

# tic\_tac\_toe.cpp 보고서

183014 이해건

## 1. 서론

A. 프로젝트 및 배경 :

4주차까지 배운 내용에 대한 실습을 위한 진행

B. 목표 :

Tic tac toe 게임의 구현

## 2. 요구사항

A. 사용자 요구사항 :

두 명의 사용자가 번갈아가며 o 와 x를 놓기

B. 기능 요구사항:

1. 누구의 차례인지 출력
2. 좌표 입력받기
3. 입력받은 좌표 유효성 체크
4. 좌표에 o / x놓기
5. 현재 보드판 출력
6. 빙고시 승자 출력 후 종료
7. 모든 칸이 찼으면 종료

### 3. 설계 및 구현

#### A. 기능 별 구현사항:

##### 1. 누구의 차례인지 출력

```
//1. 몇번 유저의 차례인지 출력
switch (k % 2) { // 나머지를 0 1로
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;
}
```

#### A. 입력

K

#### B. 결과

Case 0 의 경우: "1번유저(X)의 차례입니다->" 출력 후 currentUser 변수를 X로 변경함.

Case 1 의 경우: "2번유저(O)의 차례입니다->" 출력 후 currentUser 변수를 O로 변경함.

#### C. 설명

while문으로 게임이 진행됨에 따라 플레이어 턴이 끝날 때 마다 k++로 차례가 넘어오고 그것을 2로 나눈 값의 나머진 0과 1을 case로 하는 switch 문을 작성, 이를 통해 몇번 플레이어의 턴이고, 그 플레이어의 tictactoe 부호 값까지 변경함.

##### 2. 좌표 입력받기

```
//2.유저에게 좌표입력받기
cout << "(x,y) 좌표를 입력하세요: ";
cin >> x >> y;
```

#### A. 입력

X = 좌표 x값

Y = 좌표 y값

#### B. 결과

유저에게 입력 받은 두 값을 x와 y값으로 함.

#### C. 설명

유저에게 받은 두 값을 x, y로 지정하여 받은 위 두 좌표를 이용하여 추후 유저의 돌을 놓

을 예정

### 3. 입력 받은 좌표 유효성 체크

```
//3.입력받은 좌표의 유효성 체크. 보드판 밖의 값이거나 이미 값이 있을 때
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;
}
```

#### A. 입력

X = 입력 받은 좌표 x값

Y = 입력 받은 좌표 y값

numCell=가로세로 보드판의 크기

#### B. 결과

칸을 놓을 수 없는 이유를 출력하고 다시 while문 처음으로 돌아감.

#### C. 설명

1. 사용자가 입력한 값이 설정한 게임 보드판을 벗어나는지 if문을 통하여 체크 후 그렇다면 안내문구 출력
2. 사용자가 입력한 값이 이미 돌을 입력한 장소인지 if문을 통하여 체크 후 그렇다면 안내문구 출력

### 4. 좌표에 o/x 놓기

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

#### A. 입력

좌표값 x,y

#### B. 결과

배열 board 게임판에 사용자 부호 값 기입

#### C. 설명

사용자의 입력값  $x,y$ 를 통해 2차원 배열 `board[x][y]`의 값으로 찾아서 그 값을 현재 플레이하고 있는 유저의 부호로 변경하여 돌을 놓는 것을 구현

#### 5. 현재 보드판 출력

```
//5.현재 보드판 출력

for (int i = 0; i < numCell; i++) {
    cout << "----|----|----" << endl; //맨 위 보드칸 그리기
    for (int j = 0; j < numCell; j++) {
        cout << board[i][j]; //칸당 하나씩 그리고
        if (j == numCell - 1) {
            break; //마지막은 | 안그리고 탈출
        }
        cout << "  |";
    }
    cout << endl;
}
cout << "----|----|----" << endl; // 맨 밑칸 보드칸 그리기
k++;
```

##### A. 입력

i값 x좌표의 해당 값

j값 y좌표의 해당 값

##### B. 결과

상단과 하단에 보드칸을 그리고, 각 칸별로 `board[i][j]`에 해당되는 공백이 주어지며, 유저의 입력값인  $x,y$  좌표에는 해당 유저의 마크가 표시

##### C. 설명

`board[i][j]`를 for문을 통하여 각 x좌표와 y좌표를 읽어오는 보드판을 생성한다. 위와 아래, 그리고 각 칸이 기입된 뒤 칸을 구분하는 | 표시도 해주면서 보드판을 출력한다.

보드판이 출력된 이후는 `k++`을 통해 다음 턴으로 턴이 넘어갔음을 알려준다.

#### 6. 빙고시 승자 출력 후 종료

```
//7. 승자 체크
bool isWin = false; //승리 확인을 위한 변수

for (int i = 0; i < numCell; i++) { // 각각 X와 Y 좌표의 i가 증가할 때 해당 줄의 모든 값이 현재유저 값이면 승리 확인
    if (board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser) {
        cout << "가로에 모두 돌이 놓였습니다! ";
        isWin = true;
    }
    if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser) {
        cout << "세로에 모두 돌이 놓였습니다! ";
        isWin = true;
    }
}
// 대각선의 모든 값이 현재 유저의 값일 때 유저값과 같으면 승리확인
if (board[0][0] == currentUser && board[1][1] == currentUser && board[2][2] == currentUser) {
    cout << "왼쪽 위에서 오른쪽 아래 대각선으로 모두 돌이 놓였습니다! ";
    isWin = true;
}
if (board[0][2] == currentUser && board[1][1] == currentUser && board[2][0] == currentUser) {
    cout << "오른쪽 위에서 왼쪽 아래 대각선으로 모두 돌이 놓였습니다! ";
    isWin = true;
}
```

A. 입력

x좌표나 y좌표, 좌우 대각선에 통일된 하나의 값이 들어옴

B. 결과

변수 isWin의 값을 true로 변경

C. 설명

가로 세로 대각선 중 하나의 줄이라도 한 개의 부호로만 이루어진 것이 확인되면 승리를 확인하기 위한 변수 isWin의 값을 true로 변경하여 승리임을 확인시킨다.

7. 모든 칸이 찼으면 종료

```
//6.모든 칸이 찼으면 종료
int checked = 0; //체크를 위한 변수를 만들고
for (int i = 0; i < numCell; i++) {
    for (int j = 0; j < numCell; j++) {
        if (board[i][j] == ' ') { // 빈칸이 있을때마다 변수 값을 추가!!
            checked++;
        }
    }
}
if (checked == 0) { //매번 반복으로 돌리면서 마지막에 체크가 더이상 추가되지 않으면
    cout << "모든 칸이 다 찼습니다. 종료합니다.<<endl";
    break; // 끝내고 반복문 탈출
}
```

A. 입력

2차원배열 값(각 좌표)에 ' '공백이 존재하는지 확인

변수 checked의 값이 0과 같은지 확인

B. 결과

존재한다면 변수 checked의 값을 1 증가

변수 checked의 값이 0과 같다면 종료 멘트를 출력하고 반복문을 탈출함

#### C. 설명

각 값에 사전에 초기화했던 값, 즉 공백이 있는지 이중반복문과 if문을 통하여 확인하고, 그것을 통하여checked변수를 1씩 증가한다. 경기가 진행됨에 따라서 빈칸이 없어질 것이며, checked 변수가 0과 같아지는 턴에 모든 수가 끝난것이므로 "모든 칸이 다 찼습니다. 종료합니다. 멘트를 출력하고 게임을 진행하는 while 반복문에서 탈출한다.

## 4. 테스트

### A. 기능 별 테스트 결과

#### 1. 누구의 차례인지 출력

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

#### 2. 좌표 입력받기

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

#### 3. 입력받은 좌표 유효성 체크

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

#### 4. 좌표에 o/x놓기

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

#### 5. 현재 보드판 출력

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

#### 6. 빙고시 승자 출력 후 종료

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

X	0	0
	X	
		X

왼쪽 위에서 오른쪽 아래 대각선으로 모두 돌이 놓였습니다!: 2번 유저(X)의 승리입니다!  
종료합니다

#### 7. 모든 칸이 찼으면 종료

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

0	X	0
0	X	X
X	0	X

모든 칸이 다 찼습니다. 종료합니다.<<endl

#### 8. 최종테스트 스크린샷

```
Microsoft Visual Studio 디버그 콘솔
1번 유저(X)의 차례입니다 -> (x,y) 좌표를 입력하세요: 0 0
X
|
|
|
|
2번 유저(O)의 차례입니다 -> (x,y) 좌표를 입력하세요: 1 1
X
|
| 0
|
|
1번 유저(X)의 차례입니다 -> (x,y) 좌표를 입력하세요: 0 1
X X
| 0
|
|
2번 유저(O)의 차례입니다 -> (x,y) 좌표를 입력하세요: 0 2
X X 0
| 0
|
|
1번 유저(X)의 차례입니다 -> (x,y) 좌표를 입력하세요: 2 0
X X 0
| 0
|
|
2번 유저(O)의 차례입니다 -> (x,y) 좌표를 입력하세요: 1 0
X X 0
0 0
|
|
1번 유저(X)의 차례입니다 -> (x,y) 좌표를 입력하세요: 2 1
X X 0
0 0
X X
|
|
2번 유저(O)의 차례입니다 -> (x,y) 좌표를 입력하세요: 1 2
X X 0
0 0 0
X X
가로에 모두 돌이 놓였습니다!: 1번 유저(O)의 승리입니다!
종료합니다
C:\Users\haegun2\Desktop\Project1\Debug\Project1.exe(프로세스 25900개)이(가) 종료되었습니다(코드: 0개).
```

## 5. 결과 및 결론

### A. 프로젝트 결과

C++을 이용하여 tic tac toe 게임을 구현

### B. 느낀점:

다른 학우들은 상당히 쉽게 잘 하고 금방금방하는데 저만 너무 이해와 적용이 늦나 싶어 상당히 걱정 중입니다. 그래도 이번 시간 tic tac toe 게임을 이것저것 만져보면서 추후 프로젝트 제작을 어떤 것 해야 할 지 대략적인 감이 생겨 좋은 것 같습니다.

예시에 "교수님이 1+1을 알려줬는데 곱하기를 시키셨다."라는 멘트가 참으로 웃기면서도 슬픈 웃픈 상황입니다...!!