**OpenGL 초기화**

OpenGL의 초기버전에서 사용한 함수 세트는 현대 OpenGL의 함수 세트와는 매우 다르다.

OpenGL은 3개의 프로파일을 지원한다.(코어, 호환성, ES)

파라미터 설정을 위해 SDL\_GL\_SetAttribute(설정할 속성,해당 속성의 값); 함수를 호출한다.

Context는 OpenGL이 인식하는 모든 상태나 오브젝트를 포함하는 OpenGL의 세계이다. 색상깊이, 로드된 이미지나 모델, 그리고 여러 다양한 OpenGL 오브젝트를 포함한다.

윈도우를 생성하고 제거하는 것처럼 OpenGL 콘텍스트를 소멸자에서 제거해야 한다.

**삼각형 기초**

현대의 그래픽 하드웨어에서는 폴리곤 렌더링이 매우 효율적이어서 2D이든 3D이든 최근의 모든 게임에서는 궁극적으로 폴리곤을 사용한다.

다른 3D 그래픽 테크닉과 비교해보면 폴리곤은 런타임 시 많은 계산이 필요하지 않다.

크기를 가변적으로 조절할 수 있다.

대부분의 3D오브젝트를 폴리곤으로 표현할 수 있다.

그 중 삼각형은 3개의 버텍스만 필요로 하고 그렇기 때문에 한 평면안에만 놓일 수 있다.

삼각형은 쉽게 테셀레이션이 가능하다(복잡한 3D물체를 여러 개의 삼각형으로 쉽게 나눌 수 있다.)(다른 폴리곤(ex. 사각형)이 한 평면상에 놓일 수 있다면 다른 폴리곤도 사용 가능)

정규화된 장치 좌표

OpenGL에서는 정규화된 장치 좌표를 사용한다. 윈도우의 중심이 장치좌표의 중심

버텍스 버퍼와 인덱스 버퍼

여러 갱의 삼각형으로 구성된 3D 모델이 있다고 하면 메모리상에 저장해야 하는데 가장 간단한 방법은 인접한 배열이나 버퍼 형태로 직접 저장하는 것이다.

하지만 버텍스의 배열은 일부 중복 데이터를 갖고 있다. 해결 방법으로 버텍스 버퍼에는 3D 기하에서 사용한 고유한 좌표만을 담고 버퍼에 인덱스를 붙인다. 인덱스 버퍼는 인덱스 3개로 구성된 개별 삼각형 정보를 배열 형태로 저장한다.

**셰이더**

현대의 그래픽스 파이프라인은 단순히 버텍스/ 인덱스 버퍼만 제공받아서 삼각형을 그리는 것이 아니라 삼각형은 고정된 색상으로 그릴 것인지, 텍스처에서 얻은 색상을 버텍스에 사용할 것인지, 그려야하는 모든 픽셀에 광원 계산을 할 것인지 등 어떻게 버텍스를 그려야 할지 지정하는 작업이 필요하다.

셰이더 프로그램은 그래픽 하드웨어 상에서 특정 태스크를 수행할 때 실행되는 작은 프로그램으로 자신만의 메인 기능을 가진 별도의 프로그램이다.

버텍스 셰이더

버텍스 셰이더 프로그램은 그려질 모든 삼각형의 모든 버텍스에 대해 한 번 씩 실행된다.

버텍스 셰이더는 입력으로 버텍스 속성 데이터를 받아서 적절하게 수정한다.

프래그먼트 셰이더

삼각형의 버텍스가 버텍스 셰이더를 거친 후에 OpenGL은 삼각형에 해당하는 픽셀이 어떤 색상을 가지는지를 결정해야 한다. 삼각형을 픽셀로 변환하는 과정을 래스터 변환이라 한다.

프래그먼트 셰이더의 역할을 각 픽셀의 색상을 결정하는 것이다. 따라서 모든 픽셀마다 한 번씩 실행된다. 이 색상은 텍스처나 색상, 재질 같은 표면 속성을 고려해서 결정된다.조명이 존재한다면 광원계산도 고려해야한다.

셰이더 로딩

고수준 레벨에서는 다음과 같은 단계를 수행해야 한다.

1. 버텍스 셰이더를 로드하고 컴파일 한다.
2. 프래그먼트 셰이더를 로드하고 컴파일한다.
3. 2개의 셰이더를 ‘셰이더 프로그램’에 서로 연결시킨다.