

렌더링 파이프 라인.

렌더링은 원래 CPU에서만 처리하고 있었으나, GPU가 추가되면서 이를 이용한 하드웨어 가속이 가능해졌다.

동시에 렌더링 파이프라인이라는 개념이 쓰이게 되었고, 여러 그래픽 API의 기반이 되었다.

CPU가 GPU에 명령을 보내어, 각각 자신의 일을 병렬적으로 처리한다.

GPU에는 전용 메모리인 VRAM이 존재하는데, 여기에는 텍스처나 메시 데이터 혹은 버퍼 등이 저장된다.

-기기별로 하나의 물리적인 RAM을 CPU+GPU가 나눠서 쓰기도 한다.

렌더링 파이프라인

- 래스터(Raster) : 화상정보를 표현하기 위해 이미지를 2차원 배열 형태의 픽셀로 구성하여 표현하는 것.

즉; 연속된 픽셀들의 집합.

- 렌더링 파이프 라인 : 3차원 이미지를 2차원 래스터 이미지로 표현하기 위한 단계적 방법

* 오브젝트를 렌더링 하기 위해서는 많은 데이터가 필요하다.

1. 오브젝트의 입체적 형태를 나타내는 메시 정보
2. 텍스처 정보
3. Lighting
4. Transform
5. 기타

이러한 정보들을 이용하여 오브젝트가 렌더링 되는 과정을 렌더링 파이프라인이라고 함.

렌더링 파이프 라인은 크게 3가지로 나눌 수 있다.

1. 애플리케이션
2. 지오메트리
3. 래스터라이즈

1. 애플리케이션 스테이지

애플리케이션 상에서 처리되는 단계를 말한다.

해당 단계는 보통 CPU에서 처리하는 단계이기 때문에 엄밀히 따지면, 렌더링 파이프라인이라고 하기에는 애매한 단계이다.

이 단계에서 **컬링**을 사용하여 렌더링 될 오브젝트를 선별한다.

배칭도 이 단계중 하나라고 볼 수 있다.

2. 지오메트리 스테이지

버텍스, 폴리곤을 처리한다.

vertex transform / vertex shader / geometry

버텍스 트랜스폼 - VRAM에 저장된 버텍스 데이터는 단순히 메시 형태만 반영되어 있는 데이터이다.

모든 버텍스와 노멀을 월드 좌표계로 옮겨 주는 월드 트랜스폼,

다시 이것을 카메라 스페이스로 이동시켜주는 뷰 트랜스폼,

이것을 2D 상의 위치로 매칭 시켜주는 프로젝션이 있다.

월드 트랜스폼 -> 뷰 트랜스폼 -> 프로젝션

버텍스 셰이더 - 트랜스폼 변환들은 여기서 일어난다.

버텍스들을 처리하는 셰이더이다. 버텍스 셰이더는 버텍스의 위치 변환이나 노멀 및 컬러도 결정함

지오메트리 생성

- 버텍스 셰이더가 트랜스폼을 결정하고 나면, 버텍스들을 연결하여 도형을 만든다.

이때 생성된 메시 도형을 지오메트리라 부른다.

최근에는 셰이더로 지오메트리 생성에 관여해 테셀레이션을 수행할 수 있다.

- 파이프 라인에 입력된 버텍스들은 모두 지오메트리화 된다.

따라서, 카메라 밖에 있어서 렌더링 되지 않는 버텍스들도, 지오메트리화 되기에 카메라 영역 밖에 있는 오브젝트들이 렌더링 파이프라인을 거치지 않도록 하는 컬링이 필요함.

래스터라이저 스테이지

메시의 폴리곤을 픽셀로 매핑시키는 과정을 rasterization이라고 한다.

래스터라이제이션이 끝나면 최종적으로 각 픽셀의 최종 색이 결정된다.

프래그먼트 (픽셀) 셰이더에서 결정된 픽셀은 RGBA외에도 픽셀과 카메라의 거리를 별도로 저장하며 이때 저장되는 버퍼를 뎁스 버퍼라고 한다.

뎁스 버퍼를 통해 픽셀을 렌더링할 때 깊이 판정을 수행한다. (뎁스 테스트)

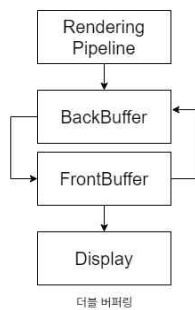
현대에는 프래그먼트 셰이더로 넘어가기전에 뎁스 테스트를 수행해 필요없는 정보를 걸러냄

*프래그먼트 셰이더를 픽셀 셰이더라고 하는데,

엄밀히 말하면 프래그먼트는 최종 픽셀이 되기 전의 데이터라고 함.

픽셀들의 최종 렌더링 색을 결정함.

랜더루프



프레임이 시작되고 로직을 처리한다.

컬링 연산을 수행해 불필요한 렌더링 부하를 방지한다. (유니티는 기본적으로 프러스텀 컬링 수행)

백 버퍼에 씬을 렌더링한다. 각 오브젝트마다 드로우콜이 발생하며 버텍스셰이더와 프래그먼트 셰이더와 같은 GPU 파이프라인을 거친다.

포스트 프로세싱을 수행한다. 포스트프로세싱 역시 버퍼에 렌더링을 하는 것이므로 한 번 이상의 드로우 콜이 발생한다.

백 버퍼와 프론트 버퍼를 스왑한다.