

# C++프로그래밍및실습 보고서

소프트웨어학과

213161 이재혁

## 1. 프로젝트 목적 및 배경

: 4주차까지 배운 내용에 대한 실습을 위해 진행

- 목표 : Tic Tac Toe 게임 구현

## 2. 요구사항

- 사용자 요구사항 : 두 명의 사용자가 번갈아가며 O와 X를 놓기
- 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

## 3. 설계 및 구현

1. 기능 별 구현 사항

### ① 누구의 차례인지 출력

```
//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;
}
}
```

### ② 좌표 입력 받기

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

③ 입력 받은 좌표 유효성 체크

```
//3. 입력받은 좌표의 유효성 체크
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;
}
}
```

④ 좌표에 O / X 놓기

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

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];
        if(j == numCell - 1){
            break;
        }
        cout << " |";
    }
    cout << endl;
}
cout << "---|---|---" << endl;
k++, times++;
```

⑥ 빙고 시 승자 출력 후 종료

```
//6. 빙고 시 승자 출력 후 종료
for(int i = 0; i < numCell; i++){
    if(board[i][0] == board[i][1] && board[i][1] == board[i][2]){
        if(board[i][0] == 'X'){
            cout << "첫번째 유저(X)가 승리했습니다.";
        }
        else if(board[i][0] == 'O'){
            cout << "두번째 유저(O)가 승리했습니다.";
        }
        else
            continue;
    }
    if(board[0][i] == board[1][i] && board[1][i] == board[2][i]){
        if(board[0][i] == 'X'){
            cout << "첫번째 유저(X)가 승리했습니다.";
        }
        else if(board[0][i] == 'O'){
            cout << "두번째 유저(O)가 승리했습니다.";
        }
        else
            continue;
    }
}
if((board[0][0] == board[1][1] && board[0][0] == board[2][2]) || (board[0][2] == board[1][1] && board[0][2] == board[2][0])){
    if(board[1][1] == 'X'){
        cout << "첫번째 유저(X)가 승리했습니다.";
    }
    else if(board[1][1] == 'O'){
        cout << "두번째 유저(O)가 승리했습니다.";
    }
    else
        continue;
}
break;
}
```

⑦ 모든 칸이 찼으면 종료

```
//7. 모든 칸이 찼으면 종료
if(times == 9){
    cout << "남은 칸이 없어 종료하겠습니다.";
    break;
}
```

## 4. 테스트

- ① 누구의 차례인지 출력
- ② 좌표 입력 받기

```
왼쪽 위가 (0,0)이고 오른쪽 아래가 (2,2)입니다.첫번째 유저(x)의 차례입니다 -> (x,y) 좌표를 입력하세요 : 0
0
---|---|---
x  |   |
---|---|---
   |   |
---|---|---
   |   |
---|---|---
두번째 유저(o)의 차례입니다 -> (x,y) 좌표를 입력하세요 : 
```

- ③ 입력 받은 좌표 유효성 체크

```
왼쪽 위가 (0,0)이고 오른쪽 아래가 (2,2)입니다.첫번째 유저(x)의 차례입니다 -> (x,y) 좌표를 입력하세요 : 3
3, 3: x 와 y 둘 중 하나가 칸을 벗어납니다.
첫번째 유저(x)의 차례입니다 -> (x,y) 좌표를 입력하세요 : 
```

- ④ 좌표에 0 / X 놓기
- ⑤ 현재 보드판 출력

```
첫번째 유저(x)의 차례입니다 -> (x,y) 좌표를 입력하세요 : 1
1
---|---|---
   |   |
---|---|---
   |x  |
---|---|---
   |   |
---|---|---
두번째 유저(o)의 차례입니다 -> (x,y) 좌표를 입력하세요 : 
```

- ⑥ 빙고 시 승자 출력 후 종료

```
---|---|---
x  |o  |x
---|---|---
o  |o  |x
---|---|---
x  |x  |o
---|---|---
남은 칸이 없어 종료하겠습니다.
```

- ⑦ 모든 칸이 찼으면 종료

```
---|---|---
x  |o  |
---|---|---
   |x  |o
---|---|---
   |  |x
---|---|---
첫번째 유저(x)가 승리했습니다.
```

## 5. 결과 및 결론

: 2차원 배열과 지금까지 배운 기능들을 이용해 Tic Tac Toe게임을 만들었음

- 느낀점

: 처음에는 쉽다 생각했는데 생각보다 시간이 걸려서 힘들었지만 시간 안에 할 수 있어서 뿌듯했습니다.