

Fragments = piece of pixel.

Raster operation = combine fragments to generate pixels on the screen

Rasterization of all primitives are independent.

Ordering gets important when transparencies are present.

OpenGL → Vulkan (Ordering is flexible)

→ Care about throughput of each pixel. (Once last pixel is computed, only then we can switch frames).

Vertex Processor

Transformation in vertices: Change their coordinates
Introduce new vertices in between } Geometry manipulation

Fragment Processor

→ Filtering operations, I/O operations from memory.

Should these two be homogeneous/heterogeneous?

Let's have 2 different types of cores for vertex & fragment processors.

Texture: Processing on CPU, pass it to GPU.

↓ Next

Let's make fragment & vertex processors programmable

Amount of efforts on fragments processing