3D Game Programming 02

afewhee@gmail.com

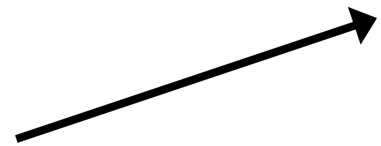
- **Scalar**
  - 대상을 수학적으로 표현하기 위해 1개의 요소로 표현할 수 있으면 scalar

- **Vector**
  - 아일랜드의 수학자 해밀턴이 창안
  - 간단하게 정리하면 대상을 수학적으로 표현하기 위해 최소한 1개 이상의 쌍으로 표현을 해야 하는 경우에는 vector

  - 벡터의 표현 방법
    - 기하학적 표현: 크기와 방향을 가진 화살표로 표시
    - 대수학적 방법: 원소의 나열로 표시
    - 벡터 대수학적 표현: 벡터 기호로 표시

1. 벡터

- 벡터 표현
  - 기하학적 표현
    - 차원이 없이 머리와 꼬리로 구성된
      직선 화살표로 표시

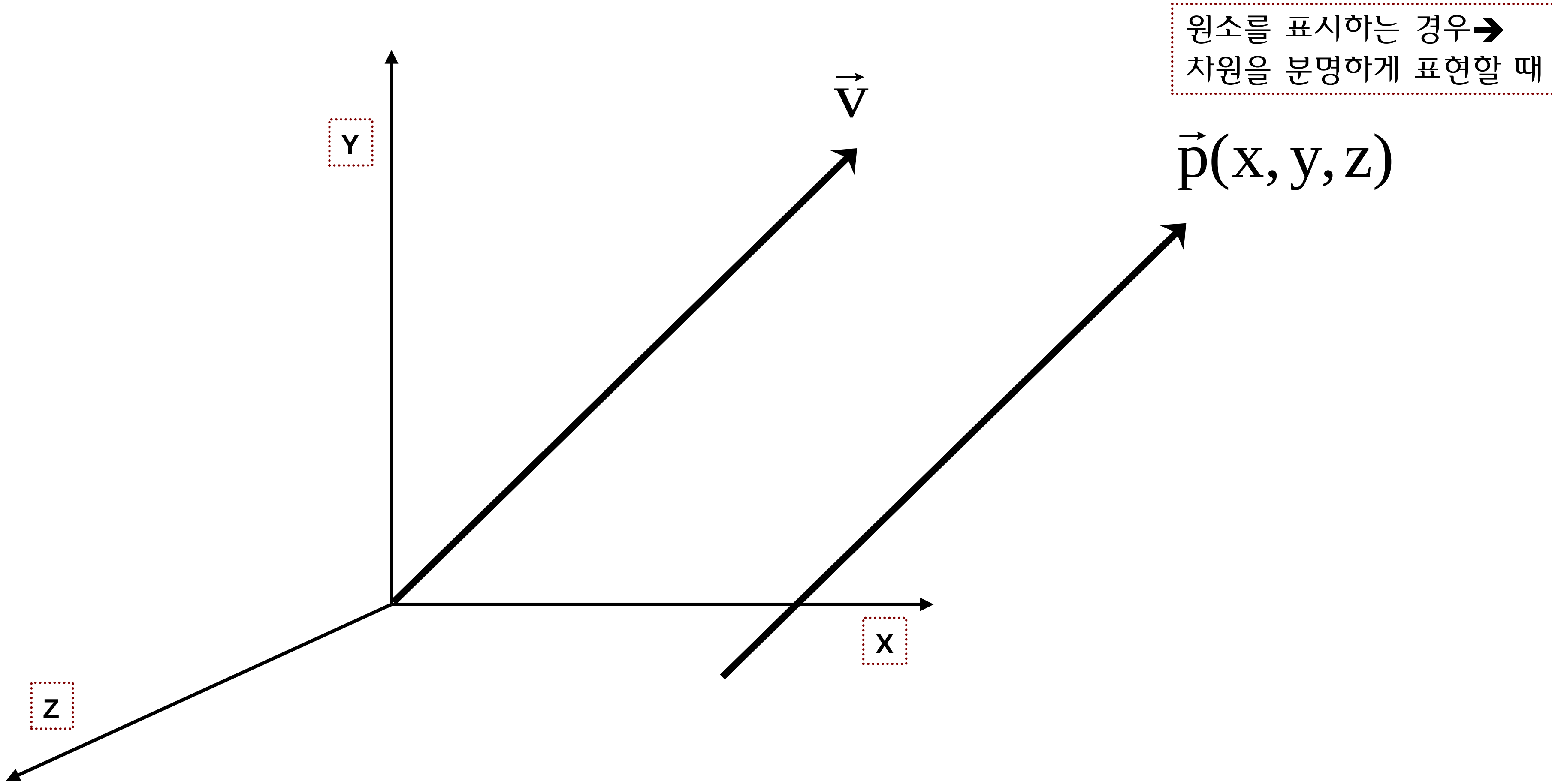


  - 대수학적 표현
    - 차원에 따라 n개의 원소로 표현
      1차원: (a), 2차원: (a, b), 3차원: (a,b,c), n차원: (a,b,…,n)
  - 벡터기호 표시:
    - 굵은 글씨: 인쇄술이 발달하지 못했던 시절에 유행.
      지금도 많이 사용 ex) **V, M, T**
    - 문자 위의 화살표: $\vec{V}$ $\vec{\mathrm{V}}$
    - 간결한 표시: 주로 물리학, 수학자들이 애용: $\underset{\sim}{V}$

- 벡터는 그림, 기호, 대수학적 방법을 혼용해서 사용
  - EX)

$$\vec{P}(x, y, z) \qquad \mathbf{M}(x, y, z)$$

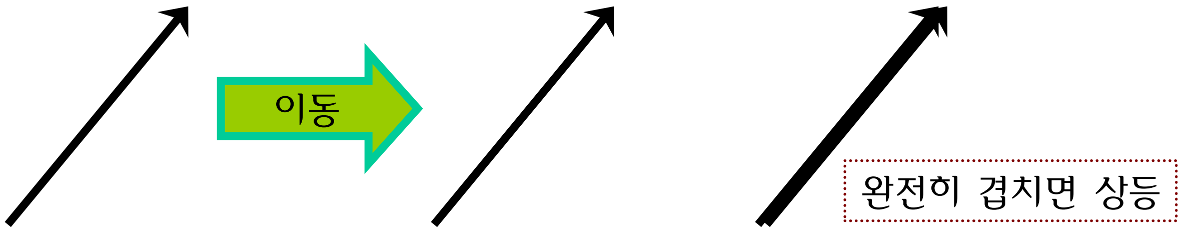
- 기하학, 대수학, 기호를 가지고 동시에 표현



벡터의 상등

- 기하학적 표현: 화살표의 길이와 방향이 같으면 상등

➔ 하나의 벡터를 이동해서 다른 벡터와 완전히 겹치면 상등



- 대수학적 표현: 차원이 같고 각각의 차원에 대응되는 원소의 값이 같은 경우

  EX)

  vector a=(a_1, a_2, a_3, a_4)

  vector b=(b_1, b_2, b_3, b_4)

  a_1==b_1, a_2==b_2, a_3==b_3, a_4==b_4

- 벡터 기호 표현: $\vec{V}_1 = \vec{V}_2$

프로그램에서 벡터의 상등

- 벡터는 대부분 FLOAT 형을 이용해서 표현. 이 때 FLOAT형은 오차가 존재하므로 벡터의 상등을 3차원에서 다음과 같이 비교하면 좋지 않음

```
if(v1.x == v2.x && v1.y == v2.y && v1.x == v2.x)
{
	상등;
}
```

- Epsilon(작은 편차 값)을 이용해서 비교

```
const FLOAT EPSILON = 0.001f;

if(fabsf(v1.x - v2.x)< EPSILON &&
   fabsf(v1.y - v2.y)< EPSILON &&
   fabsf(v1.z - v2.z)< EPSILON)
{
	상등;
}
```

벡터의 크기

- 기하학적 표현: 선분의 길이

- 대수적 표현: $(v_1, v_2, \ldots v_n)$ ➔ $\sqrt{v_1^2 + v_2^2 + v_3^2 + \ldots + v_n^2} = \left(\sum v_i^2\right)^{\frac{1}{2}}$

- 기호적 표현: $\|\vec{v}\|$, $|\vec{v}|$

- "Norm" 또는 "Length" 로 읽음 ➔ "Norm"은 어감상 좋지 않아 주로 "Length"로 읽음

- D3DXVec3Length(), D3DXVec3LengthSq()

- 벡터의 정규화: Normalize
  - 벡터의 크기를 1로 만듦
  - 단위 벡터: 크기가 1인 벡터 = Unit Vector
  - 단위 벡터의 기호: $\hat{v} = \dfrac{\vec{v}}{\|\vec{v}\|}$
  - 3차원의 경우: $\hat{v} = \left(\dfrac{v_x}{\|\vec{v}\|}, \dfrac{v_y}{\|\vec{v}\|}, \dfrac{v_z}{\|\vec{v}\|} \right)$
  - D3DXVec3Normalize(*pOut, *pIn), D3DXVec2Normalize()
  - 표준 기저 벡터(Base Vector)
    - n 차원의 벡터에서 k 번째 성분만 1이고 나머지는 0인 벡터
      Ex) 3차원 데카르트 좌표계의 기저벡터(1,0,0), (0,1,0), (0,0,1)
    - 단위벡터의 한 종류
    - 벡터는 기저 벡터에 실수를 곱하고 이들을 더해서 만들 수 있음
    - Ex) (2,3,4) ➔ 2*(1,0,0) + 3*(0,1,0) 4*(0,0,1)
      ➔ $2*\hat{i}+3*\hat{j}+4*\hat{k} \Rightarrow \sum v_i \hat{e}_i$

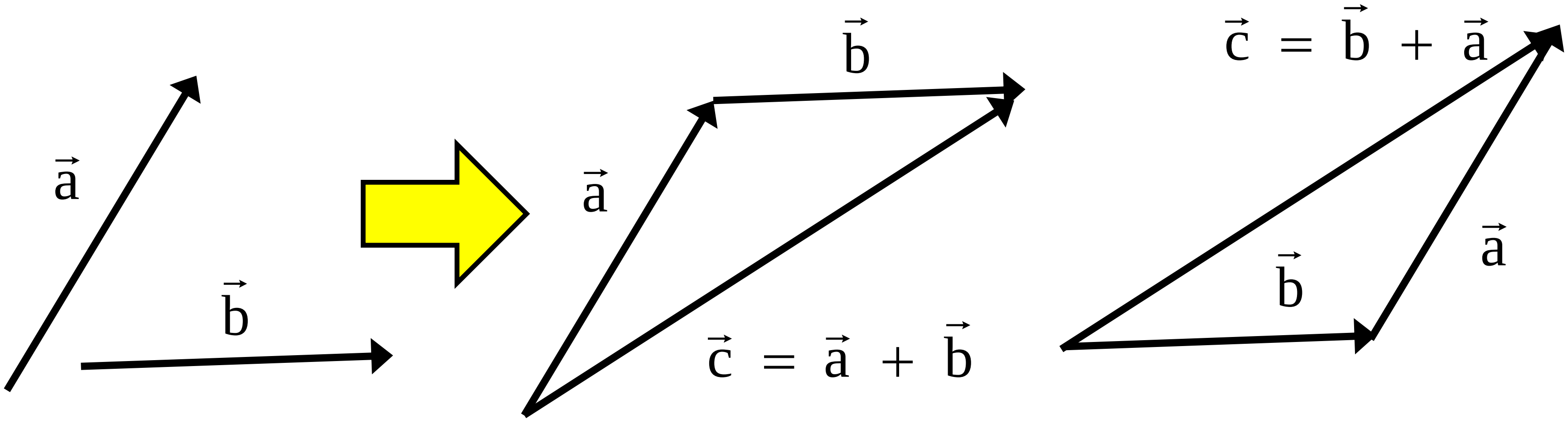
1. 벡터

- 벡터의 연산
  - 차원이 같고, 물리량이 같은 벡터 사이에서만 가능
  - 더하기, Negative, 빼기, 스칼라 곱, 내적, 외적

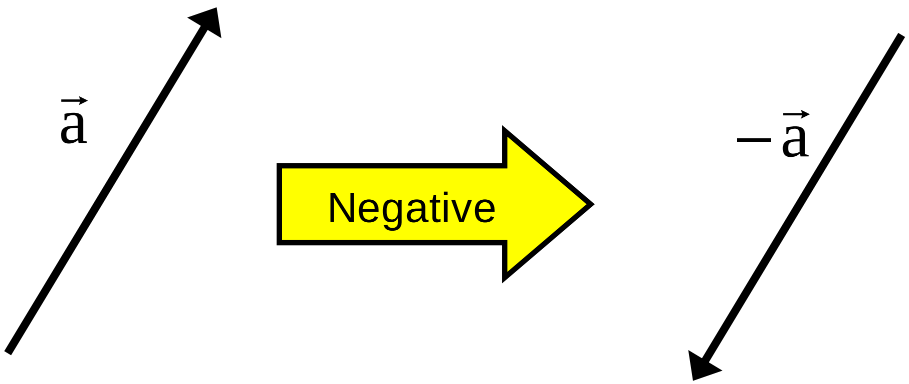
- 더하기
  - 대수적: 두 벡터의 성분끼리 더해 하나의 벡터를 만듦.
    a=(a1, a2, a3), b= (b1, b2, b3) a+b = (a1+b1, a2+b2, a3+b3)
  - 기호적:

$$\vec{c} = \vec{a} + \vec{b}$$

  - 기하학적:
    - 삼각형법: 벡터의 합성
    - 평행사변형법: 벡터의 분해

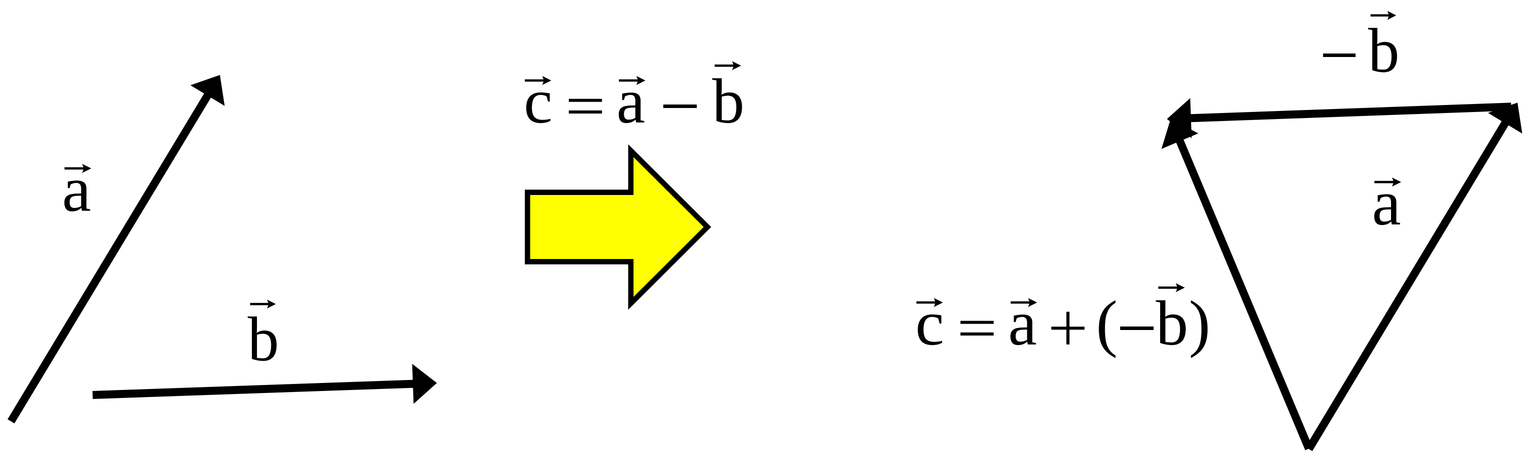


- Negative:
  - 벡터 성분의 부호를 반전
    a=(a1, a2, a3), ➔ -a = -(a1, a2, a3) ➔ -a = (-a1, -a2, -a3)
  - 기호적: $\text{Negative } \vec{a} = -\vec{a}$
  - 기하학적 의미: 선분의 길이는 그대로 유지하고 방향만 반대로 바꿈

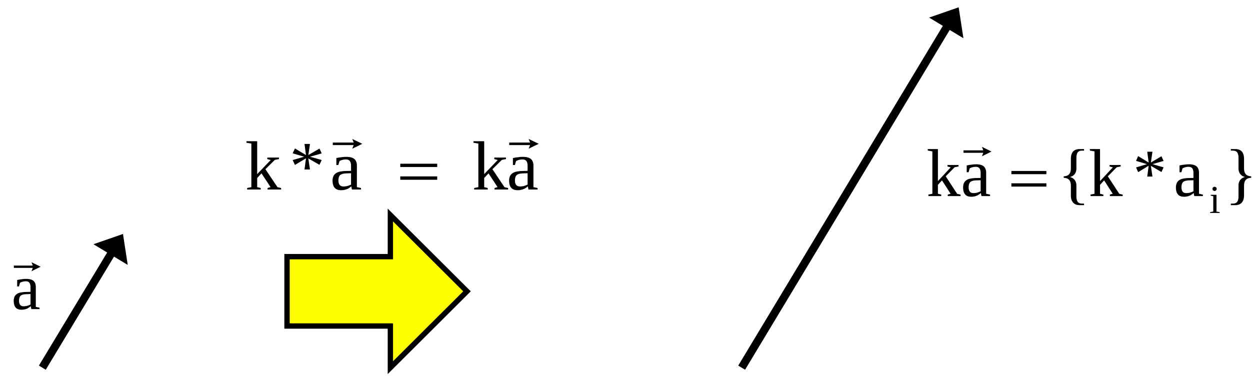


- 빼기
  - 대수적: 두 벡터의 성분끼리 뺄셈을 수행
    a=(a1, a2, a3), b= (b1, b2, b3) a-b = (a1-b1, a2-b2, a3-b3)
  - 기호적: $\vec{c} = \vec{a} - \vec{b}$
  - 기하학적 의미: 하나의 벡터를 Negative로 한 다음 덧셈을 수행

$$\vec{c} = \vec{a} - \vec{b} = \vec{a} + (-\vec{b})$$



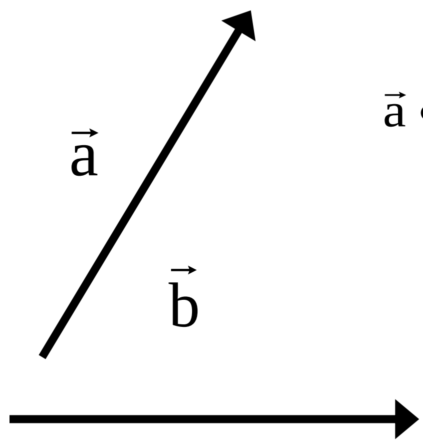
- **Scalar 곱**
  - 벡터의 크기를 변화
  - 대수적 : 각각의 성분에 scalar값을 곱함
    a=(a1, a2, a3), k= 실수, k*a = k*(a1, a2, a3) = (k*a1, k*a2, k*a3)
  - 기호적: 곱셈 기호 이용.
  - 무 방향 벡터인 영 벡터(모든 성분이 0), Negative를 정의 Scalar 곱으로 정의 할 수 있음
  - 기하학적 의미: 벡터의 길이에 k배 함



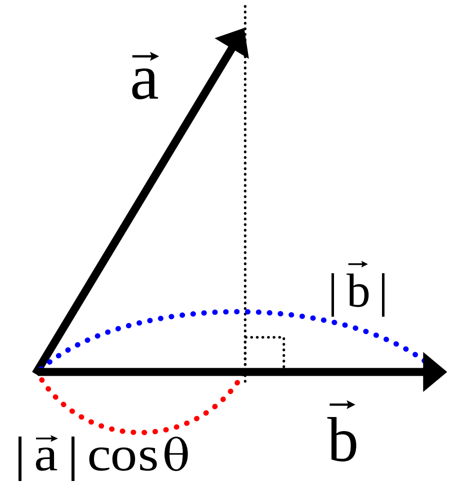
- 내적(Inner Product)
  - 두 벡터의 길이와 벡터 사이의 각도에 대한 cos 값을 곱한 량
  - 기호적: $f = \vec{a} \bullet \vec{b} = |\vec{a}||\vec{b}|\cos(\vec{a}, \vec{b})$
  - 두 벡터의 연산 결과가 scalar여서 scalar product(연산, 곱)이라 부름
  - 연산의 기호를 dot(점)을 사용해 dot product 라 부름
  - 대수적: 두 벡터의 성분끼리 곱하고 이들을 다시 더해 스칼라 값을 만듦
    a=(a1, a2, a3), b= (b1, b2, b3)   dot(a,b) = a.b = a1*b1 + a2*b2 + a3*b3

$$f = \vec{a} \bullet \vec{b} = \sum a_i * b_i$$

  - 기하학적 의미: 사상(Projection: 투영, 사영)
  - 벡터의 크기 계산에도 사용 : $a.a = \Sigma a_i * a_i = |a|^2$
  - D3DXVector3Dot()

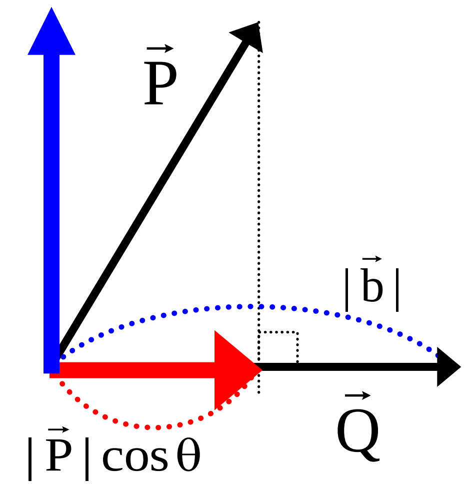


$$\vec{a} \bullet \vec{b} = |\vec{a}||\vec{b}|\cos(\vec{a}, \vec{b})$$



벡터 투영(Projection)

- 하나의 벡터를 다른 벡터의 평행 과 수직 성분으로 분해
- 평행 사변형법 이용



$$\vec{P}_{proj\vec{Q}} = |\vec{P}|\cos\theta * \hat{Q} = |\vec{P}|\left(\frac{\vec{P}\bullet\vec{Q}}{|\vec{P}\|\vec{Q}|}\right) * \hat{Q}$$

$$= \left(\frac{\vec{P}\bullet\vec{Q}}{|\vec{Q}|}\right) * \hat{Q} = \left(\frac{\vec{P}\bullet\vec{Q}}{|\vec{Q}|}\right) * \frac{\vec{Q}}{|\vec{Q}|}$$

$$= \frac{(\vec{P}\bullet\vec{Q})}{|\vec{Q}|^2}\vec{Q}$$

$$\therefore \vec{P}_{proj\vec{Q}} = \vec{P}\bullet\frac{\vec{Q}\vec{Q}}{|\vec{Q}|^2}$$

$$\vec{P}'_{proj\vec{Q}} = \vec{P} - \vec{P}_{proj\vec{Q}}$$

$$= \vec{P} - \frac{(\vec{P}\bullet\vec{Q})}{|\vec{Q}|^2}\vec{Q}$$

$$= \vec{P} - \vec{P}\bullet\frac{\vec{Q}\vec{Q}}{|\vec{Q}|^2}$$

$$= \vec{P}\bullet\left(\vec{I} - \frac{\vec{Q}\vec{Q}}{|\vec{Q}|^2}\right)$$

$$\therefore \vec{M}'_{\vec{P}proj\vec{Q}} = \vec{I} - \frac{\vec{Q}\vec{Q}}{|\vec{Q}|^2}$$

$$\vec{Q}\vec{Q} = \begin{bmatrix} Q_xQ_x & Q_xQ_y & Q_xQ_z \\ Q_yQ_x & Q_xQ_x & Q_yQ_z \\ Q_zQ_x & Q_zQ_y & Q_zQ_z \end{bmatrix}$$

$$\therefore \vec{M}_{\vec{P}proj\vec{Q}} = \frac{1}{|\vec{Q}|^2}\begin{bmatrix} Q_x{}^2 & Q_xQ_y & Q_xQ_z \\ Q_yQ_x & Q_y{}^2 & Q_yQ_z \\ Q_zQ_x & Q_zQ_y & Q_z{}^2 \end{bmatrix}$$

$$= \frac{1}{|\vec{Q}|^2}\begin{bmatrix} 0 & -Q_xQ_y & -Q_xQ_z \\ -Q_yQ_x & 0 & -Q_yQ_z \\ -Q_zQ_x & -Q_zQ_y & 0 \end{bmatrix}$$

- 외적(Outer Product)
  - 법선 벡터(Normal Vector): 두 벡터에 수직인 벡터
  - 외적: 두 벡터의 단위 법선 벡터에 두 벡터의 길이와 벡터 사이의 각도에 대한 sin 값을 벡터의 크기로 정한 벡터

$$\vec{N} = \vec{a} \times \vec{b} = |\vec{a}||\vec{b}|\sin(\vec{a}, \vec{b})\,\hat{n}, \quad \hat{n} : \vec{a} \perp \vec{b} \text{ unit vector}$$

  - 방향을 나타내는 단위벡터 n 은 오른손 법칙(Right Hand Rule)을 사용해서 정함
  - 두 벡터의 연산 결과가 vector여서 vector product(연산, 곱)이라 부름
  - 연산의 기호가 X(cross)이어서 cross product 라 부름
  - 대수적: $\vec{C} = \vec{A} \times \vec{B}$

$$C_k = \sum_i \sum_j a_i * b_j * \varepsilon_{ijk}$$

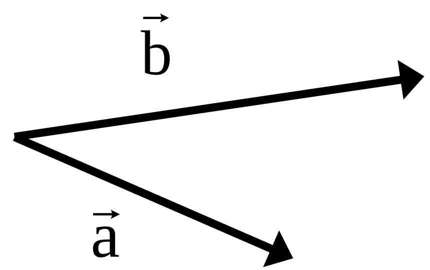
$$i = j \text{ or } j = k \text{ or } k = i : \varepsilon_{ijk} = 0$$

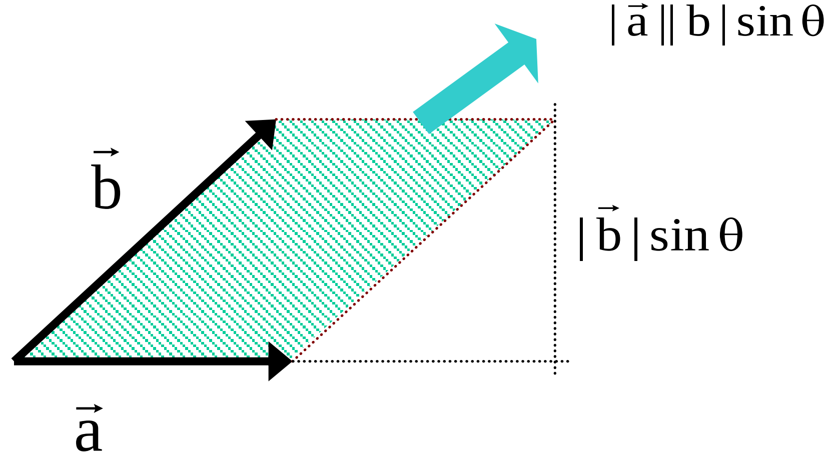
$$i \to j \to k,\ j \to k \to i,\ k \to i \to j : \varepsilon_{ijk} = 1$$

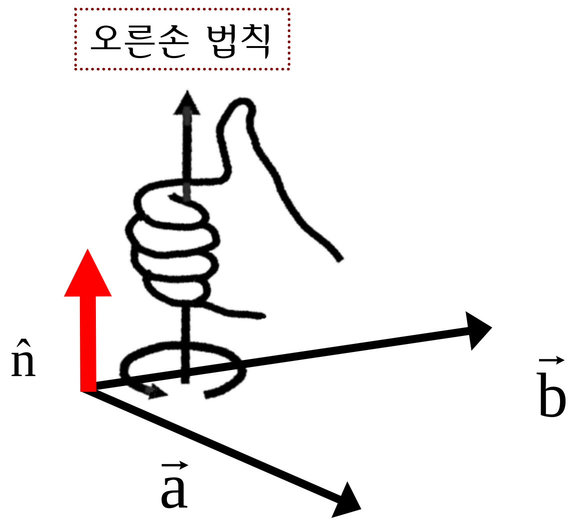
$$i \to k \to j,\ j \to i \to k,\ k \to j \to i : \varepsilon_{ijk} = -1$$

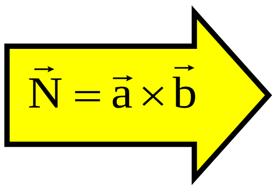
  - a=(a1, a2, a3), b= (b1, b2, b3)
    cross(a,b) = a x b = (a2*b3 - a3*b2, a3*b1 - a1*b3, a1*b2 - a2*b1)
  - 기하학적 의미: 두 벡터가 이루는 평행사변 형 넓이
  - D3DXVec3Cross()

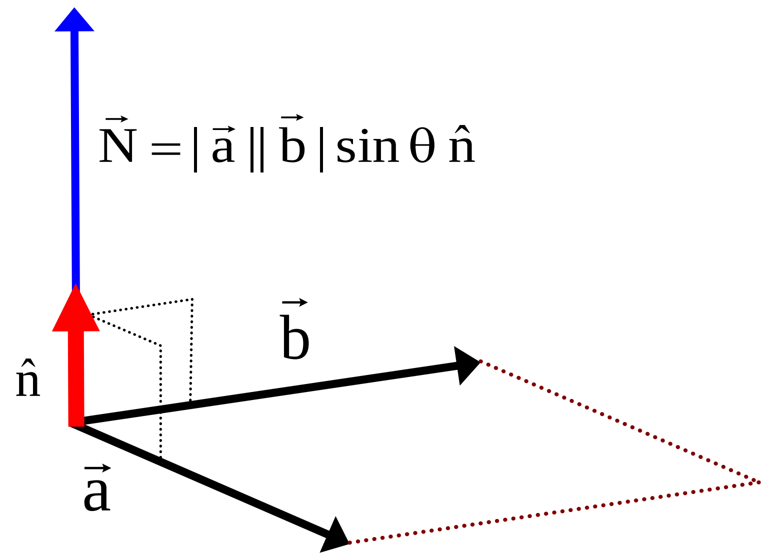
외적의 형성과정


$\vec{b}$
$\vec{a}$


$|\vec{a}||\vec{b}|\sin\theta$
$\vec{b}$
$|\vec{b}|\sin\theta$
$\vec{a}$


오른손 법칙
$\hat{n}$
$\vec{b}$
$\vec{a}$


$\vec{N} = \vec{a} \times \vec{b}$


$\vec{N} = |\vec{a}||\vec{b}|\sin\theta\,\hat{n}$
$\vec{b}$
$\hat{n}$
$\vec{a}$

외적에 대한 행렬 표현

$$\vec{C} = \vec{P} \times \vec{Q}$$

$$= (P_y Q_z - P_z Q_y,\ P_z Q_x - P_x Q_z,\ P_x Q_y - P_y Q_x)$$

$$= (P_x, P_y, P_z)\begin{pmatrix} 0 & -Q_z & Q_y \\ Q_z & 0 & -Q_x \\ -Q_y & Q_x & 0 \end{pmatrix}$$

$$\therefore M_{\vec{P}\times\vec{Q}} = \begin{pmatrix} 0 & -Q_z & Q_y \\ Q_z & 0 & -Q_x \\ -Q_y & Q_x & 0 \end{pmatrix}$$

외적에 대한 행렬식 표현

$$\vec{C} = \vec{P} \times \vec{Q}$$

$$= \begin{vmatrix} \hat{x} & \hat{y} & \hat{z} \\ P_x & P_y & P_z \\ Q_x & Q_y & Q_z \end{vmatrix}$$

$$= \left|\begin{array}{ccc|cc} \hat{x} & \hat{y} & \hat{z} & \hat{x} & \hat{y} \\ P_x & P_y & P_z & P_x & P_y \\ Q_x & Q_y & Q_z & Q_x & Q_y \end{array}\right.$$

$$= \hat{x}(P_y Q_z - P_z Q_y)$$
$$+ \hat{y}(P_z Q_x - P_x Q_z)$$
$$+ \hat{z}(P_x Q_y - P_y Q_x)$$

- Matrix
  - 대상을 수학적으로 N x M 의 행과 열의 2차원으로 표현한 것
  - 대수학: 대괄호[], 또는 중괄호() 안에 원소 표시. 또는 소괄호{} 안에 원소에 아래 첨자를 붙임➔ 소괄호 없이 원소의 아래 첨자만 붙여서도 사용

2X3 $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ 3X2 $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ $\{a_{ij}\}$ a_{ij}

  - 기호: 굵은 고딕의 주로 대문자 사용➔ 굵은 소문자는 벡터 ex) **S, R, T**
  - 벡터기호: 문자 위에 양방향의 화살표 표시
    ➔ 공학, 물리학에서 주로 사용

$\overleftrightarrow{M}, \overleftrightarrow{S}, \overleftrightarrow{R}, \overleftrightarrow{T}$

- 행 벡터(Low Vector)
  - 하나의 행으로 구성된 행렬: 1xn 행렬

$\begin{bmatrix} x & y \end{bmatrix}$ $\begin{bmatrix} x & y & z \end{bmatrix}$ $\begin{bmatrix} x & y & z & w \end{bmatrix}$

- 열 벡터(Column Vector)
  - 하나의 열로 구성된 행렬: nX1 행렬

$\begin{bmatrix} V_x \\ V_y \end{bmatrix}$ $\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}$ $\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$

- Scalar, Vector, Matrix
  - Scalar: 1차원 벡터 v=(a), 또는 1x1행렬 {a}
  - Vector: 행 또는 열 벡터, 1xN 또는 Nx1 행렬 $\{M_{n1}\}$, $\{M_{1n}\}$

- 상등
  - 차원이 같은 두 행렬의 원소가 같은 경우 $\forall_{ij}(a_{ij} == b_{ij} \rightarrow true)$
- 스칼라 곱
  - 행렬에 대한 Scalar 곱은 각각의 모든 원소에 Scalar를 곱한 것 $k\overleftrightarrow{M} = \{k * M_{ij}\}$ or kM_{ij}
- 행렬의 덧셈, 뺄셈
  - 차원이 같은 두 행렬의 덧셈은 각각의 원소를 더해서 새로운 행렬을 만드는 것 $\overleftrightarrow{M} = \overleftrightarrow{R} \pm \overleftrightarrow{S} \Rightarrow M_{ij} = R_{ij} \pm S_{ij}$
- 행렬 사이의 곱셈
  - 행렬 M과 행렬 S의 곱은 M의 행렬의 열과 S의 행이 같은 차원일 때만 성립하고 최종 행렬의 차원은 M의 행, S의 열이 됨 $\overleftrightarrow{M} * \overleftrightarrow{S} \neq \overleftrightarrow{S} * \overleftrightarrow{M}$
  - 계산식 $\overleftrightarrow{T} = \overleftrightarrow{M} * \overleftrightarrow{S} \Leftrightarrow T_{ij} = \sum_k M_{ik} S_{kj}$

Ex)

$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} g & h \\ i & j \\ k & l \end{bmatrix} = \begin{bmatrix} a*g+b*i+c*k & a*h+b*j+c*i \\ d*g+e*i+f*k & d*h+e*j+f*i \end{bmatrix}$$

$$\begin{bmatrix} 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 4*1+5*2+6*3 \end{bmatrix} = \begin{bmatrix} 32 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \begin{bmatrix} 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 1*4 & 1*5 & 1*6 \\ 2*4 & 2*5 & 2*6 \\ 3*4 & 3*5 & 3*6 \end{bmatrix}$$

- 항등 행렬(Identity Matrix: 단위(Unit) 행렬)
  - 가로 세로의 크기가 같은 행렬 중에서 대각선 모두가 1이고, 나머지는 0인 행렬
  - 항등 행렬에 어떤 행렬을 곱해도 자기 자신이 됨
  - 기호: I
  - D3DXMatrixIdentity()

$$\overleftrightarrow{\mathrm{I}}$$

- 역 행렬(Inverse Matrix) $\overleftrightarrow{\mathrm{M}}^{-1}$
  - 어떤 행렬에 행렬을 곱하고 그 결과가 항등 행렬인 행렬
  - 기호 : -1

$$\overleftrightarrow{\mathrm{M}}\overleftrightarrow{\mathrm{M}}^{-1} = \overleftrightarrow{\mathrm{M}}^{-1}\overleftrightarrow{\mathrm{M}} = \overleftrightarrow{\mathrm{I}}$$

  - 회전행렬의 역 행렬: 각도를 반대로 적용한 행렬
  - 이동 행렬의 역 행렬: 반대로 이동한 행렬
  - 크기 변환행렬의 역행렬: Scale 값을 1/Scale로 적용한 행렬
  - 행렬 곱에서의 역 행렬 ➔

$$\left(\overleftrightarrow{\mathrm{M}}\overleftrightarrow{\mathrm{S}}\right)^{-1} = \overleftrightarrow{\mathrm{S}}^{-1}\overleftrightarrow{\mathrm{M}}^{-1}$$

  - D3DXMatrixInverse()

2. 행렬

- 전치 행렬 (Transpose)
  - 행렬의 열과 행을 교환
  - D3DXMatrixTranspose()

$$M_{ij}^{T} = M_{ji}$$

- 행렬식(Determinant)
  - 행렬에서 유도되는 Scalar 값
  - 기호: **det M, |M|** **det M = ΣM$_{ij}$C$_{ij}$(M), C$_{ij}$(M):** Cofactor-여인수
  - Sarrus 방법1

2x2

$$|\overleftrightarrow{M}| = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

$$= a * d - b * c$$

3x3

$$|\overleftrightarrow{M}| = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} \begin{matrix} a & b \\ d & e \\ g & h \end{matrix}$$

$$= aei + bfg + cdh$$

$$- ceg - afh - bdi$$

$$= a(ei - fh) + b(fg - di) + c(eg - dh)$$

D3DXMatrixDeterminant()

- 벡터의 3대 변환
  - 크기(Scaling), 회전(Rotation), 이동(Translation)
  - 기하 변환(Geometric Transform): 위치 이동

- 변환에 대한 행렬
  - 3D는 3개의 좌표로 구성되어 있어 변환에 대한 행렬이 3x3 행렬이 필요
  - 변환에 대해서 크기 변환 행렬, 회전 행렬, 이동 행렬이 존재
  - 행렬을 4x4로 가지면 3차원에 대해서 Scaling, Rotation, Translation을 하나의 행렬에서 처리 가능 ➔ 동차 좌표계가 필요

  - 4x4행렬 연산을 위해서 3차원 좌표(x,y,z)를 등가인 4차원 homogeneous 좌표(x,y,z,w=1)를 이용함

- Direct3D의 행렬
  - Direct3D는 왼손 좌표계 사용 ➔ 벡터의 변환은 행 벡터 * 행렬 연산을 수행 ➔ 행 중심 배열

왼손 좌표계 벡터 변환 = 벡터*행렬

$$[x' \quad y' \quad z' \quad 1] \Leftarrow [x \quad y \quad z \quad 1]\begin{bmatrix} R_{11} & R_{12} & R_{13} & 0 \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix}$$

오른손 좌표계 벡터 변환 = 행렬*벡터

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \Leftarrow \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

기하 변환 예

$$\begin{bmatrix} x' & y' & z' & w' \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & M_{14} \\ R_{21} & R_{22} & R_{23} & M_{24} \\ R_{31} & R_{32} & R_{33} & M_{34} \\ T_x & T_y & T_z & M_{44} \end{bmatrix}$$

$$x' = x*R_{11} + y*R_{21} + z*R_{31} + T_x$$
$$y' = x*R_{12} + y*R_{22} + z*R_{32} + T_y$$
$$z' = x*R_{13} + y*R_{23} + z*R_{33} + T_z$$
$$w' = x*R_{14} + y*R_{24} + z*R_{34} + R_{44}$$
$$(\text{if } R_{14} \leftarrow 0 \text{ and } R_{24} \leftarrow 0 \text{ and } R_{34} \leftarrow 0 \text{ and } R_{44} \leftarrow 1 \text{ then } w' = 1)$$

$$x'/= w', x'/= w', x'/= w', x'/= w'$$
$$\begin{bmatrix} x' & y' & z' & w' \end{bmatrix} = [x'/w' \quad y'/w' \quad z'/w' \quad 1]$$

- 정점의 평행 이동(Translation)
  - 정점의 위치를 상대적으로 이동
  - V' = V(x,y,z)+T(x,y,z)
    = (Vx+Tx, Vy+Ty, Vz+Tz)

- 행렬을 이용한 변환

$$\begin{bmatrix} V_x & V_y & V_z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix} = \begin{bmatrix} V_x + T_x & V_y + T_y & V_z + T_z & 1 \end{bmatrix}$$

- 이동 행렬

$$T(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

$$T^{-1}(t) = T(-t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -t_x & -t_y & -t_z & 1 \end{bmatrix}$$

  - D3DXMatrixTranslation(*pOutMatrix, Tx,Ty,Tz)

- 크기변환 (Scaling)
  - 정점의 위치에 scalr 곱을 적용한 것
  - V' = S(x,y,z) ⊗ V(x,y,z)
    = (Vx*Sx, Vy*Sy, Vz*Sz)

- 행렬을 이용한 변환

$$\begin{bmatrix} V_x & V_y & V_z & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} V_x * S_x & V_y * S_y & V_z * S_z & 1 \end{bmatrix}$$

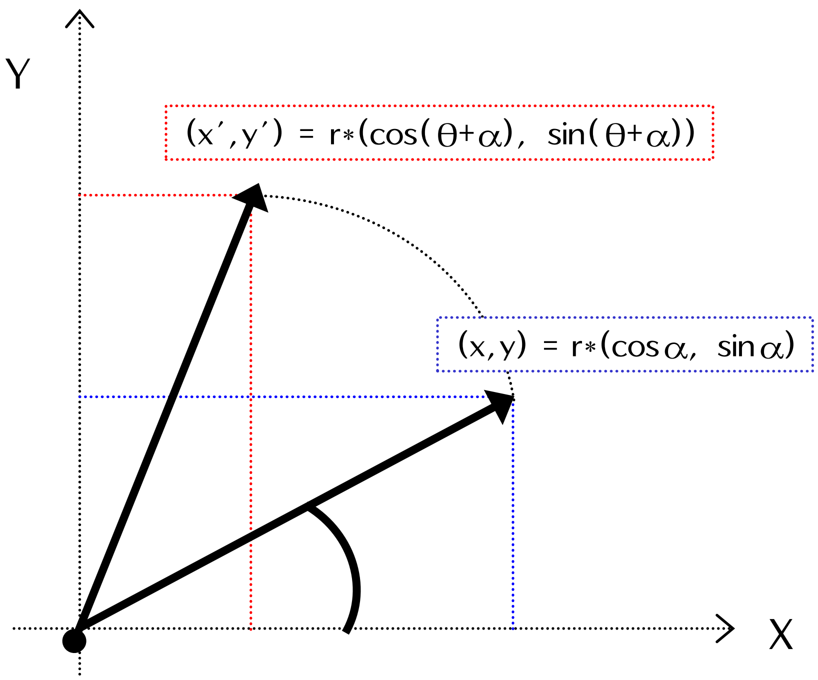
- 크기 변환 행렬 (Scaling Matrix)

$$S(s) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S^{-1}(s) = S(\frac{1}{s_x} \quad \frac{1}{s_y} \quad \frac{1}{s_z}) = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 & 0 \\ 0 & 0 & \frac{1}{s_z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - D3DXMatrixScaling(*pOutMatrix, Sx,Sy,Sz)

- 정점의 회전(Rotation)
  - 좌표 축에 대한 회전 ➔ 구하기 쉬움
  - 임의의 축에 대한 회전 ➔ Quaternion 사용
- 2차원 회전 변환



$$x' = r*\cos(\theta+\alpha)$$
$$= r*\cos\theta\cos\alpha - r*\sin\theta\sin\alpha$$
$$= \cos\theta*(r*\cos\alpha) - \sin\theta*(r*\sin\alpha)$$
$$= \cos\theta*x - \sin\theta*y$$

$$y' = r*\sin(\theta+\alpha) = r*\sin\theta\cos\alpha + r*\cos\theta\sin\alpha$$
$$= \sin\theta*r*\cos\alpha + \cos\theta*r*\sin\alpha$$
$$= \sin\theta*x + \cos\theta*y$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} r*\cos(\theta+\alpha) \\ r*\sin(\theta+\alpha) \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\begin{pmatrix} x' & y' \end{pmatrix} = \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

- X, Y, Z축에 대한 회전 행렬

$$M_{RotX}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{RotY}(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{RotZ}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - D3DXMatrixRotationX()
  - D3DXMatrixRotationY()
  - D3DXMatrixRotationZ()

- 회전 행렬의 역 행렬
  - 각도에 반대 각도(-θ)를 적용한 행렬
  - 전치. Transpose(R) = Inverse(R) ← Orthogonal Matrix 특징

- 임의의 축에 대한 회전
  - 행렬의 순서가 바뀌면 두 행렬의 곱 M*N ≠N*M이므로 정점의 변환을 회전행렬에서 적용할 때 회전에 대한 행렬을 연속적으로 적용할 때 문제가 발생 →짐벌락(Gimbal Lock)

  - 각도에 대해서 누적으로 하고 임의의 축에 대한 회전으로 변경 →Quaternion 사용

- Quaternion(사원수)
  - 해밀턴이 창안
  - 회전에 대한 문제를 풀기 위해 주로 이용
  - 하나의 실수, 3개의 허수로 구성되어 있는 형태로 복소수의 연장

$$\hat{q} = w + x\hat{i} + y\hat{j} + z\hat{k} = \cos\theta + \sin\theta\hat{v} = [x, y, z, w] = (s, \hat{v})$$

$$s = w$$

$$\hat{v} = [x, y, z]$$

- 사원수의 성질

▶좌표축 단위벡터들에 대한 곱셈 정의

$$\hat{i} * \hat{i} = -1 \qquad \hat{i} * \hat{j} = -\hat{j} * \hat{i} = \hat{k}$$

$$\hat{j} * \hat{j} = -1 \qquad \hat{j} * \hat{k} = -\hat{k} * \hat{j} = \hat{i}$$

$$\hat{k} * \hat{k} = -1 \qquad \hat{k} * \hat{i} = -\hat{i} * \hat{k} = \hat{j}$$

▶복소 공액(Conjugate)과 크기

$$\hat{q} = w + x\hat{i} + y\hat{j} + z\hat{k}$$

$$\hat{q}' = w - x\hat{i} - y\hat{j} - z\hat{k}$$

$$\|\hat{q}\| = \sqrt{\hat{q} * \hat{q}'} = \sqrt{w^2 + x^2 + y^2 + z^2}$$

▶사원수 사이의 곱셈

$$\hat{q}_1 = (s_1, \hat{v}_1)$$

$$\hat{q}_2 = (s_2, \hat{v}_2)$$

$$\hat{q}_1 * \hat{q}_2 = (s_1 s_2 - \hat{v}_1 \cdot \hat{v}_2,\ s_1\hat{v}_2 + s_2\hat{v}_1 + \hat{v}_1 \times \hat{v}_2)$$

▶사원수 사이의 곱은 결합법칙만 성립(증명가능)

$$\hat{q}_1 * \hat{q}_2 * \hat{q}_3 = (\hat{q}_1 * \hat{q}_2) * \hat{q}_3 = \hat{q}_1 * (\hat{q}_2 * \hat{q}_3)$$

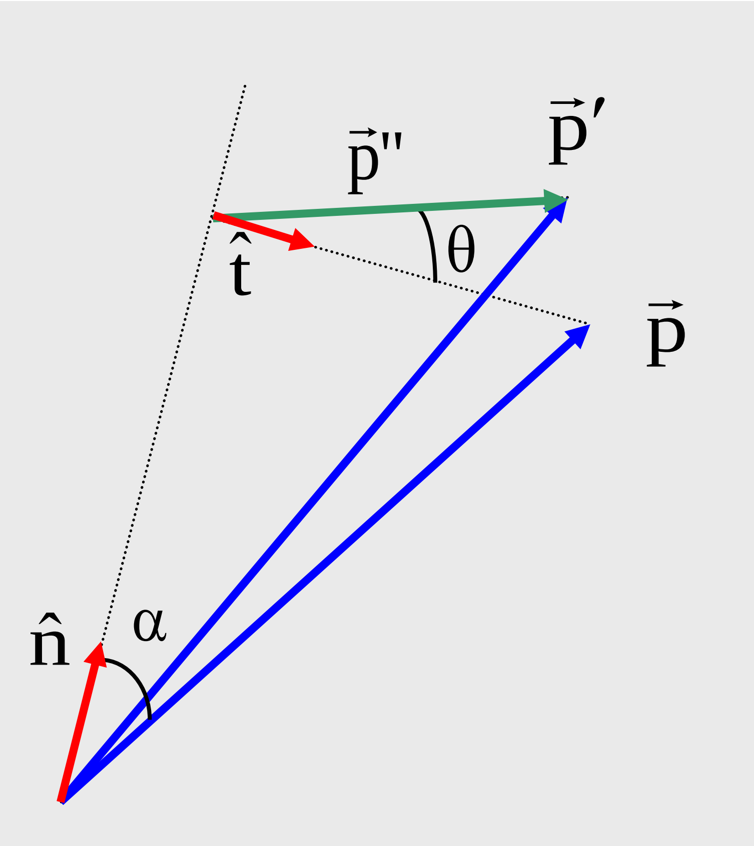
$$\hat{q}_1 * \hat{q}_2 \neq \hat{q}_2 * \hat{q}_1$$

$$\|\hat{q}\| = 1$$

$$\hat{q}' = \hat{q}^{-1}$$

회전과 Quaternion

임의의 축(n)에 대한 점 p의 회전



$$\vec{p}' = (\vec{p}\cdot\hat{n})\hat{n} + \vec{p}''$$

$$\vec{p}'' = \|\vec{p}''\|\cos\theta\hat{t} + \|\vec{p}''\|\sin\theta\frac{(\hat{n}\times\vec{p})}{\|\hat{n}\times\vec{p}\|}$$

$$\|\vec{p}''\| = \|\vec{p}\|\sin\alpha$$

$$\|\vec{p}\|\sin\alpha = \|\hat{n}\times\vec{p}\|$$

$$\vec{p}'' = \|\vec{p}''\|\cos\theta\hat{t} + \sin\theta * (\hat{n}\times\vec{p})$$

$$\|\vec{p}''\|\hat{t} = \vec{p} - (\vec{p}\cdot\hat{n})\hat{n}$$

$$\vec{p}'' = \cos\theta(\vec{p} - (\vec{p}\cdot\hat{n})\hat{n}) + \sin\theta(\hat{n}\times\vec{p})$$

$$\vec{p}' = (\vec{p}\cdot\hat{n})\hat{n} + \cos\theta(\vec{p} - (\vec{p}\cdot\hat{n})\hat{n}) + \sin\theta(\hat{n}\times\vec{p})$$

$$\therefore \vec{p}' = \cos\theta\vec{p} + \sin\theta(\hat{n}\times\vec{p}) + (1-\cos\theta)(\hat{n}\cdot\vec{p})\hat{n}$$

회전과 Quaternion

- 2개의 quaternion q, P

  q∗P∗q-1

$$\vec{p}_{rotated} = q * \vec{p} * q^{-1}$$

$$= \left(\cos\frac{\theta}{2},\ u\sin\frac{\theta}{2}\right) * (0, \vec{v}) * \left(\cos\frac{\theta}{2},\ -\hat{u}\sin\frac{\theta}{2}\right)$$

$$= \left(-\sin\frac{\theta}{2}(\hat{u}\cdot\vec{v}), \cos\frac{\theta}{2}\vec{v} + \sin\frac{\theta}{2}(\hat{u}\times\vec{v})\right) * \left(\cos\frac{\theta}{2}, -\hat{u}\sin\frac{\theta}{2}\right)$$

$$= \left(-\sin\frac{\theta}{2}(\hat{u}\cdot\vec{v}), \cos\frac{\theta}{2}\vec{v} + \sin\frac{\theta}{2}(\hat{u}\times\vec{v})\right) * \left(\cos\frac{\theta}{2}, -\hat{u}\sin\frac{\theta}{2}\right)$$

$$= \left(\begin{array}{l} -\sin\frac{\theta}{2}\cos\frac{\theta}{2}(\hat{u}\cdot\vec{v}) + \sin\frac{\theta}{2}\cos\frac{\theta}{2}(\vec{v}\cdot\hat{u}) - \sin^2\frac{\theta}{2}(\hat{u}\times\vec{v})\cdot\hat{u}, \\ \sin^2\frac{\theta}{2}(\hat{u}\cdot\vec{v})\hat{u} + \cos^2\frac{\theta}{2}\vec{v} + \sin\frac{\theta}{2}\cos\frac{\theta}{2}(\hat{u}\times\vec{v}) \\ -\sin\frac{\theta}{2}\cos\frac{\theta}{2}(\vec{v}\times\hat{u}) - \sin^2\frac{\theta}{2}(\hat{u}\times\vec{v})\times\hat{u} \end{array}\right)$$

$$(\hat{u}\times\vec{v})\cdot\hat{u} = 0,\quad \vec{A}\times\vec{B} = -\vec{B}\times\vec{A}$$

$$\vec{A}\times(\vec{B}\times\vec{C}) = (\vec{A}\cdot\vec{C})\vec{B} - (\vec{A}\cdot\vec{B})\vec{C}$$

$$-(\hat{u}\times\vec{v})\times\hat{u} = \hat{u}\times(\hat{u}\times\vec{v})$$

$$\hat{u}\times(\hat{u}\times\vec{v}) = (\hat{u}\cdot\vec{v})\hat{u} - (\hat{u}\cdot\hat{u})\vec{v}$$

$$= (\hat{u}\cdot\vec{v})\hat{u} - \vec{v}$$

$$\cos(a+b) = \cos(a)\cos(b) - \sin(a)\sin(b)$$

$$\cos\theta = \cos\left(\frac{\theta}{2}+\frac{\theta}{2}\right)$$

$$= \cos\frac{\theta}{2}\cos\frac{\theta}{2} - \sin\frac{\theta}{2}\sin\frac{\theta}{2}$$

$$= \cos^2\frac{\theta}{2} - \sin^2\frac{\theta}{2}$$

$$= \cos^2\frac{\theta}{2} - \left(1-\cos^2\frac{\theta}{2}\right)$$

$$= 2\cos^2\frac{\theta}{2} - 1 \quad = \quad 2\left(1-\sin^2\frac{\theta}{2}\right) - 1$$

$$= 1 - 2\sin^2\frac{\theta}{2}$$

회전과 Quaternion

- 2개의 quaternion q, P, q^{-1} 의 곱(계속)

$$=\left(0, \sin^2\frac{\theta}{2}(\hat{u}\cdot\vec{v})\hat{u}+\cos^2\frac{\theta}{2}\vec{v}+\sin\theta(\hat{u}\times\vec{v})+\sin^2\frac{\theta}{2}\hat{u}\times(\hat{u}\times\vec{v})\right)$$

$$\vec{p}'=\sin^2\frac{\theta}{2}(\hat{u}\cdot\vec{v})\hat{u}+\cos^2\frac{\theta}{2}\vec{v}+\sin\theta(\hat{u}\times\vec{v})+\sin^2\frac{\theta}{2}\hat{u}\times(\hat{u}\times\vec{v})$$

$$=\sin^2\frac{\theta}{2}(\hat{u}\cdot\vec{v})\hat{u}+\cos^2\frac{\theta}{2}\vec{v}+\sin\theta(\hat{u}\times\vec{v})+\sin^2\frac{\theta}{2}(\hat{u}\cdot\vec{v})\hat{u}-\sin^2\frac{\theta}{2}\vec{v}$$

$$=(\cos^2\frac{\theta}{2}-\sin^2\frac{\theta}{2})\vec{v}+\sin\theta(\hat{u}\times\vec{v})+2\sin^2\frac{\theta}{2}(\hat{u}\cdot\vec{v})\hat{u}$$

$$\therefore\vec{p}'=\cos\theta\vec{v}+\sin\theta(\hat{u}\times\vec{v})+(1-\cos\theta)(\hat{u}\cdot\vec{v})\hat{u}$$

벡터를 이용한 것과 동일

축에 대한 정점 P의 회전은 회전 각이 ½로 만든 사원수 q(θ/2)의 곱 qPq-1 와 동일

- Quaternion을 이용한 회전
  - 임의 축(Axis)과 각도를 사원수로 바꾼다. 이 때 각도는 ½배 한다
    q=(cos(θ/2), sin(θ/2)*AxisX, sin(θ/2)*AxisY, sin(θ/2)*AxisZ)
  - 회전을 적용할 정점을 사원수로 바꾼다
    P=(0, Px, Py, Pz)
  - 사원수 곱 q*P*q-1 연산을 수행한다
  - 최종 연산의 결과도 사원수 이므로 이중에서 허수 부를 선택한다
- 임의의 축이 여러 개 일 때 회전
  - q1, q2을 순서대로 적용하면

$$
\begin{aligned}
&q_2 * (q_1 * \vec{P} * q_1^{-1}) * q_2^{-1} \\
&= (q_2 * q_1) * \vec{P} * (q_1^{-1} * q_2^{-1}) \\
&= (q_2 * q_1) * \vec{P} * (q_2 * q_1)^{-1}
\end{aligned}
$$

  - q2 * q1을 먼저 계산한 후 이의 공액을 구하고 다시 qPq-1을 계산

Quaternion의 회전 행렬 전환

$$\vec{p}_{rotated} = q * \vec{p} * q^{-1}$$
$$= \cos\theta\vec{v} + \sin\theta(\hat{u}\times\vec{v}) + (1-\cos\theta)\hat{u}(\hat{u}\cdot\vec{v})$$
$$= \left(\cos^2\frac{\theta}{2} - \sin^2\frac{\theta}{2}\right)\vec{I}\vec{v}$$
$$+ 2\cos\frac{\theta}{2}\begin{bmatrix} 0 & -\sin\frac{\theta}{2}\hat{u}_z & \sin\frac{\theta}{2}\hat{u}_y \\ \sin\frac{\theta}{2}\hat{u}_z & 0 & -\sin\frac{\theta}{2}\hat{u}_x \\ -\sin\frac{\theta}{2}\hat{u}_y & \sin\frac{\theta}{2}\hat{u}_x & 0 \end{bmatrix}\vec{v}$$
$$+ 2\left(\sin\frac{\theta}{2}\hat{u}\right)\left(\sin\frac{\theta}{2}\hat{u}\right)\cdot\vec{v}$$

$$= \begin{bmatrix} w^2-x^2-y^2-z^2 & 0 & 0 \\ 0 & w^2-x^2-y^2-z^2 & 0 \\ 0 & 0 & w^2-x^2-y^2-z^2 \end{bmatrix}\vec{v}$$
$$+ \begin{bmatrix} 0 & -2wz & 2wy \\ 2wz & 0 & -2wx \\ -2wy & 2wx & 0 \end{bmatrix}\vec{v} + \begin{bmatrix} 2x^2 & 2xy & 2xz \\ 2wz & 2y^2 & 2yz \\ 2xz & 2yz & 2z^2 \end{bmatrix}\vec{v}$$
$$= \begin{bmatrix} 1-2y^2-2z^2 & 2xy-2wz & 2xz+2wy \\ 2xy+2wz & 1-2x^2-2z^2 & 2yz-2wx \\ 2xz-2wy & 2yz+2wx & 1-2x^2-2y^2 \end{bmatrix}v$$

$$\therefore R_q = \begin{bmatrix} 1-2y^2-2z^2 & 2xy-2wz & 2xz+2wy \\ 2xy+2wz & 1-2x^2-2z^2 & 2yz-2wx \\ 2xz-2wy & 2yz+2wx & 1-2x^2-2y^2 \end{bmatrix}$$

- 선형 보간 (Linear Interpolation)
  - p = (1-t) * p1 + t*p2
    t: weight, p1, p2 시작, 끝의 물리량

- 회전에서 보간 (Interpolation)
  - 선형 보간 원리를 이용해서 각도에 적용한 후 외적 등을 이용해서 중간 위치를 구함

$$\theta = \theta(1-t) + \theta t$$

$$\vec{p} = a \cdot \vec{p}_i + b \cdot \vec{p}_f$$

$$\vec{p}_i \times \vec{p} = a \cdot \vec{p}_i \times \vec{p}_i + b \cdot \vec{p}_i \times \vec{p}_f$$

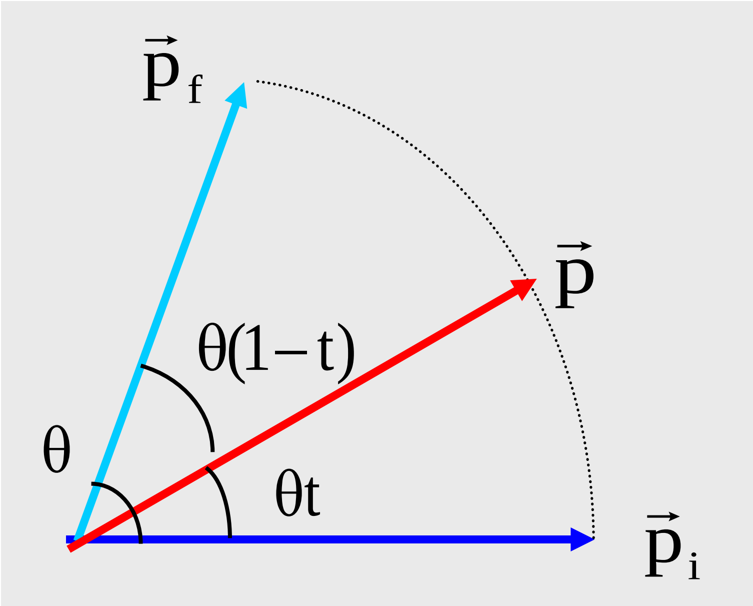
$$\sin\theta t \cdot \hat{n} = a \cdot \vec{0} + b \sin\theta \cdot \hat{n}$$

$$\sin\theta t = b \sin\theta \quad \therefore b = \frac{\sin\theta t}{\sin\theta}$$

$$\vec{p} \times \vec{p}_f = a \cdot \vec{p}_i \times \vec{p}_f + b \cdot \vec{p}_f \times \vec{p}_f$$

$$\sin\theta(1-t) = a \sin\theta \quad \therefore a = \frac{\sin\theta(1-t)}{\sin\theta}$$

$$\therefore \vec{p} = \frac{1}{\sin\theta}\left[\sin\theta(1-t)\ \vec{p}_i + \sin(\theta t)\ \vec{p}_f\right]$$



θ ~ 0 인 경우…
직선에 대한 선형 보간과 유사

$$\sin\theta \approx \theta, \sin\theta t \approx \theta t, \sin\theta(1-t) \approx \theta(1-t)$$

$$\therefore \vec{p} = (1-t) \cdot \vec{p}_i + t \cdot \vec{p}_f$$

- 정점의 변환 순서
  - 회전 중심점으로 상대적으로 이동: T′
  - 크기 변환 적용: S
  - 회전 변환 적용: R
  - 다시 원래의 위치로 이동: T
  - 최종 행렬: T′SRT

  - 3차원 물체들은 원점을 중심으로 작업을 하므로
    최종 행렬 = SRT
  - 최종행렬에 정점을 곱하면 변환된 정점의 위치가 구해짐

- 정점 변환에 대한 DirectX함수
  - D3DXVec3TransformCoord(): w=1로 놓고 변환 → 크기+회전+이동 변환
  - D3DXVec3TransformNormal(): w=0로 놓고 변환 → 회전과 크기만 변환

  - 위의 두 함수는 V′ = V * Matrix 를 계산한 결과와 동일

Affine변환

- $T'_i = \Sigma M_{ij} T_j + V_i$
- 3D 프로그램의 정점의 변환은 전부 Affine 변환
- 아핀 변환은 이전의 특성을 그대로 유지
  - 평행선은 변환 후 평행선에 대응, 유한한 점은 이전의 유한 점에 대응
- 이동, 회전, 크기, 대칭, 밀림(Shear) 변환은 아핀 변환의 한 종류
- 단순한 이동, 회전, 대칭 변환의 경우 각도, 길이, 평행관계가 그대로 유지
  ➔ 선분의 길이, 각도는 변환 후에도 유지

4. 실습

- Float형 3차원 벡터 클래스를 작성하시오.
- Operator Overloading을 이용해서 벡터 사이의 연산, Scalar 곱을 구현하시오.
  또한 내적은 *, 외적은 ^ 연산자로 구현하시오.
- Float형 4x4, Matrix 클래스를 작성하시오.
- Scalar 곱, 행렬 사이의 연산을 Operator Overloading을 이용해서 구현하시오.
- 크기, 이동, X축, Y축, Z축에 대한 회전 변환 행렬을 구현하시오.
  ➔ 구현의 결과가 DirectX SDK함수와 비슷한지 비교해보시오.