삼각형과 광선의 충돌

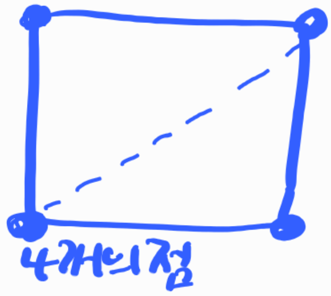
그래픽스에서 삼각형은 매우 중요한데 삼각형들로 여러 물체들을 그리기 때문이다 (Triangle Mesh)

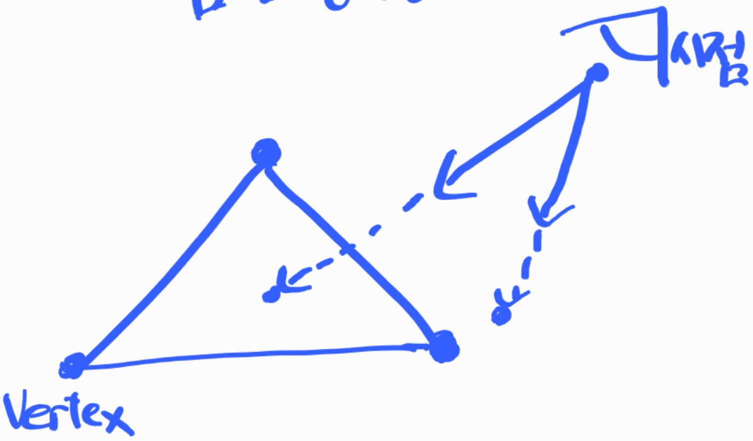
삼각형 기초(Basic)

삼각형은 그 자체가 평면(2D)이다. 그래서 접을 수 없는 평면이다.

접을 수 있는 평면은 사각형 등으로 접으면 2개의 삼각형이 된다

3개의 점으로 정의된다.





Raytracing을 위해 Ray와 삼각형의 교점을 찾는다

Vertex들의 좌표에 따라 삼각형의 위치와 형태가 결정된다

또한, Vertex들의 나열되는 순서에 따라 앞면이 어느 방향인지 (그뒷면의 방향)을 정의,결정된다.

Vertex 나열 순서를 Winding Order라 부른다

1) 삼각형이 놓여있는 무한히 넓은 평면이라 생각한 뒤 Ray와의 무한히 넓은 평면과의 충돌 지점을 찾는다.

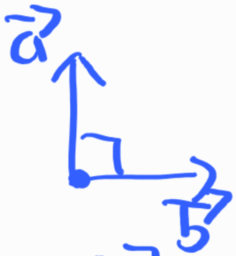
2) 충돌 지점이 삼각형의 안에 있는지 밖에 있는지 구별한다

벡터 복습 - Dot Product ($\vec{a}$, $\vec{b}$ 둘다 Unit Vector인 경우)

$\vec{a}\cdot\vec{b}=\cos\theta$

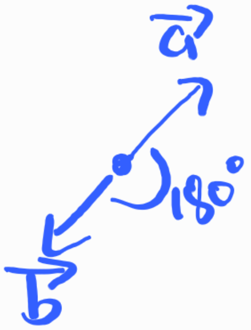
$\vec{a}\cdot\vec{b}=1$

같은방향



$\vec{a}\cdot\vec{b}=0$

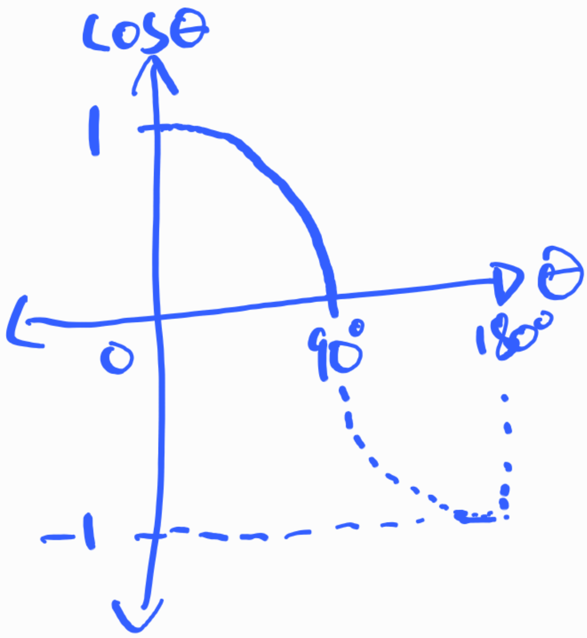
수직



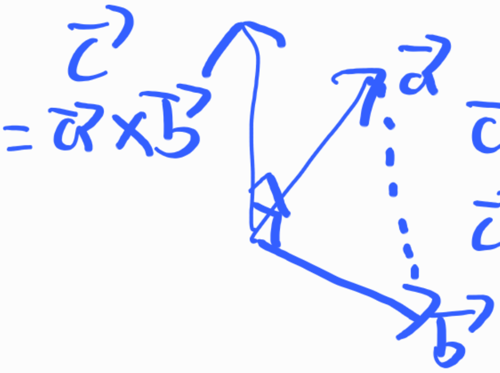
$\vec{a}\cdot\vec{b}=-1$

$\vec{b}=-\vec{a}$

완전 반대방향



벡터복습 - Cross Product



$\vec{a}\times\vec{b}=\vec{c}$ (평면에 수직인 Normal vector)

$\vec{c}$는 $\vec{a}$, $\vec{b}$ 둘다와 수직이다 ($\vec{a}\cdot\vec{c}=\vec{b}\cdot\vec{c}=0$)

$\vec{a}\cdot\vec{c}=0$, $\vec{b}\cdot\vec{c}=0$

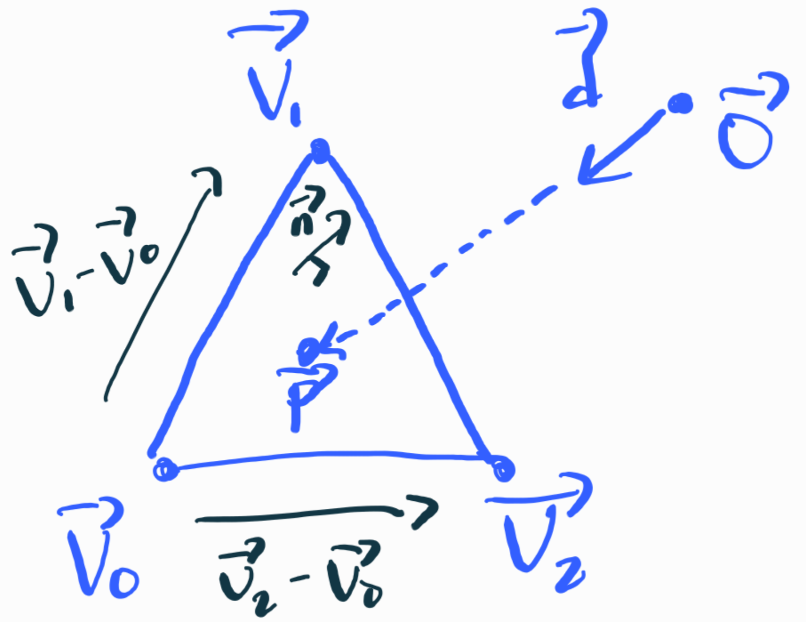
$\vec{a}\neq\vec{b}$, $\vec{a}$와 $\vec{b}$로 평면을 정의할 수 있다

삼각형의 넓이 $=\dfrac{\|\vec{a}\times\vec{b}\|}{2}$

주의! Cross Product에서 곱하는 순서가 바뀌면 결과가 달라진다.

$\vec{a}\times\vec{b}\neq\vec{b}\times\vec{a}$

삼각형과 광선의 충돌 - 1) 무한히 넓은 평면과의 충돌지점



$\vec{V}$ = vertex

$\vec{O}$ = Ray 시작점

$\vec{d}$ = Ray의 방향 (Unit Vector)

$\vec{P}$ = 충돌 지점

$$\vec{P} = \vec{O} + t\vec{d}$$

, t = ? (충돌지점을 알기위해 구해야 하는 값)

삼각형으로 정의되는 평면에 부딪히는 지점이 어딘지를 알기 위해서는 삼각형이 놓여있는 평면의 Normal Vector를 구해야 한다

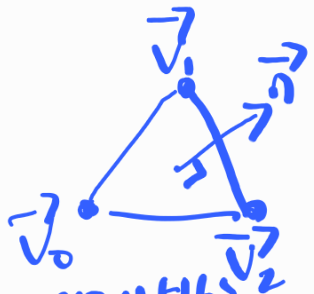
삼각형의 Normal Vector (unit vector)

$$\vec{n} = \frac{(\vec{V_1}-\vec{V_0})\times(\vec{V_2}-\vec{V_0})}{\|(\vec{V_1}-\vec{V_0})\times(\vec{V_2}-\vec{V_0})\|}$$

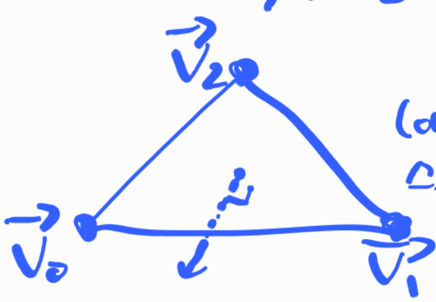
길이

$\vec{V_1}-\vec{V_0}$과 $\vec{V_2}-\vec{V_0}$은 0 벡터가 나오면 안된다. ($x/0$ = error)

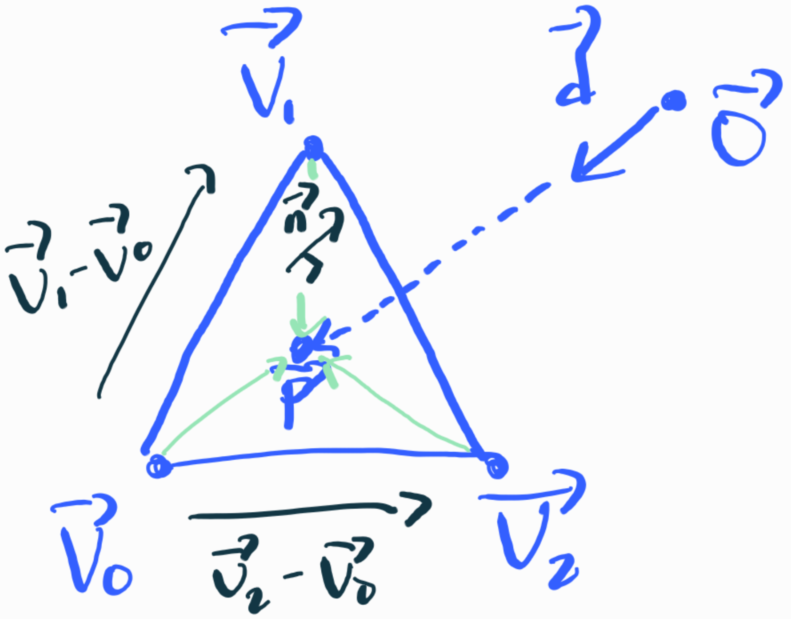
왼손 좌표계에서 $\vec{V_0}$, $\vec{V_1}$, $\vec{V_2}$가 시계 방향(clock-wise)라면 $\vec{n}$(normal vector)가 스크린에서 눈쪽으로 올라오는 방향이다



시계방향



Counter Clock-wise(반시계)
스크린에서 눈으로부터 멀어지는 반대 방향으로 들어가는 방향임



$\vec{P}$가 $\vec{V_0}, \vec{V_1}, \vec{V_2}$와 같은 평면에 있다는 가정하에 진행중

$$\vec{P} = \vec{O} + t\vec{d}$$

$(\vec{P}-\vec{V_0})\cdot\vec{n} = 0$, $\vec{P}-\vec{V_0}$와 $\vec{n}$은 수직이다

$$(\vec{O}+t\vec{d}-\vec{V_0})\cdot\vec{n}=0$$

$$t = \frac{\vec{V_0}\cdot\vec{n} - \vec{O}\cdot\vec{n}}{\vec{d}\cdot\vec{n}},$$

삼각형이 놓여있는 무한히 넓은 평면과의 충돌 지점을 찾을 수 있다.

만약, 삼각형이 뒷면일 때 그리지 않는 것을 Backface culling이라 한다

$-\vec{d} \cdot$ 삼각형 $\vec{n} < 0.0f$ 인경우 뒷면이다.

(광선이 오고있는 반대방향)

(두 벡터가 반대 방향이라는 의미
-1은 완전반대방향)

평면은 두께가 없으므로 평면과 광선이 수평에 매우 가까우면 충돌하지 못한 것으로 판단하고 그리지 않는다.

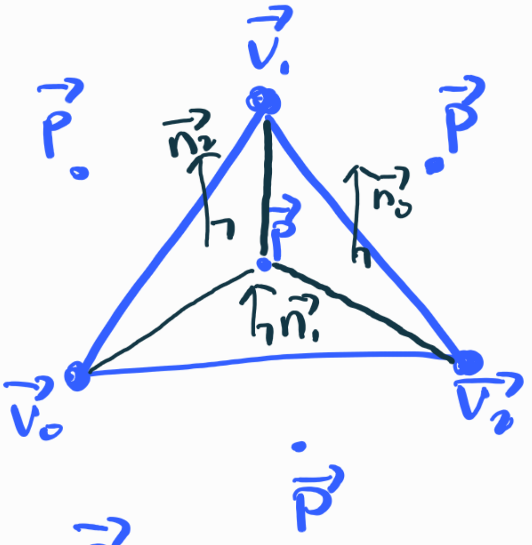
$\text{abs}(\vec{n}\cdot\vec{d}) < 1e\text{-}2f$, normal vector와 ray direction이 수직에 가깝다면 제외

(t값 계산전에 검사해주면 x/0 error를 방지할 수 있다)

광선의 시작점 이전에 충돌 했다면 렌더링하지 않는다.
(광선의 진행방향이 아닌 반대 방향에서 부딪혔다는 의미가 된다)

$t < 0.0f$, 광선의 진행 방향이 아니다

삼각형과 광선의 충돌 -2) 충돌지점이 삼각형 안에 있는지 판단



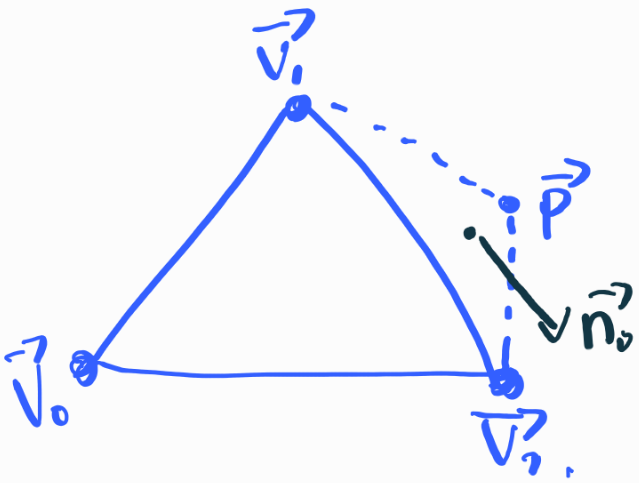
1) 삼각형을 3개의 작은 삼각형으로 나눈뒤 각각의 작은삼각형의 $\vec{n}$, normal vector를 구한다
($\vec{n}$ 계산시 $\vec{v}$들이 시계방향이도록 주의,)

$\vec{P}$가 삼각형 안에 있는 경우 $\vec{n}_0$, $\vec{n}_1$, $\vec{n}_2$가 $\vec{n}$, 평면의 normal vector와 같은 방향을 향한다.

ex) $$\vec{n}_0 = \frac{(\vec{P}-\vec{v}_2)\times(\vec{v}_1-\vec{v}_2)}{\|(\vec{P}-\vec{v}_2)\times(\vec{v}_1-\vec{v}_2)\|},$$

결국 +인지 -인지만 확인하면 되므로 Normalize는 안해도 된다

$\vec{n}_0 \cdot \vec{n} > 0$, $\vec{n}_0$와 $\vec{n}$이 같은 방향이라는 의미



$\vec{P}$가 삼각형의 밖에 있는 경우 $\vec{n}_0$가 $\vec{n}$과 반대 방향이 된다

$\vec{n}_0 \cdot \vec{n} < 0$, 서로 반대방향

$\vec{n}_1 \cdot \vec{n} < 0$

$\vec{n}_2 \cdot \vec{n} < 0$

$\vec{n}_0$, $\vec{n}_1$, $\vec{n}_2$와 $\vec{n}$의 Dot Product가 하나라도 0보다 작다면 $\vec{P}$는 삼각형 밖에 있다는 의미로 렌더링 하지 않는다

$\vec{P}$가 밖인 경우 $\vec{n}$를 계산할 때 $\vec{v}$의 순서가 반시계 방향으로 적용되 반대방향이 된다