



3D게임프로그래밍

-CHAPTER9-

SOULSEEK



목차

1. 절두체 컬링(**Frustum culling**)

절두체 컬링(**FRUSTUM CULLING**)

1. 절두체 컬링(FRUSTUM CULLING)

- 실제로 카메라의 시야 범위에 포함되는 것들만 렌더링하고, 나머지 것들은 렌더링 하지 않는 기법
- **3D**엔진 개발에 있어 가장 중요한 속도 증가 기법 중 하나

절두체를 이루는 6개의 평면

근평면 : 카메라와 수직하며 제일 가까운 곳의 시야 범위를 나타내는 평면

원평면 : 카메라와 수직하며 제일 먼 곳의 시야 범위를 나타내는 평면

좌평면 : 카메라의 좌측 시야 범위를 나타내는 평면

우평면 : 카메라의 우측 시야 범위를 나타내는 평면

상평면 : 카메라의 상단 시야 범위를 나타내는 평면

하평면 : 카메라의 하단 시야 범위를 나타내는 평면

렌더링 파이프라인에 의하면, 이렇게 **6**개의 평면에 포함되는 영역만을 렌더링하게 되므로 나머지 영역에 포함된 폴리곤들은 **D3D**의 **DrawPrimitive()**함수에게 부담만 줄 뿐이다. 이때, **6**개의 평면 내부와 외부를 판단하기 위해 **평면방정식이라는 수학적 도구를 사용**한다.

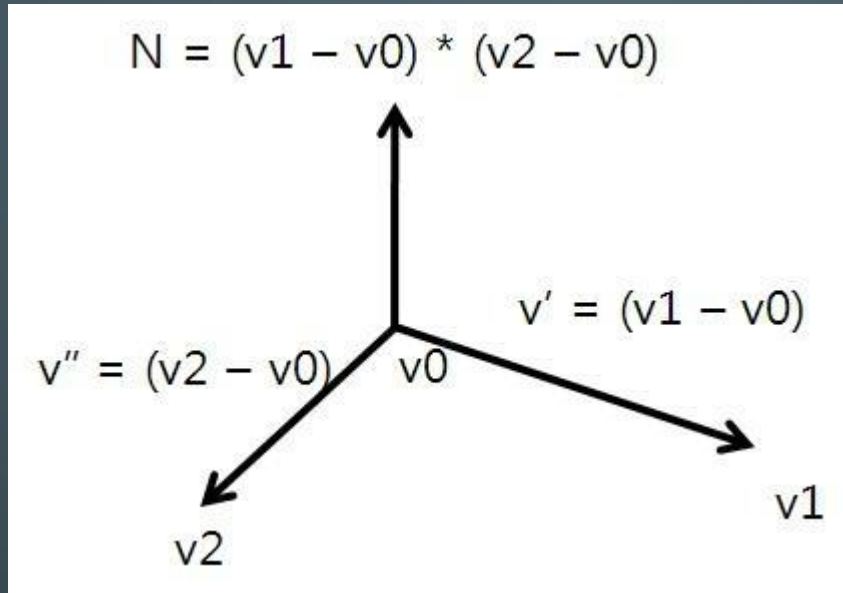
평면방정식의 정의

평면방정식이란 무한 평면을 정의하는 수학적 도구로, 간단한 수식으로 평면을 표현한 것이다.

$$ax + by + cz + d = 0$$

1. 절두체 컬링(FRUSTUM CULLING)

세개의 점으로부터 평면의 법선 벡터값을 구하는 방법



- 세개의 점 v_0, v_1, v_2 로부터 v' 와 v'' 를 만들고, 이들 2개의 벡터를 외적연산하면 직교하는 벡터 $N=(a,b,c)$ 가 나오는데, 이렇게 얻어진 벡터 $N=(a,b,c)$ 가 평면 방정식의 법선 벡터다
- d 값은 v_0, v_1, v_2 중의 어느 하나의 값을 평면방정식에 대입해서 얻어내면 된다.
- 예를 들면, $ax_1 + by_1 + cz_1 + d = 0$ 이므로 $d = -(ax_1 + by_1 + cz_1)$ 으로 구할 수 있다

D3DX 라이브러리에서..

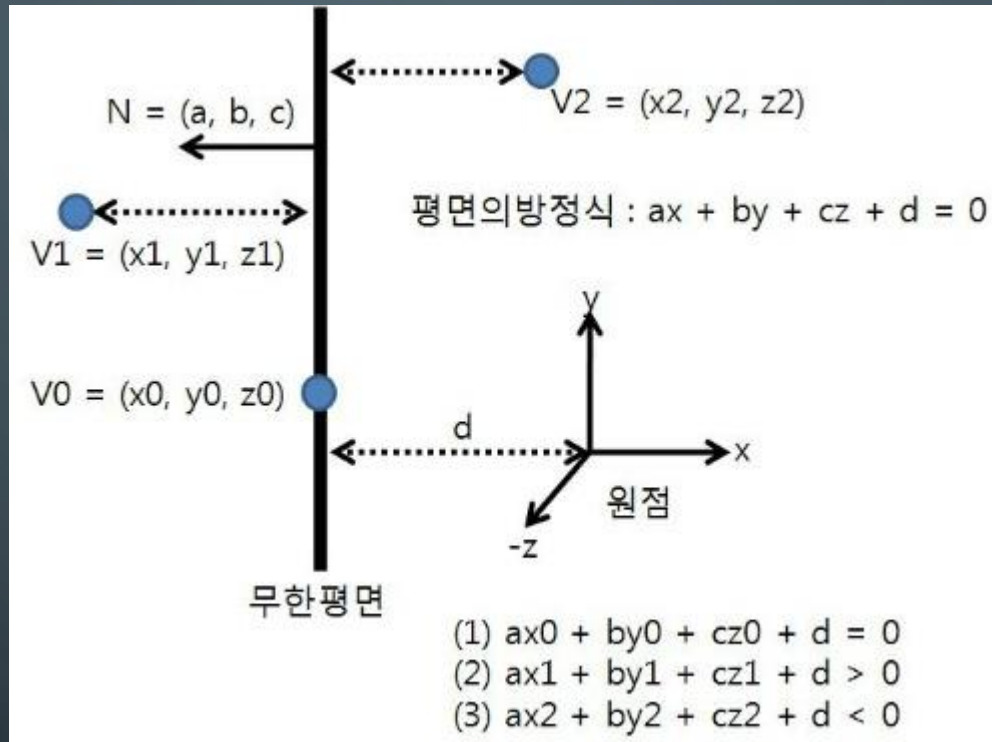
```
D3DXPLANE *D3DXPlaneFromPoints(D3DXPLANE *pOut, CONST D3DXVECTOR3  
*pV1, CONST D3DXVECTOR3 *pV2, CONST D3DXVECTOR3 *pV3);
```

- $pV1, pV2, pV3$ 에 정점의 값을 대입해서 호출하면 $pOut$ 에 평면 방정식의 a, b, c, d 값을 구해서 돌려준다.

1. 절두체 컬링(FRUSTUM CULLING)

평면 방정식의 성질

- 우리는 평면 방정식의 정의에서 평면에 방향이 있음을 알 수 있었는데, 평면에 방향이 있다는 것은 어떠한 점이 평면의 앞에 있는지, 뒤에 있는지를 판단할 수 있다



평면방정식의 연산 결과

- 위 그림의 (1) 방정식 : 점이 평면 위에 있다.
- 위 그림의 (2) 방정식 : 점이 평면 앞에 있다.
- 위 그림의 (3) 방정식 : 점이 평면 뒤에 있다.

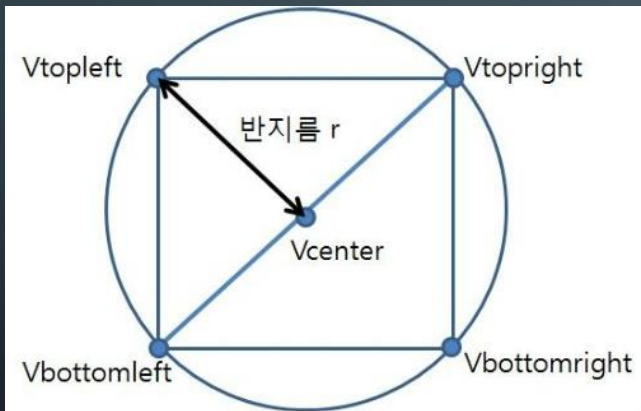
1. 절두체 컬링(FRUSTUM CULLING)

절두체 컬링 원리

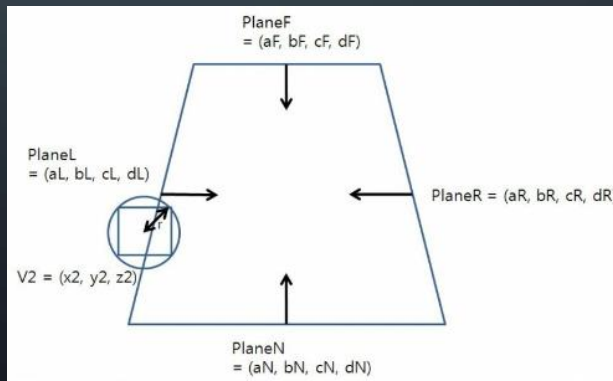
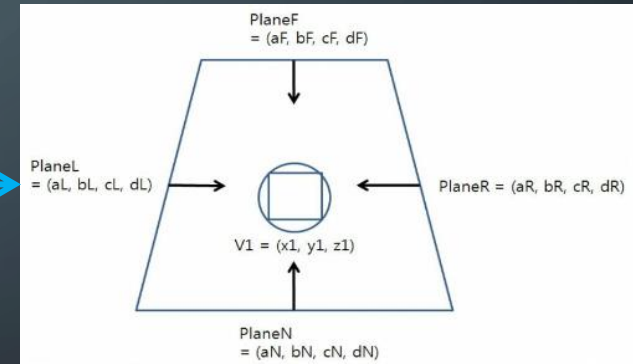
- 절두체 내부에 점이 포함되는지를 판단하기 위해 6개의 평면 방정식에 점의 좌표를 대입해서 모든 평면방정식의 결과값이 양수면 이 점은 절두체 내부에 포함되어 있는 것이다.
- 실제로 이것을 그대로 적용한다면 수백, 수천개의 정점을 체크하게 된다. 그래서 적절한 방법을 찾아야 한다.

경계구 컬링 기법

- 평면 방정식 $ax + by + cz + d = 0$ 에 경계구의 중심점 $V_{center} = (X_c, Y_c, Z_c)$ 의 값을 대입해서 나온 결과값 $AX_c + BY_c + CZ_c + D$ 가 경계구의 반지름 r 보다 작다면 평면과 중심점의 거리가 반지름보다 작은것이므로 경계구에 평면이 포함되는 것이다.



경계구가 절두체에 포함된 경우



경계구가 절두체에 포함되지 않는 경우

1. 절두체 컬링(FRUSTUM CULLING)

실제구현 - **Zfrustum Class**를 분석해 보자.

- 절두체를 구성하는 **8**개의 정점과 **6**개의 평면 방정식을 저장하기 위한 멤버변수를 갖고 있다.
- **Make()**함수는 **Mview * Mproj** 행렬을 받아 월드 좌표계에서의 절두체 정점 좌표를 구한다.
- **sin()**함수는 절두체에 점이 포함되어 있는지를 판정하며, **IsInSphere()**함수는 경계구의 반지름을 사용하여 절두체 포함 여부를 판정한다.
- 이해를 돕기위해 절두체를 그려주는 **Draw()**부분이 있다 **key**컨트롤러 체크 하는 부분이니 이부분을 빼도 무방하다.