Introduction to OpenGL

Raghavendra G S

Contents

- Why OpenGL?
- What is OpenGL?
- OpenGL rendering pipeline
- Documentation
- Sample code walkthrough
- Tasks
- Index buffers
- Uniforms
- Error handling

Why OpenGL?

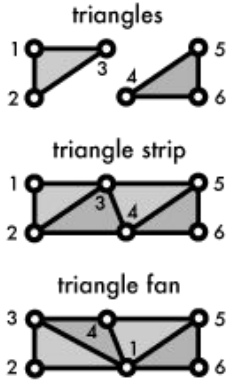
- To interact with GPU.
- Why GPU?
  - The parallel processing architecture allows to perform multiple calculations at the same time.
  - Suitable for graphics-intensive tasks, rendering images.

What is OpenGL?

- Not a library.
- Just a specification of APIs.
- Implemented code exists in GPU driver.
- Mesa3D is an open source implementation of OpenGL standard.
  - Implementations vary on basis of GPU. AMD and intel promote mesa drivers.
  - For NVIDIA, Nouveau driver is mostly developed by community.

OpenGL rendering pipeline

- Vertex specification
  - position, color, normal etc.
- Vertex shader
  - for each vertex, determines location in screen space.
- Primitive assembly.
- Clipping
  - remove the parts not visible, going out of screen.
- Rasterization
  - break the remaining parts in pixel sized fragments.
- Fragment shader
  - Color is decided for each fragment based on vertex colors, light, texture, depth etc.
  - Fragment shader runs independently for each fragment.
  - The most performance-sensitive part of pipeline.



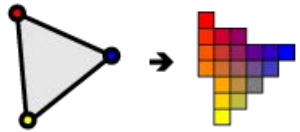


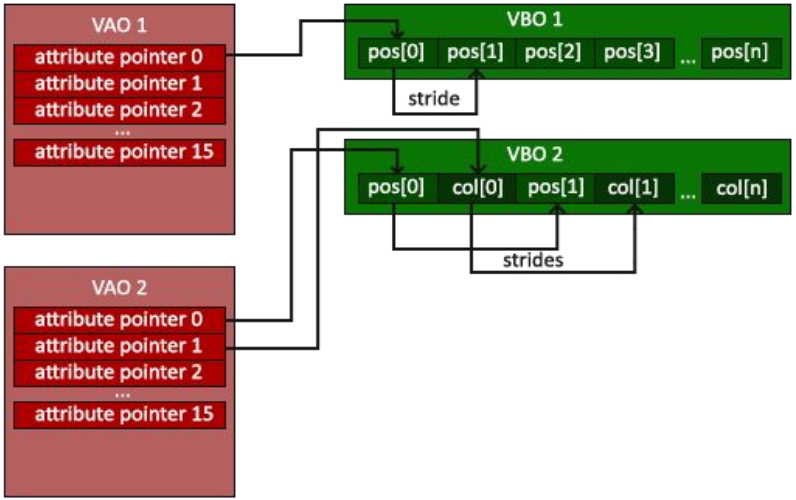
Image ref : http://duriansoftware.com/joe/An-intro-to-modern-OpenGL.-Chapter-1:-The-Graphics-Pipeline.html

Documentation

1. Docs.gl
2. https://learnopengl.com/
   first 2 chapters
1. https://open.gl/
   first 3 chapters
1. http://ogldev.atspace.co.uk/
   first 5 tutorials

Sample code walkthrough

Vertex buffer object (VBO) and Vertex array object (VAO)



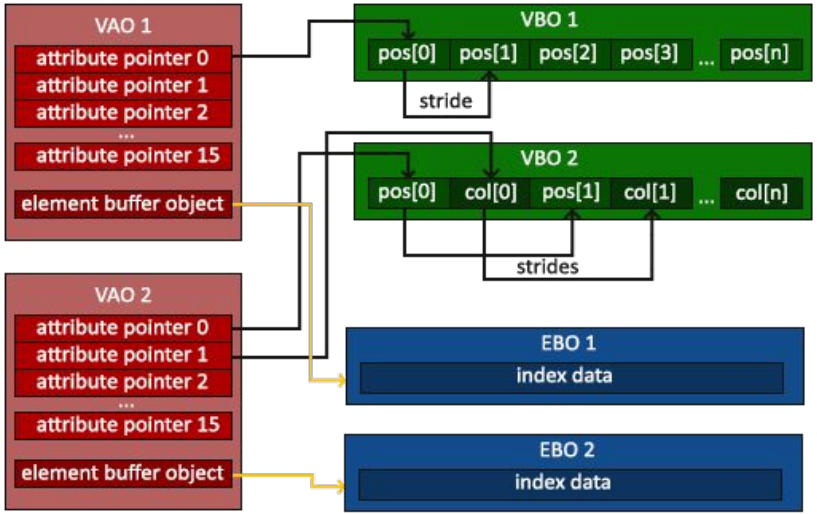


Image ref : https://learnopengl.com/

Thank you!!!