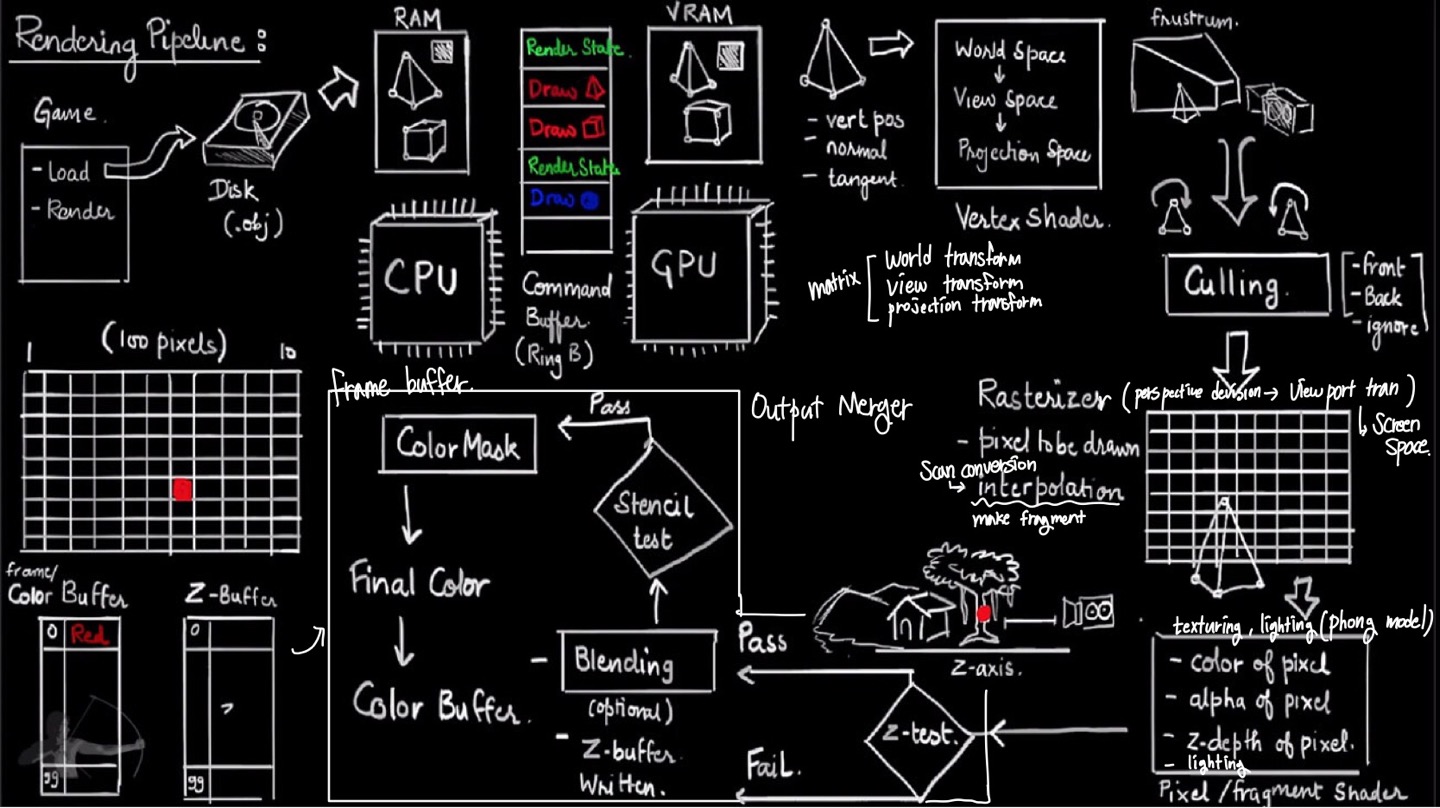
Graphics rendering pipe line



출처 : <https://gasbebe.github.io/shader/>

프로그램이 실행일 될때. 사용한 리소스를 저장장치(hdd,ssd??)에서 불러와 램에 저장

cpu에서 렌더링상태와 메쉬들을 커맨드버퍼에 Queue함

gpu는 커맨드 버퍼에 있는 큐된 메쉬, 렌더링상태 명령을 순차적으로 실행,

gpu처리방식은 만드는 회사마다 command queue의 처리방식은 다를수 있다.

그 다음부터는 우리가 배웠던 렌더링 파이프라인의 모습이다.

Triangle 정보들을 배열 형태로 불러와서 연산을 수행한다.

Vertex shader에서는 좌표를 clip space로 보내 결국엔 Screen에 띄우기 위한 matrix 연산을 거친다.

그리고는 나온 output vec틀을 rasterizer input에 넣어주면 알아서 culling과 scan conversion을 통해 interpolation하고 fragment를 인쇄해준다,

그 다음 fragment shader 가 보간된 normal and texture coord 를 input으로 하여 실제 색상을 결정하며 phong 모델로 lighting을 처리한다.

마지막으로 output merger에서는 fragment별로. z-buffering, depth , alpha blending같은 애들을 수행한다. 이들을 총칭해서 frame buffer라고 한다. 결론적으로 최종적으로 화면에 보일 fragment를 생성한다