

# 그래픽스 관련 용어

2021/5/8

류한진

# Projection

- Perspective
  - 원근감을 갖는 투사
  - 주로 시점의 각도변환이 잦거나 z 거리감이 중요한 경우 사용
  - 대부분의 컴퓨터 3D 렌더링은 이 방법을 사용하는 것이 일반적
  - 적용되는 투사행렬에 따라 보이는 방식이 다름 (주로 절두체 행렬)
- Isometric (Orthographic)
  - 원근감이 없는 투사
  - 주로 시점의 각도변환이 없는 경우 사용
  - 멀리 있는 물체도 크기가 같아 물체 간의 크기를 왜곡없이 확인가능

# Sprite vs Texture

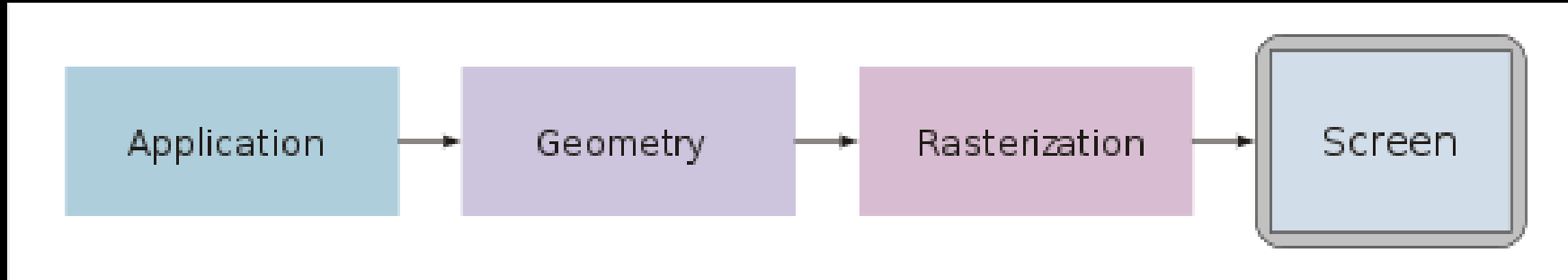
- 스프라이트

- 3D이든, 2D이든 평면으로 존재하는 객체 그 자체
- 스프라이트는 간단한 이미지와는 다르게 개별적으로 메타데이터를 가지고 월드에 참여할 수 있다. (위치, 회전, 스케일 등)

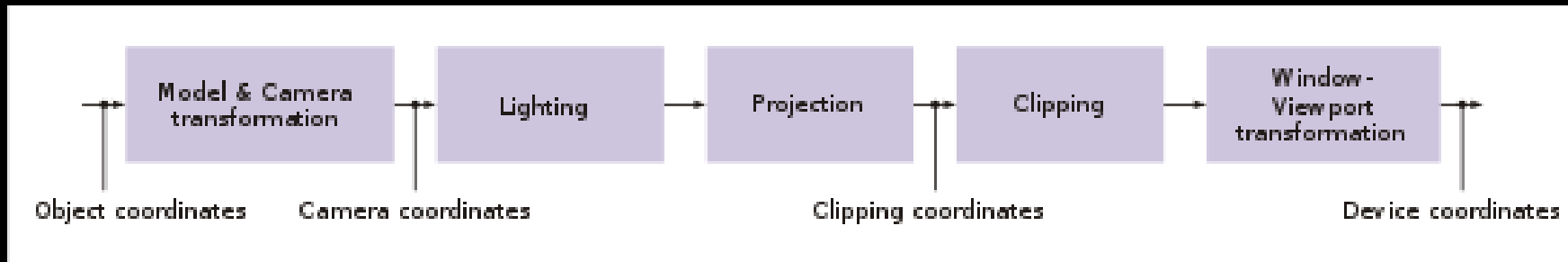
- 텍스처

- 특정 객체에 덮여질 객체의 구성품 (주로 이미지)
- 개별적으로 월드에 참여하지 못하고 다른 객체에 덮여서 사용
- 그래픽 파이프라인에서 주로 2차원 이미지로 활용되지만 셰이더가 사용하는 모든 구조화된 데이터를 통용하기도 함.

# Rendering pipeline



- 렌더링은 크게 3가지로 나뉜다.



- 그중 Geometry 구간의 작업을 주로 그래픽 프로그래밍으로 취급한다.
  - Application은 그래픽이 발생하는 원인, 즉 프로그래밍이 진행되며 요구되는 그래픽 연산의 원인이라 할 수 있다.
  - Rasterization은 자동적으로 하드웨어에 맞추어 처리된다.

# Rendering pipeline

Geometry 구간의 작업은 셰이더라는 소형 병렬 프로그램들에서 처리된다.

- **버텍스**
  - 정점들과 그 위치, 노말 등을 받아들인다.
  - 버텍스 셰이더는 받아들인 정보를 가공한다. (반필수)
- 테셀레이션
  - 정점으로 이루어진 패치를 프리미티브로 나눈다.
  - 주로 삼각형을 사용한다
  - TCS에서 처리수준 결정, TES에서 처리된 결과에 대한 작업 명시
- 지오메트리
  - 테셀레이션에서 나뉜 프리미티브들이나 그대로 온 패치들을 받아들여 각각의 정점들을 처리한다.
- 래스터라이징
  - 자동적으로 수행됨
- **프래그먼트(픽셀)**
  - 화면상에 출력되는 픽셀들에 하나하나 실행되는 셰이더
  - 색데이터를 다루기에 가장 표시가 크게나는 셰이더이다.