Graphics Pipeline Study With MESA On Ubuntu

* Team members:

Ryan Polley, Pranjal Abhyankar, Vivek Mishra, Wanxin Wang

* Hardware requirement:

Graphic card installed that supports 3D acceleration, graphics driver supports DX11 (Windows) or OpenGL 4.0 (Linux, Mac)

* Introduction:

Inspired by: <https://fgiesen.wordpress.com/2011/07/01/a-trip-through-the-graphics-pipeline-2011-part-1/>, we would work on inspecting graphics chain on operating system, from high-level graphics application, down to the graphical driver in the operating system kernel. Mesa + Linux (<https://en.wikipedia.org/wiki/Free_and_open-source_graphics_device_driver>) gives feasibility for it. While through studying the code in different levels of graphics chain, we could log the graphics system calls (OpenGL API, EGL API…), analyze the calls, profile for OpenGL calls, and customize the components in chain to yield a graphical output from different levels.

* Approach

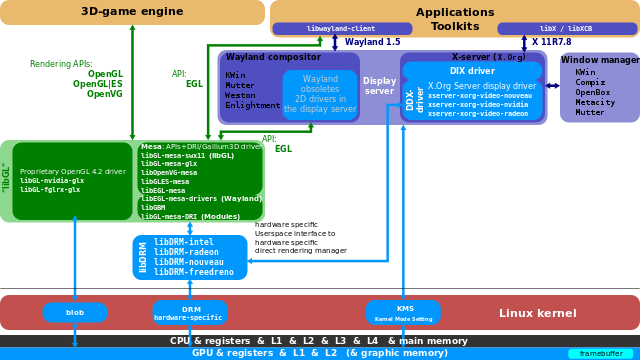


Figure1: The graphics stack on Linux

The inspection would be along paths:

* + OpenGL Application->libGL + MESA driver->libDRAM
  + EGL application(2D GUI)-> libGL + MESA driver->libDRAM
  + (Optional) 3D-game engine-> libGL + MESA driver->libDRAM

To secure the progress of the project, small but rapid iteration approach will be adopted, to guarantee each step is do-able: move from a proved firm existing base through a small step toward uncertainty, once a step move is completed, commit document and source code into git with brief description to share with other team members. The major challenge is the time limit on the project, we need make our work time efficient, to be producible all the time, till the end of the semester time. The secondary challenge the team management, everyone would work on his own branch which is off from master branch, commit every small change with a meaningful log, with every member’s consent, work could merge into master branch for everyone to rebase on. The third challenge is from technical, most of the work prospected is from environment setup: OpenGL application is ported from windows to Ubuntu Linux running on VMware, and then to Ubuntu Linux running on a bare metal; build MESA GL driver for Ubuntu VMware, and Ubuntu a bare metal, …, the flexibility of the environment requires us the move with scrutiny. Overall, the small but rapid incremental steps would secure the progress of the project.

* Experiment

The original Mesa driver will be modified, and generates trace log information for the usage of OpenGL and EGL from graphical application, and yields graphical output, for example to add a water mark on the display for every graphical application that uses Mesa driver, to profile hardware, and graphical driver information, to compare performance of mesa driver + Ubuntu with performance of other platforms.

* Milestones
  + Port OpenGL demo code from Windows to Ubuntu (VMware version)
  + Upgrade Ubuntu (VMware guest) graphics driver with latest Mesa solution
  + Add FPS(frame per second) report
  + Add logging functionality to OpenGL APIs
  + Initiate EGL demo code
  + Add logging functionality to EGL APIs
  + Set up Ubuntu environment on a bare metal with latest Mesa solution
  + Log function calls to libDRM (hardware specific interface in user space)