유니티로 배우는 게임 수학

함수, 매개변수 방정식 개념

아주대학교 미디어학과

오규환 (drghoh@ajou.ac.kr)

함수 (FUNCTION) 의 정의

- **집합 A와 B : 함수는 두 개의 집합 A와 B 사이의 관계**
  - 여기서 A를 정의구역(domain), B를 공변역(codomain)이라고 호칭함
- **연결 규칙 : 함수는 A의 각 원소를 B의 한 원소에 대응시키는 규칙이 존재**
- **단일성 : 정의역 A의 각 원소에 대해 공변역 B의 단 하나의 원소가 대응**
  - A의 어떤 원소 x에 대해 B의 두 개 이상의 다른 원소가 대응할 수 없음
- **함수는 보통 $f : A \rightarrow B$ 로 표기**
  - "함수 f는 집합 A의 원소를 집합 B의 원소로 보낸다"는 의미
- **A의 원소 x 가 B의 원소 y 에 대응될 때, 이를 f(x) = y 로 나타냄**

함수 (FUNCTION) 예시

- **수의 제곱 함수**
  - 정의구역 : 모든 실수 (R)
  - 공변역 : 모든 양의 실수
  - $f(x) = x^2$
- **Sin 함수**
  - 정의구역 : 모든 실수 (R)
  - 공변역: [-1, 1] 구간의 실수
  - 함수: $f(x) = \sin(x)$

함수 (FUNCTION) 예시

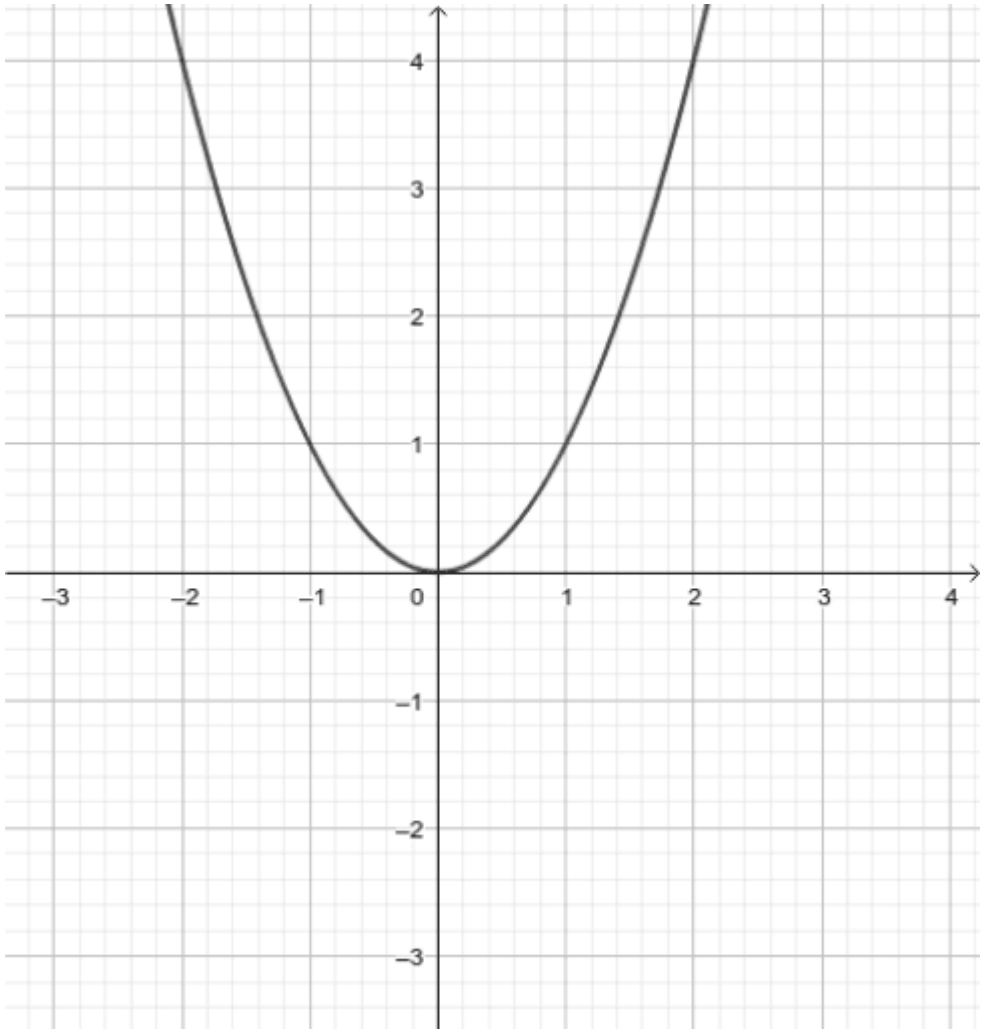
- **일대일 함수 (Injective function) : 정의구역의 서로 다른 원소들이 공역의 서로 다른 원소들에 대응되는 함수**
  - $f(x) = 2x$
- **전사 함수 (Surjective function) : 공변역의 모든 원소가 정의구역의 적어도 하나의 원소와 연결되는 함수**
  - $g(x) = sin(x)$
  - 예를 들어, y = 0.5 라면 sin(π/6) = 0.5
- **전단사 함수 (Bijective function) : 일대일 수이면서 동시에 전사 함수인 함수. 즉, 정의구역의 모든 원소가 공변역의 유일한 원소와 일대일 대응**
  - $f(x) = x + 3$

역함수 (INVERSE FUNCTION)

- **f : A → B가 주어졌을 때, 다음 조건을 만족하는 함수 f^{-1}: B → A가 존재하면, 이 함수를 f의 역함수라고 한다.**
  - $f\left(f^{-1}(y)\right) = y$ for all y ∈ B
  - $f^{-1}\left(f(x)\right) = x$ for all x ∈ A
- **역함수를 구하기 위해서는 주어진 함수의 식을 y = f(x)형태로 쓰고, 이를 x에 대해 풀어 x를 y의 함수로 나타내면 됨**
  - $\boldsymbol{y = 2x + 3}$ 의 역함수
  - $y - 3 = 2x$
  - $x = \frac{y-3}{2}$
  - $f^{-1}(y) = \frac{y-3}{2}$
- **역함수의 그래프 예시**
  - $\boldsymbol{y = 2x + 3}$ 와 $f^{-1}(y) = \frac{y-3}{2}$ 는 y = x에 대해 대칭

$f(x) = x$

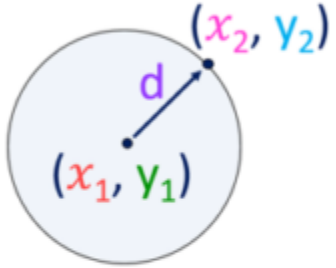
- (x, y) 가 대응되나 ? / 주어진 (x, y) 가 그래프 위의 점인가 ?



원 : $x^2 + y^2 = r^2$

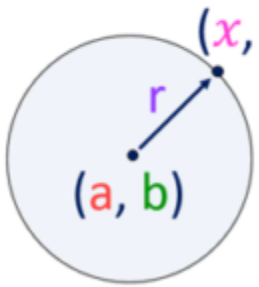
Deriving the Equation of a Circle

1. Start with the Pythagorean equation for distance between 2 points



$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = d$$

2. Substitute in the values of (a, b) as the centre and r as the radius



$$\sqrt{(x - a)^2 + (y - b)^2} = r$$

3. Square both sides of the equation $(x - a)^2 + (y - b)^2 = r^2$

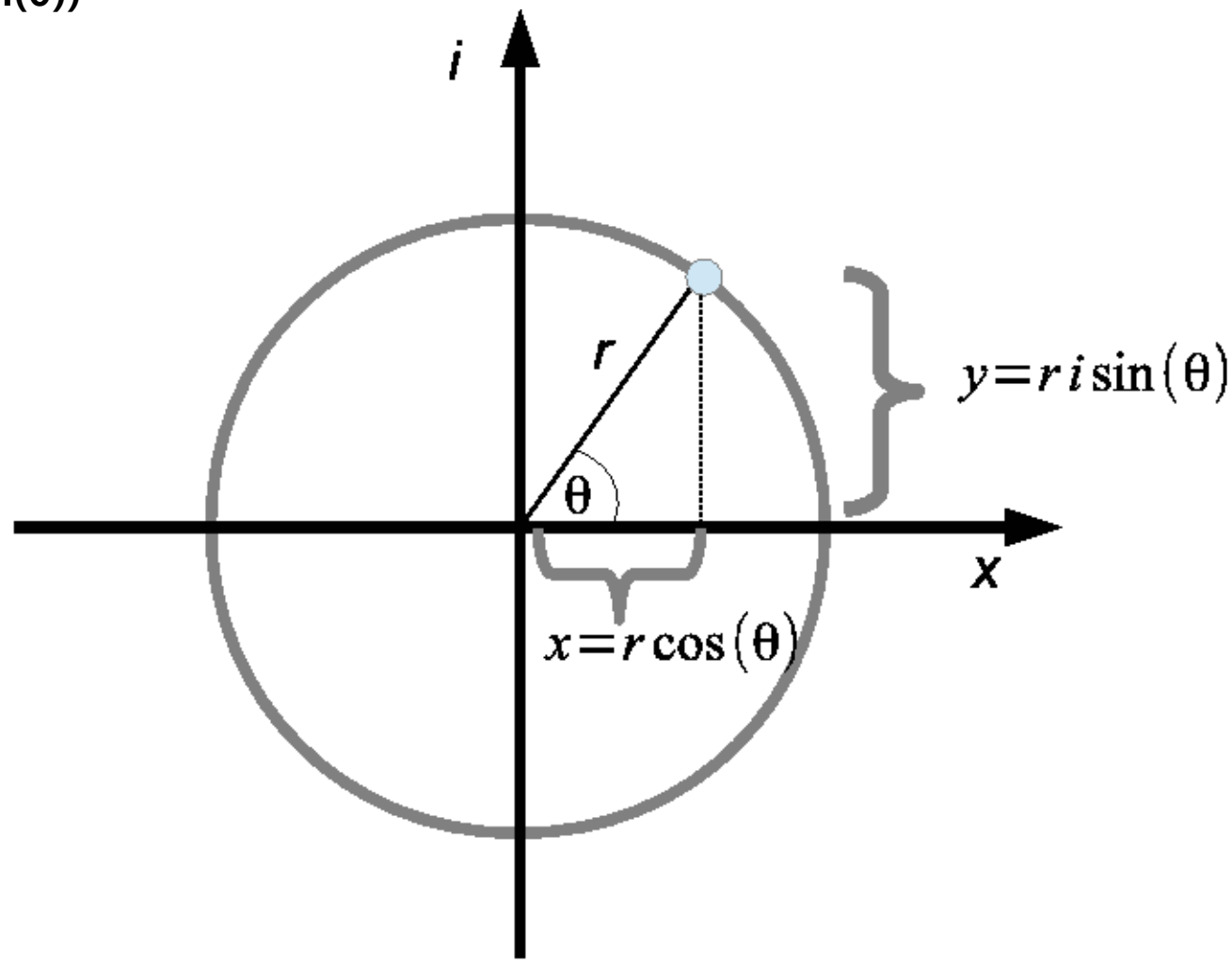
© Maths at Home

www.mathsathome.com

https://mathsathome.com/the-equation-of-a-circle/

삼각함수를 이용한 원의 표현

- (x, y) = (rcos(θ), rsin(θ))



직선의 방정식

- **게임 화면의 한 점 에서 또 다른 한 점 까지 직선으로 이동하는 궤적은 어떻게 만들 수 있을까요?**

첫 번째 방법은 (x_1,y_1), (x_2,y_2)를 지나는 직선의 방정식을 이용하는 것입니다. 두 점을 지나는 직선의 방정식은 다음과 같습니다.

$x_1 \neq x_2$ 일 때 $y-y_1=\frac{y_2-y_1}{x_2-x_1}(x-x_1)$

$x_1=x_2$ 일 때 $x=x_1$ (또는 $x=x_2$)

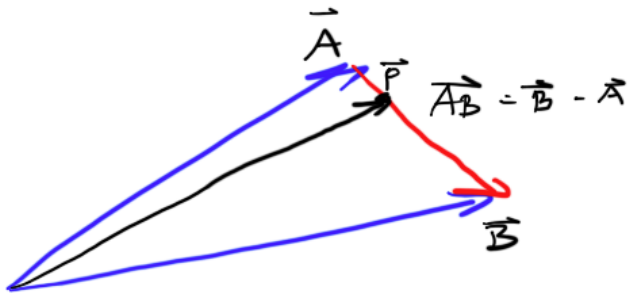
이 수식을 이용하여 프로그래밍을 할 경우,

$x_1 \neq x_2$과 $x_1 = x_2$를 나눠서 코드를 짜야 함

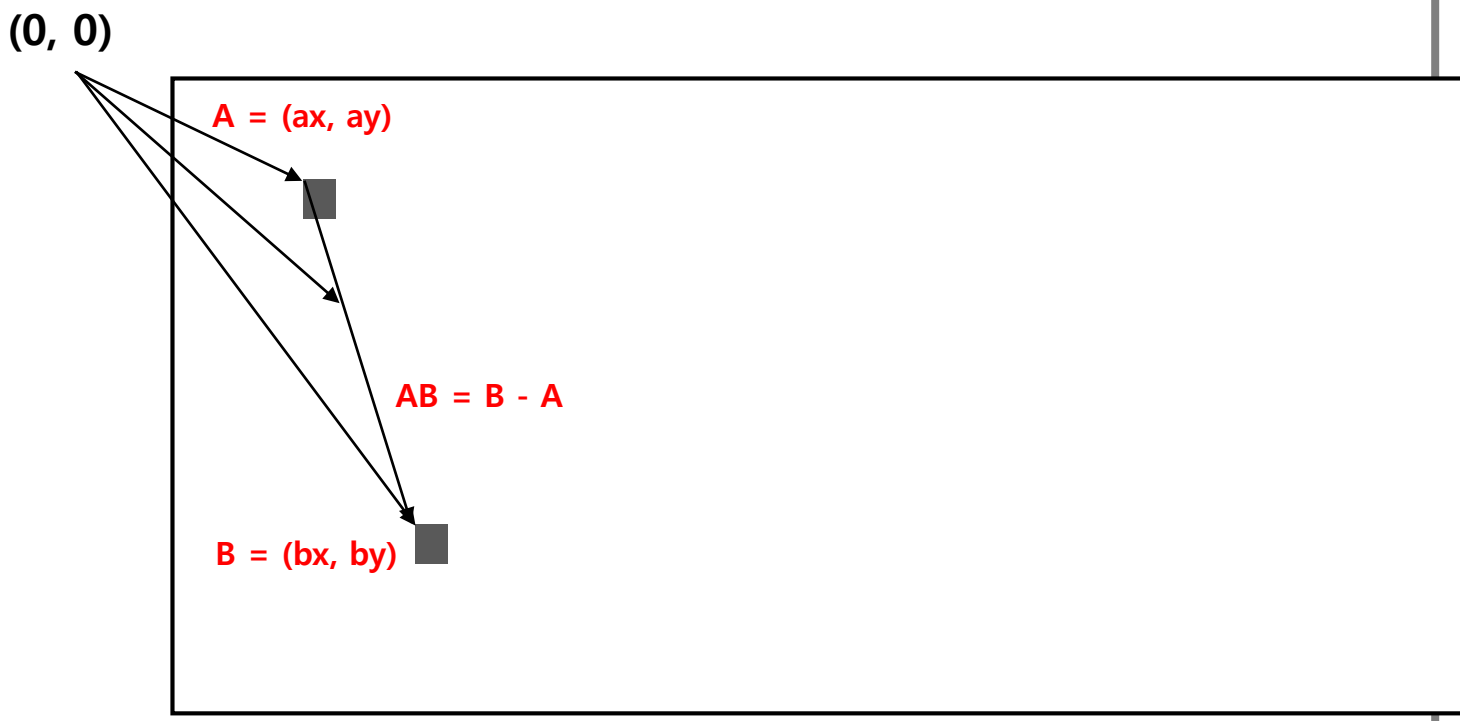
반복 루프 세팅을 위해 x_1과 x_2의 대소비교도 해야 함

두 점(벡터)간 선형 보간 (Linear Interpolation) : C(t) = A*(1-t) + t*B, 0 <= t <=1

- A = (ax, ay), A = (bx, by) 사이 반복 운동을 하기 위해서는
- A에서 B 방향으로 ¼ 떨어진 지점의 벡터
  - A + (B - A) / 4 = 3/4 A + B / 4
- A와 B사이 선분 : C(t) = A*(1-t) + t*B, 0 <= t <=1



https://gamedev.stackexchange.com/questions/18615/how-do-i-linearly-interpolate-between-two-vectors



두 점(벡터)간 선형 보간

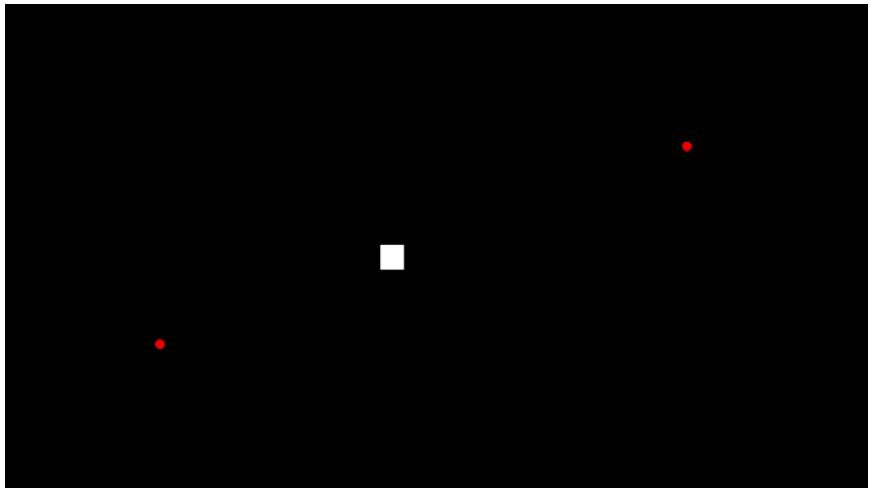
```
using UnityEngine;

public class LerpOneWay : MonoBehaviour
{
    public Transform[] twoPoints;

    private float duration = 2f;
    private Transform square;
    private float ellipsedTime;
    private float moveSpeed = 2f;

    private void Update() {
        ellipsedTime += Time.deltaTime;

        float t = ellipsedTime / moveSpeed;
        if(t >= 1f) {
            ellipsedTime = 0f;
        }
        transform.position = (1f - t) * twoPoints[0].position + t * twoPoints[1].position;
    }
}
```



두 점(벡터)간 선형 보간 (왕복)

```
using UnityEngine;

public class LerpPingPong : MonoBehaviour
{
    public Transform[] twoPoints;

    private float duration = 2f;
    private Transform square;
    private float ellipsedTime;
    private float moveSpeed = 2f;
    private int sign = 1;

    private void Update() {
        ellipsedTime += Time.deltaTime;

        float t = ellipsedTime / moveSpeed;
        if(t >= 1f || t <= 0f) {
            ellipsedTime = 0f;
            sign *= -1;
        }

        t = sign > 0 ? t : (1f - t);
        transform.position = (1f - t) * twoPoints[0].position + t * twoPoints[1].position;
    }
}
```

두 점(벡터)간 선형 + SIN 보간 (왕복)