

# C++ 프로그래밍 및 실습

## Tic Tac Toe 게임 구현

## 목차

### i. 서론

- 프로젝트 목적 및 배경
- 목표

### ii. 요구사항

- 사용자 요구사항
- 기능 요구사항

### iii. 설계 및 구현

- 기능 별 구현 사항

### iv. 테스트

- 기능 별 테스트 결과
- 최종 테스트 스크린샷

### v. 결과 및 결론

- 프로젝트 결과
- 느낀 점

## 1. 서론

### 1) 프로젝트 목적 및 배경

5주차까지 배운 내용에 대한 실습을 위해 'Tic Tac Toe' 게임의 코드를 작성한다.

### 2) 목표

'Tic Tac Toe' 게임을 프로그램으로 구현한다.

## 2. 요구 사항

### 1) 사용자 요구사항

두 명의 사용자가 번갈아가며 O와 X를 놓는다.

### 2) 기능 요구사항

-누구 차례인지 출력

-좌표 입력 받기

-입력받은 좌표의 유효성 체크

-입력받은 좌표에 현재 유저의 돌 넣기

-현재 보드 판 출력

-빙고 시 승자 출력 후 종료 (가로, 세로, 대각선)

-모든 칸이 찼으면 종료

### 3. 설계 및 구현

#### 1) 기능 별 구현 사항

-요구 사항 별 코드

```
switch (k % 2) {  
    case 0:  
        cout << "첫 번째 유저(X)의 차례입니다. ->  
";  
        currentUser = 'X';  
        break;  
    case 1:  
        cout << "두 번째 유저(O)의 차례입니다. ->  
";  
        currentUser = 'O';  
        break;  
}
```

누구 차례인지 출력하는 코드이다. 서로 번갈아가며 유저의 차례를 출력해준다.

```
cout << "(x, y) 좌표를 입력하세요: ";
```

```
cin >> y >> x;
```

좌표를 입력받는다. x와 y의 입력을 바꾸어 입력했을 때 출력값이 입력한 그대로 나오게 한다.

```
if (x >= numCell || y >= numCell) {
```

```
    cout << x << ", " << y << ": ";
```

```
    cout << "x 와 y 둘 중 하나가 칸을 벗어났니  
다." << endl;
```

```
        continue;
    }
```

```
        if (board[x][y] != ' ') {
            cout << x << ", " << y << ": 이미 돌이 차있습
니다." << endl;
            continue;
        }
```

칸을 벗어나는지 돌이 차있는지 변수 체크한다.

```
board[x][y] = currentUser;
입력받은 좌표에 현재 유저의 돌을 놓는다.
```

```
for (int i = 0; i < numCell; i++) {
    cout << "---|---|---" << endl;
    for (int j = 0; j < numCell; j++) {
        cout << board[i][j] << ' ';
        if (j == numCell - 1) {
            break;
        }
        cout << " |";
    }
    cout << endl;
}
cout << "---|---|---" << endl;
```

현재 보드판을 출력한다.

```

int checked = 0;
    for (int i = 0; i < numCell; i++) {
        for (int j = 0; j < numCell; j++) {
            if (board[i][j] == ' ') {
                checked++;
            }
        }
    }

    if (checked == 0) {
        cout << "모든 칸이 다 찼습니다. 종료합니다."
<< endl;
        break;
    }

```

모든 칸 다 찼는지 체크한다.

```

for (int i = 0; i < numCell; i++) {
    if (board[i][0] == currentUser && board[i][1]
== currentUser && board[i][2] == currentUser) {
        cout << "가로에 모두 돌이 놓였습니다!." <<
endl;
        isWin = true;
    }

```

```

    if (board[0][i] == currentUser && board[1][i]
== currentUser && board[2][i] == currentUser) {
        cout << "세로에 모두 돌이 놓였습니다!." <<

```

```
endl;
```

```
        isWin = true;
```

```
    }
```

```
}
```

```
if (board[0][0] == currentUser && board[1][1] ==  
currentUser && board[2][2] == currentUser) {
```

```
    cout << "왼쪽 위에서 오른쪽 아래 대각선으로  
모두 돌이 놓였습니다!: " << endl;
```

```
    isWin = true;
```

```
}
```

```
if (board[0][2] == currentUser && board[1][1] ==  
currentUser && board[2][0] == currentUser) {
```

```
    cout << "오른쪽 위에서 왼쪽 아래 대각선으로  
모두 돌이 놓였습니다!: " << endl;
```

```
    isWin = true;
```

```
}
```

```
if (isWin == true) {
```

```
    cout << k % 2 + 1 << "번 유저(" <<  
currentUser << ")의 승리입니다!" << endl;
```

```
    cout << "종료합니다" << endl;
```

```
    break;
```

```
}
```

가로, 세로, 대각선으로 돌이 모두 놓였는지 체크한다.

## 4. 테스트

### 1) 기능 별 테스트 결과

#### 스크린샷

#### -누구의 차례인지 출력

```
첫 번째 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요:   
두 번째 유저(o)의 차례입니다 -> (x, y) 좌표를 입력하세요: 
```

#### -좌표 입력 받기

```
첫 번째 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 0  
두 번째 유저(o)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 1
```

#### -입력받은 좌표의 유효성 체크

```
첫 번째 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요: 3 4  
3, 4: x 와 y 둘 중 하나가 칸을 벗어납니다.  
첫 번째 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요:   
첫 번째 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2  
0  
2, 0: 이미 돌이 차있습니다.
```

#### -입력받은 좌표에 현재 유저의 돌 넣기

```
첫 번째 유저(x)의 차례입니다. -> (x, y) 좌표를 입력하세요: 2 0  
---|---|---  
  |  |x  
---|---|---  
  |  |  
---|---|---  
  |  |  
---|---|---
```

#### -현재 보드 판 출력

```
첫 번째 유저(x)의 차례입니다. -> (x, y) 좌표를 입력하세요: 1 1  
---|---|---  
  |  |x  
---|---|---  
  |x |  
---|---|---  
  |  |o  
---|---|---
```



-모든 칸이 찼으면 종료

```
---|---|---
X  |O  |O
---|---|---
O  |X  |X
---|---|---
X  |X  |O
---|---|---
무승부입니다!
```

-빙고 시 승차 출력 후 종료 (가로, 세로, 대각선)

```
---|---|---
   |O  |X
---|---|---
   |X  |
---|---|---
X  |   |O
---|---|---
오른쪽 위에서 왼쪽 아래 대각선으로 모두 돌이 놓였습니다!:
1번 유저(X)의 승리입니다!
종료합니다
```

## 2) 최종 테스트 스크린샷

### 프로그램 전체 동작 스크린샷

```
첫 번째 유저(x)의 차례입니다. -> (x, y) 좌표를 입력하세요: 2 0
---|---|---
|   |x  |
---|---|---
|   |   |
---|---|---
|   |   |
---|---|---
두 번째 유저(o)의 차례입니다. -> (x, y) 좌표를 입력하세요: 2 2
---|---|---
|   |x  |
---|---|---
|   |   |
---|---|---
|   |o  |
---|---|---
첫 번째 유저(x)의 차례입니다. -> (x, y) 좌표를 입력하세요: 1 1
---|---|---
|   |x  |
---|---|---
|x  |   |
---|---|---
|   |o  |
---|---|---
두 번째 유저(o)의 차례입니다. -> (x, y) 좌표를 입력하세요: 1 0
---|---|---
|o  |x  |
---|---|---
|x  |   |
---|---|---
|   |o  |
---|---|---
첫 번째 유저(x)의 차례입니다. -> (x, y) 좌표를 입력하세요: 0 2
---|---|---
|o  |x  |
---|---|---
|x  |   |
---|---|---
x  |   |o
---|---|---
오른쪽 위에서 왼쪽 아래 대각선으로 모두 놓였습니다!:
1번 유저(x)의 승리입니다!
종료합니다
```

## 5. 결과 및 결론

### 1) 프로젝트 결과

- 'Tic Tac Toe' 게임을 만들었다.

### 2) 느낀 점

1~5주차 practice는 할만하다고 생각했고, 그동안 해온 challenge 결과물도 만족스럽게 잘 작동하여 나는 c++에 재능이 있는 줄 알았는데 이번 실습을 하면서 잘못 생각하고 있음을 느꼈다. 게임판이나 입력받고, 출력하는 건 할 수 있는데 bool 함수 등 게임 진행에 필요한 요소들은 어려웠다. 입출력 외에 배열, 리스트, 조건문, 반복문을 더 공부해야겠다.