Vertex Buffer / Arrays Objects->

(Textures ->) Vertex Shader -> Primitive Assembly -> Rasterization ->

Textures -> Fragment Shader -> Per-Fragment Operations -> Framebuffer -> API

#) Vertex Shader

input : Attribute / Uniforms / Samplers(Texture) / Temporary Variables

output : Varying / gl\_Position / gl\_FrontFacing / gl\_PointSize

#) Fragment Shader

input : Varying / Uniforms / Samplers / Temporary Variables

output : gl\_FragColor

#) Per-Fragment Operation의 순서

Fragment Data -> Pixel Ownership Test -> Scissor Test -> Stencil Test -> Depth Test ->

Blending -> Dithering -> Framebuffer

#) Framebuffer란?

그래픽 pipeline이 실행되어 최종적으로 그린 결과물을 저장하는 메모리

drawArray를 하면 그제서야 Pipeline이 실행

VG vs Raster // 파이프라인정리 한글강의 9분~10분경

# Fragment Shader에서는 각 vertex의 pixel 값을 정하고 interpolate하여 pixel들의 color를 결정한다.

Framebuffer에 있는 걸 가져오는 작업 : Read Back Path / Read Back Pixel

정확한 함수명은 readPixels