

# Tic\_Tac\_Toe 보고서

203566 서현호

## 1, 프로젝트 목표 및 배경

프로젝트 배경 : 1~4주차 수업 내용에 대한 실습 진행

프로젝트 목표 : Tic Tac Toe 게임 구현

## 2. 프로젝트 요구사항

### 2.1 사용자 요구사항 :

두명의 사용자가 번갈아 가며 O와 X를 놓기

### 2.2 기능 요구사항 :

1. 누구의 차례인지 출력
2. 좌표 입력받기
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

### 3. 설계 및 구현

#### 3.1 기능 별 구현 사항

##### 1. 누구의 차례인지 출력

```
14 }
15 //게임하는 코드
16 int k = 0; // 누구 차례인지 체크하기 위한 변수
17 char currentUser = 'X'; //현재 유저의 돌을 저장하기 위한 문자 변수
18 while (true) {
19     //1. 누구차례인지 출력
20     switch (k % 2) {
21         case 0:
22             cout << k % 2 + 1 << "번 유저 (O)의 차례입니다 ->";
23             currentUser = 'O';
24             break;
25         case 1:
26             cout << k % 2 + 1 << "번 유저 (X)의 차례입니다 ->";
27             currentUser = 'X';
28             break;
29     }
30 }
31 }
```

K라는 차례체크를 위한 변수를 선언한 후 switch문을 사용하여 K를 2로 나누어 나머지가 0 이 될 때와 1이 될 때 즉 홀수 번째와 짝수 번째 순서를 정해준다.

##### 2. 좌표 입력 받기

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

Cout을 통하여 화면에 따옴표 안 문자를 출력하고 cin을 통해서 좌표를 입력받은 후 x와 y에 저장한다

##### 3. 입력 받은 좌표의 유효성 체크

```
36 //3. 입력받은 좌표의 유효성 체크
37 //정해진 칸 이외에 칸을 입력한 경우
38 if (x >= numCell || y >= numCell) {
39     cout << x << ", " << y << ": ";
40     cout << "x와 y 둘 중 하나가 칸을 벗어납니다." << endl;
41     continue;
42 }
43 //둘이 차 있는 경우
44 if (board[x][y] != ' ') {
45     cout << x << ", " << y << ":이미 돌이 차있습니다." << endl;
46     continue;
47 }
48 }
```

if 문을 통해서 만약 x가 numCell보다 크거나 y가 numCell보다 작은경우 둘 중 하나라도 해당된다면 "x와y 둘 중 하나가 칸을 벗어납니다"를 출력한다 또한 if문을 통해 배열 board에 있는 칸이 빈칸이 아니라면 " 이미 돌이 차있습니다" 를 출력한다.

##### 4. 입력 받은 좌표에 현재 유저의 돌 놓기

```
cout << x << ", " << y << ":이미 돌이 차있
continue;
}

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

위에서 선언한 돌을 저장하기 위한 문자 변수 currentUser를 배열 board에 대입하여 유저의 돌을 놓는다

## 5. 현재 보드판 출력하기

```
52 //5. 현재 보드판 출력
53 for (int i = 0; i < numCell; i++) {
54     cout << "----|----|----" << endl;
55     for (int j = 0; j < numCell; j++) {
56         cout << board[i][j];
57         if (j == numCell - 1) {
58             break;
59         }
60         cout << " |";
61     }
62     cout << endl;
63 }
64 cout << "----|----|----" << endl;
65 k++;
```

For 문을 사용하였고 변수 i를 0으로 초기화한 후 만약 i가 numCell보다 작다면 아래에 문장을 수행하고 그 안에 있는 for문으로 넘어가서 for문을 수행한다.

안에 있는 for문은 j를 선언 후 j가 numcell 보다 작다면 아래에 문장을 수행하고 아래에 문장에 if 문이 포함되어 있으므로 if문도 수행하게 된다. If 문은 j가 numCell에서 1을 뺀 것과 같다면 l을 출력하지 않는다.

If 문까지 수행 후 안에 있는 for문을 마쳐 수행하고 다시 밖에 있는 for문의 i가 하나 증가된 상태로 다시 문장을 수행하면서 반복하여 보드판을 출력하게 된다.

## 6. 빙고 시 승자 출력 후 종료

```
//6. 빙고 시 승자 출력 후 종료
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;
}
```

Checked라는 변수를 선언하고 0으로 초기화 시키면서 시작된다. 밖에 있는 for문을 먼저 수행하게 된다. i를 선언하고 numCell보다 작으면 아래에 문장을 수행하는데 수행해야 할 문장에 for문이 포함되어 있다. 안에 있는 for문을 보면 j라는 변수를 선언하고 j가 numCell보다 작다면 안에있는 if 문을 수행하고 안에 있는 if 문은 board[i][j]가 빈칸이라면 checked를 1 증가시킨다 그 후에 안에 있는 for문의 j가 1 증가되고 마쳐다 수행한 후에 밖에 있는 i가 1 증가된 후 다시 for문이 수행된다. 맨 아래에 있는 if 문을 통해서 만약 checked가 0 이 된다면 "모든 칸이 다 찼습니다"를 출력하고 종료한다,

## 7. 승자 체크하기

```
80 //7. 승자 체크하기
81 bool isWin = false;
82 //7.1 가로/세로줄 채움체크
83 for (int i = 0; i < numDel; i++) {
84     if (board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser) {
85         cout << "가로에 모두 놓였습니다!" << endl;
86         isWin = true;
87     }
88     if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser) {
89         cout << "세로에 모두 놓였습니다!" << endl;
90         isWin = true;
91     }
92 }
93 //7.2 대각선들 체크하기
94 if (board[0][0] == currentUser && board[1][1] == currentUser && board[2][2] == currentUser) {
95     cout << "왼쪽 위에서 오른쪽 아래 대각선으로 모두 놓였습니다!" << endl;
96     isWin = true;
97 }
98 if (board[0][2] == currentUser && board[1][1] == currentUser && board[2][0] == currentUser) {
99     cout << "오른쪽 위에서 왼쪽 아래 대각선으로 모두 놓였습니다!" << endl;
100    isWin = true;
101 }
102 if (isWin == true) {
103     cout << k * 2 + 1 << "번 유저(" << currentUser << ") 의 승리입니다!" << endl;
104     cout << "종료합니다" << endl;
105     break;
106 }
```

Bool 타입 isWin 변수를 선언하고 false로 초기화 시킨다. For문과 if문을 통해서 승자를 체크한다. 먼저 for문에 i라는 변수를 선언하고 조건식과 비교한 후 조건에 충족 되면 아래에 if문을 수행한다. If문은 board[i][0]이 currentUser와 같으면서 board[i][1]과 같고 board[i][2]와도 같다면 "가로에 모두 놓였습니다"를 출력하고 isWin을 true로 초기화한다. 아래에 있는 if문도 마찬가지로 조건식에 해당한다면 "세로에 모두 놓였습니다"를 출력하고 isWin을 true로 초기화한다. 아래 대각선 둘 체크하기도 마찬가지로 수행 한 후 마지막 if문을 보면 만약 isWin이 true와 같다면 아래에 문장을 출력하고 break문을 통해 종료한다.

## 4. 테스트

### 4.1 기능 별 테스트결과

#### 1. 누구의 차례인지 출력

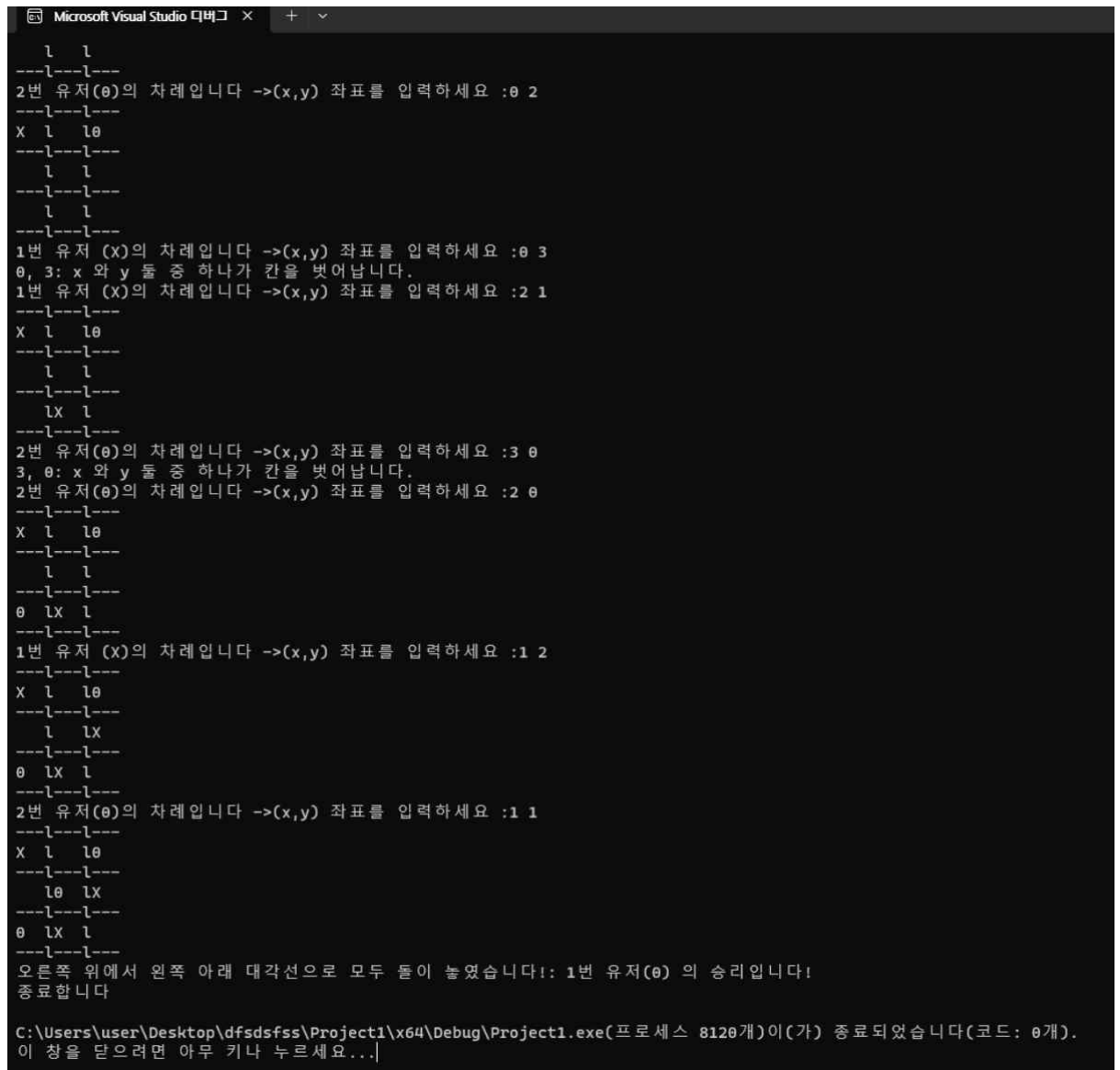
```
1번 유저 (X)의 차례입니다 ->(x,y) 좌표를 입력하세요 :|
```

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

#### 2. 좌표 입력받기

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

### 3. 최종 테스트 스크린 샷



```
Microsoft Visual Studio 디버그 x + v

  l  l
---l---l---
2번 유저(0)의 차례입니다 ->(x,y) 좌표를 입력하세요 :0 2
---l---l---
X l  l0
---l---l---
  l  l
---l---l---
  l  l
---l---l---
1번 유저 (X)의 차례입니다 ->(x,y) 좌표를 입력하세요 :0 3
0, 3: x 와 y 둘 중 하나가 칸을 벗어납니다.
1번 유저 (X)의 차례입니다 ->(x,y) 좌표를 입력하세요 :2 1
---l---l---
X l  l0
---l---l---
  l  l
---l---l---
 lX l
---l---l---
2번 유저(0)의 차례입니다 ->(x,y) 좌표를 입력하세요 :3 0
3, 0: x 와 y 둘 중 하나가 칸을 벗어납니다.
2번 유저(0)의 차례입니다 ->(x,y) 좌표를 입력하세요 :2 0
---l---l---
X l  l0
---l---l---
  l  l
---l---l---
0 lX l
---l---l---
1번 유저 (X)의 차례입니다 ->(x,y) 좌표를 입력하세요 :1 2
---l---l---
X l  l0
---l---l---
  l  lX
---l---l---
0 lX l
---l---l---
2번 유저(0)의 차례입니다 ->(x,y) 좌표를 입력하세요 :1 1
---l---l---
X l  l0
---l---l---
 l0 lX
---l---l---
0 lX l
---l---l---
오른쪽 위에서 왼쪽 아래 대각선으로 모두 둘이 놓였습니다!: 1번 유저(0)의 승리입니다!
종료합니다

C:\Users\user\Desktop\dfsdssfss\Project1\x64\Debug\Project1.exe(프로세스 8120개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

### 5. 프로젝트 결과 및 결론

프로젝트 결과 : Tic Tac Toe 게임을 만들었음

프로젝트 후 느낀 점 : 호기심에 선택한 부전공이지만 현재 전공공부보다 더 많은 공부량이 요구되는 것을 깨닫고 열심히 공부 해야 할 것을 느꼈습니다. 하지만 프로그래밍이 신기하고 재미있어서 견딜 수 있습니다. 많이 부족한 질문에도 항상 열심히 알려주시는 교수님 감동입니다... 더 노력하겠습니다!