

# tic\_tac\_toe 프로젝트 보고서

---



수 강 과 목 : C++ 프로그래밍 기초 및 실습

담 당 교 수 : 김미수

학 과 : 분자생명공학과

학 번 : 222204

이 름 : 은아영

제 출 일 : 2023.10.11

## 목차

1. 서론
  - 1-1. 프로젝트 목적 및 배경
  - 1-2. 목표
2. 요구사항
  - 2-1. 사용자 요구사항
  - 2-2. 기능 요구사항
3. 설계 및 구현
  - 3-1. 기능 별 구현 사항
4. 테스트
  - 4-1. 기능 별 테스트 결과
  - 4-2. 최종 테스트 스크린샷
5. 결과 및 결론
  - 5-1. 프로젝트 결과
  - 5-2. 느낀 점

## 1. 서론

### 1-1. 프로젝트 목적 및 배경

4주차까지 배운 내용에 대한 실습을 위해 진행되었다.

### 1-2. 목표

Tic Tac Toe 게임을 구현하는 것이 프로젝트의 목표이다.

## 2. 요구사항

### 2-1. 사용자 요구사항

두명의 사용자가 번갈아 가며 X와 Y를 놓기

### 2-2. 기능 요구사항

- a) 누구의 차례인지 출력하기
- b) 좌표 입력 받기
- c) 입력 받은 좌표 유효성 체크하기
- d) 좌표에 X, Y 놓기
- e) 현재 보드판 출력하기
- f) 빙고 시 승자 출력 후 종료하기
- g) 모든 칸이 찼으면 종료하기

### 3. 설계 및 구현

#### 3-1. 기능 별 구현 사항

##### a) 누구의 차례인지 출력하기

```
16 //게임하는 코드
17 int k = 0; //누구 차례인지 체크하기 위한 변수
18 char currentUser = 'X'; //현재 유저의 돌을 저장하기 위한 변수 선언 및 초기화
19 while (true) {
20     //1. 누구의 차례인지 출력
21     switch (k % 2) {
22     case 0: //k/2의 나머지가 0인 경우 (k가 짝수인 경우)
23         cout << k % 2 + 1 << "번 유저 (X)의 차례입니다. -> "; //1번 유저 차례 출력
24         currentUser = 'X'; //1번 유저의 기호 저장
25         break;
26     case 1: //k/2의 나머지가 1인 경우 (k가 홀수인 경우)
27         cout << k % 2 + 1 << "번 유저 (Y)의 차례입니다. -> "; //2번 유저 차례 출력
28         currentUser = 'Y'; //2번 유저의 기호 저장
29         break;
30     }
```

while문

입력 : true

출력 : currentUser의 기호(X 또는 Y)

Switch문을 이용해서 k가 홀수인 경우 1번 유저의 차례를 안내하는 문장을 출력하고 k가 짝수인 경우에는 2번 유저의 차례를 안내하는 문장을 출력한다. 또한 1번 유저의 차례인 경우 변수 currentUser에 X를 2번 유저의 차례인 경우 Y를 저장한다.

##### b) 좌표 입력 받기

```
32 //2. 좌표 입력받기
33 cout << "(x, y)좌표를 입력하세요: ";
34 cin >> x >> y; //입력받은 좌표를 x, y에 차례대로 저장
```

cout을 통해 안내 문구를 출력한 후 입력 받은 두 수를 변수 x, y에 차례대로 저장한다.

c) 입력 받은 좌표 유효성 체크하기

```
36 //3. 입력받은 좌표의 유효성 체크
37 if (x >= numCell || y >= numCell) { //x 또는 y가 numCell의 범위를 벗어난다면(칸에서 벗어난다면) 문구 출력
38     cout << x << ", " << y << " ";
39     cout << "x와 y 둘 중 하나가 칸을 벗어납니다." << endl;
40     continue;
41 }
42 if (board[x][y] != ' ') { //사용자에게 입력받은 좌표에 이미 값이 차있을 경우 문구 출력
43     cout << x << ", " << y << " : 이미 돌이 차있습니다." << endl;
44     continue;
45 }
```

If문1

입력 : 변수 x, y의 값

출력 : 없음

If문2

입력 : board[x][y] 배열의 값

출력 : 없음

입력 받은 x, y 값이 유효하지 않은 경우는 x, y의 값이 numCell의 범위를 벗어나거나 입력 받은 좌표에 이미 값이 차 있는 경우이다.

첫 번째 if문을 통해 x 또는 y의 값이 numCell의 범위를 넘어서는 경우 경고 문구를 출력하고 좌표를 다시 입력받는다.

두 번째 if문을 통해 입력 받은 좌표의 값이 공백이 아니라면 이미 값이 차 있다는 경고 문구를 출력하고 다시 입력받는다.

d) 좌표에 X, Y 놓기

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

사용자로부터 입력받은 좌표에 currentUser에 저장된 사용자 기호를 저장한다.

e) 현재 보드판 출력하기

```
50 //5. 현재 보드판 출력
51 for (int i = 0; i < numCell; i++) {
52     cout << "---|---|---" << endl;
53     for (int j = 0; j < numCell; j++) {
54         cout << board[i][j];           //배열에 저장된 값(유저기호 또는 공백) 출력
55         if (j == numCell - 1) {       //j좌표가 배열 벗어나면 반복문 종료
56             break;
57         }
58         cout << " |";                 //칸 나누는 기호
59     }
60     cout << endl;
61 }
62 cout << "---|---|---" << endl;
```

입력 : numCell

출력 : k

cout으로 첫번째 줄을 출력한다. 그 아래 각 좌표의 값과 공백, 칸 나누는 기호를 출력하여 보드판의 모양을 잡는다. j가 numCell-1일 때 for문을 중단하고 마지막 줄을 출력한다.

f) 빙고 시 승자 출력 후 종료하기

```
//7. 승자 체크하기
bool iswin = false;
//7.1. 가로/세로들 체크하기
for (int i = 0; i < numCell; i++) {
    if (board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser) { //가로들 체크
        cout << "가로에 둘이 모두 놓였습니다! ";
        iswin = true;
    }
    if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser) { //세로들 체크
        cout << "세로에 모두 둘이 놓였습니다! ";
        iswin = true;
    }
}
//7.2. 대각선들 체크하기
if (board[0][0] == currentUser && board[1][1] == currentUser && board[2][2] == currentUser) { //왼쪽 위에서 오른쪽 아래 대각선 체크
    cout << "왼쪽 위에서 오른쪽 아래 대각선으로 모두 둘이 놓였습니다! ";
    iswin = true;
}
if (board[0][2] == currentUser && board[1][1] == currentUser && board[2][0] == currentUser) { //오른쪽 위에서 왼쪽 아래 대각선 체크
    cout << "오른쪽 위에서 왼쪽 아래 대각선으로 모두 둘이 놓였습니다! ";
    iswin = true;
}

if (iswin == true) {
    cout << k * 2 + 1 << "번 유저(" << currentUser << ")의 승리입니다!" << endl; //승자 출력하기
    cout << "종료합니다" << endl;
    break;
}
k++;
```

가로/세로들 체크하기

If문

입력 : board[x][y] 배열의 값

출력 : iswin = true 또는 iswin = false

If문을 이용하여 가로, 세로 3개의 기호가 현재 유저의 기호와 일치하는지 확인한 후 일치하면 `iswin = true`값을 일치하지 않으면 `iswin=false`값을 출력한다.

대각선돌 체크하기

If문

입력 : `board[x][y]` 배열의 값

출력 : `iswin = true` 또는 `iswin = false`

If문을 이용하여 왼쪽 위에서 오른쪽 아래 대각선 3개의 기호 또는 오른쪽 위에서 왼쪽 아래 대각선 3개의 기호가 일치하면 `iswin = true` 일치하지 않으면 `iswin = false`값을 출력한다.

승자 출력하기

If문

입력 : `iswin`

출력 : 없음

변수 `iswin`에 저장된 값이 `true`일 경우 현재 유저를 승자로 출력하고 프로그램을 종료한다.

K에 `k+1` 값을 저장하여 다음 유저의 차례로 넘긴다.

g) 모든 칸이 찾으면 종료하기

```
65 //6. 모든 칸 다 찾는지 체크하기
66 int checked = 0; //변수 선언, 초기화
67 for (int i = 0; i < numCell; i++) {
68     for (int j = 0; j < numCell; j++) {
69         if (board[i][j] == ' ') {
70             checked++;
71         }
72     }
73 }
74 if (checked == 0) {
75     cout << "모든 칸이 다 찾습니다. 종료합니다." << endl;
76     break;
77 }
```

If문

입력 : numCell, board 배열의 값

출력 : 없음

Board 배열의 칸이 공백일 경우 변수 checked에 checked + 1을 저장하고 만약 checked의 값이 0이라면 모든 칸이 찾음을 알리는 문구를 출력한 후 프로그램을 종료한다.



#### 4. 테스트

##### 4-1. 기능 별 테스트 결과

###### a) 누구의 차례인지 출력하기

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

###### b) 좌표 입력받기

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

###### c) 현재 보드판 출력하기

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

###### d) 빙고 시 승자 출력 후 종료하기

###### ㄱ. 가로줄을 완성하여 승리한 경우

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

ㄴ. 세로줄을 완성하여 승리한 경우

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

ㄷ. 대각선 줄을 완성하여 승리한 경우

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

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

e. 모든 칸이 다 찼지만 승자가 결정되지 않은 경우

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

## 5. 결과 및 결론

### 5-1. 프로젝트 결과

4주차까지 학습한 내용을 활용하여 차례 출력 기능, 좌표 입력 받기, 입력 받은 좌표 유효성 확인, 좌표에 사용자 기호 저장, 현재 보드판 출력, 빙고시 승자 출력 후 종료, 모든 칸이 찬 경우 프로그램 종료 등의 기능을 추가해 Tic Tac Toe 게임을 만들었다.

### 5-2. 느낀 점

C++이라는 프로그래밍 언어를 처음 접하는 입장에서 조금 난이도가 있다고 느껴졌지만 보고서 제출 기한이 넉넉하고 정답을 제공받아 과제를 완료하는데에는 무리가 없었다. 복습의 중요성을 다시 한번 느낄 수 있었다.