

보고서

학과 : 소프트웨어공학과

학번 : 213182

이름 : 박태은

주제 : Tic Tac Toe 게임 C++

## 1. 서론

1. 프로젝트 목적 및 배경: 4주차까지 배운 내용에 대한 실습을 위해 진행
2. 목표 : Tic Tac Toe 게임 구현

## 2. 요구사항

1. 사용자 요구사항: 두 명의 사용자가 번갈아가며 O와 X를 놓기
2. 기능 요구사항:
  - 1) 누구의 차례인지 출력
  - 2) 좌표 입력 받기
  - 3) 입력 받은 좌표 유효성 체크
  - 4) 좌표에 O/X놓기
  - 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. 코드블록 스크린샷

```
while(true){  
    // 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. 입력

- currentUser = 'X'
- currentUser = 'Y'

## 3. 결과

- currentUser의 변수값에 따라서 어떤 유저 차례인지 출력

## 4. 설명

- switch문으로 차례변수 k가 짝수일때는 X의 차례, k가 홀수인 사람이 O의 차례라고 한다.
- while문 안에서 게임을 돌린다.

### 1. 코드블록 스크린샷

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

## 2. 입력

- x = 행 좌표
- y = 열 좌표

## 3. 결과

- 행 좌표와 열 좌표를 x와 y 변수에 입력받음

## 4. 설명

- cin으로 x와 y값을 입력받는다.

### 1. 코드블록 스크린샷

```
// 3. 입력받은 좌표의 유효성 체크
if (x >= numCell || y >= numCell){
    cout << x << ", " << y << ":";
    cout << " x와 y 둘 중 하나가 칸을 벗어납니다."<< endl;
    continue;
}
if (board[x][y] != ' '){
    cout << x << ", " << y << ": 이미 돌이 차있습니다."<< endl;
    continue;
}
```

## 2. 입력

- x = 행 값
- y = 열 값

-numCell = 가로/세로 칸 개수

### 3. 결과

- 칸을 놓을 수 없는 이유를 출력
- 출력 후 while문 초반으로 이동

### 4. 설명

- 1. 사용자가 입력한 좌표가 게임 판을 벗어나는지 if로 체크
- 2. 사용자가 입력한 좌표에 돌이 이미 있는지 if로 체크

#### 1. 코드블록 스크린샷

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

### 2. 입력

- x는 행 값, y는 열 값

### 3. 결과

- board의 x,y위치에 유저 값을 대입

### 4. 설명

- 1. 'X'나 'Y'를 board의 x와 y위치에 대입시킨다.

#### 1. 코드블록 스크린샷

```
// 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){ // 판의 제일 오른쪽에는 " |"가 없어서 2가 만나면 탈출  
            break;  
        }  
        cout << " |";  
    }  
    cout << endl;  
}  
cout << "---|---|---" << endl;
```

### 2. 입력

- numCell=가로/세로 칸 개수

### 3. 결과

- 보드판을 출력 "---|---|---"를 경계로한다.
- board[i][j]를 출력후 "--|" 경계 생성

### 4. 설명

- 판의 제일 오른쪽에는 "--|"가 없어서 numCell==2일때는 break하도록 if 문을 생성한다.

- "---|---|---"가 행의 경계선으로 있으니 마지막에도 "---|---|---"를 출력

#### 1. 코드블록 스크린샷

```
// 6-1. 빙고 시 승자 출력 후 종료(가로,세로)

bool flag = false; // 승자 확인 변수
for (int i = 0; i < numCell; i++) {
    if (board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser) {
        flag = true;
    }
    if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser) {
        flag = true;
    }
}

// 6-2. 빙고 시 승자 출력 후 종료(대각선)
if (board[0][0] == currentUser && board[1][1] == currentUser && board[2][2] == currentUser) {
    flag = true;
}
if (board[0][2] == currentUser && board[1][1] == currentUser && board[2][0] == currentUser) {
    flag = true;
}

// 6-3. 빙고 시 승자 출력 후 종료(승자 출력 후 종료)
if (flag) {
    cout << currentUser << "가 승리했습니다!" << endl;
    break;
}
```

#### 2. 입력

-flag = false

-numCell=가로/세로 칸 개수

-currentUser=유저의 돌

#### 3. 결과

- 가로의 돌이 다 같거나, 세로의 돌이 다 같거나, 대각선의 돌이 다 같아서 승자가 나오는 경우를 출력

- flag를 true로 바꿔서 승자 출력 후 break

#### 4. 설명

- flag를 설정해서 false로 만들어 놓고, for문안에 if를 돌려서, 가로와 세로가 일치하면 true로, if문에 대각선이 일치하면 true로 바꾼다.

- if문에 flag가 true면 currentUser가 승리했음을 출력.

#### 1. 코드블록 스크린샷

```
// 7. 모든 칸이 찼으면 종료(무승부)
if (k==8){
    cout<< "무승부입니다.";
    break;
}
```

#### 2. 입력

- k= 차례변수

#### 3. 결과

- k==8이면 무승부 출력 후 break

#### 4. 설명

- k==8이면 위에서 승리 요건이 안찬 상태로 보드판이 가득 찬것이니까. 무승부입니다를 출력한다.

#### 4. 테스트

##### 1. 기능 별 테스트 결과:

##### 1. 누구 차례인지 출력

첫번째 유저(x)의 차례입니다 -> (행,열) 좌표를 입력하세요:

##### 2. 좌표 입력받기

첫번째 유저(x)의 차례입니다 -> (행,열) 좌표를 입력하세요: 1 1

##### 3. 입력 받은 좌표 유효성 체크

두번째 유저(o)의 차례입니다 -> (행,열) 좌표를 입력하세요: 1 1  
1, 1: 이미 둘이 차있습니다.  
두번째 유저(o)의 차례입니다 -> (행,열) 좌표를 입력하세요: 2 4  
2, 4: x와 y 둘 중 하나가 칸을 벗어납니다.

##### 4. 좌표에 O/X놓기

첫번째 유저(x)의 차례입니다 -> (행,열) 좌표를 입력하세요: 1 1

```
---|---|---
|   |   |
---|---|---
| x |   |
---|---|---
|   |   |
---|---|---
```

##### 5. 현재 보드판 출력

첫번째 유저(x)의 차례입니다 -> (행,열) 좌표를 입력하세요: 1 1

```
---|---|---  
  |  |  |  
---|---|---  
  |x |  |  
---|---|---  
  |  |  |  
---|---|---
```

##### 6. 빙고 시 승자 출력 후 종료

첫번째 유저(x)의 차례입니다 -> (행,열) 좌표를 입력하세요: 0 0

```
---|---|---  
x |  |  |  
---|---|---  
  |x |o |  
---|---|---  
  |o |x |  
---|---|---
```

x가 승리했습니다!

첫번째 유저(x)의 차례입니다 -> (행,열) 좌표를 입력하세요: 0 2

```
---|---|---  
x |x |x |  
---|---|---  
  |o |  |  
---|---|---  
  |  |o |  
---|---|---
```

x가 승리했습니다!

첫번째 유저(x)의 차례입니다 -> (행,열) 좌표를 입력하세요: 2 0

```
---|---|---  
x |  |  |  
---|---|---  
x |  |o |  
---|---|---  
x |  |o |  
---|---|---
```

x가 승리했습니다!

## 7. 모든 칸이 찼으면 종료

```
첫번째 유저(x)의 차례입니다 -> (행,열) 좌표를 입력하세요: 1 0
---|---|---
X  |O  |X
---|---|---
X  |O  |O
---|---|---
O  |X  |X
---|---|---
무승부입니다.
```

## 2. 최종 테스트 스크린샷:

```
첫번째 유저(x)의 차례입니다 -> (행,열) 좌표를 입력하세요: 0 0
---|---|---
X  |  |
---|---|---
  |  |
---|---|---
  |  |
---|---|---
두번째 유저(o)의 차례입니다 -> (행,열) 좌표를 입력하세요: 2 3
2, 3: x와 y 둘 중 하나가 칸을 벗어납니다.
두번째 유저(o)의 차례입니다 -> (행,열) 좌표를 입력하세요: 0 0
0, 0: 이미 둘이 차있습니다.
두번째 유저(o)의 차례입니다 -> (행,열) 좌표를 입력하세요: 1 2
---|---|---
X  |  |
---|---|---
  |O  |
---|---|---
  |  |
---|---|---
첫번째 유저(x)의 차례입니다 -> (행,열) 좌표를 입력하세요: 1 1
---|---|---
X  |  |
---|---|---
  |X  |O
---|---|---
  |  |
---|---|---
두번째 유저(o)의 차례입니다 -> (행,열) 좌표를 입력하세요: 2 1
---|---|---
X  |  |
---|---|---
  |X  |O
---|---|---
  |O  |
---|---|---
첫번째 유저(x)의 차례입니다 -> (행,열) 좌표를 입력하세요: 2 2
---|---|---
X  |  |
---|---|---
  |X  |O
---|---|---
  |O  |X
---|---|---
x가 승리했습니다!
```

## 5. 결과 및 결론

1. 프로젝트 결과: Tic Tac Toe 게임을 만들었음

2. 느낀 점:

- 6번 기능을 수행할 때 교수님이 효율적으로 짜려 해보라는 말이 기억나서 flag=false를 설정하고 for문 안에서 가로와 세로를 반복적으로 확인시키게 했다. 만약 조건이 충족되면 flag를 true값으로 바꿔서 승리를 출력하고 while문을 break하게 했다. 대각선은 2가지 경우의 수만 있어서 if문을 2번 써서 했다.
- '모든 칸이 찼으면 종료' 기능에서 어차피 앞의 승리요건이 충족하면 break로 while문을 빠져나가니, k==8일 경우 무승부를 출력하도록 했다.
- Tic tac toe 게임을 객체지향 설계 프로젝트에서 미니맥스 알고리즘 배울 때 예시에 나왔어서, tic tac toe 게임 코드를 짜면 어떻게 짤 수 있을까라는 생각을 한 적 있는데, 직접 코드를 짜보는 경험을 하니 좋았던 것 같다. 저번부터 flag 사용법이 익숙하지 않았는데 실습을 하며 계속 연습해보니 나중에 코드를 짤 때 잘 써먹을 수 있을 것 같다.