# Tic-tac-toe 게임 구현 보고서

223445 빅데이터융합학과 김채흔

## 1. 서론

### 1-1. 프로젝트 목적 및 배경

- C++의 기본 구조, 자료형, 조건문, 반복문, 2차원 배열 등 4주차 동안 배운 것들을 종합하여 활용하기 위한 목적
- 배운 것들을 실습해보며 C++ 프로그램 구현에 익숙해지기 위한 목적

### 1-2. 목표

- Tic-tac-toe 게임을 C++를 이용해 구현

## 2. 요구 사항

### 2-1. 사용자 요구사항

- 두 명의 사용자가 번갈아가며 자신의 차례 일 때 O와 X를 놓기

### 2-2. 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) 모든 칸이 찼으면 종료

(7) 빙고 시 승자 출력 후 종료

## 3. 설계 및 구현

3-1. 누구의 차례인지 출력

```
//1.누구 차례인지 출력
switch (k % 2) {
case 0:
    cout << k % 2 + 1 << "번 유저(X)의 차례입니다";
    currentUser = 'X';
    break;              //첫번째 유저 차례 출력
case 1:
    cout << k % 2 + 1 << "번 유저(0)의 차례입니다";
    currentUser = '0';
    break;  //두번째 유저 차례 출력
```

- 입력
  - K = 누구 차례 인지 알기 위한 변수
  - currentUser = 현재 유저가 놓는 돌(X/O)
- 결과
  - 놓을 차례인 유저 출력
- 설명
  - 몇 번 유저의 차례인지 k를 이용해 출력

  (k는 처음에 0으로 초기화, 코드 마지막 부분에서 1을 더해 유저가 번갈아가며 나올 수 있도록 함)

3-2. 좌표 입력 받기

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

- 입력
  - x = x좌표 x 값
  - y = y좌표 y 값
- 결과
  - X와 y 사용자로부터 입력 받기
- 설명
  - 사용자로부터 x좌표와 y좌표를 입력 받아 저장

3-3. 입력 받은 좌표 유효성 체크

```
//3. 입력받은 좌표의 유효성 체크
//3-1. 칸을 벗어나는 경우 메시지 출력
if (x >= numCell or y >= numCell) {
    cout << x << ", " << y << ": ";
    cout << "x와 y 둘 중 하나가 칸을 벗어납니다." << endl;
    continue;
}
//3-2.돌이 모두 차있는 경우 메시지 출력
if (board[x][y] != ' ') {
    cout << x << ", " << y << ":이미 돌이 차있습니다." << endl;
    continue;
}
```

- 입력
  - x = x좌표 x 값
  - y = y좌표 y 값
  - numCell =가로와 세로 칸 개수
  - board = 게임 보드판 (배열)
- 결과
  - 칸을 벗어 났거나 이미 돌이 있는 경우, 돌을 놓을 수 없는 이유 출력
- 설명
  - 유저가 입력한 좌표가 보드판을 벗어나는 지 if문을 활용하여 체크
  - 유저가 입력한 좌표에 이미 돌이 놓여 있는 지 If 문을 활용하여 체크
  - 체크하여, 위의 두 경우 중 하나에 해당하면 이유 출력

3-4. 좌표에 O/X 놓기

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

- 입력
  - X = x좌표 x 값
  - y = y좌표 y 값
  - currentUser = 유저가 놓은 돌(O/X)
  - board = 게임 보드판 (배열)
- 결과
  - 현재 보드판에 유저가 입력한 좌표에 현재 유저의 돌 놓기
- 설명
  - 현재 배열의 유저에게 입력 받은 좌표에 현재 유저의 돌 저장

3-5. 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];  //보드판에 있는 o or x or 공백 출력
        if (j == numCell - 1) {
            break;
        }
        cout << "  |";
    }
    cout << endl;
}
cout << "---|---|---" << endl;
```

- 입력
  - numCell =가로와 세로 칸 개수
  - board = 게임 보드판 (배열)
  - i, j = 반복문을 돌릴 때 줄, 칸 변수

- 결과
  - 현재 보드판(배열) 출력
- 설명
  - 이중 for문을 활용하여 첫 번째 줄, 칸부터 현재 보드판의 상태를 출력

3-6. 모든 칸이 찼으면 종료

```
//6.모든 칸이 찼으면 종료
for (int s = 0; s < numCell; s++) {
    for (int t = 0; t < numCell; t++) {
        if (board[s][t] != ' ') {
            u = u + 1; //칸이 찼다며 u에 1 더하기
        }
    }
}
if (u >= 9) {
    cout << "칸이 다 찼습니다. 종료합니다.";
    break;   //u가 9 이상이 되면 종료
}
```

- 입력
  - numCell =가로와 세로 칸 개수
  - board = 게임 보드판 (배열)
  - s, t = 반복문을 돌릴 때 줄, 칸 변수
  - u = 칸이 찬 걸 확인하기 위한 변수 (코드 처음에 0으로 초기화)

- 결과
  - 칸이 다 찼다면 다 찼음을 출력하고 프로그램 종료
- 설명
  - 이중 for문을 활용하여 첫 번째 줄, 칸부터 칸이 비었는지를 체크
  - If 문을 활용하여 모든 칸이 다 찼다면 이를 출력하고 종료

3-7. 빙고 시 승자 출력 후 종료

```
//7.빙고 시 승자 출력 후 종료
bool win = false;
for (int c = 0; c < numCell; c++) {
    if (board[c][0] == currentUser && board[c][1] == currentUser && board[c][2] == currentUser) {
        cout << "가로에 모두 돌이 놓였습니다.";
        win = true;  //가로에 놓인 게 모두 같을 시 출력
    }
    if (board[0][c] == currentUser && board[1][c] == currentUser && board[2][c] == currentUser) {
        cout << "세로에 모두 돌이 놓였습니다.";
        win = true; //세로에 놓인 게 모두 같을 시 출력
    }
}
if (board[0][0] == currentUser && board[1][1] == currentUser && board[2][2] == currentUser) {
    cout << "왼쪽 위에서 오른쪽 아래 대각선으로 모두 돌이 놓였습니다.";
    win = true;
}
if (board[0][2] == currentUser && board[1][1] == currentUser && board[2][0] == currentUser) {
    cout << "오른쪽 위에서 왼쪽 아래 대각선으로 모두 돌이 놓였습니다.";
    win = true;  // 대각선에 놓인 게 모두 같을 시 출력
}
if (win == true) {
    cout << k % 2 + 1 << "번 유저(" << currentUser << ")의 승리입니다" << endl;
    cout << "종료합니다" << endl;
    break;
```

- 입력
  - numCell =가로와 세로 칸 개수
  - board = 게임 보드판 (배열)
  - c = 반복문을 돌릴 때 줄, 칸 변수
  - K = 누구 차례 인지 알기 위한 변수
  - currentUser = 유저가 놓은 돌(O/X)
  - win = 빙고를 판단하기 위한 변수(처음엔 false로 지정, 빙고 시 true로 변경)
- 결과
  - 가로/세로/대각선 중 하나로 빙고가 완성된다면 이를 출력
  - 빙고 시, 승자 유저를 출력하고 프로그램 종료
- 설명
  - for문과 if문을 활용하여 가로/세로에 모두 같은 돌이 놓였는지 체크
  - If 문을 활용하여 대각선에 모두 같은 돌이 놓였는지 체크
  - 빙고 시(win 변수가 true일 때) 승리한 유저를 출력하고 프로그램 종료

## 4. 테스트

### 4-1. 기능별 테스트 결과

(1) 누구의 차례인지 출력

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

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

(2) 좌표 입력 받기

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

```
2번 유저(O)의 차례입니다(x,y)좌표를 입력하세요: 2 1
```

(3) 입력 받은 좌표 유효성 체크

```
2번 유저(O)의 차례입니다(x,y)좌표를 입력하세요: 0 1
0, 1:이미 돌이 차있습니다.
```

```
2번 유저(O)의 차례입니다(x,y)좌표를 입력하세요: 0 3
0, 3: x와 y 둘 중 하나가 칸을 벗어납니다.
```

(4) / (5) 좌표에 O/X 놓기 / 현재 보드판 출력

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

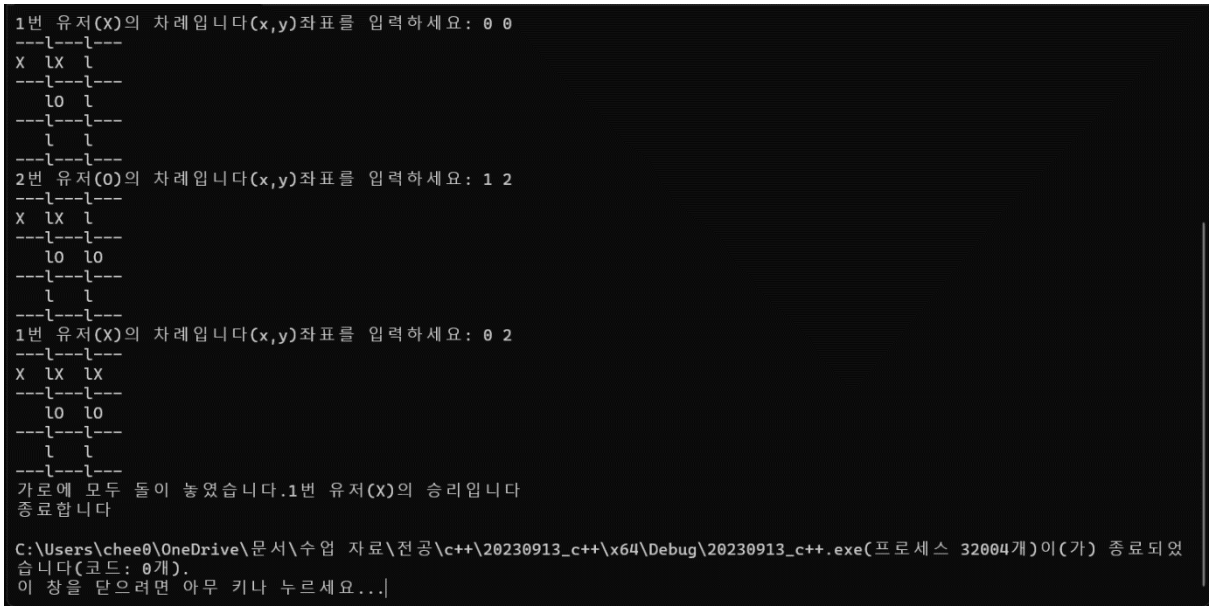
(6) 모든 칸이 찼으면 종료

```
---|---|---
O  |X  |O
---|---|---
X  |O  |X
---|---|---
X  |O  |X
---|---|---
칸이 다 찼습니다. 종료합니다.
```

(7) 빙고 시 승자 출력 후 종료

```
---|---|---
X  |X  |X
---|---|---
   |O  |O
---|---|---
   |   |
---|---|---
가로에 모두 돌이 놓였습니다.1번 유저(X)의 승리입니다
종료합니다
```

4-2.최종 테스트 스크린샷




## 5. 결과 및 결론

5-1. 프로젝트 결과

- Tic-tac-toe 게임을 C++로 구현하였음.

5-2. 소감

- 교수님께서 차근차근 알려주셔서 완성된 코드는 이해하기 어렵지 않았지만, C++을 이 수업으로 처음 접해본 저에게는 아직 이런 코드를 짜기에는 조금 어려운 것 같습니다.