정점(vertex)

* 다양한 정보를 가지고 있는 점(위치, normal, UV 등등[위치는 필수이지만 나머진 필요에 따라 사용.]).상황에 따라 다름

엣지(edge)

* 정점 두 개를 연결한 것.

삼각형, 폴리곤, 페이스

* 정점 세 개를 연결해서 만든 삼각형

메시(Mesh)

* 삼각형들을 모아 만든 형태

// 이 항목까지 외형의 담당

조명(Light)

* 종류[아래의 세 종류는 3D그래픽스에서 항상 포함된 종류]
  + Directional Light : 특정 방향으로 평행하게 빛을 내보내는 광원(방향) ex)햇빛
  + Point Light : 특정 위치에서 특정 반지름만큼 모든 방향으로 빛을 균등하게 내보내는 광원(위치, 반지름)
  + Spot Light : 특정 위치에서 특정 방향으로 바라보는 원뿔 모양으로 빛을 내보내는 광원(위치, 방향, 반지름, 원뿔각도(내각, 외각))
  + Ambient Light : 환경광. 씬의 모든 오브젝트에 동일하게 영향을 끼치는 전방향, 고정강도, 고정 색상의 광원[조명이라기보단 전체적인 톤을 조절할 때 사용
  + Area Light : 사각형 조명. 일정 영역이 빛나는 것 처럼 보인다.

셰이더(Shader)

* 컴퓨터 그래픽스에서 3D 장면의 음영계산, 특수효과, 화면 후처리 등의 작업을 수행하기 위해 사용하는 소프트웨어 명령의 집합
* GPU의 프로그래밍 가능한 랜더링 파이프라인을 사용하는 소프트웨어 명령의 집합

Unity 셰이더용 언어

* HLSL(High Level[레벨이 높을수록 사람이 알아보기 쉬워짐] Shading Language), CG(C for Gaphics)
* Surface shader
* Shader Graph

셰이더의 종류

* 픽셀 셰이더(Pixel Shader) -> 프래그맨트 세이더 (Fargment Shader)
  + 스크린에 최종 출력되는 픽셀들의 색상이나 다른 특성들을 계산하여 변경하는데 사용되는 셰이더
* 정점 셰이더(Vertex Shader)
  + 3D 모델이 가지는 각 정점을 입력으로 받은 다음 변경하는데 사용되는 셰이더
* 테셀레이션 셰이더
  + 필요에 따라 폴리곤을 더 작은 폴리곤으로 조각내어 여러 이득을 얻는 기법
* 지오메트리 셰이더
  + 정점 셰이더가 수행된 이후 추가적으로 점, 선, 삼각형 같은 프리미티브를 생성

랜더링

* 3D월드를 이미지로 변환하는 작업

랜더링 파이프라인

* 3D오브젝트들로 구성된 월드를 디스플레이에서 출력하기 위해 2차원 래스터 이미지로 변경하는 단계적인 작업

Built-in : 기존의 유니티 랜더링 파이프라인

SRP(Scriptable Rendering Pipeline) – URP, HDRP

Forward Rendering : 고전적인 랜더링(Diffuse, Specular, Ambient)

Physically Based Rendering : 물리기반 랜더링(Albedo, Metallic, Smoothness)

LOD(Level of Detail)

* 3D오브젝트를 표현할 때 가까이 있으면 하이 폴리곤 모델을 사용하고 멀어질수록 로우 폴리곤 모델을 사용하는 방법ㄴ