

1. 서론

1. 프로젝트 목적 및 배경: 4주차까지 배운 내용에 대한 복습검 실습
2. 목표: 요구사항에 대한 Tic Tac Toe 게임 구현

2. 요구사항

1. 사용자 요구사항: 두 명의 사용자가 번갈아가며

O와 X를 놓고 먼저 빙고를 달성한 승자를 가려냄

2. 기능 요구사항: (앞에 있는 내용)

- ① 누구의 차례인지 출력
- ② 좌표 입력 받기
- ③ 입력 받은 좌표 유효성 체크
- ④ 좌표에 O / X 놓기
- ⑤ 현재 보드판 출력
- ⑥ 빙고 시 승자 출력 후 종료
- ⑦ 모든 칸이 찼으면 종료

3. 설계 및 구현

1. 기능 별 구현 사항: (요구사항 별 코드)

- ① 누구의 차례인지 출력

```
// 게임 시작
while (true) {
    // 1. 누가 차례인지 출력
    switch (k % 2) {
        case 0:
            cout << "첫번째 유저(X)의 차례입니다 -> ";
            currentUser = 'X';
            break;
        case 1:
            cout << "두번째 유저(O)의 차례입니다 -> ";
            currentUser = 'O';
            break;
    }
}
```

- ② 좌표 입력 받기, ③ 입력 받은 좌표 유효성 체크

```
// 2. 좌표 입력 받기
cout << "(x, y) 좌표를 입력하세요: ";
cin >> x >> y;

// 3. 입력받은 좌표의 유효성 체크
if (x >= numCell || y >= numCell || board[x][y] != ' ') {
    cout << "잘못된 입력입니다. 다시 입력하세요." << endl;
    continue;
}
```

④ 좌표에 O / X 놓기

```
// 4. 입력받은 좌표에 현재 유저의 돌 추가
board[x][y] = currentUser;
```

⑤ 현재 보드판 출력

```
// 5. 현재 보드 출력
for (int i = 0; i < numCell; i++) {
    cout << "----" << endl;
    for (int j = 0; j < numCell; j++) {
        cout << board[i][j] << " ";
    }
    cout << endl;
}
cout << "----" << endl;
```

⑥ 빙고 시 승자 출력 후 종료

```
// 6. 빙고 체크 (가로, 세로, 대각선)
int winner = 0;

// 가로 체크
for (int i = 0; i < numCell; i++) {
    if (board[x][i] != currentUser) break;
    if (i == numCell - 1) winner = 1;
}

// 세로 체크
for (int i = 0; i < numCell; i++) {
    if (board[i][y] != currentUser) break;
    if (i == numCell - 1) winner = 1;
}

// 대각선 체크 (왼쪽 위에서 오른쪽 아래)
if (x == y) {
    for (int i = 0; i < numCell; i++) {
        if (board[i][i] != currentUser) break;
        if (i == numCell - 1) winner = 1;
    }
}

// 대각선 체크 (오른쪽 위에서 왼쪽 아래)
if (x + y == numCell - 1) {
    for (int i = 0; i < numCell; i++) {
        if (board[i][numCell - 1 - i] != currentUser) break;
        if (i == numCell - 1) winner = 1;
    }
}
```

⑦ 모든 칸이 찼으면 종료

```
// 7. 보드가 꽉 찼는지 확인
int isFull = 1; // 꽉 찼음을 의미하는 변수
for (int i = 0; i < numCell; i++) {
    for (int j = 0; j < numCell; j++) {
        if (board[i][j] == ' ') {
            isFull = 0;
            break;
        }
    }
    if (isFull == 0) break;
}

if (isFull == 1) {
    cout << "보드가 꽉 찼습니다!" << endl;
    break;
}

// 8. 차례 교체
k++;
```

4. 테스트

- (입력에 따라 원하는 결과나 나오는지 확인하는 과정)

1. 기능 별 테스트 결과: (요구사항 별 스크린샷

① 누구의 차례인지 출력

첫번째 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요:

② 좌표 입력 받기 , 좌표에 O / X 놓기현재 보드판 출력

```
첫번째 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요: 00
1
-----
X
-----
-----
-----
-----
```

③ 입력 받은 좌표 유효성 체크

```
두번째 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1
4
잘못된 입력입니다. 다시 입력하세요.
```

④ 빙고 시 승자 출력 후 종료

```
두번째 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2
2
----
O X X
----
X O
----
    O
----
승리자: 0
```

⑤ 모든 칸이 찼으면 종료

```
첫번째 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2
2
----
X O X
----
X O O
----
O X X
----
보드가 꽉 찼습니다!
```

2. 최종 테스트 스크린샷: (프로그램 전체 동작 스크린샷)

```
첫번째 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0
0
----
X
----

----
두번째 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0
1
----
X O
----

----
첫번째 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0
2
----
X O X
----

----
두번째 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1
1
----
X O X
----
    O
----

----
첫번째 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1
0
----
X O X
----
X O
----

----
```

```

첫번째 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1
0
----
X O X
----
X O
----

두번째 유저(o)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2
0
----
X O X
----
X O
----
0
----

첫번째 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2
1
----
X O X
----
X O
----
O X
----

두번째 유저(o)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1
2
----
X O X
----
X O O
----
O X
----

첫번째 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2
2
----
X O X
----
X O O
----
O X X
----
보드가 꽉 찼습니다!

```

5. 결과 및 결론

1. 프로젝트 결과: 승자와 패자가 있는 Tic Tac Toe 게임을 만들었다
2. 느낀 점: 지금까지 배운 내용들로 요구사항에 맞춰 Tic Tac Toe 게임을 만드는 과정에서 생각보다 많은 어려움이 있었으며 지금까지 배운 내용들과 원래 알았다고 생각한 지식들을 다시한번 점검하는 과정이 있어야 한다고 생각이 들었음