

렌더링 파이프라인

2022년 5월 16일 월요일 오후 4:28

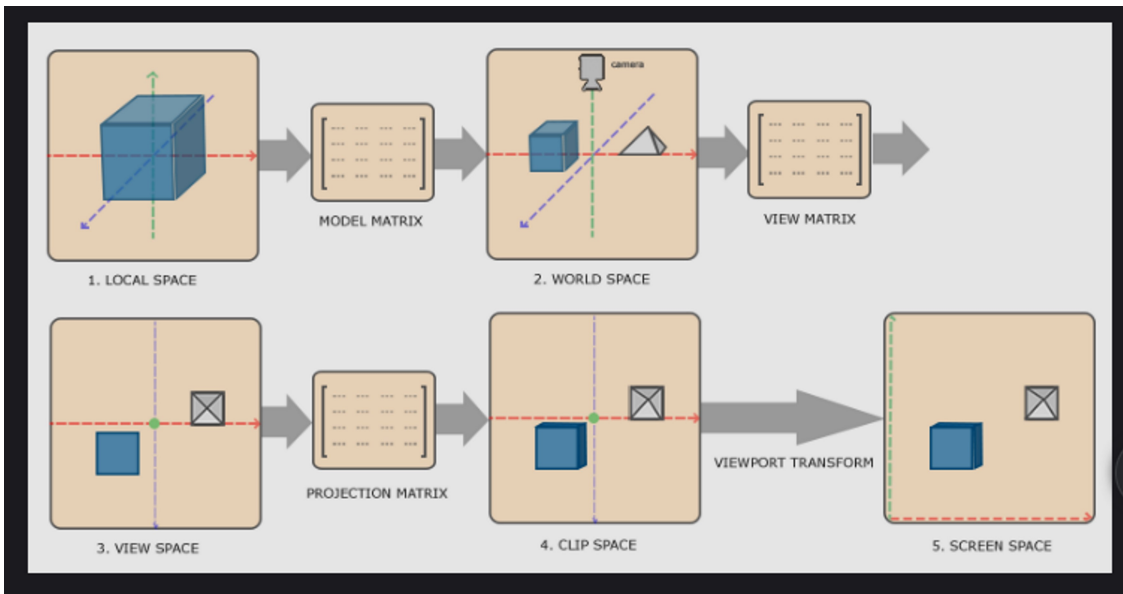
1. 입력조립 : CPU가 GPU로부터 정점 데이터를 전달 받아서 프리미티브(삼각형)들을 만든다.
2. 정점 셰이더 : Object Space에서 Clip Space까지 정점들의 공간 변환을 수행한다.
3. 래스터라이저 : Clip Space의 정점 데이터를 Viewport로 변환하고, 정점 데이터를 기반으로 보간된 프래그먼트(픽셀 데이터)를 생성한다.
4. 픽셀(프래그먼트)셰이더 : 프래그먼트를 입력 받아 화면에 그려질 모든 픽셀의 색상과 깊이 값을 출력한다.
5. 출력 병합 : Z-Test, Stencil Test, Alpha Blending등을 통해 최종적으로 화면에 그려질 색상을 결정한다.

입력 조립 (Input Assembly)

- A. 렌더링 파이프라인의 첫 단계
- B. 정점 데이터를 CPU에서 CPU로 운반하기 위한 자료구조를 정점 버퍼라고 함.
- C. 정점 버퍼는 위치, 노말, 색상, UV를 담고있음.
- D. GPU에서는 이렇게 전달 받은 정점 버퍼를 벡터 명세를 통해 정점 데이터(구조체)로 조립함.
- E. 삼각형과 같은 기본 도형(프리미티브, primitive)로 조립하기 때문에 입력 조립 단계(Input Assembly)라고 함.
- F. 조립된 프리미티브가 정점 셰이더의 입력이 됨.

정점 셰이더(Vertex Shader)

- A. 정점 데이터를 입력 받아 공간 변환을 수행함.



B. 공간 변환 과정

- a. 공간 변환은 각각의 행렬 연산을 통해 이루어짐.
- b. 변환 행렬은 총 3가지로 Model(M), View(V), Projection(P)
- c. Model(M) : 로컬, 월드 공간, 이동, 회전, 크기 등을 통해서 변환이 이루어짐 (T, R, S 순서로 곱)
- d. View(V) : 카메라 공간
- e. Projection(P) : 투영 (클립) 공간, 카메라 기준의 정점 위치를 화면에 보이기 위한 정점 위치로 변환한다.
 - a) 렌더링 영역의 절두체가 정의됨
 - b) 절두체는 Near, Far, Field을 통해 정의
 - c) 절두체를 완전히 벗어난 곳은 모두 버려짐
 - d) 직교 투영, 원근 투영이 있음
 - e) 클립 공간은 4D 좌표계이다. 이를 동차 좌표계라고 함 (x,y,z,w w가 1: 점 0: 벡터)

래스터라이저(Rasterizer)

- A. 하드웨어 자체 알고리즘을 통해 동작됨
- B. 프로그래밍이 불가, 고정 파이프라인 단계
- C. 클립 스페이스(투영)의 정점 데이터를 전달 받아 프래그먼트를 구성하고 화면에 출력할 픽셀을 찾는다.
- D. 픽셀 색상의 데이터는 정점의 데이터를 기준으로 보간된다.

E. 프래그먼트(Fragment)란, 픽셀 하나의 색상을 화면에 그려내기 위한 정보를 담고 있는 데이터이다.

F. 레스터라이저의 역할

i. 클리핑

- a) 버텍스 셰이더의 마지막 단계에서 절두체에 벗어난 정점들은 버려졌지만 걸쳐있는 건 아직 버려지지 않음
- b) 걸쳐있는 폴리곤을 지우기, 절두체 외부, 내부 영역을 분리하여 외부 영역은 버린다.

ii. 원근 분할

- a) 클립 스페이스 좌표의 모든 요소를 w 값으로 나누게 되는데, 이를 통해 모든 원근법 구현이 완료되면 이를 원근 분할이라고 한다.

iii. 후면 컬링

- a) View 벡터와 Normal(법선) 벡터의 관계를 통해 후면을 찾아내어 렌더링 되지 않도록 한다. (View, Normal 내적 각도를 통해서 포함되어 있는지 아닌지 체크)

iv. 뷰포트 변환

- a) 3D 공간 상의 좌표를 2D 스크린 좌표로 변환

v. 스캔 변환

- a) 프리미티브(도형, 삼각형 등)를 통해 프래그먼트를 생성하고 프래그먼트를 채우는 픽셀을 찾는다.
- b) 각 픽셀마다 정점 데이터(위치, 색상, 노멀, UV)들을 보간하여 할당한다.

픽셀 셰이더(Pixel Shader)

1. DirectX에선 픽셀 셰이더, OpenGL과 유니티에선 프로그래머트 셰이더
2. 화면에서 차지하는 픽셀의 개수만큼 픽셀 셰이더가 실행됨
3. 투명도, 라이팅, 그림자, 색상을 모두 픽셀셰이더가 결정함.
4. 픽셀 셰이더는 각 픽셀들의 색상과 깊이 값을 출력으로 전달한다.
5. 깊이 값은 Z-Buffer, 색상 값은 Color Buffer에 저장된다.
6. 버퍼는 텍스처, 이런 버퍼들을 통칭하여 스크린 버퍼(Screen Buffer) 이라고 한다.

출력 병합(Output Merge)

1. 렌더링 파이프라인 마지막 단계
2. 픽셀들을 화면에 출력하기 위한 마지막 연산을 수행
 - A. Z-Test
 - B. Stencil Test
 - C. Alpha Blending
3. 각각의 픽셀 위치마다 여러 오브젝트의 픽셀이 겹쳐있을 수 있기 때문에 겹치는 픽셀들을 연산 및 판단하여 최종적인 색상을 결정한다.
4. 프레임 버퍼 (Frame Buffer)
 - A. 한 프레임의 스크린 버퍼
 - B. Color Buffer : 색상 값 텍스처
 - C. Z-Buffer : 깊이 값 텍스처
 - D. Stencil Buffer : 픽셀을 렌더링 또는 폐기하기 위한 마스크 텍스처