '16 report

Alexander Trukhin

Supervised by Martin Peres

XDC 2016

Introduction

GPU Profiling tools

- Intel GPA
- NVIDIA Nsight (NVIDIA Linux Graphics Debugger lately)
- AMD CodeXL (GPUOpen initiative)

Have

- Complex piece of software
- Limited counters

Want to see

- Generic counters' support
- Keep it simple
 - Collect metrics
 - Visualize, inspect metrics
- Vendor-agnostic

Apitrace is an open source program that allows tracing, replaying, inspecting different graphics API calls.

<https://apitrace.github.io>

Old profiling (Aptrace)

- Profiling calls
- Available set of metrics:

```
--pcpu (CPU times)  
--pgpu (GPU times)  
--ppd (Pixels Drawn)  
--pmem (Memory usage)
```

New profiling (Aptrace)

- Metric abstraction system with backends
- Backends added:
 - GL_AMD_performance_monitor*,
 - GL_INTEL_performance_query*,
 - ...
- New CLI options

```
--pframes="backend1: metric1, metric2, ...; backend2: ...; ..."  
--pdrawcalls="backend1: metric1, metric2, ...; backend2: ...; ..."
```

Existing profiling view (QApitrace)



Existing profiling view (QApitrace)

- Only old set of counters supported
- Some performance problems with large data sets

Proposed profiling view (QApitrace)

Key ideas:

- Make use of the new metric abstraction system
- OpenGL accelerated graphs for visualization

Demo

- Need feedback
- Need some additional work

Repository: <https://github.com/trtt/apitrace>

Contact: *trtt* @ IRC/Freenode

alxtry@gmail.com

Questions?
