

서론

프로젝트 목적 및 배경: 4주차까지 배운 내용에 대한 실습을 위해 진행

목표: Tic Tac Toe 게임 구현

요구사항

사용자 요구사항: 두 명의 사용자가 번갈아 가며 O와 X를 놓기

기능 요구사항: (앞에 있는 내용)

설계 및 구현

```
if (x >= numCell || y >= numCell) {  
    cout << x << ", " << y << ": ";  
    cout << " x와 y 둘 중 하나가 칸을 벗어납니다." << endl;  
    continue;  
}  
if (board[x][y] != ' ') {  
    cout << x << ", " << y << ": 이미 돌이 차있습니다." << endl;  
    continue;  
}
```

입력: x=좌표 x값

Y= 좌표 y값

numCell = 가로/세로 칸 개수

결과:

칸을 놓을 수 없는 이유를 출력

출력 후 while문 초반으로 이동

설명:

사용자가 입력한 좌표가 게임 판을 벗어나는지 if로 체크

사용자가 입력한 좌표에 돌이 이미 있는지 if로 체크

```

while (true) {
    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;
    }

    cout << "(x, y) 좌표를 입력하세요: ";
    cin >> x >> y;
}

```

입력:

X= 좌표 x값

Y= 좌표 y값

numCell = 가로/세로 칸 개수

결과

K%2의 나머지가 0이면 첫번째 문장 출력 아니면 두번째 문장 출력

좌표를 입력하세요를 출력하고 입력 받은 값을 변수에 저장

설명:

switch문을 통해 누구 차례인지 출력함

```

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;
k++;

```

입력: bord[i][j]값

결과 : 입력받은 값과 보드판 모양을 출력

출력 후 다시 k++증감 연산하여 switch 문으로 돌아감

설명 : 보드판 모양의 개수를 제한하기 위해 if로 체크

for문을 통해 문장 출력 제한을 둠

```

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;
}

```

입력: board[i][j]

결과: checked가 0이면 종료 문자 출력 후 멈춤

모든 칸이 공백이 아니면 다른 문장 출력됨

설명: 칸이 공백인지 확인하기 위해 if로 체크

칸이 공백이면 checked++해줌

```
// 7.1. 가로/세로돌 체크하기
for (int i = 0; i < numCell; i++) {
    if (board[i][0] == currentUser && board[i][1] == currentUser) {
        cout << "가로에 모두 돌이 놓였습니다! "; // 가로에 입
        isWin = true; // true로 초기화
    }
    if (board[0][i] == currentUser && board[1][i] == currentUser) {
        cout << "세로에 모두 돌이 놓였습니다! ";
        isWin = true;
    }
}

// 7.2 대각선돌 체크하기
if (board[0][0] == currentUser && board[1][1] == currentUser && board[2][2] == currentUser) {
    cout << "왼쪽 위에서 오른쪽 아래 대각선으로 모두 돌이 놓였습니다! ";
    isWin = true; // 대각선에 입력받은 값에 저장된 값과 동일하면 true로 초기화
}
if (board[0][2] == currentUser && board[1][1] == currentUser && board[0][0] == currentUser) {
    cout << "오른쪽 위에서 왼쪽 아래 대각선으로 모두 돌이 놓였습니다! ";
    isWin = true;
}
if (isWin == true) {
    cout << "k % 2 + 1" << "번 유저(" << currentUser << ")의 승리가 되었습니다." << endl;
    break; // 만약 true이면 이문장이 출력되며 종료된다.
}
```

입력: board[i][j]

결과: 만약 가로 또는 세로 대각선으로 빙고가 만들어 진다면 문장 출력 및 true와 같아 종료문장 출력

설명: 빙고가 만들어졌는지 확인하기위해 변수에 저장된 값과 주어진값이 같은지 if문으로 체크

만약 빙고라면 false가 true로 초기화 되서 if문이 true와 같아지게되어 종료 문장이 출력됨

테스트

-기능 별 테스트 결과:

누구 차례인지 출력

```
1번 유저(X)의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 2
```

```
2번 유저(O)의 차례입니다 ->(x, y) 좌표를 입력하세요:
```

좌표 입력 받기

```
1번 유저(X)의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 2
---|---|---
```

아직 빙고가 되지 않았기에 차례차례 실행

```

2 1번 유저(X)의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 1
---|---|---
   |X  |
---|---|---
   |   |
---|---|---
   |   |
---|---|---
2번 유저(O)의 차례입니다 ->(x, y) 좌표를 입력하세요: 2 1
---|---|---
   |X  |
---|---|---
   |   |
---|---|---
   |O  |
---|---|---
1번 유저(X)의 차례입니다 ->(x, y) 좌표를 입력하세요: 2 2
---|---|---
   |X  |
---|---|---
   |   |
---|---|---
   |O  |X
---|---|---
2번 유저(O)의 차례입니다 ->(x, y) 좌표를 입력하세요: 2 0
---|---|---
   |X  |
---|---|---
   |   |
---|---|---

```

범위에 벗어난 값을 입력하여 다시 실행

```

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

```

모든 칸이 다참, 종료됨

```

1번 유저(x)의 차례입니다
---|---|---
x  |o  |x
---|---|---
x  |o  |x
---|---|---
o  |x  |o
---|---|---
모든 칸이 다 찹습니다.

```

최종 테스트 스크린샷

```

0  |0  |X
---|---|---
1번 유저(X)의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 2
---|---|---
  |X  |X
---|---|---
  |X  |0
---|---|---
0  |0  |X
---|---|---
2번 유저(0)의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 0
---|---|---
0  |X  |X
---|---|---
  |X  |0
---|---|---
0  |0  |X
---|---|---
1번 유저(X)의 차례입니다 ->(x, y) 좌표를 입력하세요: 1 0
---|---|---
0  |X  |X
---|---|---
X  |X  |0
---|---|---
0  |0  |X
---|---|---
모든 칸이 다 찹습니다. 종료합니다.

```

```

---|---|---
1번 유저(X)의 차례입니다 ->(x, y) 좌표를 입력하세요: 1 1
---|---|---
X |   |
---|---|---
0 |X  |
---|---|---
---|---|---
2번 유저(0)의 차례입니다 ->(x, y) 좌표를 입력하세요: 2 0
---|---|---
X |   |
---|---|---
0 |X  |
---|---|---
0 |   |
---|---|---
1번 유저(X)의 차례입니다 ->(x, y) 좌표를 입력하세요: 2 2
---|---|---
X |   |
---|---|---
0 |X  |
---|---|---
0 |   |X
---|---|---
왼쪽 위에서 오른쪽 아래 대각선으로 모두 돌이 놓였습니다! 2번 유저(X)의 승리입니다!
종료합니다

```

```

---|---|---
1번 유저(X)의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 0
---|---|---
X |   |0
---|---|---
X |   |
---|---|---
---|---|---
2번 유저(0)의 차례입니다 ->(x, y) 좌표를 입력하세요: 2 1
---|---|---
X |   |0
---|---|---
X |   |
---|---|---
  |0  |
---|---|---
1번 유저(X)의 차례입니다 ->(x, y) 좌표를 입력하세요: 2 0
---|---|---
X |   |0
---|---|---
X |   |
---|---|---
X |0  |
---|---|---
세로에 모두 돌이 놓였습니다!: 2번 유저(X)의 승리입니다!
종료합니다

```


게 만들어진다는 것을 깨닫고 룰 같이 이보다 더 복잡한 게임은 어떤 언어의 구조로 되어 있을지 새삼 궁금해진다.