

1. 서론

- (1). 프로젝트 목적 및 배경: 4주차까지 배운 내용에 대한 실습을 위해 진행
- (2). 목표: Tic Tac Toe 게임 구현

2. 요구사항

- (1). 사용자 요구사항: 두 명의 사용자가 번갈아 O와 X를 놓기
- (2). 기능 요구사항: 빙고 시 승자 출력 후 종료, 모든 칸이 찼으면 종료

3. 설계 및 구현

- (1). 기능 별 구현 사항:

[빙고(가로, 세로, 대각선)]

```
// 빙고를 체크하는 함수 추가
bool checkWin(char board[3][3], char currentUser) {
    // 가로와 세로 확인
    for (int i = 0; i < 3; i++) {
        if (board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser)
            return true; // 가로 빙고
        if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser)
            return true; // 세로 빙고
    }
    // 대각선 확인
    if (board[0][0] == currentUser && board[1][1] == currentUser && board[2][2] == currentUser)
        return true; // 왼쪽 상단에서 오른쪽 하단 대각선
    if (board[0][2] == currentUser && board[1][1] == currentUser && board[2][0] == currentUser)
        return true; // 오른쪽 상단에서 왼쪽 하단 대각선
    return false;
}
```

<입력/결과>

- currentUser 값, bool함수인 checkWin를 입력
- 가로, 세로, 대각선 빙고 시 bool함수 true 출력

<설명>

- 빙고를 체크하는 함수를 표현하기 위해서 bool함수를 사용
- 빙고시 true, 아닐 시 false

[모든 칸이 다 찼는지 확인]

```
// 모든 칸이 다 찼는지 확인하는 함수 추가
bool BoardFull(char board[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if (board[i][j] == ' ') return false; // 빈칸이 있으면 아직 차지 않음
        }
    }
    return true; // 빈칸이 없으면 true
}
```

<입력/결과>

- bool 함수에 2차원배열 board입력
- 빈칸이 없다면 true 출력

<설명>

- 빈칸이 있는지 없는지를 if문을 통해 확인
- 빈칸이 없다면 bool함수 BoardFull이 true를 나타냄

[안내값 출력]

```
// 빙고 체크 추가
if (checkWin(board, currentUser)) {
    cout << "승리자: " << currentUser << endl;
    break; // 빙고 시 게임 종료
}
```

```
// 모든 칸이 찼는지 확인하는 부분 추가
if (BoardFull(board)) {
    cout << "모든 칸이 다 찼습니다. 무승부입니다!" << endl;
    break; // 모든 칸이 차면 게임 종료
}
```

<입력/결과>

- bool함수 checkWin,BoardFull에 현재 배열 입력
- bool함수 값에 따라서 각각의 안내문 출력

<설명>

- bool함수를 이용하여 각각의 상황이 참일 때만 if문을 출력할 수 있도록 설정함

4. 테스트

1. 기능 별 테스트 결과

(대각선 빙고)

```
첫번째 유저의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 0
---|---|---
X |   |   |
---|---|---

두번째 유저의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 1
---|---|---
X | 0 |   |
---|---|---

첫번째 유저의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 2
---|---|---
X | 0 | X |
---|---|---

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

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

두번째 유저의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 1
---|---|---
X | 0 | X |
O |   |   |
X | 0 |   |
---|---|---

첫번째 유저의 차례입니다 ->(x, y) 좌표를 입력하세요: 1 1
승리자: X
```

(가로/세로 빙고)

```
첫번째 유저의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 0
X |   |   |
---|---|---

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

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

두번째 유저의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 0
X | X |   |
O |   |   |
O |   |   |
---|---|---

첫번째 유저의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 2
승리자: X
```

(칸이 다 찾을때)

```
첫번째 유저의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 2
---|---|---
X | 0 | X
---|---|---
X | X | 
---|---|---
O |  | O
---|---|---
두번째 유저의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 2
---|---|---
X | 0 | X
---|---|---
X | X | O
---|---|---
O |  | O
---|---|---
첫번째 유저의 차례입니다 ->(x, y) 좌표를 입력하세요: 2 1
---|---|---
X | 0 | X
---|---|---
X | X | O
---|---|---
O | X | O
---|---|---
모든 칸이 다 찹습니다. 무승부입니다!
```

2. 최종 테스트 스크린샷

```
첫번째 유저의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 2
---|---|---
  |  | X
---|---|---
  | X | 
---|---|---
  | O | 
---|---|---
두번째 유저의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 0
---|---|---
  |  | X
---|---|---
  | X | 
---|---|---
O | O | 
---|---|---
첫번째 유저의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 0
---|---|---
X |  | X
---|---|---
  | X | 
---|---|---
O | O | 
---|---|---
두번째 유저의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
승리자: O
```

5. 결과 및 결론

1. 프로젝트 결과: Tic Tac Toe 게임을 만들었음
2. 느낀 점: 교수님이 1+1을 알려줬는데 곱하기를 시키셨다.