

1. 서론

프로젝트 목적 및 배경: 4주차까지 배운 if문 for문 while문 등을 활용하여 직접 작업물을 만들어내기 위함

목표: Tic Tac Toe 게임 구현

2. 요구사항

사용자 요구사항: 두 명의 사용자가 (x, y) 위치를 고려하여 번갈아가며 O와 X를 놓는다

기능 요구사항:

- 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]을 구현한 코드입니다.

```
47 while(true) {
48     switch (k % 2) {
49         case 0:
50             cout << "첫번째 유저(X)의 차례입니다 -> ";
51             currentUser = 'X';
52             break;
53         case 1:
54             cout << "두번째 유저(O)의 차례입니다 -> ";
55             currentUser = 'O';
56             break;
57     }
```

② [좌표 입력 받기]를 구현한 코드입니다.

```
59     cout << "(x, y) 좌표를 입력하세요: ";
60     cin >> y >> x; // x, y 좌표를 입력받은대로 반영하기 위하여 이 코드를 수정했습니다.
```

③ [입력 받은 좌표 유효성 체크]를 구현한 코드입니다.

```

62         if (x >= numCell || y >= numCell) {
63             cout << x << ", " << y << ": ";
64             cout << "x와 y 둘 중 하나가 칸을 벗어납니다." << endl;
65             continue;
66         }
67         if (board[x][y] != ' ') {
68             cout << x << ", " << y << ": 이미 돌이 차있습니다." << endl;
69             continue;
70         }
71     }

```

④ [좌표에 O / X 놓기]를 구현한 코드입니다.

```

72     board[x][y] = currentUser;
73 }

```

⑤ [현재 보드판 출력]를 구현한 코드입니다.

```

74     for (int i = 0; i < numCell; i++){
75         cout << "---|---|---" << endl;
76         for (int j = 0; j < numCell; j++){
77             cout << board[i][j];
78             if (j == numCell - 1){
79                 break;
80             }
81             cout << " |";
82         }
83         cout << endl;
84     }
85     cout << "---|---|---" << endl;

```

⑥번 [빙고 시 승자 출력 후 종료] 를 구현한 코드입니다.

```

7 // 승리 조건 체크
8 bool checkWin() {
9     for (int i = 0; i < numCell; i++) {
10         if (board[i][0] == board[i][1] && board[i][1] == board[i][2] && board[i][0] != ' ') // x축이 빙고인 경우
11             return true;
12         if (board[0][i] == board[1][i] && board[1][i] == board[2][i] && board[0][i] != ' ') // y축이 빙고인 경우
13             return true;
14     }
15     if (board[0][0] == board[1][1] && board[1][1] == board[2][2] && board[0][0] != ' ') //우하강 대각선 빙고
16         return true;
17     if (board[0][2] == board[1][1] && board[1][1] == board[2][0] && board[0][2] != ' ') //우상향 대각선 빙고
18         return true;
19     return false;
20 }
21 }

```

Main 함수에 들어가기 전 승리 조건을 체크하기 위해 x축이 빙고인 경우, y축이 빙고인 경우, 대각선이 빙고인 경우를 if문을 통해 확인하였고, 만약 하나라도 해당하는 경우 true를 반환하여 나중에 출력할 승자를 정하는 코드입니다.

```

87         // 승리 체크
88         if (checkWin()) {
89             cout << "Player " << players[turn] << "가 이겼습니다!" << endl;
90             break;
91         }

```

⑦번 [모든 칸이 찼으면 종료]를 구현한 코드입니다.

```

23         // 무승부 검사
24         bool checkDraw() {
25             for (int i = 0; i < numCell; i++) {
26                 for (int j = 0; j < numCell; j++) {
27                     if (board[i][j] == ' ') return false; // 빈 칸이 있으면 아직 게임이 끝나지 않음
28                 }
29             }
30             return true;
31         }
32     }

```

Main 함수에 들어가기 전 무승부인지를 체크하기 위해 구현한 코드입니다. 즉 모든 칸이 찼지만 승부가 정해지지 않았을 경우를 검사하며 true인 경우 이후에 무승부임을 알리기 위해 사용될 코드입니다.

```

93         // 무승부 체크
94         if (checkDraw()) {
95             cout << "무승부 입니다!" << endl;
96             break;
97         }

```

4. 테스트

- (입력에 따라 원하는 결과나 나오는지 확인하는 과정)

1. 기능 별 테스트 결과:

① 누구의 차례인지 출력

```

kimhyeshin@gimhyesin-ui-MacBookAir week5 % cd "/Users/kimhyeshin/Documents/CPP240
/project/week5/"tic_tac_toe
첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요 : 
두 번째 유저 (O)의 차례입니다 -> (x, y) 좌표를 입력하세요 : 

```

② 좌표 입력 받기 , ④ 좌표에 O / X 놓기, ⑤ 현재 보드판 출력

첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요 : 0 0

X		

③ 입력 받은 좌표 유효성 체크

첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요 : 3 3

3, 3: x와 y 둘 중 하나가 칸을 벗어납니다.

첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요 : █

⑥ 빙고 시 승자 출력 후 종료

첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요 : 2 2

X		
0	X	
	0	X

Player X가 이겼습니다 !

⑦ 모든 칸이 찼으면 종료

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

X	0	0
0	X	X
X	X	0

무승부 입니다 !

2. 최종 테스트 스크린샷: (프로그램 전체 동작 스크린샷)

첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요 : 0 0

X		

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

X		
O		

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

X		
O		
X		

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

X	O	
O		
X		

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

X	O	
O		
X	X	

두 번째 유저 (O)의 차례입니다 -> (x, y) 좌표를 입력하세요 : 2 2

X	O	
O		
X	X	O

첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요 : 1 1

X	O	
O	X	
X	X	O

두 번째 유저 (O)의 차례입니다 -> (x, y) 좌표를 입력하세요 : 0 2
2, 0: 이미 둘이 차 있습니다.

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

X	O	O
O	X	
X	X	O

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

X	O	O
O	X	X
X	X	O

무승부 입니다 !

5. 결과 및 결론

1. 프로젝트 결과:

Tic Tac Toe 게임을 만들었음

(+무승부인 경우 무승부라고 알려주는 기능을 추가했다)

2. 느낀 점: 아무래도 교수님 코드를 보고 시작해서 추가로 할 부분만 기존 코드에 끼워 맞추는 식으로 구현하다보니 main함수가 너무 길어지게 되었던 것 같습니다.