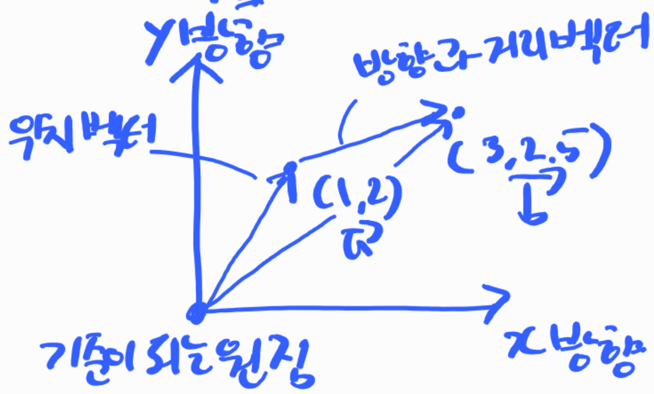
벡터와 포인트

일반적인 벡터 : $\vec{a}$, $\vec{b}$ 위치벡터와 벡터

그래픽스 수학

- P, Q, R : 포인트(Point, 위치)
- u, v, w : 벡터(vector) - 방향, 절대값
- α, β, γ : 스칼라(Scalar)
- **a** : 벡터를 행렬로 표기할 경우
- M : 행렬

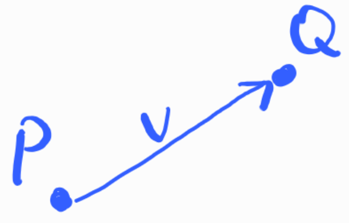
벡터 복습



포인트와 벡터 (사용법은 기존과 같으나 생각하는 방식이 다르다)

- 포인트 •
  - 위치만 표현 ex) a의 위치, b의 위치
- 벡터 ↗
  - 방향과 크기 표현 ex) 속도, 힘

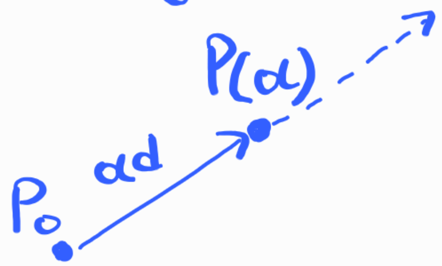
포인트 - 벡터 연산



연산	결과	ex)
포인트 - 포인트	= 벡터	Q - P = v
포인트 ± 벡터	= 포인트	P + v = Q

* 대부분의 API에선 포인트와 벡터를 구별하지 않는다.

직선 (Line)

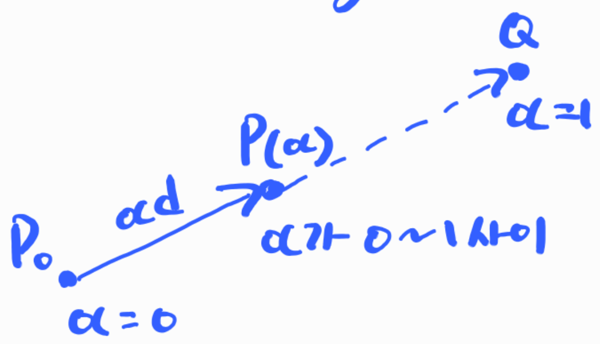


직선위의 점들

$P(\alpha) = P_0 + \alpha d$, $P(\alpha)$ = 포인트

P_0 = 위치, α = 스칼라, d = 벡터

선분 (Line Segment) - 직선의 일부분

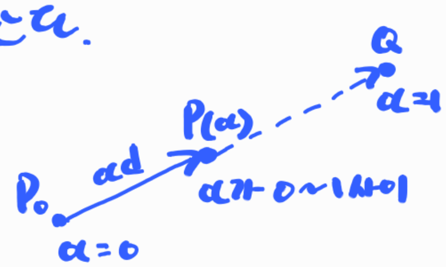


$$P(\alpha) = P_0 + \alpha(Q - P_0),\ 0 \le \alpha \le 1$$
$$= P_0 + \alpha d$$
$$= (1-\alpha)P_0 + \alpha Q$$

P_0 = 시작점, Q = 끝점

여러개의 포인터들 앞에 각각 스칼라를 곱한 뒤 더하는 것을 Affine Combination (Affine Sum) 이라고 한다.

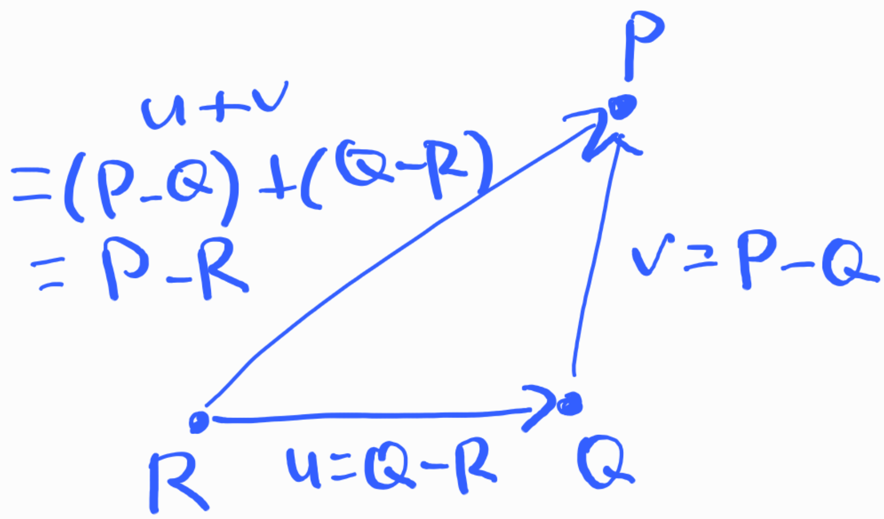
$$P(\alpha) = \underbrace{(1-\alpha)P_0 + \alpha Q}_{\text{가중치}},\ 0 \le \alpha \le 1$$



조건 : 포인터들에 곱해지는 스칼라 값들의 합이 1이 되어야한다

$$(1-\alpha) + \alpha = 1$$

벡터 더하기



행렬 벡터

2차원에서는 숫자 2개, 3차원에서는 숫자 3개

$$u = (1, 2, 3)$$

$$u = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^T, \quad T = \text{Transpose}$$

Homogeneous Coordinates (동차 좌표) - 벡터와 포인트 개념을 같이 사용하는 좌표계

$$u = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 0 \end{bmatrix} \quad u = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}$$

4차원 x
벡터와 포인트 구별

마지막이 0이면 벡터 아니면 포인트 (보통 1)

코드에선 포인트와 벡터의 구별이 없으므로 개발자가 결정한다

- GLM
  - glm::vec2, glm::vec3, glm::vec4 등
- DirectXMath
  - XMFLOAT4
  - XMVECTORF32
- SimpleMath
  - Vector2, Vector3, Vector4

* 포인트는 보는 사람과 관계없이 물체의 고유 성질로 위치를 다룰 때 사용하는 개념
위치 벡터는 특정 좌표계를 기준으로 물체의 위치를 정의할 때 사용하는 도구

* 동일한 포인트가 좌표계에 따라서 서로 다른 위치벡터로 표현된다.

* 벡터로 위치를 표현하기 위해선 기준점과 기준방향이 필요하다 (벡터만으로 위치표현 x)
그래서 어떤 좌표축의 원점과 좌표축 방향을 기준으로 위치를 표현하는 걸 위치벡터라 한다