

보고서 작성(Tic Tac Toe)

황윤규(214968)

1.서론

1-1 프로젝트 목적 및 배경

4주차까지 배운 내용에 대한 실습을 위해 진행합니다.

1-2 목표

Tic Tac Toe 게임 구현

2.요구사항

1-1 사용자 요구사항

두 명의 사용자가 번갈아가며 O와 X를 놓기

1-2 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

2) 제약 조건

보드판은 2차원 배열 사용

3.설계 및 구현

1 기능 별 구현 사항:(요구사항 별 코드)

```
project> g++ tic_tac_toe.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      const int numCell = 3;
6      char board[numCell][numCell]{};
7      int x, y; // 사용자에게 받는 x,y 좌표를 저장할 변수
8
9      //보드판 초기화
10     for( x = 0; x < numCell; x++){
11         for(y = 0; y < numCell; y++){
12             board[x][y] = ' ';
13         }
14     }
15 }
```

2. 변수

const int numCell = 틱택토 칸의 개수

board = 판 내용을 저장

x, y = 사용자가 입력한 x,y좌표를 저장

3. 설명

for문을 통해서 x, y를 정해진 칸의 개수 만큼 판을 초기화 시킴

```

// 게임하는 코드
int k = 0; // 누구 차례인지 체크하기 위한 변수
char currentUser = 'X';
char checkUser;

while(true){
    //1,누구 차례인지 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;
    }
}

```

1. 변수

- k = 플레이어를 구별하기 위한 변수
- currentUser = 현재 플레이어 지정, X를 통해 플레이어 1을 지정한다.
- checkUser = 후에 사용하려고 했으나 사용하지 않은 변수

2. 설명

- k를 나중에 1씩 증가 시킴으로써 플레이어 1과 2를 k%2를 통해 차례를 구분하였습니다.
- k가 0일때는 currentUser에게 X를 대입하여 플레이어 1로 구분, 1일때는 O를 넣어 플레이어 2로 구분하였습니다,

```

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

//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 차있습니다." << endl;
    continue;
}

```

1. 입력

- x = 좌표 x의 값
- y = 좌표 y의 값
- numCell = 가로/세로 칸 개수

2. 결과

- 칸을 놓을 수 없는 이유를 출력
- 출력 후 while문 초반으로 이동

3. 설명

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

```

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

//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){
            break;
        }
        cout << " |";
    }
    cout << endl;
}
cout << "---|---|---" << endl;

```

1. 입력

- x = 좌표 x의 값
- y = 좌표 y의 값

2. 결과

- 입력한 좌표에 현재 플레이어의 문자를 남긴다.
- 이중 for문을 통해 판의 현재 현황을 출력한다.

2. 설명

- if 문을 통해 안쪽 for문이 먼저 종료되어 “ |”문장을 한번 더 출력되지 않도록 조정

```

// 6. 승자 체크
bool winnerFound = false;

// 가로 및 세로 체크
for (int i = 0; i < numCell; i++) {
    // 가로 체크
    // 현재 유저의 값과 board의 값이 모두 같다면 winnerFound를 true로 변환한다.
    if (board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser) {
        winnerFound = true;
    }
    // 세로 체크
    // 현재 유저의 값과 board의 값이 모두 같다면 winnerFound를 true로 변환한다.
    if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser) {
        winnerFound = true;
    }
}

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

```

1. 변수

- bool winnerFound = false값을 넣는다.

2. 결과

- 가로와 세로를 먼저 체크하여 승리조건 달성시 winnerFound를 true로 바꾼다.
- 대각선을 체크하여 승리조건 달성시 winnerFound를 true로 바꾼다.

2. 설명

- for문에서 가로, 세로 한줄 씩 모두 if문을 통해 각 줄을 체크하여 승리 조건인 같은 문자가 3개 연달아 있을 시에 winnerFound를 true로 전환합니다.
- 대각선을 모두 if문을 통해 각 줄을 체크하여 승리 조건인 대각선에 같은 문자가 3개 있을 시에는 winnerFound를 true로 전환합니다.

```

// winnerFound가 true라면 출력하게 합니다.
if (winnerFound) {
    if (currentUser == 'X') {
        cout << "첫번째 유저의 승리입니다." << endl;
    } else {
        cout << "두번째 유저의 승리입니다." << endl;
    }
    break;
}

// 무승부 판별
bool checkDraw = true;

for(int i = 0; i < numCell; i++){
    for(int j = 0; j < numCell; j++){
        if(board[i][j] == ' '){
            checkDraw = false;
        }
    }
}

if(checkDraw){
    cout << "무승부 입니다!" << endl;
    break;
}

```

1. 변수

- bool winnerFound = false값을 넣는다.
- bool checkDraw = true값을 넣는다.

2. 결과

- 승리 시 결과를 출력하고 while문을 종료합니다.
- 모든 칸이 찰 경우에 무승부를 출력합니고 while문을 종료합니다.

2. 설명

- 사용자가 플레이어 1인지 2인지를 판별해 승리자를 출력합니다.
- 2중 for문을 통해 if문으로 하나의 빈칸이라도 있을 시에 checkDraw를 false로 바꿉니다.
- checkDraw가 앞 조건을 넘어 true일 시 무승부를 출력합니다.

4.테스트

1 기능별 테스트 결과

1-1 누구의 차례인지 출력

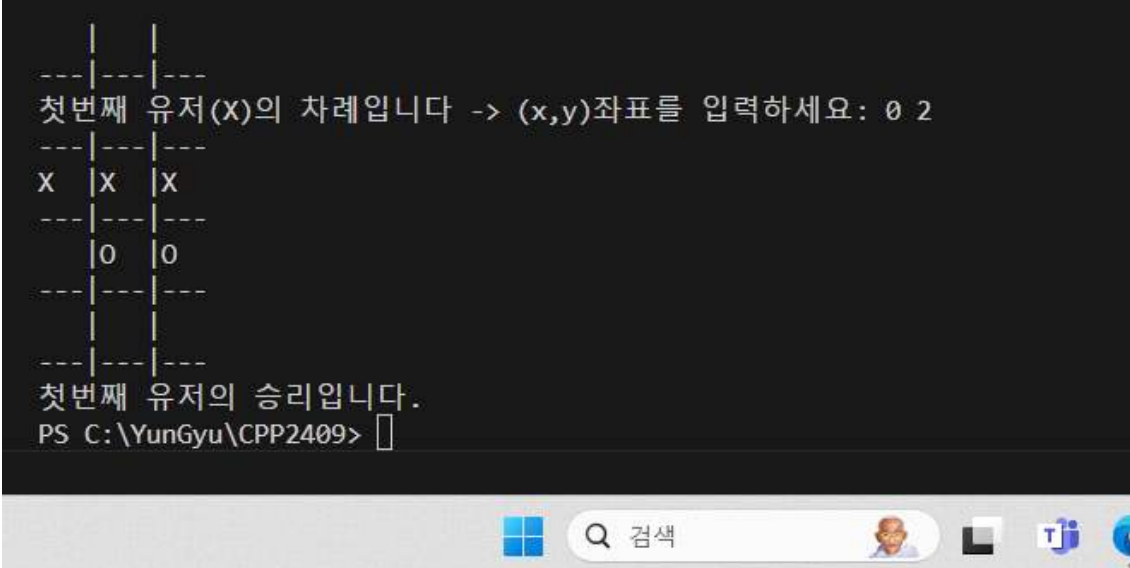
```
두번째 유저(0)의 차례입니다 -> (x,y)좌표를 입력하세요:   
첫번째 유저(x)의 차례입니다 -> (x,y)좌표를 입력하세요:   
1-2 좌표 입력 받기
```

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

2 최종 테스트 스크린샷

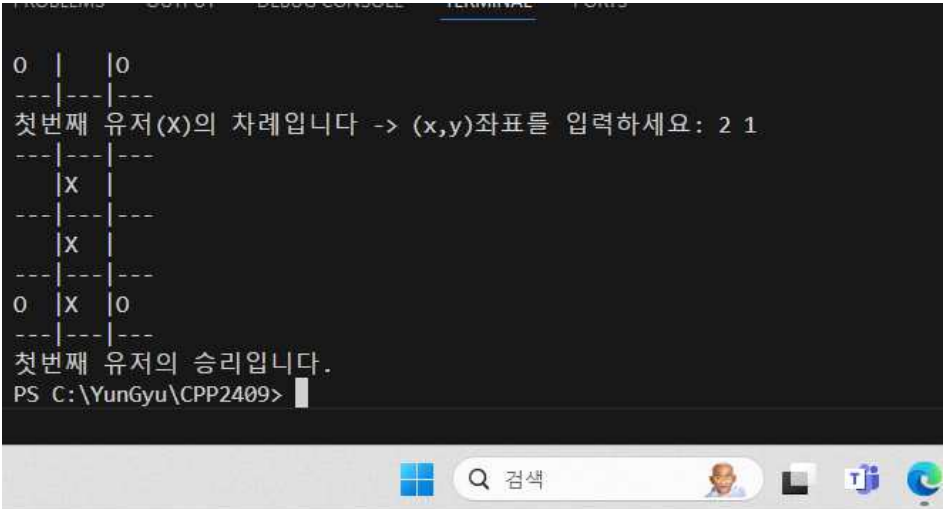
1-1 가로 승리

```
---|---|---  
첫번째 유저(x)의 차례입니다 -> (x,y)좌표를 입력하세요: 0 2  
---|---|---  
X  |X  |X    
---|---|---  
   |0  |0    
---|---|---  
---|---|---  
첫번째 유저의 승리입니다.  
PS C:\YunGyu\CPP2409> 
```



2-2 세로 승리

```
0  |  |0  
---|---|---  
첫번째 유저(x)의 차례입니다 -> (x,y)좌표를 입력하세요: 2 1  
---|---|---  
   |X  |  
---|---|---  
   |X  |  
---|---|---  
0  |X  |0  
---|---|---  
첫번째 유저의 승리입니다.  
PS C:\YunGyu\CPP2409> 
```



2-3 대각선(1) 승리

```

  |  |
---|---|---
두번째 유저(0)의 차례입니다 -> (x,y)좌표를 입력하세요: 2 0
  |  |
X  |X |O
  |  |
X  |O |
  |  |
O  |  |
  |  |
두번째 유저의 승리입니다.
PS C:\YunGyu\CPP2409>

```

2-4 대각선(2) 승리

```

  |  |
---|---|---
첫번째 유저(X)의 차례입니다 -> (x,y)좌표를 입력하세요: 2 2
  |  |
X  |O |
  |  |
O  |X |
  |  |
  |  |X
  |  |
첫번째 유저의 승리입니다.
PS C:\YunGyu\CPP2409>

```

5.결과 및 결론

1. 프로젝트 결과 : Tic Tac Toe 게임을 제작하였습니다.
기본 승리 조건에 맞추어 제작하였고 무승부또한
고려하여 제작하게 되었습니다.
2. 느낀 점 : 처음에 checkuser라는 변수를 통해 플레이어를 체크하고
했습니다. 하지만 이미 currentUser에서 현재 플레이어가 정해져 있어서 쓸모없는

변수라는 사실을 깨달았습니다. 해당 변수를 사용하면 더 쉬운 방법이 있기에
처음부터 다시 짜느라 시간이 더 오래걸리게 되어 아쉬웠습니다. 이를 통해 처음부터
기존에 있던 변수들을 명확하게 알고, 이를 잘 활용하는 것이 중요하다는 사실을
다시 한번 느끼게 되었습니다.