For The CG Artists

First Edition

By Patrick Song

Supervisor
Yong-jae Nam
Myoung-Chul Kim

CURSE OF ANGEL

Contents

Introduction

CG란?

2D 3D공통

색

삼원색

CG의 방식

Animation

2D

Vector & Raster

Resolution

Antialiasing

Layer

Channel

3D

Modeling

Lighting

Camera

Mapping

Rendering

참고문헌

책

Internet Web Site

Introduction

CG란?

CG란 무엇인가?

CG란 Computer Graphics의 준말이다. Computer로 그린 그림이란 소리다. 그림은 즉 미술이다. 미술은 인간이 문명을 가지기 시작하면서 시작된 것이라고 한다. 그때에는 제대로 된 언어체계가 없었으므로 그림으로써 기록을 남겼고, 그것이 계속 발전하면서 미학적인 개념이 추가되며, 현대의 CG까지 넘어오게 된 것이다.

CG는 Digital이라는 개념에서 중요하다. 일반적으로 사람이 손으로 그리는 그림들은 Analogue이다. 이것들은 사람의 손의 미세한 힘의 차이로 생동감을 느끼게 해줄 수는 있지만, 그것을 변형, 복제하기는 매우 곤란하다. 가령 그림을 그렸는데 좀더 밝게 하고 싶은 충동을 느꼈을 때 손으로 그린 그림들은 전부 다시 그려야 하는 경우까지 발생하지만, CG라면 몇 번의 마우스 클릭만으로 변화를 줄 수 있는 것이다. 정말 혁신적인 것이다.

CG를 하는 이유는 미술과 마찬가지로 무엇인가를 표현하고 싶다거나, 현실에서 존재하기 불가능한 것을 창조하기 위해서이다. 그러나 CG에서는 후자가 더 강한 목적을 띈다. 최근에 CG가 가장 두드러지게 사용되는 곳은 영화의 F/X이다. 여기서는 현실에서는 촬영이 거의 불가능 한 것을 CG로써 제작하여 합성하는 것이다.(93년에 나왔던 Jurassic Park에서 T-Rex가 날뛰는 장면에 많은 사람들이 충격을 받았었다. 65000년 전의 과거를 그대로 재현했다고 날뛰면서 말이다.)

왜 최근에 CG가 붐을 일으키는가?

요 근래 몇 년 사이에 갑자기 CG의 붐이 일었다. Programming만 가르쳤던 Computer학원들이 죄다 CG를 가르치기 시작했고, 짜증이 날만큼 학원에서 광고성 전화가 온다. 왜 이렇게 갑자기 붐을 일으키는 것인가? 이 이야기를 하려면 Windows 95의 출현을 빼놓을 수 없다. 예전 DOS의 CUI(Characteristic User Interface)에서 GUI(Graphical User Interface)로의 변화였으니 말이다. 이때부터 사람들은 예전에는 꿈도 꿀 수 없었던, Computer의 바탕화면을 꾸미기 시작했으며, Game을 할 때도 Game의 재미뿐만 아니라, Graphic의 Quality를 따지기 시작했다. 이것은 컴퓨터의 성능이 발전하였기에 가능한 변화였다. 예전의 PC에서는 아주 간단한 Image하나를 생성하는데도 속도가 느렸기에 짜증이 났었다. 그리고 말이 생성이지, 일명 'Dot 노가다'라고 불리는 것인데, 점을 하나씩 찍어가며 Image를 만들어 갔다. 그러나 성능이 좋아짐에 따라 점을 하나씩 찍는 것이 아니라 색칠을 하며, 명암을 줄 수 있게 되었다. Internet의 확산 또한 중요하다. Internet은 HTML기반으로 글뿐만 아니라 그림이 같이 뜬다. 그러기에 사람들은 간편하게 Computer로 Image를 다운 받을 수 있게 되었으며, 그것을 넘어서, 개인이 Homepage를 구축하기도 하여, 그것을 위해서 Homepage에 넣을 Image를 만들기 위해서 CG를 배우는 사람이 늘어났다. 그리고 기업체 등의 Homepage를 제작하기 위해서 Web Designer라는 신종 직업까지 탄생하게 되었다. 마지막으로 Computer의 용도의 확대를 꼽을 수 있다. 예전에 Computer는 연구소에서는 산술계산을 빨리 하기 위해서, 기업체에서는 DB(Data Base)를 구축하기 위해서 사용되었다. 개인이 쓰는 경우는 많이 희귀한 경우였다. 누구 집에 Computer있다 하면 그 집에는 항상 사람이 많았다. 뭐… 그때도 게임만했지만. (개인적인 사담인데, 당시 유행했던 Price of Persia 나 Prehistory는 명작이었다.) 하튼 지금은 PC가 거의 생활화 되어있다. 집집마다 PC가 거의 한대씩은 있으며 심한 경우는 집안에 PC가 여러 대여서 Network까지 깔려있다. (우리 집이 그렇다.) 이렇게 PC가 생활 속에 자리 잡았으며, 아름다움을 추구하는 인간의 본능이 CG에 대한 붐을 일으켰다고 생각된다.

2D 3D 공통

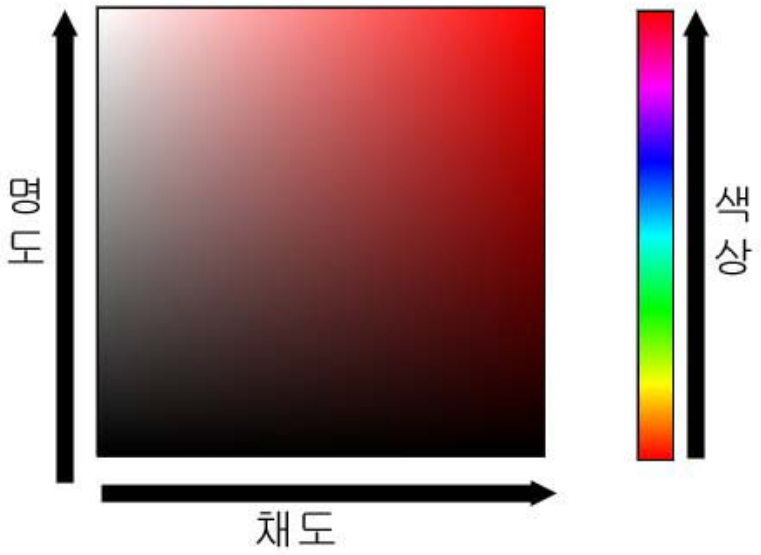
색

색이란?

색이라는 건 일단 빛이 물체에 부딪히고, 반사되어 나온 빛이 눈에 들어가서 뇌에서 인지하는 것을 색이라고 할 수 있다. (물론 그 빛의 영역은 가시광선영역이다.)

색의 3요소

색은 3가지로 구성되어 있다. 색상(Hue), 채도(Saturation), 명도(Brightness or Lightness) 이다. 그림으로 보자면 다음과 같다.



색상은 그냥 색이다. 즉 색의 종류를 나타내는 것이다.

채도는 색의 탁하기라고 표현할 수 있다. 즉 채도가 높을수록 색이 진해지며, 낮을수록 색이 옅어지다가 결국에는 흰색, 회색, 검은색(이것은 명도가 높아지면 흰색 낮아지면 검은색이 된다. - 이 색들을 무채색이라고 한다.)이 되는 것이다.

명도는 색의 밝기이다. 명도가 높다면 색이 밝아진다. 반대로 낮아진다면 검은색이 되는 것이다.

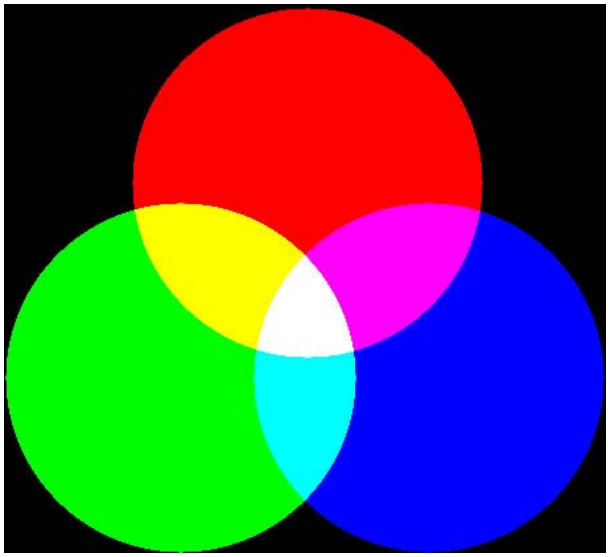
삼원색

삼원색이란?

색은 세가지 색의 조화로써 모든 색을 표현할 수 있다. 그 세가지 색에 대해서 알아보자.

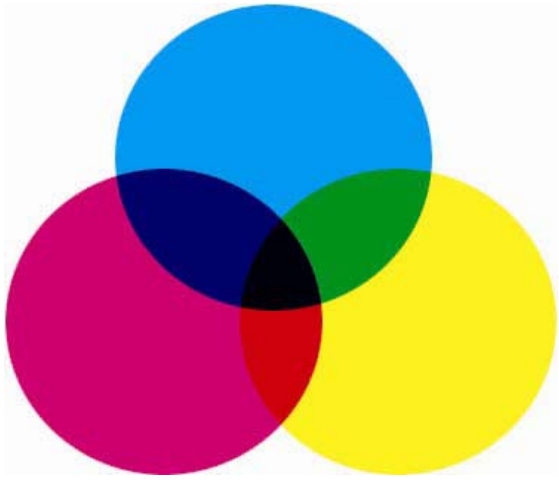
빛의 삼원색 RGB

빛으로써 표현하는 방식 즉 TV나 영사 스크린이나 모니터 등의 경우에 쓰이는 방식이다. RGB라고 말하는데 이것은 Red(빨간색), Green(초록색), Blue(파란색) 이 세가지 색을 지칭하는 말이다. RGB는 빛이기에 색이 섞이면 섞일수록 명도가 증가한다. 이것을 보고 가산혼합이라고 한다. 그림으로 보면 다음과 같다.



색의 삼원색 CMYK

물감이나 프린터 등으로 표현하는 색이다. CMYK라고 말하며 Cyan(밝은 파란색), Magenta(자주색), Yellow(노란색) blacK(검은색) 이렇게 이다. 원래는 CMY까지 이고 CMY를 섞은 값이 이론상으론 검은색이 나와야 정상이나, 실제적으로 해보면 잉크의 화학성분 등의 이유로 어둡고 진한 갈색이 나온다. 그리고 검은색의 출력이 빈번하기에 CMYK로 따로 잉크를 만든다. 그리고 이것은 색이 섞일수록 명도가 감소하기에 감산혼합이라고 한다. 그림으로 보면 다음과 같다.



CG의 방식

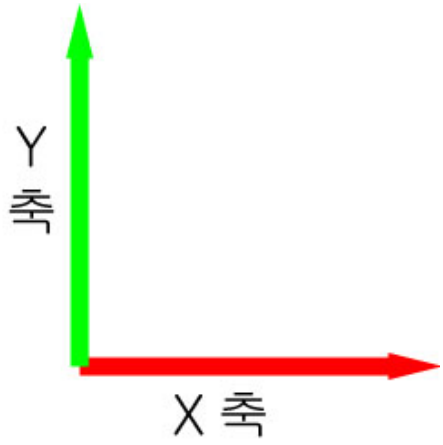
CG의 방식에 대해서

CG에서는 크게 2가지 방식의 방식이 있다. 2D와 3D인데 여기서 D는 Dimension 즉 차원이다. 2차원과 3차원이다.

2D

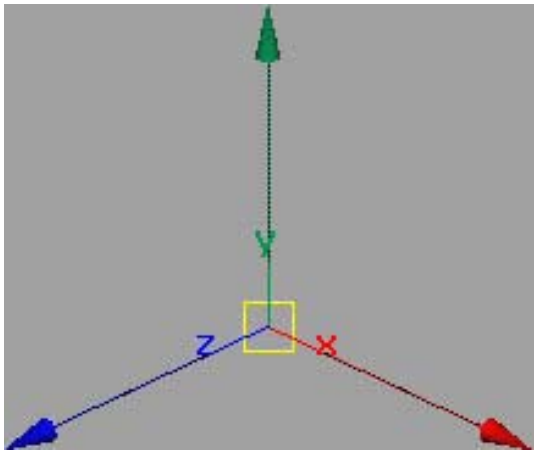
X,Y축 이렇게 두 개의 축을 사용하는 방식이다. 그래서 평면적인 Image가 가능하다.

그림으로 보면 다음과 같다.



3D

X,Y,Z축 이렇게 세가지 축을 사용하는 방식이다. 그래서 입체적인 표현이 가능하다. 그림으로 보면 다음과 같다.



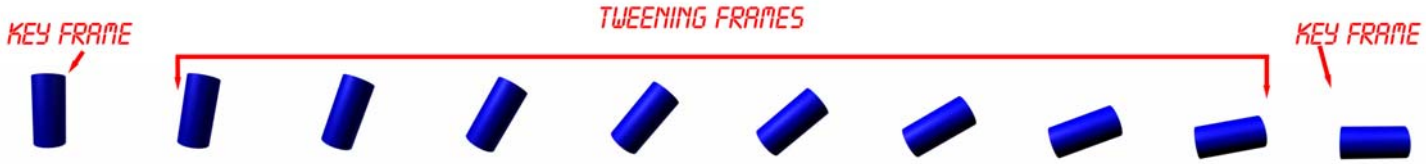
Animation

Animation이란?

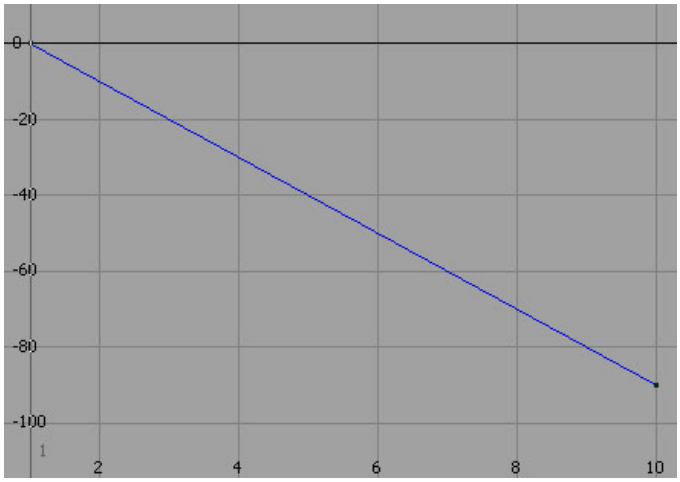
'생명을 불어넣다' 라는 뜻의 animate라는 단어를 명사형으로 만든 것 이다. 즉 생명이 불어넣어져서 움직이는 Image들이라고 볼 수 있다. 그냥 움직이는 그림이라고 생각하면 이해가 쉽다. CG상에서 Animation의 기법은 Sprite와 Vector등이 있으나 결국엔 Tweening이라는 기법을 사용하여 만들게 된다. 그러므로 Tweening이라는 기법에 대해서만 알아보자.

Tweening

In-bewteening 의 약자로 Key frame(여기서 frame은 그림 한 장이라는 뜻이다.)과 Key frame사이를 자동적으로 채워주는 방식이다. 그림으로 예를 들어보겠다.



맨 처음에는 서있는 Cylinder 맨 끝에는 누워있는 Cylinder 이렇게 2개의 Frame을 만들었다. 그러면 그 사이에 8개의 넘어져가는 Cylinder그림을 자동으로 계산하여 만들어 낸다. 이런 것이 Tweening 이다. 그리고 이렇게 만들면 자동적으로 다음과 같은 Graph가 생성된다.



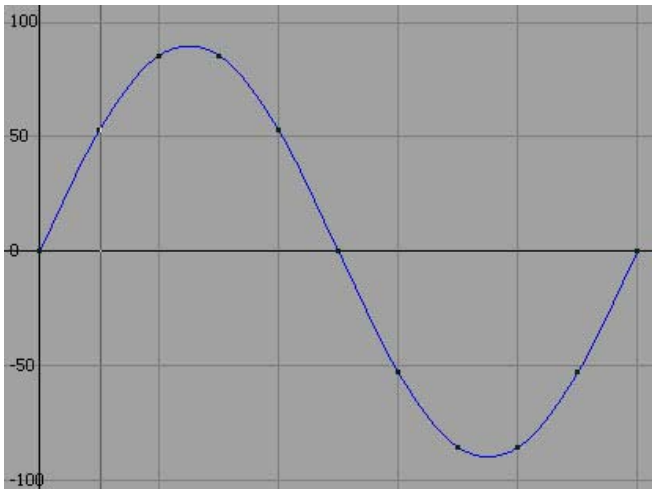
보면 알겠지만 검은 점은 Key frame의 값이고 파란선은 Tweening된 값들이다. 이것을 각 Frame의 수치에 맞게 미분하여, 그 값으로 Image를 출력하는 것이다. 그리고 이 Graph는 Bezier함수를 기본적으로 사용하는데 수정을 가해서 가속, 감속이 되는 Animation을 만들 수도 있다.

이 방법을 응용한 것이 Expression을 사용한 Animation이다. 이것은 그래프에 직접적으로 수학적 공식을 삽입하여, 그래프를 만드는 것이다.

다음과 같은 수식을 했다고 해보자.

pCylinder1.rotateZ = sin(frame*3.14159265358979323846264338327 95/5)*90

그럼 다음과 같은 그래프를 그린다.



어디서 많이 본 그래프다. 중학교 때 배웠던 sine함수 그래프이다. 이렇게 수학식을 사용하여 Animation이 가능하다. (이런 것 써서 어디다 써먹느냐고 물어보는 사람이 있을 수도 있기에 사족을 붙인다. 자연계의 움직임에는 의외로 sine의 움직임이 많다. 사람 숨쉬는 것도 sine곡선의 움직임이다. 공이 튀는 것도 sine함수를 변형하면 충분히 가능하다.)

FPS

이것은 CG뿐만 아니라 영화나 TV에서도 쓰이는 말인데, 동영상은 그림 여러 장을 순차적으로 나오게 함으로써 잔

상효과를 이용하여 움직이는 것처럼 보이게 하는 것이다. 이때 초당 나오는 그림의 수를 FPS(Frame Per Second)라고 한다. 정해진 규격 FPS로는 일단 영화에서는 24fps 우리나라와, 미국 등의 TV규격인 NTSC은 29.97fps 유럽쪽 TV 규격인 PAL은 25fps 일반적인 컴퓨터 동영상은 15fps이다.

2D

Vector와 Raster

Vector

Vector는 기하학적으로 그림을 표현하는 것이다. 즉 수학적인 연산으로써(그래프를 그린다.)그림을 그리는 것이다. (보통 Bezier 함수를 사용한다.) 그러기에 확대를 시켜도 수학적인 계산이 계속 되며 그림이 커지기에 그림이 깨지는 현상이 없다. 0과 1사이에 무수한 숫자가 존재한다는 것을 안다면 쉽게 그 원리를 알 것이다. 확대를 시켜도 깨지는 현상이 없다는 강력한 장점은 있지만, 대신에 사람 손으로 직접 제작하여야 하며, 도형이 많아지면 많아질수록 용량이 커진다는 단점이 있다. 그래서 기업의 로고와 캐릭터 일러스트 같은 곳에 주로 쓰인다.

Raster

Bitmap이라고도 한다. 즉 작은 사각형들(Pixel)들의 모임으로써, 그림이 이루어 지는 것이다. 그래서 확대를 시키면 그림이 깨지게 된다. 대신에 많은 색을 한꺼번에 표현이 가능하므로, 사진 등에 적합하다.
(* Pixel이란- Picture + Element 라는 두 단어의 합성어이며 Raster Image의 기본 단위이며, 점(사각형)하나를 나타낸다.)

Resolution

Resolution이란?

우리말로 해상도라고 한다. 이것은 Raster방식의 Image에서만 쓰이는 말이다. Resolution이라는 것은 쉽게 말해 Pixel의 수다. 표현방법은 가로Pixel수 * 세로Pixel수 이다.

규격 Resolution

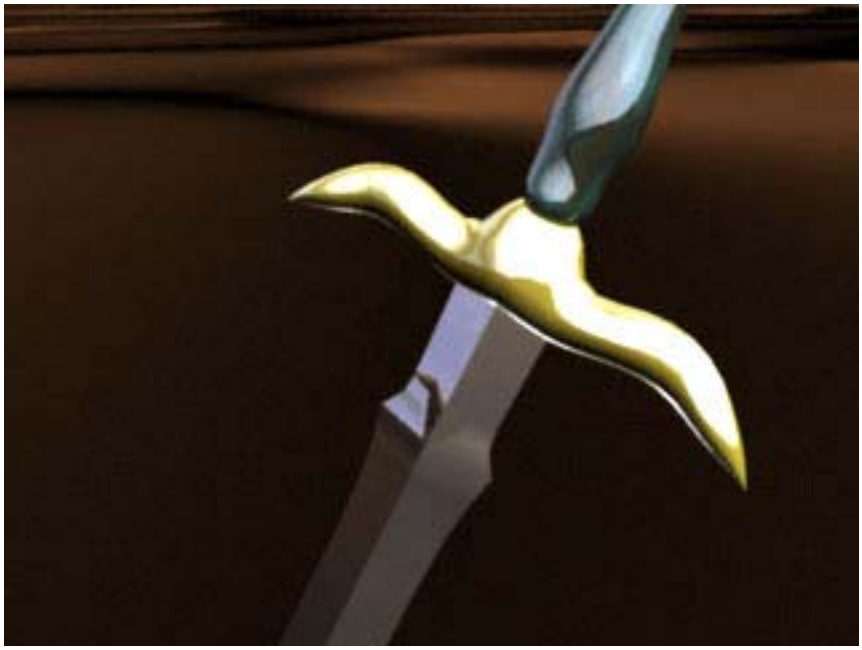
Resolution에도 정해진 규격이 있다. NTSC는 720*480(이나486) PAL은 768*576이다. 이외에도 정해진 규격의 Resolution이 많이 있다.

Anti-aliasing Technology

Anti-aliasing이란?

이것은 Raster Image에서만 적용되는 기술이다. Raster Image는 Pixel구조이기에 필연적으로 계단현상(aliasing)이 발생한다. 그림과 같은 현상이다.

이것을 제거(라기 보단 좀 덜 보이게 해주는)해주는 기술이 Anti-aliasing 기술이다. 이 기술을 적용한 그림은 다음과 같다.



확대하면 분명히 계단현상이 나타나기는 한다. 그러나 그냥 보기에는 적용되지 않은 Image에 비해 훨씬 더 깔끔하다. 이래서 이 기술의 필요성이 있는 것이다. 이 기술의 원리는 의외로 간단하다. Pixel과 Pixel사이에 중간 값을 채워 넣는 것이다.

Layer

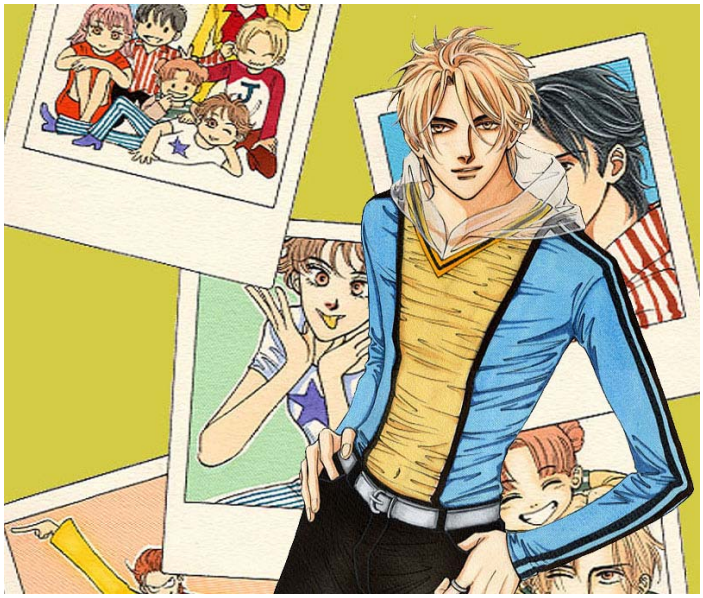
Layer란?

Layer란 사전적 의미로 '층'이라는 뜻을 가졌다. 말 그대로 Image들을 층층이 쌓아 올려 결국엔 하나의 Image를 만드는 것이다. Cell Animation의 기법을 생각하면 이해하기 쉽다. Cell Animation은 투명한 Film에다가 그림을 그리고 그것을 겹쳐서 현상하여 최종 Image들을 만들어낸다.

Layer에서는 Layer의 순서를 바꾸거나 몇 개의 Layer를 보이지 않게 하며 불투명도(Opacity)를 주는 등의 효과를 주어 여러 가지 효과를 줄 수 있다. 그림을 보자.

이와 같은 그림이 있다. 여기서 Layer의 순서를 바꿔 보겠다.

이렇게 사람의 위치가 바뀌었다. 이제 Layer를 보이지 않게 하겠다.



사람이 사라졌다. 이제 불투명도(Opacity)를 써보겠다.

한 Layer의 불투명도(Opacity)를 50%로 조정하여, 투과되는듯한 Image를 만들었다.
Layer는 즉 여러 그림을 한 Image에 써야 할 때 이런 식으로 쓰는 것이다.

Channel

Channel 이란?

Channel의 개념을 이해하기 위해선 출판용 인쇄의 방법에 대해서 알아야 한다. 일반적인 가정용 PC의 경우 고속의 Color Printer가 있다고 해도 소량의 인쇄용으로 만들어졌기에 한 장 한 장 조금씩 잉크를 뿌려가며 인쇄하는 방법이다. 그러기에 속도가 느리다. 그런데 이런 방식으로 대량으로 인쇄(책이나 광고지 등)하는 곳에서 했다간 정말로 시간 낭비가 심할 것이다. 그런 곳에서는 인쇄를 최소 100장 이상씩 하는데 한 장당 1분씩 걸린다고 해도 100장이면 100분이다. 너무 오래 걸린다. 그래서 CMYK를 분판하여 인쇄를 하는데, CMYK로 분판하면 OHP용지 같은 투명한 Film이 제작된다. 그 Film을 사용하여 찍어내는 듯한 인쇄를 한다. 분판하는 방법은 다음과 같다.

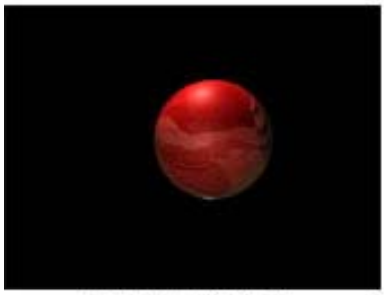
이와 같이 분판을 하고 검은 부분에 색이 들어가도록 인쇄를 하는 것이다. 그렇다고 해서 Channel이라는 것이 출판만을 위해서 있는 건 아니다. Channel은 색의 구조를 보여주기 위한 존재이다. 즉 어느 부분에 얼마만큼의 색이 들어가는지 그것을 판단하기 위한 것이다. (참고로 CMYK의 경우는 감산혼합이므로 검은 부분에 색이 들어가지만 RGB의 경우 가산혼합이므로 흰 부분에 색이 들어가게 된다.)

Alpha Channel

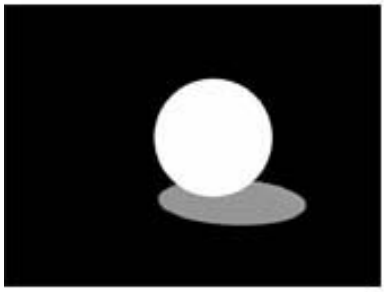
Photoshop의 범주에서 아주 간단하게 말하자면 선택영역을 저장하는 방법이라고 할 수 있다. 즉 Grayscale(색상이 없는, 즉 채도가 0인 색들)로써 선택영역을 지정하는데, 명도가 가장 높은 부분인 흰 부분은 선택되는 부분이며 명

도가 가장 낮은 검은 부분은 선택이 되지 않는 부분이라고 생각하면 된다. 어중간한(?) 명도인 회색부분은 Feather를 준 것과 같은 것으로써 뿌옇게 선택이 되는 것이다. 그러나 이런 식으로 설명을 하면 Alpha Channel 의 개념을 제대로 잡기가 힘들다. 나도 처음에 이렇게 이해했다가 나중에 다른 것을 할 때 이해가 안돼서 큰 고생을 했다. Alpha Channel이란 위에서 설명한 출판 방식의 이론을 다시 언급해서 말하자면, 분판된 Film에 하나의 Film(Alpha Channel)을 더 추가하여 그 부분이 가려져서 인쇄가 안되도록 하는 개념이다. 그러나 출판에서 Alpha Channel은 거의 쓰이지 않고 Photoshop에서 선택영역을 저장하거나 아니면 영화 등에서 합성을 할 때 쓰인다. 간단한 예를 그림으로 보여주겠다.

배경 이미지



합성될 이미지



Alpha Channel

최종 합성된 이미지.

이와 같은 방법으로 합성을 하는 것이다. 배경 Image에 합성될 Image를 놓고 합성될 Image에서 Alpha Channel의 영역을 뺀 부분이 합성이 되는 것이다.

3D

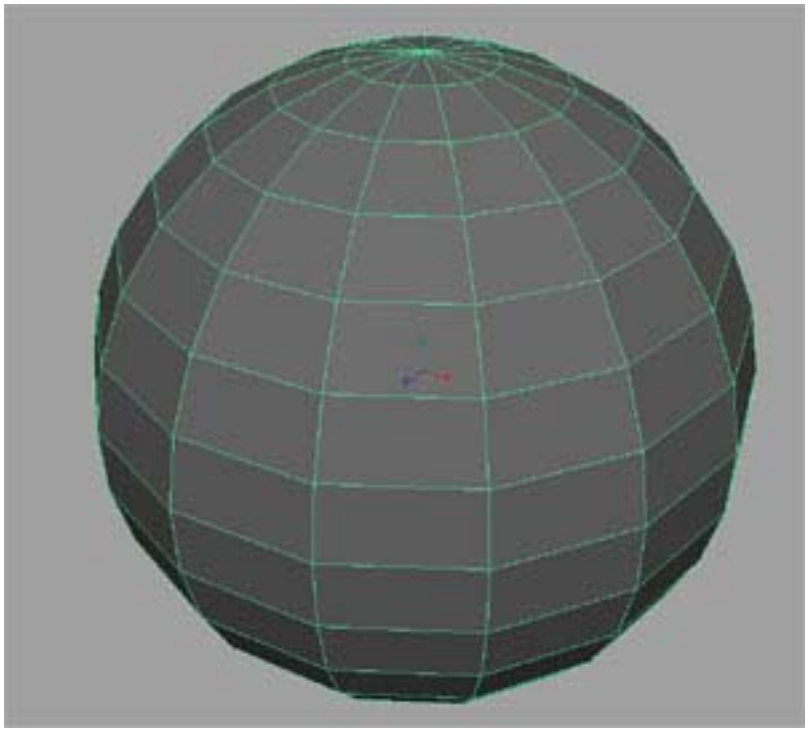
Modeling

Modeling 이란?

간단히 말해서 3D공간상에 물체(Object)를 만들고 그것을 고치는 것을 Modeling이라고 한다. 즉 어떠한 형태를 만드는 작업을 지칭한다. 방식에는 크게 Polygon방식과 NURBS방식이 있으나, 3D 프로그램마다 고유한 방식을 지원하기도 한다. (예: Patch, Subdiv 등)

Polygon

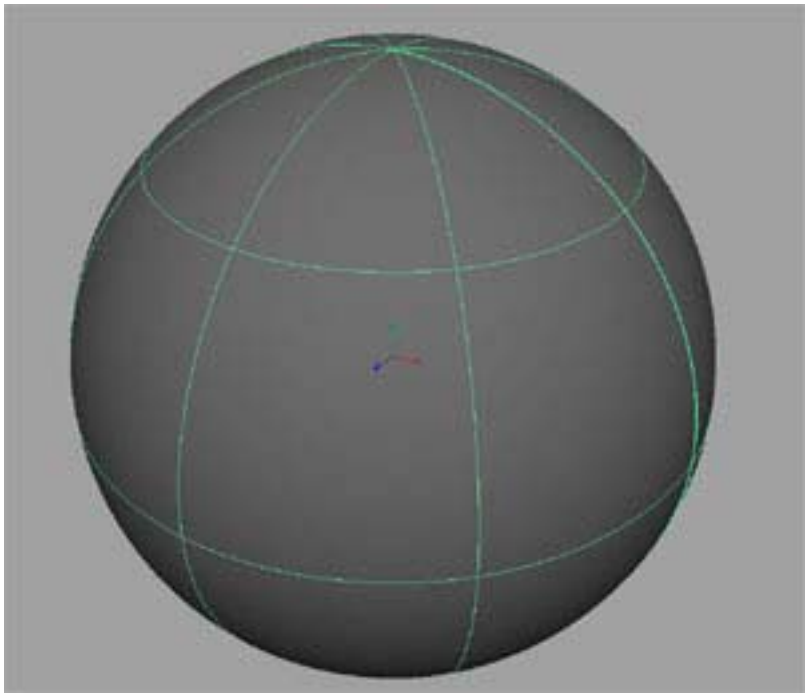
평면 다각형을 계속 붙여가며 도형을 만드는 것이다. 그림과 같다.



이 방식은 속도가 빠르며, Modeling을 하는데 큰 제한이 없다. (어떻게 하든 점의 수만 같으면 면이 붙는다. 점의 수가 같지 않다면 점을 더 만들면 된다.) 그러나 직선적인 면이기에 거친 느낌이 들 수 있다. 그리고 품질을 높이기 위해서는 면이 계속 많아져야 하고, 그러다 보면 Modeling 하기가 상당히 까다로워질 수 있다. 그러나 빠른 속도라는 장점 때문에 실시간 3D게임에서 쓰인다.

NURBS

Non-Uniform B-Spline의 약자이다. Polygon에 비해 엄청나게 복잡한 알고리즘으로 되어있다. 수학적인 연산으로 면(곡면)을 만드는 것이다. 그림과 같다.



여하튼 수학적인 연산으로 곡면을 만들어 면이 부드럽다. 그러나 면과 면을 붙일 때 상당히 까다롭다.(붙인다기보단 이어주는 것이다.)Polygon의 경우 면과 면을 완벽히 붙이면 하나의 Object를 형성하지만 NURBS의 경우는 면과 면을 붙이면 붙인 것처럼 보이게 해주는 것이지 붙이는 것은 아니다. 그러다 보니 Modeling과정이 복잡해 진다.

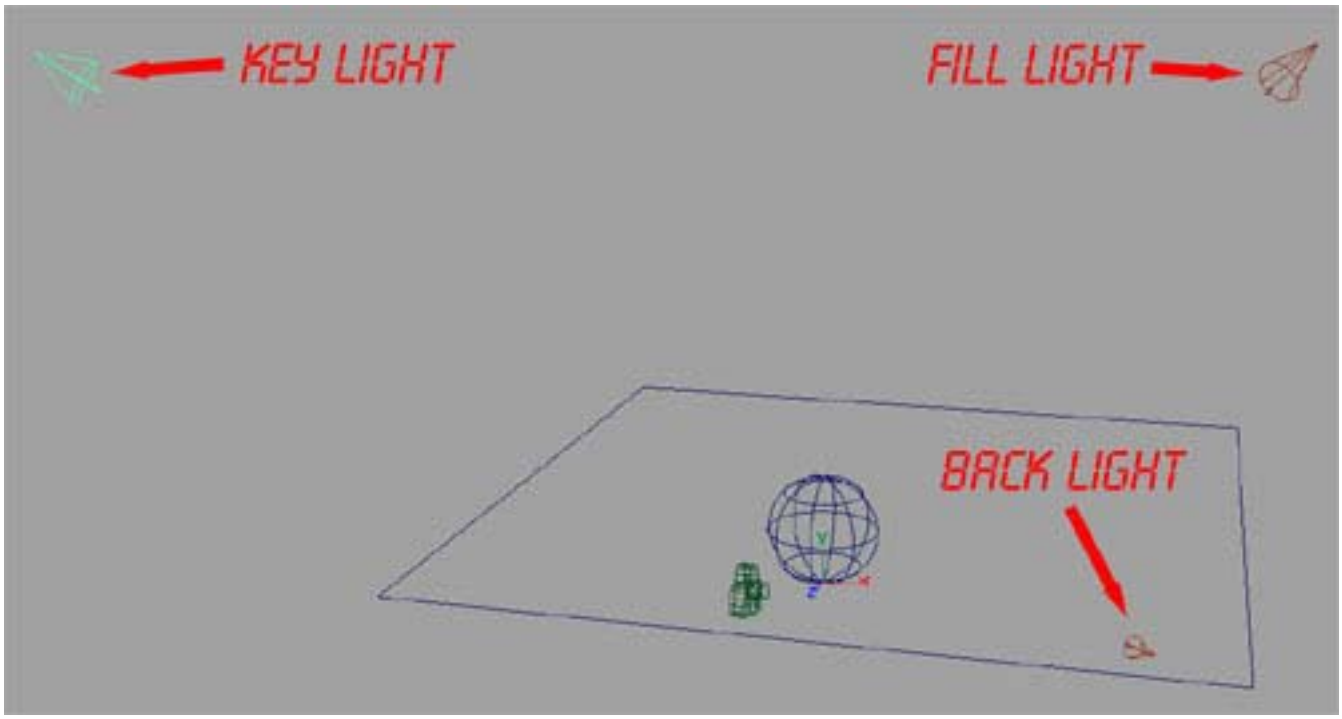
Lighting

Lighting이란?

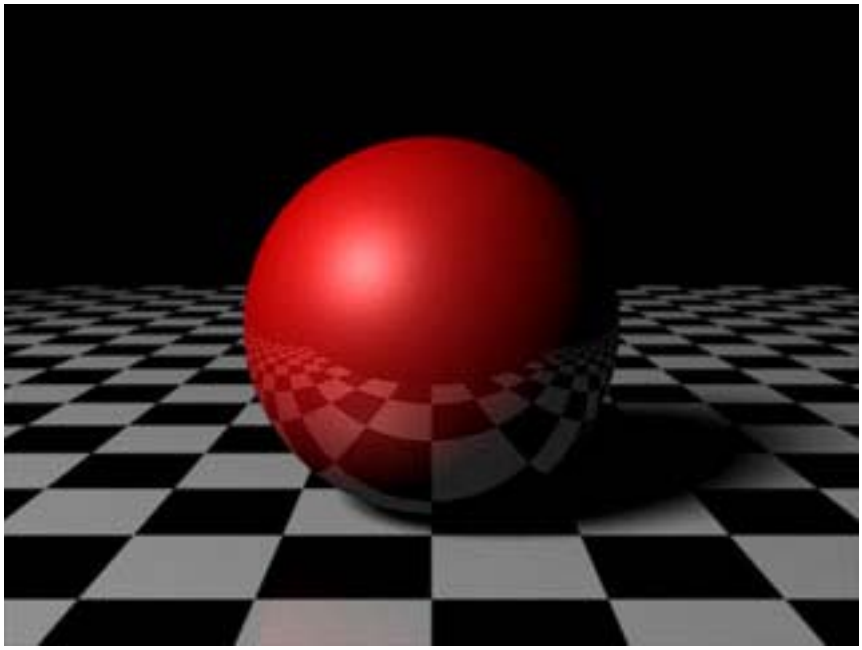
우리말로 이야기하면 조명이다. 3D공간은 빛이 없는 암흑의 공간이다. 여기에 Light를 추가함으로써 물체를 보이게 하는 것이다.

기본적인 3가지 Light

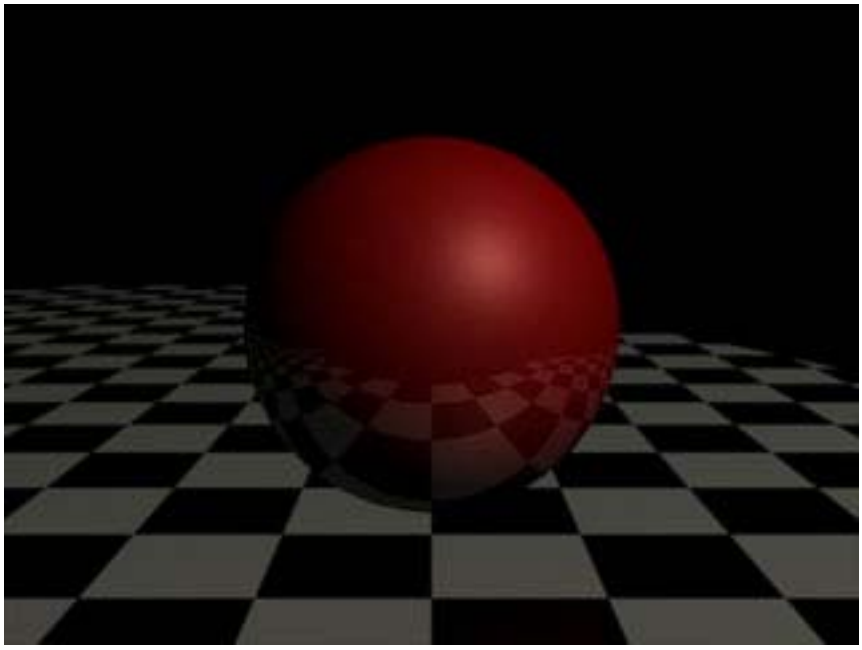
3D에서는 기본적으로 3가지 Light가 필요하다. Key Light, Fill Light, Back Light 이렇게 3가지 이다.
대강의 구조는 다음과 같다.



이제 하나하나 그림으로 설명하겠다.

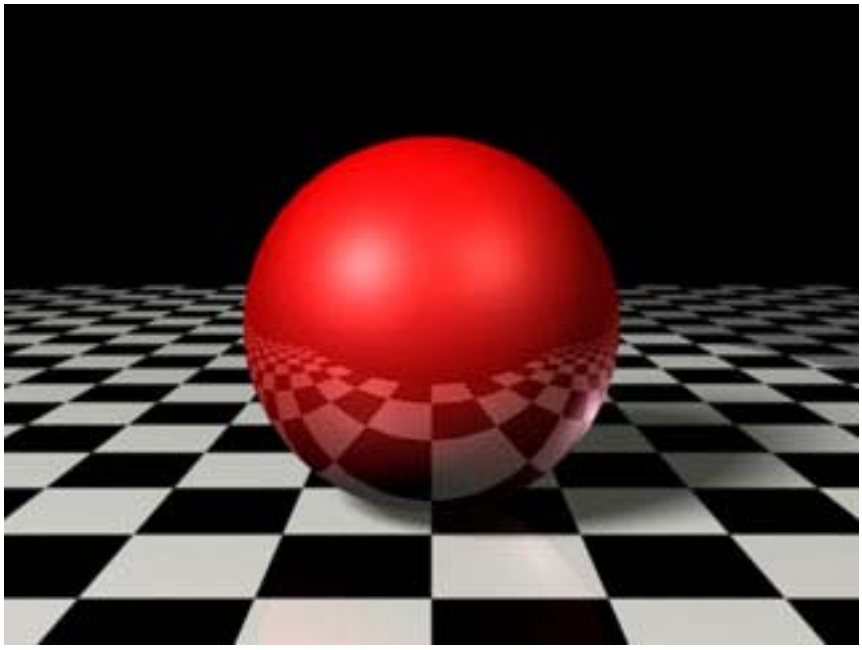


Key Light는 가장 밝기가 센 Light이다. 전체적으로 봤을 때 주가 되는 빛이다. 그리고 이 Light로써 그림자 처리를 한다.



Fill Light는 Key Light를 보조해주는 역할을 한다. 즉 빛을 약하게 뿌려서, 좀더 부드러운 느낌을 내게 한다.

Back Light는 반사광을 만들어준다. 3D에서는 반사광을 알아서 계산하는 경우가 드물기에, 반사광을 인위적으로 만들어 줘야 한다.

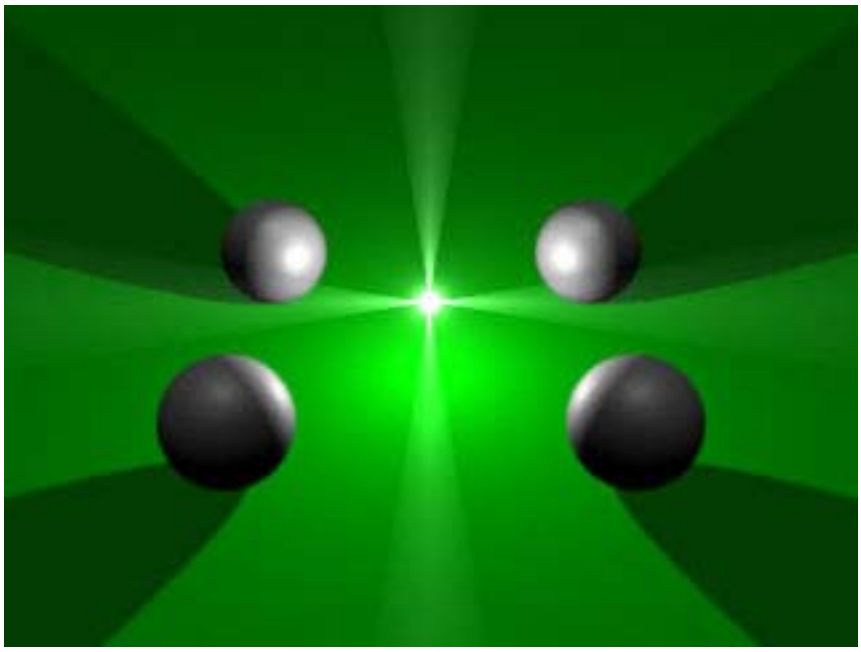


이 그림이 3가지 조명을 전부 다 쓴 최종 결과물이다. 하나씩 썼을 때 보다 훨씬 더 현실감이 느껴지는 것을 느낄 수 있을 것이다.

Light의 종류

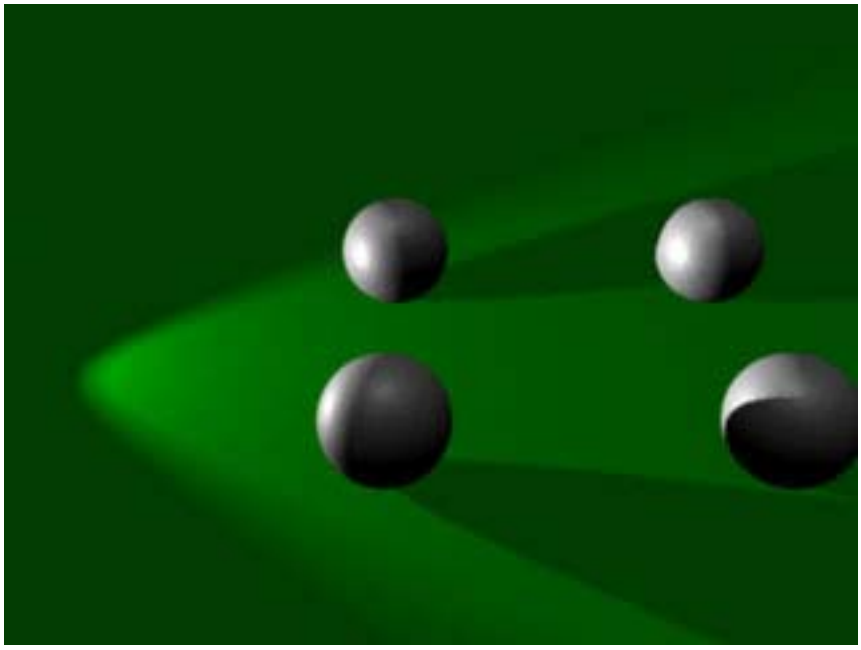
3D에서 쓰이는 Light의 종류는 크게 5가지이다. 하나하나 그림과 함께 설명하도록 하겠다.

- Point (Omni directional) Light



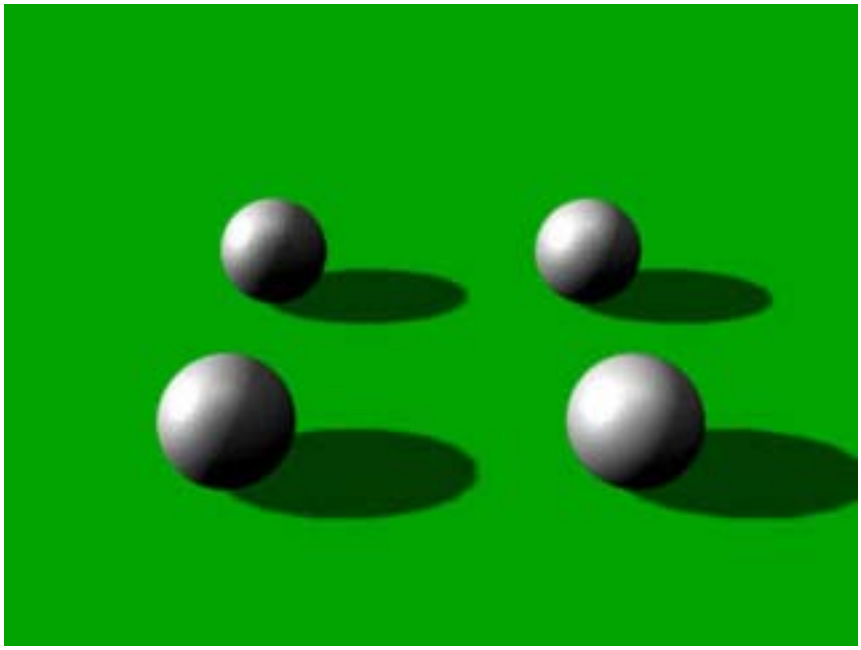
Point Light는 모든 방향으로 빛이 퍼져 나가는 Light이다. 즉 백열등이나 촛불 같다고 생각하면 쉽다. 빛이 퍼져 나가기에 물체의 그림자가 광원에서 먼 부분은 커지는 것을 볼 수 있다.

- Spot Light



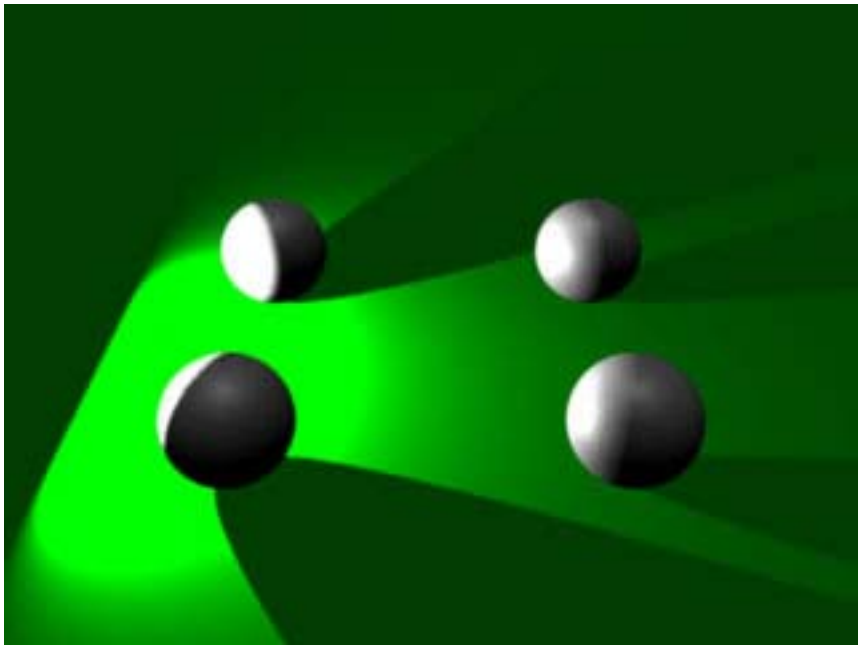
Spot Light는 한 방향으로만 빛이 퍼져 나간다. 가장 기본적인 Light이다. Point Light와 같이 역시 빛이 퍼져나가는 것이기에 광원에서 먼 그림자는 점점 커지는 것을 확인할 수 있다.
(참고로 Point Light의 원리는 Spot Light를 3개 붙여서 모든 방향으로 빛이 뻗어나가도록 만든 것이다.)

-Directional Light



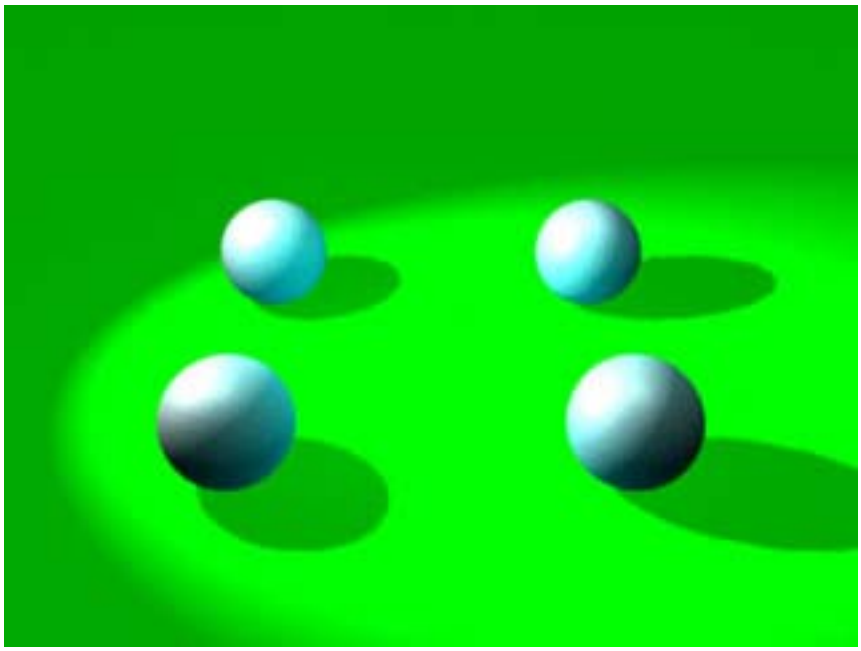
Directional Light는 한 방향으로 빛을 쏘는 방식이다. 그래서 광원과 그림자가 멀어질수록 그림자의 크기에 차이가 없으며, 또한 빛이 지나가는 영역이 지정된 것이 아니라 무한대의 영역으로 빛이 흐르기에, 그림자를 빼고는 어두운 부분이 없는 것을 확인할 수 있다. 그러나 모든 빛은 퍼져나가는 것인데 왜 Directional Light같은 것이 필요할까? 이것은 태양같이 아주 멀리 떨어진 광원을 표현하기 위해 존재하는 것이다. 태양의 경우에는 너무나 먼 광원이기에, 그 빛이 지구에 도착했을 때에는 빛이 퍼짐이 아주 작다. 그래서 이런 Directional Light가 존재하는 것이다.

-Area Light



Area Light는 2차원 적인 평면에서 빛을 발산 하는 것이다. 이것도 역시 빛을 퍼뜨리는 방식이다. 한 영역을 지정해주고 그 부분에서 빛이 나오게 할 수 있어, 빛이 강하게 들어오는 창문 등을 표현할 수 있다.

-Ambient Light



Ambient Light는 이해하기가 좀 까다로울 수 있다. 이유는 현실세계에서는 존재할 수 없는 Light이기 때문이다. Ambient Light는 어두운 부분의 색을 바꾸는 Light이다. 빛이 퍼지는 방식은 Point Light와 거의 같지만, 그 기능이 다르다. Light는 일반적으로 밝은 부분을 만드는 것이지만 Ambient Light는 가장 어두운 부분의 색을 바꾸는 것이기 때문이다. 위의 Image에선 Ambient Light의 색을 밝은 파란색으로 지정했다.

Camera

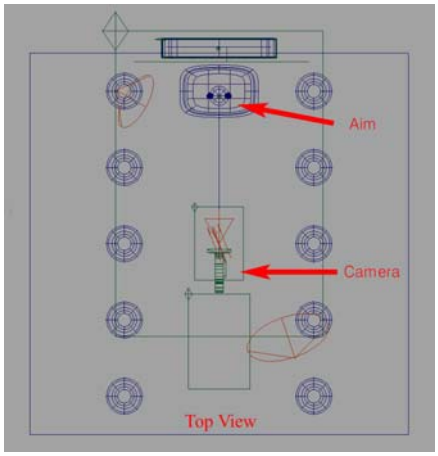
Camera란?

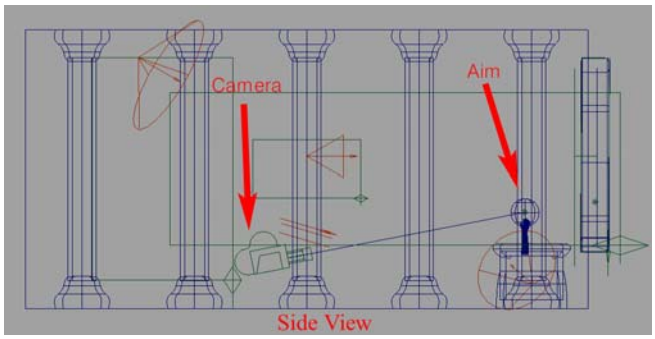
3D상의 Camera라고 일반적인 Camera와 특별히 다를 건 없다. 3D공간은 아무것도 없는 공간이기에, 보는 시선이 필요함으로, 그러기에 Camera라는 개념이 존재할 뿐이다. Camera의 Option도 일반적인 Camera Option과 동일하다고 보면 된다. 그러기에 여기서의 설명도 일반적인 Camera Option에 대한 설명을 하겠다.

Lens Size

Camera Lens의 직경이다. 단위는 mm이며, 크기가 작을수록 Zoom Out이 되며, 원근감 표현이 더 잘 나타난다. 즉, 화면이 약간 왜곡 되어서 표현이 된다. 수치가 높아질수록 Zoom In이 된다. 일반적으로 35mm가 사람의 시야와 비슷하며 9mm는 물고기의 눈처럼 왜곡이 심해진다. 그림을 보면 쉽게 이해가 될 것이다.

구조도





테스트된 화면

9mm(Fish Eye)

18mm

25mm

35mm

50mm

80mm

85mm

Camera Size

화면에 나타나는 피사체의 Size를 뜻한다. 각 Size별로 그림과 함께 설명하겠다.

ELS - Extreme Long Shot의 약어로써, 매우 먼 거리에서 촬영한 것이다. 일반적으로, 장소나 시간에 대한 정보를 제공할 때 쓰인다.

LS - Long Shot의 약어로써, 먼 거리에서 촬영하는 것이다. 인물간이나 인물과 사물간의 횡간 관계를 나타낸다. 관객에게 상황을 설명해주기에 Establish Shot이라고도 한다.

FS - Full Shot의 약자로써 인물의 전신을 다 보여주는 것이다. 그 인물에게만 관심을 보이도록 유도하는 목적을 가지고 있다.

MS - Medium Shot의 약어로써 크게 3가지로 나뉜다. Knee-Medium Shot(첫 번째 그림), Waist-Medium Shot(두 번째 그림), Breast -Medium Shot(세 번째 그림)이다.

Medium Shot은 인물의 동작을 자세히 보여줌으로써, 관객이 인물의 동작에 큰 영향을 받도록 하는 목적을 가지고 있다.

인물의 동작뿐 아니라, 옷차림과 소품을 비교적 자세히 보여주기에 관객의 궁금증을 해결하기도 한다.

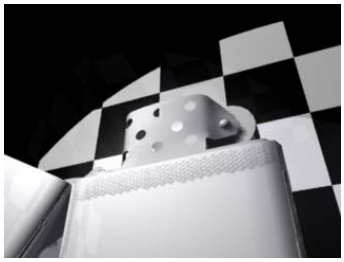
CS - Close Shot의 약어로써 얼굴이 화면의 대부분을 채우고 몸이 살짝 보이도록 촬영 하는 것이다. 인물의 심리 상태를 표정과 약간의 몸의 움직임을 통해서 나타낸다.

CU - Close Up의 약어로써 얼굴이 화면에 꽉 채우도록 촬영하는 것이다. 인물의 심리상태를 표정으로써 나타낸다.

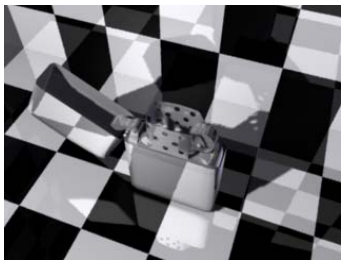
ECU - Extreme Close Up의 약어로써, 눈, 코, 입 등 표정을 나타내는 곳 중 어느 한 부분만 강조 하여 화면에 꽉 채우는 것이다. 것이다. 즉 중요한 말을 할 때에는 입만을 화면에 꽉 채우는 식이다.

Camera Angle

Camera의 Angle(각도)이다. Angle에 따라 화면의 느낌이 다르다. 각각의 느낌을 그림으로 보면서 설명하겠다.

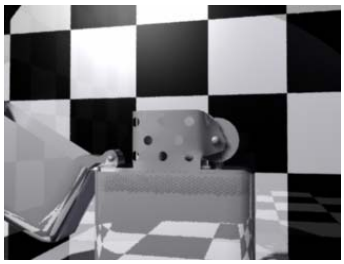


ELA - Extreme Low Angle의 약어로 아주 낮은 앵글로 찍는 것이다.
상대적으로 거대해 보인다. 위압감과 위엄을 많이 준다.



HA - High Angle의 약어로 높은 앵글로 찍는 것이다.
풍경을 찍을 때 많이 사용되며, 상대적으로 왜소해 보인다.

LA - Low Angle의 약어로 낮은 앵글로 찍는 것이다.
ELA보다 적으나 거대해 보이며, 위압감과 위엄을 준다.



DA - Direct Angle로 Eye Level이라고도 한다. 눈의 시점과 유사한 앵글로 찍는 것이다.
사람의 시점과 비슷하기에 현실적인 느낌을 준다.

Camera Work

Camera가 움직이는 방법이다.

Fix - Camera를 고정하는 것이다.

Pan - Camera를 왼쪽이나 오른쪽(Right, Left)으로 돌리는 것이다.

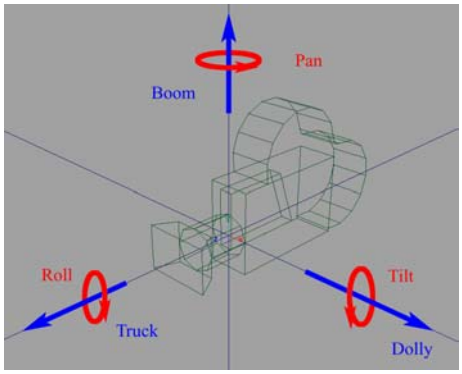
Tilt - Camera를 위나 아래(Up, Down)로 돌리는 것이다.

Track - Camera를 전진이나 후진(In, Out)시키는 것이다. 보통 Track을 설치하여 움직인다.

Zoom - Camera 자체를 움직이지 않고 렌즈를 사용하여 줌을 하는 것이다. (Track과 마찬가지로 In, Out이다.)

H-H - Hand Held의 약어이다. 손으로 직접 Camera를 잡고 찍는 것이다. 다큐멘터리같이 아주 사실적인 분위기가 난다. 그러나 Camera가 심하게 흔들릴 수도 있다는 단점이 있다. 3D상에서 표현하려면 상당히 손이 많이 가는 Animation을 해야 한다.

Camera Work의 방향에 대한 용어를 그림으로 설명하면 다음과 같다.



Mapping

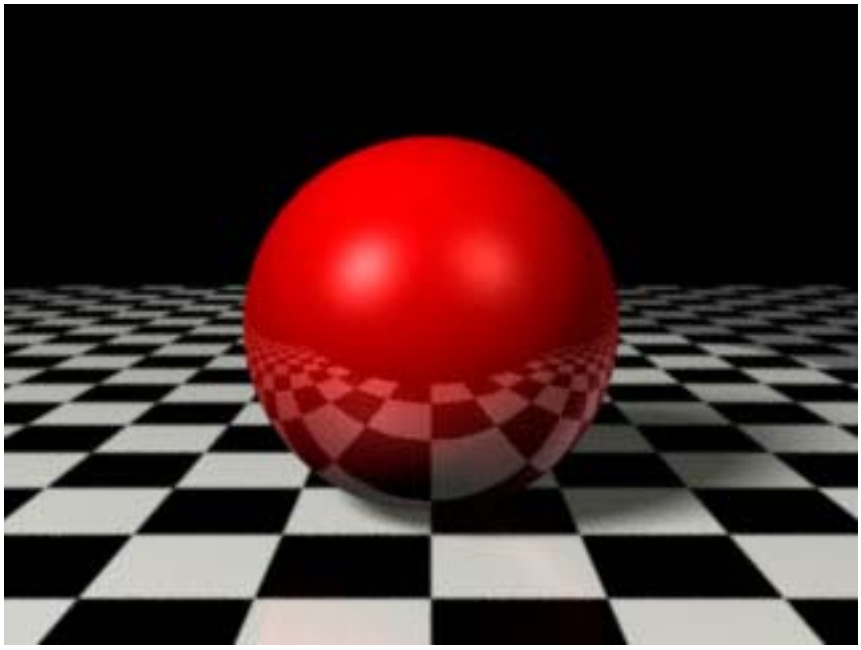
Mapping이란?

질감을 주는 작업이다. Modeling을 하여 만든 Object에 질감을 입혀 원하는 Image를 창조 하는 것이다. Mapping을 할 때, 크게 2가지의 개체가 필요한데 Material과 Texture다.

Material

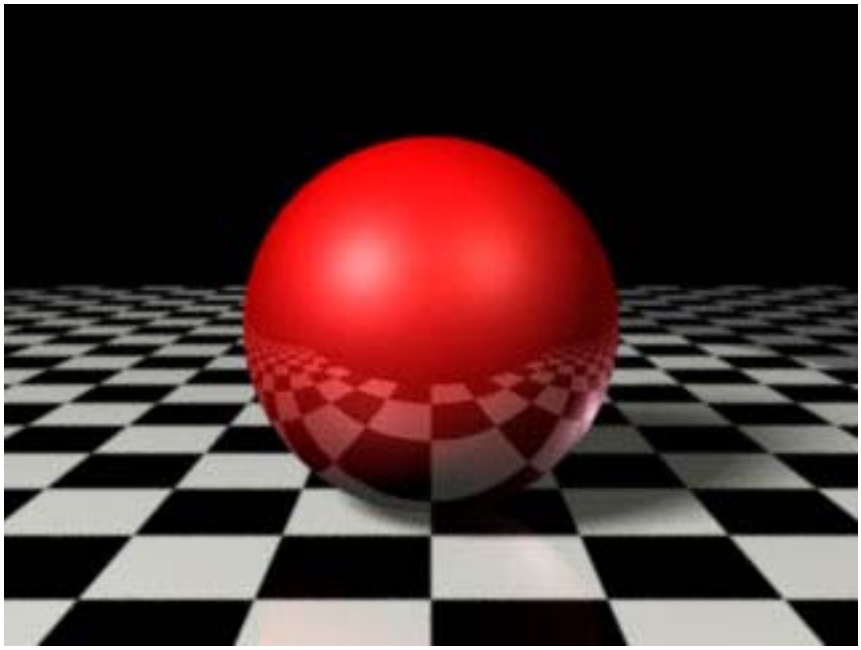
Material은 Shading을 하는 방법이라고 볼 수 있다. Shading이란 빛이 물체에 반사되는 방식이다.(그림자 처리라고도 한다.) 플라스틱의 경우에는 빛의 반사가 강하지만, 면으로 된 옷 같은 경우는 빛의 반사가 약하다. 이런 식으로 Shading의 방법을 정해주는 것이다. Shading의 방법에는 여러 가지가 있지만 가장 대표적인 Phong, Blinn, Lambert 이렇게 3가지만 그림과 함께 알아보겠다. (이것의 구동 알고리즘은 상당히 복잡하다. 그러므로 그것에 대한 설명은 생략하거나 간단히 하겠다.)

-Phong



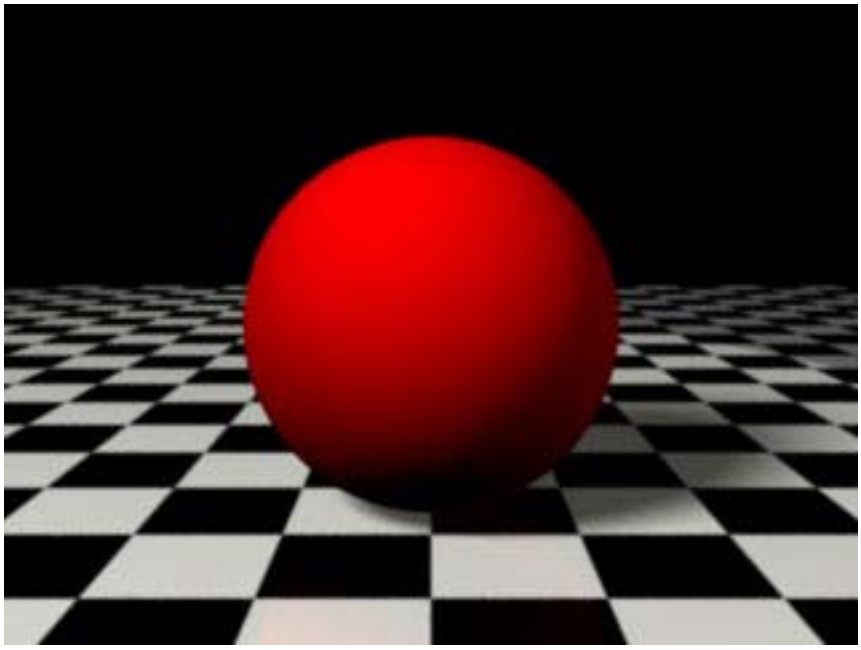
Bui Tuong Phong이라는 학자가 개발한 빛의 반사법. 빛의 반사를 도형의 각도 별로 계산하여, (원래는 Vertex의 Normal값과 Normal값 사이를 Interpolate해주고… 어쩌고 하는 원리가 있지만 어려우니 넘어가자.)빛의 반사를 표현하는 방법이다. 플라스틱과 같은 질감을 낼 때 유용하다.

-Blinn



James Blinn이라는 학자가 개발한 빛의 반사법. Phong에서의 빛의 반사에 Highlight(가장 밝은 부분)부분을 좀 더 부드럽게 해준 것이다. 그래서 결과적으로 금속적인 느낌이 든다.

-Lambert



Johann Heinrich Lambert라는 학자가 개발한 빛의 반사법. Phong은 각도 별로 다른 빛의 반사가 일어나지만, Lambert는 그와 달리, 빛이 어느 방향으로 오던, 같은 방향으로 빛의 반사가 일어난다. 그래서 분필이나 파우더 같은 느낌이 든다.

이런 Material들의 세부적인 Option으로 Ambient, Diffuse, Specular가 있다. 그냥 가볍게 알아보고 가자.
-Ambient는 가장 어두운 곳의 색을 말한다. 보통 검은색이다.
-Diffuse는 보통부분의 색을 말한다. 이것의 색에 따라, Object의 색이 결정된다고 생각하면 된다.
-Specular는 가장 밝은 부분의 색이다. 보통 회색에서 흰색 사이 값이다.

Texture

Texture는 Material로 빛의 반사법이 적용된 Object에 Image 등을 입혀서 그 고유의 색등을 입히는 작업이라고 생각 하면 된다. Texture의 방법으로는 2D와 3D가 있다.

-2D

Image 파일을 직접 씌우거나, 2D함수로 계산된 Image를 씌우는 것이다. 2D를 3D에 씌우다 보니 어느 한 부분은 반드시 Image가 꼬여있게 되어있다. 그러나 종류가 많고, 실제 작업 시 2D Image 파일을 씌워버리는 경우가 많기에 많이 쓰이고 있다.

-3D

3D함수로 계산된 Image를 씌우는 것이다. Image의 꼬임은 없지만, 종류가 많지 않다.

Rendering

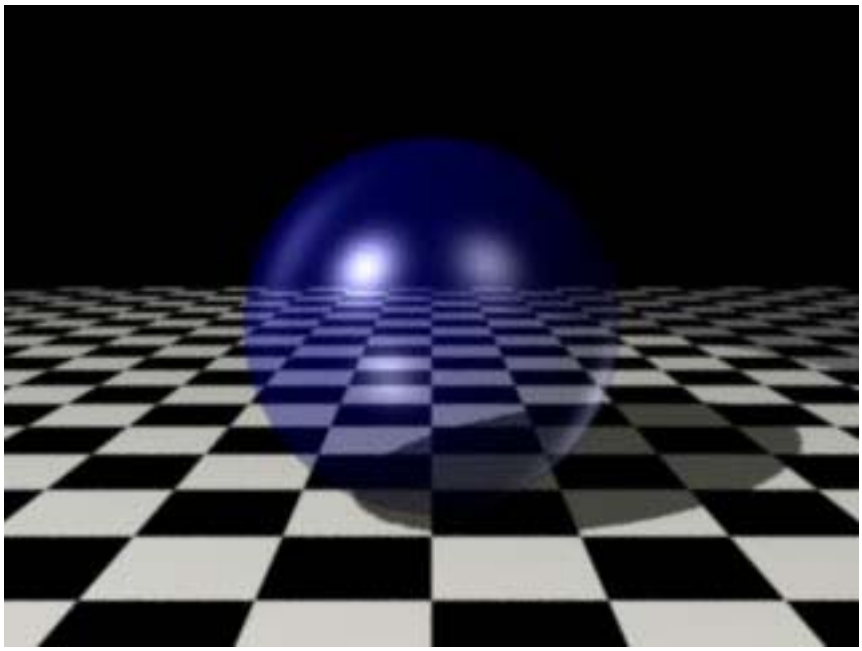
Rendering이란?

3D로 작업된 것을 2D그림으로 결과물을 뽑아 내는 작업이다. Rendering이라는 말은 일단 사전적으로는 “되다”라는 뜻이 있다. Rendering이라는 말은 많은 곳에서 쓰인다. 건축설계에서 설계도를 가지고 조감도를 그리는 것도

Rendering이라고 하고, Computer이론에서는 Computer가 처리한 것을 사람이 볼 수 있도록 만드는 것을 Rendering 이라고 한다. 3D에서의 Rendering은 이 두 가지의 Rendering의 의미를 둘 다 가지고 있다고 볼 수 있다. 3D좌표상에서 설계된 것을 2D 그림으로 뽑아내고, Computer가 3D 좌표상에 산재해 있는 많은 데이터들을 총 집합하여 연산함으로써, 결과물을 사람이 볼 수 있게 뽑아내므로 둘 다의 의미를 내포하고 있다는 것이라고 말할 수 있는 것이다.

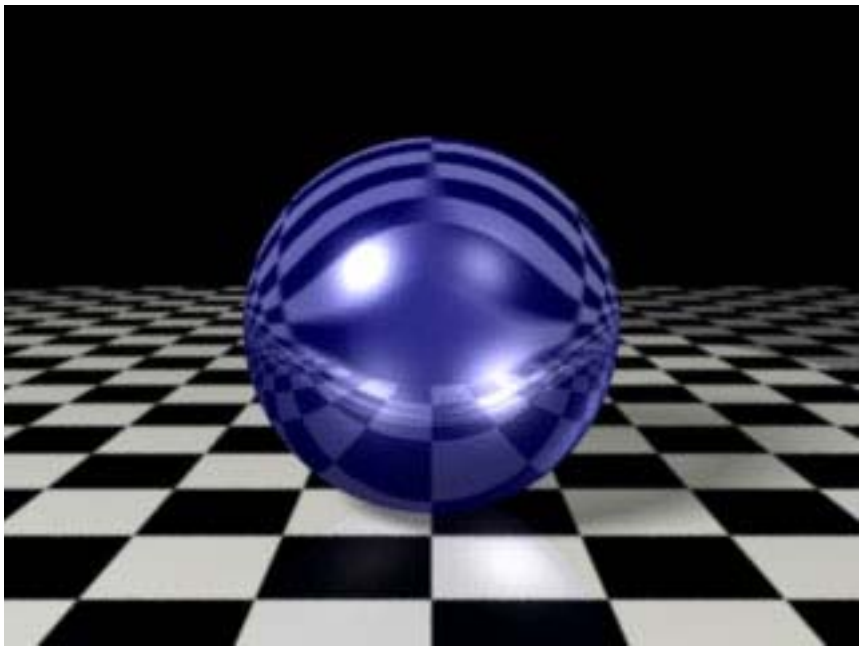
Ray cast

Ray cast란 Object를 그냥 한 덩어리로 보고 Rendering하는 방법이다. 따라서 투명한 물체라도 그림자는 그 Object의 Opacity를 무시하고 그냥 검은색으로 나오게 된다. 그리고 Reflection(반사), Refraction(굴절)등의 표현이 되지를 않는다. (굳이 한다면 Texture를 입히는 방법이 있다.) Ray cast로 Rendering 한 예는 다음과 같다.



Ray trace

Ray trace는 '광추적' 이라고도 하는데 이유는 빛을 추적하여, Reflection, Refraction, Opacity등을 계산하기 때문이다. 따라서 Rendering시간이 몇 배는 더 걸린다. Ray trace로 Rendering을 한 예는 다음과 같다.



참고문헌

책

서명 - 저자명 - 출판사 순

멀티미디어 배움터 - 최윤철, 고건 - 생능출판사
최신 비디오 촬영 교실 - 아트워크 - 하서
The Art of 3-D Computer Animation and Imaging - Isaac Victor Kerlow - WILEY
Digital Lighting & Rendering - Jeremy Birn - New Rider
3D Lighting History, Concepts, & Techniques - Arnold Gallardo - Charles River Media

Internet Web Site

http://www.filmmakers.co.kr - 필름 메이커스 커뮤니티
http://myhome.hananet.net/~hanibal/ - 다큐멘터리 제작자 한성수님의 홈페이지
http://madang.ajou.ac.kr/~iklee/ - 아주대학교 이인권 교수 홈페이지
http://www.softjobplaza.com/ - 소프트잡 플라자
http://escience.anu.edu.au - The Australian National University eScience

그리고…
감수를 맡아주신 아주대학교 미디어학과의 남용재 형, Motion Factory의 김명철 선생님께 감사를 드립니다.