FrameRetrace: A Responsive UI for Apitrace
Mark Janes, February 3, 2018
mark.a.janes@intel.com
janesma on #dri-devel
About me:

• Working on Linux platforms since 2004, with a background on embedded devices.


• Joined Mesa in 2014, working on performance tools and automation.
GPU Tools stumbling blocks

- Generally hardware-specific
- Mostly closed source
- Linux support is an afterthought
- Tracing/retracing not reliable
- Low numbers of users
- Mesa support for GPU performance counters
FrameRetrace: frame analysis based on ApiTrace

- Widely used and high quality trace/retrace
- [https://github.com/janesma/apitrace](https://github.com/janesma/apitrace)
- Cross-platform: Linux and Windows
- Upstream GPU Counter support in Mesa and Kernel for Haswell and later.
- Leveraged by Intel Mesa team to identify and fix several performance issues in i965.
- Support for Radeon and VC5 in progress.
FrameRetrace: frame analysis based on ApiTrace

- GPU Metrics for each render
- Render target visualization and experiments
- Api log
- Batch disassembly
- Shader analysis, live editing, and assembly
- Uniform constant display and live editing
- Render experiments
- State display and live editing
Windows support provides important leverage for open source driver teams seeking to find Mesa performance gaps.

Proposed features:
- Display texture state, with mip clamp experiment
- Display geometry mesh
- Depth buffer visualization
- Overdraw / hotspot rendertarget visualization
- UI improvements
- Support for more hardware (Radeon and VC5 in progress)
- Android support
Caveats

- Currently a one-person side project, with help
  - Thanks to Laura Ekstrand, Robert Bragg, Lionel Landwerlin, Eero Taminen, Pekka Jylhä-Ollila, Marek Olšák
- Experiments require intricate state tracking
- Some workloads do not have single-frame run loops
- Radeon metrics exposed by AMD_performance_monitor are not human-readable
- Please help!
Questions?