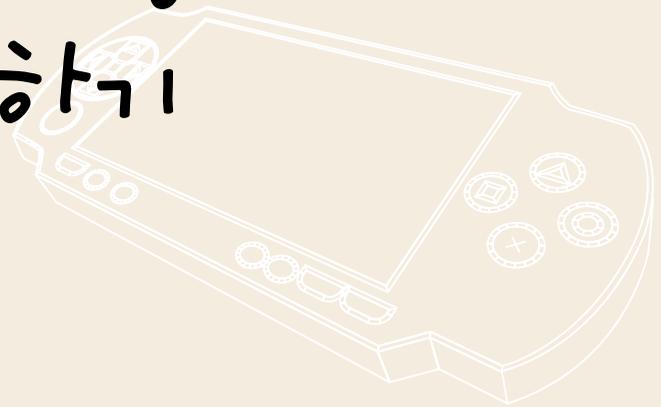


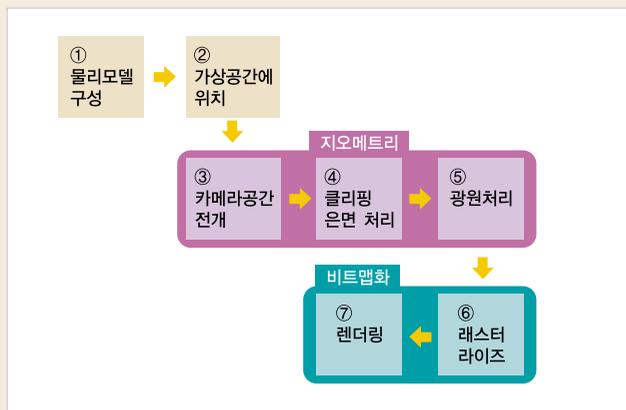
# 3D 그래픽 관련 용어 쉽게 이해하기



## 게임 엔진

일반적으로 DOOM 엔진, Unreal 엔진이라 불리는 3D게임 엔진은 CPU가 담당하는 처리부분을 말하는 것이다. 또한 이러한 게임 엔진은 3D그래픽을 표현하는데 상황에 따라 어떻게 표현하는지를 설정하는 부분까지 포함한다. 즉, 같은 DirectX를 사용하고 같은 그래픽 카드를 사용하는 데에도 게임 타이틀마다 3D 그래픽의 표현이 다른 것은 게임 엔진의 차이 때문이다.

## 지오메트리 처리



위 그림에서 ③④⑤의 부분은 지오메트리처리 과정을 보여주는 것이다. 지오메트리(Geometry)는 기하학이란 뜻으로, 3D 그래픽에 있어서 생각할 수 없는 3차원

벡터에 관한 처리인 것이다. 벡터는 X, Y, Z축 세 개의 좌표로 표현되며 방향성을 가지고 있다. 또한 각각의 벡터 좌표의 기준을 좌표계라고 한다.

③의 단계에서는 각 벡터의 좌표계가 하나로 통일되며 최종적으로 카메라의 시점을 기준으로 한 좌표계로 전개된다.

예를 들어 3D캐릭터의 팔을 굽히는 장면에서 그 관절을 기준으로 한 좌표계로 계산하게 되면 간단히 처리할 수 있다. 따라서 팔을 굽히는 단계는 각 관절을 기준으로 설정된 로컬 좌표계를 이용해 계산한다. 이후 굽혀진 팔은 글로벌 좌표계라는 공간의 표준 좌표계로 변환되는 것이다.

## 클리핑 은면 처리

글로벌 좌표계로 통일된 3D 개체들은 최종적으로 카메라의 시점 좌표계로 변환되어 시점을 기준으로 투시 변환 되게 된다. 투시 변환이란 시점에서 보이는 3D 공간을 화면에 표시되는 2D로 변화하는 처리를 말한다. 이것이 클리핑 은면 처리에 해당하는 부분이다.

클리핑은 시계에 나타나는 부분을 찾는 작업이며, 은면 처리는 시점에서 보이지 않는 폴리곤을 없애는 작업이다. 또한 이 부분에서 어느 폴리곤에 어떤 텍스처를 붙이게 될지 계산하는 작업도 수행된다. 실제로 텍스처 매핑 작업은 ⑦의 단계에서 수행하지만, 이 부분에서 준비작업을 거친다고 생각하면 된다.

## 광원처리

벡터의 좌표 변환 작업이 완료되면 광원 처리가 이루어진다. 이것이 ⑤의 부분이다. 빛이 사물에 닿으면 반사되거나 흡수된다. 이러한 처리를 컴퓨터는 수학적으로 변환하여 처리하게 되는데 이 것 역시 벡터로서 처리된다. 광원 처리에는 빛의 방향벡터와 시선벡터, 빛이 닿는 면이 향하는 방향벡터의 관계로 성립된다.

이 세 개의 벡터 관계를 계산해 빛이 시점에 어떤 형태로 도달하는 방정식을 수립하고, 계산해 내는 것이다. 결국 좌표변환(Transform)과 광원(Lighting)은 모두 벡터 연산으로 처리되면 현재 하드웨어 T&L엔진이 처리하는 부분이다. 이처럼 3D그래픽은 1프레임 이미지에 수 백개의 벡터 연산이 처리되게 된다.

## 렌더링

평면인 그림에 형태·위치·조명 등 외부의 정보에 따라 다르게 나타나는 그림자·색상·농도 등을 고려하면서 실감나는 3차원 화상을 만들어내는 과정 또는 그러한 기법을 일컫는다. 즉, 평면적으로 보이는 물체에 그림자나 농도의 변화 등을 주어 입체감이 들게 함으로써 사실감을 추가하는 컴퓨터그래픽상의 과정이 곧 렌더링이다.

## DirectX

미국 마이크로소프트사가 1995년 윈도우95와 윈도우NT 4.0용으로 개발하였다. 윈도우환경에 멀티미디어 기능과 오락기능을 강화시킨 프로그램들의 집합체이다. 2차원 및 3차원의 도형·음성·동화(動畵) 등 각종 미디어를 사용한 응용프로그램이 하드웨어 장치를 직접 접촉한다고 해서 '다이렉트'라는 이름이 붙여졌다.

1995년에 X1을 발표한 이후에 기능을 계속 강화하여 1997년에 X5를 발표하였다. 컴퓨터 게임에 적합한 기능을 고루고루 갖추고 있다. 다이렉트 드로우(Direct Draw)는 그래픽 가속기 기능을 하며, 다이렉트 3D는 그래픽 가속기 없이 3차원 그래픽을 보다 자연스럽게 구현하게 해준다. 다이렉트 사운드는 사운드 기능을 직접 컨트롤하고, 다이렉트 3D 사운드는 3차원 사운드 기능을 컨트롤한다. 다이렉트 인풋은 조이스틱·마우스 등 게임에 필요한 보조장치들을 제어해 주고, 다이렉트 플레이는 멀티플레이·모뎀플레이 등을 통일된 규격상태로 즐길 수 있게 해준다.

## OpenGL

Silicon Graphic가 자사의 그래픽 워크스테이션에서 3D 그래픽을 지원하기 위해서 개발한 API로, 실제와 같은 3차원 렌더링 이미지를 생성할 수 있는 각종의 이펙터와 기능들을 복합적으로 지원하고 있다.

## DirectX7 시대의 하드웨어 T&L

하드웨어 T&L의 T는 Transform으로 좌표 변환을 뜻하고, L은 Lighting으로 광원 처리를 말하는 것이다. 하드웨어 T&L은 지금까지 CPU가 소프트웨어 레벨로 처리하던 것을 하드웨어인 비디오 칩으로 처리하는 의미를 가지고 있다. 이러한 구조의 장점은 단순히 3D그래픽 처리에서 CPU가 지오메트리 연산 처리에서 해방된다는 것이 최대 장점이다. CPU는 게임 엔진 부분에 집중할 수 있게 된 것이다. CPU와 비디오칩의 업무 분산화가 진행될 만큼 여유가 생겨 3D처리 자체가 가벼워진다. 따라서 CPU는 더욱 고도의 연산이 가능한 것이다. 렌더링뿐만 아니라 지오메트리 연산을 담당하게 된 비디오칩은 단순한 Accelerator에서 CPU와 동급의 GPU(Graphics Processing Unit)으로 불리게 된다.

## 해상도와 메모리

MDA의 경우 단순히 문자로 이루어진 정보만을 출력했었기 때문에 해상도는 큰 의미가 없었다. 그러나, VGA가 등장하면서 한 화면에 나타낼 수 있는 가로, 세로의 점의 개수를 이용하여 해상도를 표현하면서 관심을 끌기 시작하였다. 현

재 일반적인 VGA 경우 최대 1600x1200 해상도에, 최소 16MB의 그래픽 메모리를 기본으로 가지고 있다. 해상도가 높아지게 되면 그래픽 메모리 또한 많아져야 하는데, 이것은 해상도가 높아질수록 한 화면을 구성하는 픽셀(pixel) 수가 늘어나기 때문이다. 예를 들어 1600x1280의 해상도로 출력할 경우 한 화면을 구성하기 위해서는 최소 2M의 픽셀이 필요하다. 컬러 수 또한 높아지게 되면 한 픽셀을 구성하는 비트수도 늘어나기 때문에 그만큼 많은 그래픽 메모리가 필요하다.

## 픽셀 (Pixel)과 도트(Dot)의 차이점

모니터나 프린터 같은 출력장치들은 표현할 수 있는 가장 작은 점을 도트(dot)라고 하며, 도트 사이의 거리를 mm 단위로 도트 크기(dot pitch)로 규정하고 있다. 그러나, VGA에서는 그래픽 이미지를 구성하고 있는 최소 크기의 점을 나타낼 때 도트라는 용어를 사용하지 않고 픽셀(pixel, 화소)라 한다. 640x480 해상도라 한 화면은 모니터가 나타낼 수 있는 점의 크기에 관계없이 가로 640 픽셀, 수직 480 픽셀로 구성된다. 예를 들어, 모니터의 도트가 1280x960이고 해상도가 640x480이라면 아래의 그림처럼 도트 한개는 RGB 컬러 요소로 구성되어 되며, 4개의 도트가 모여서 하나의 픽셀을 구성하게 된다.

## 하이컬러 (High Color)

하이 컬러의 한 픽셀은 16비트로 구성되며 비트를 나누기 위해서 팔레트 대신 RGBA가 사용된다. 일부에서는 하이컬러를 5551이나 5650 등 여러 가지로 표현을 하게 되는데, 5551은 Red/Green/Blue에 각각 5비트씩 A(Alpha)값에 1비트를 할당하는 것을 말한다. 여기서 A는 Alpha 값으로 명도를 지정하는 값이다. 이것은 실제 채도를 결정하는 RGB에 할당된 비트 수가 15비트이기 때문에, 15bpp 모드로 구분하며 32,768가지의 색상을 표현할 수 있다고 한다. 현재 하이컬러는 보통 5650 모드를 말하며 65,536 컬러를 표현할 수 있다.

## 트루컬러 (True Color)

트루컬러는 한 픽셀을 24비트 또는 32비트로 나타내는 것으로, 24비트 모드에서 RGBA 값이 8880, 3비트 모드는 8888로 표현한다. 두 가지 모두 표현할 수 있는 색상은 RGB 비트 수가 24비트를 갖기 때문에 모두 16,777,216(224) 컬러를 표현할 수 있으며 32비트 모드의 경우 마지막 8비트로 각 픽셀의 밝기까지 조절할 수 있다.

1600x1280 해상도의 한 화면은 2M의 픽셀로 구성되는데, 이 경우에 트루컬러를 사용한다면 한 화면에 모든 컬러를 표현할 수 없는 정도의 많은 컬러수이기 때문에 사람의 눈으로는 구분할 수 없는 미세한 색상의 변화까지 표현할 수 있다. ☺

자료출처 : <http://www.hwlab.com>