

1. 서론

1. 프로젝트 목적 및 배경
: 4주차까지 배운 내용에 대한 실습을 위해 진행.
2. 목표 : TicTac Toe 구현

2. 요구사항

1. 사용자 요구사항 :
 - 1) 누구의 차례인지 출력.
 - 2) 좌표 입력 받기.
 - 3) 입력 받은 좌표 유효성 체크.
 - 4) 좌표에 O / X 놓기.
 - 5) 현재 보드판 출력.
2. 기능 요구사항 :
 - 1) 빙고 시 승자 출력 후 종료.
 - 2) 모든 칸이 찼으면 종료.

3. 설계 및 구현

1. 기능 별 구현 사항 : (요구사항 별 코드)
 - 1) 누구의 차례인지 출력.

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

- k : 놓은 횟수를 셈.
- k % 2 : 2명에서 하기 때문에 k % 2 + 1은 플레이어 번호.

- 2) 좌표 입력 받기.

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

- x : x좌표
- y : y좌표

3) 입력 받은 좌표 유효성 체크.

```
// 3. 입력받은 좌표의 유효성 체크
if (x >= numCell || y >= numCell) {
    cout << x << ", " << y << ": ";
    cout << " x 와 y 둘 중 하나가 칸을 벗어납니다." << endl;
    continue;
}

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

- x좌표와 y좌표의 입력값이 board의 크기보다 클 경우엔 반복문 다시.
- 이미 돌이 놓여있는 경우에는 반복문 다시 실행(continue).

4) 좌표에 O / X 놓기.

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

- 이 돌이 어떤player의 돌인지 board에 표시.

5) 현재 보드판 출력.

```
// 5. 현재 보드판 출력
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;
```

- board판 출력.

6) 빙고 시 승자 출력 후 종료.

```
// 게임이 끝났는지 판단하기 위한 tmp변수
int tmp = 0;

// 빙고가 됐는지 확인하기 위한 for문
for (int i = 0; i < numCell; i++) {
    // 가로가 같은 돌이 한줄로 다 놓였을 경우
    if ((board[i][0] == board[i][1]) && (board[i][0] == board[i][2])) {
        if (board[i][0] != ' '){
            cout << "가로에 모든 돌이 놓였습니다!: " << k * 2 + 1 << "번 유저(" << board[i][0] << ")의 승리입니다.";
            tmp = 1; // 게임을 끝내기 위해서 tmp변수를 1로 초기화.
            break;
        }
    }

    // 세로가 같은 돌이 한줄로 다 놓였을 경우
    else if (board[0][i] == board[1][i] && board[1][i] == board[2][i] && (board[0][i] == board[2][i])) {
        if (board[0][i] != ' '){
            cout << "세로에 모든 돌이 놓였습니다!: " << k * 2 + 1 << "번 유저(" << board[0][i] << ")의 승리입니다.";
            tmp = 1; // 게임을 끝내기 위해서 tmp변수를 1로 초기화.
            break;
        }
    }

    // 대각선을 확인하기 위한 if문
    // 배열의 범위를 벗어나지 않기 위해.
    // 오른쪽 대각선
    if (i == 0) {
        // 왼쪽 위에서 오른쪽 아래 대각선으로 모두 같은 돌로 한줄로 놓였을 경우
        if (board[i][0] == board[i + 1][1] && board[i][0] == board[i + 2][2]) {
            if (board[i][0] != ' '){
                cout << "왼쪽 위에서 오른쪽 아래 대각선으로 모든 돌이 놓였습니다!: " << k * 2 + 1 << "번 유저(" << board[i][0] << ")의 승리입니다.";
                tmp = 1; // 게임을 끝내기 위해서 tmp변수를 1로 초기화.
                break;
            }
        }
    }

    // 왼쪽 대각선
    else if (i == 2) {
        // 오른쪽 위에서 왼쪽 아래 대각선으로 모두 같은 돌로 한줄로 놓였을 경우
        if (board[0][i] == board[1][i - 1] && board[0][i] == board[2][i - 2]) {
            if (board[0][i] != ' '){
                cout << "오른쪽 위에서 왼쪽 아래 대각선으로 모든 돌이 놓였습니다!: " << k * 2 + 1 << "번 유저(" << board[0][i] << ")의 승리입니다.";
                tmp = 1; // 게임을 끝내기 위해서 tmp변수를 1로 초기화.
                break;
            }
        }
    }
}

// tmp가 1이라면 게임이 끝났음을 알리기 위해
if (tmp == 1)
    break;
```

- 빙고 규칙에 의해 가로 or 세로 or 대각선이 같은 player의 돌로 채워지면 그 해당 player 승리 -> 게임의 승리여부 or 게임이 끝났음을 확인하는 변수 tmp를 1로 초기화하여 게임 종료.

7) 모든 칸이 찼으면 종료.

```
// 보드판에 모든 돌이 찼을 경우
if (k == 9) {
    cout << "모든 칸이 다 찼습니다. 종료합니다.";
    break;
}
```

- count변수인 k가 9이면, board에 돌이 다 놓여있으며 게임을 종료.

4. 테스트

1. 기능 별 테스트 결과 : (요구사항 별 스크린 샷)

1) 누구의 차례인지 출력.

1번 유저(x)의 차례입니다 ->

2) 좌표 입력 받기.

(x, y) 좌표를 입력하세요: 0 1

3) 좌표에 O / X 놓기 및 현재 보드판 출력.

```
---|---|---
  |X|
---|---|---
  |
---|---|---
  |
---|---|---
```

3) 입력 받은 좌표 유효성 체크.

1번 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 1
0, 1: 이미 둘이 차 있습니다.

1번 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 3
0, 3: x 와 y 둘 중 하나가 칸을 벗어납니다.

4) 빙고 시 승자 출력 후 종료.

```
---|---|---
0 |X|
---|---|---
0 |X|
---|---|---
  |X|
---|---|---
세로에 모두 둘이 놓였습니다!: 1번 유저(x)의 승리입니다.
```

5) 모든 칸이 찼으면 종료.

```
---|---|---
X |0|X
---|---|---
X |0|0
---|---|---
0 |X|X
---|---|---
모든 칸이 다 찼습니다. 종료합니다.
```

2. 최종 테스트 스크린샷 :

1) 가로

```
---|---|---
X  |X  |X
---|---|---
O  |   |
---|---|---
O  |   |
---|---|---
가로에 모두 돌이 놓였습니다!: 1번 유저(X)의 승리입니다.
```

2) 세로

```
---|---|---
X  |O  |
---|---|---
X  |O  |
---|---|---
   |O  |X
---|---|---
세로에 모두 돌이 놓였습니다!: 2번 유저(O)의 승리입니다.
```

3) 오른쪽 대각선

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

4) 왼쪽 대각선

```
---|---|---
X  |X  |O
---|---|---
   |O  |X
---|---|---
O  |   |
---|---|---
오른쪽 위에서 왼쪽 아래 대각선으로 모두 돌이 놓였습니다!: 2번 유저(O)의 승리입니다.
```

5. 결과 및 결론

1. 프로젝트 결과 : Tic Tac Toe 게임을 만들었음.
2. 느낀 점 : 수업 시간 내에 하는 내용이 너무 차이가 나는 것 같다.
(어느 날은 너무 많고 어느 날은 너무 적고)