

1, 프로젝트 목적 및 배경

C++ 4주차 까지 내용 복습 및 실습

목표 : Tic Tac Toe 게임 구현

2. 요구사항

두 명의 사용자가 O, X들을 놓으며 Tic Tac Toe 게임 진행 // 보드판은 2차원 배열 사용

- 누구의 차례인지 출력
- 좌표 입력받기 (x, y)
- 입력 받은 좌표 유효성 체크
- 좌표에 O, X 놓기
- 현재 보드판 출력
- 빙고 시 승자 출력 후 종료
- 모든 칸이 찼으면 종료

3. 설계 및 구현

1. 누구의 차례인지 출력

입력

k = 플레이어

currntUser = 플레이어 둘 모양

결과

플레이어의 차례를 출력 (X or O)

플레이어 차례에 맞게 둘 입력

```
while (true) {  
    // 1. 플레이어 차례 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 입력받기 (x, y)

결과

x, y 좌표 입력

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

3. 입력 받은 좌표 유효성 체크

입력

x, y 좌표 값

numCell = 게임 판 크기

결과

칸에 놓을 수 없는 이유 출력

출력 후 while문 초반으로 이동

설명

- 사용자가 입력한 좌표가 게임 판을 벗어나는지 if로 체크
- 사용자가 입력한 좌표에 돌이 이미 있는지 if로 체크

```
// 3. 입력받은 좌표의 유효성 체크
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 있습니다." << endl;
    continue;
}
```

4. 좌표에 O, X 놓기

결과

board[x][y] = 보드판 좌표

currentUser = 돌 놓을 유저

설명

좌표에 맞게 돌 놓기 (O or X)

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

5. 현재 보드판 출력

결과

numCell = 보드판 크기

---|---|---, | = 보드판 모양

현재 보드판 출력

```
// 5. 현재 보드 판 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) {
            cout << " |";
        }
    }
    cout << endl;
}
cout << "---|---|---" << endl;
```

6. 빙고 시 승자 출력 후 종료

입력

Board[x][y] = 보드판 좌표

numCell = 보드판 크기(셀)

결과

가로, 세로, 대각선에 빙고가 생기면 프로그램 종료

설명

가로, 세로, 대각선에 빙고가 생기면 bingo 값이 1이 되면서 게임 종료

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

// 가로 체크
for (int i = 0; i < numCell; i++) {
    if (board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser) {
        bingo = 1;
        break;
    }
}

// 세로 체크
for (int i = 0; i < numCell; i++) {
    if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser) {
        bingo = 1;
        break;
    }
}

// 대각선 체크
if (board[0][0] == currentUser && board[1][1] == currentUser && board[2][2] == currentUser) {
    bingo = 1;
}
if (board[0][2] == currentUser && board[1][1] == currentUser && board[2][0] == currentUser) {
    bingo = 1;
}

if (bingo == 1) {
    cout << "플레이어 " << currentUser << " 승리!" << endl;
    break; // 게임 종료
}
```

7. 모든 칸이 찼으면 종료

입력

numCell = 보드판 크기(셀)

Board[x][y] = 보드판 좌표

결과

칸이 꽉 찬 경우 프로그램 종료 메시지 출력, 프로그램 종료

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

if (full == 1) {
    cout << "무승부입니다. 게임이 종료됩니다." << endl;
    break; // 게임 종료
}
```

설명

numCell에 더 이상 놓을 수 있는 곳이 없을 경우 if로 체크 후 프로그램 종료

4. 테스트

1. 누구의 차례인지

첫번째 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요:

2. 좌표 입력 받기

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

3. 입력 받은 좌표 유효성 체크

두번째 유저(O)의 차례입니다 -> (x, y) 좌표를 입력하세요: 4 1
4, 1: x와 y 둘 중 하나가 간을 벗어납니다.

- 좌표에 O, X 놓기
- 현재 보드판 출력
- 빙고 시 승자 출력 후 종료
- 모든 칸이 찼으면 종료

