



# **Programming - Practice 10**

- **Tic-Tac-Toe**



# ▶ Tic-Tac-Toe 게임 만들기

- 컴퓨터와 사람이 번갈아 두는 Tic-Tac-Toe를 만들어 보자
- 필요한 요소
  - 3x3 보드 그리기
  - Tic-tac-toe 승패 판정
  - 사람이 두기 (유효한 입력을 받아 내부 배열 및 보드 업데이트)
  - 컴퓨터가 두기 (유효한 입력을 랜덤하게 선택)

# Modulo (%) 활용

- 자릿수 분리

- 78분을 시간과 분으로 분리:  $(\text{int})78/60 \rightarrow$  시간,  $78\%60 \rightarrow$  분

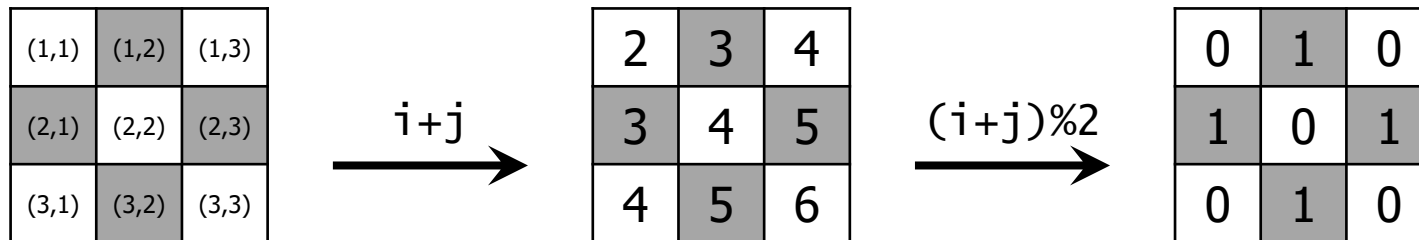
- 랜덤 숫자에서 임의의 범위 숫자 얻기

- $\text{rand()} \% (b - a + 1) + a$

- Checker pattern

- 홀짝 판정:  $(i \% 2)? \text{ odd:even}$

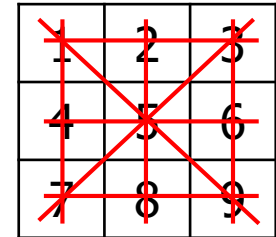
- Checker는 줄에 따라 홀짝 다름:  $((i + j) \% 2)? \text{ odd:even}$



# 소수(prime number)의 활용

## ■ Tic-Tac-Toe 승리 판정

- 대각선 포함하여 연속된 3개가 있으면 승리



1	2	3
4	5	6
7	8	9

## ■ and (&&)를 이용하여 판정

- 가로, 세로, 대각선 각각 연속 세 개가 있는지 판정
- 예) `board[i] == X && board[i+1] == X && board[i+2] == X`

## ■ 소수를 활용한다

- 두 개의 소수가 있을 때 소수의 곱을 구하면 각각이 몇 개 있는지 알 수 있다
- Empty = 1, X = 2, O = 3 으로 정의했을 때 각 줄의 곱을 구하면
  - X 연속 3개: 8, O 연속 3개: 27
- 3개 이외의 것도 판정 가능
  - X 2개: 4, O 2개: 9, X 1개 O 1개: 6

# ▶ Tic-Tac-Toe 절차

- 값을 입력 받아 내부 배열에 저장하고 보드를 업데이트하여 그린다

- 맨 처음 **player**가 먼저 둘 지(1) 나중에 둘 지(2)를 선택

```
### Tic-Tac-Toe ###  
1 : Play first (X)  
2 : Play second (O)  
3 : Exit  
Enter command:
```

- X 또는 O의 두는 자리를 입력 받는다

- 입력 값은 1~9 값으로 /, % 연산으로  
행과 열을 알 수 있다.

- 유효한 값을 얻을 때까지 반복하여 입력

- 유효한 값: 1~9 && 비어 있는 곳

- Player와 Computer가 번갈아 두며 Computer는 유효한 값 중 랜덤으로 둔다

- 연속된 3개가 되면 승리 (대각선 포함). 끝까지 승부가 안 나면 무승부.

- 승패가 결정되면 메시지 출력 후 끝냄

1	2	3
4	5	6
7	8	9

# ▶ Tic-Tac-Toe 인터페이스

## ■ 실행 예

```
### Tic-Tac-Toe ###  
Human: X, Computer: O  


|   |   |   |
|---|---|---|
| X | O | X |
| O | X | O |
| X |   |   |

  
Player1 Wins!!  
Play again ? (Y/N): _
```

# ▶ Challenge problem

## ■ Dumb AI를 만들어 보자

□ 컴퓨터가 두는 간단한 우선 순위를 만든다

1. 내가 이길 수 있는 곳에 둔다 (나 연속 2개)
2. 상대방이 두면 내가 지는 곳에 둔다 (상대방 연속 2개)
3. 5가 비어 있으면 둔다
4. 2, 4, 6, 8 순으로 비어 있으면 둔다
5. 1, 3, 7, 9 순으로 비어 있으면 둔다

1	2	3
4	5	6
7	8	9