Ⅰ 서론

4주차까지 배운 내용에 대한 실습을 위해 Tic tac toe 게임 구현을 진행했습니다.

Ⅱ 요구사항

사용자 요구사항은 두 명의 사용자가 번갈아가며 O와 X를 놓기였으며, 기능 요구사항은 사용자의 차례 출력, 좌표 입력 받기, 좌표 유효성 체크, 보드에 O / X 표시, 현재 보드판 출력, 빙고 시 승자 출력 후 종료, 모든 칸이 찼으면 종료이었습니다.

Ⅲ 설계 및 구현

기능 별 구현 사항은 아래의 순서에 따라 표로 나타냈습니다.

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

	코드블록 스크린샷
①	

```
int k = 0; // 1.누구 차례인지 체크하기 위한 변수
char currentUser = 'X';

//게임 시작

while(true){

    // 차례 확인 인원수 변동시 변동
    switch(k % 2){
    case 0:{
        cout << "첫번째 유저(X)의 차례입니다";
        currentUser = 'X';
        break;
    }

     case 1:{
        cout << "두번째 유저(Y)의 차례입니다";
        currentUser = 'O';
        break;
    }
    }
```

	입력	결과	설명
	k : 차례 확인 지표 currentUser : 보드에 표시할 각 각의 유저 고유의 기호	차례를 출력 후 좌표 입력 받기로 넘어갑니다.	1. 현재 보드판의 출력이 끝난 후 k를 1씩 증가시키고, switch 문에서 k가 짝수면 첫번째 유저, 홀수면 두번째 유저의 차례가 되도록 조정했습니다. 2. 차례를 출력한 뒤 break로 switch문을 빠져나옵니다.
②	``` //2.좌표 입력 받기 cout << "(x, y) 좌표를 입력하세요: "; cin>> x >> y; ```		
	입력	결과	설명
	x : 좌표 x 값 y :좌표 y 값	사용자에게 입력 받은 좌표를 각각 x와 y로 받습니다.	1. while문을 한번 돌때마다 매번 새로 입력 받으며, board 배열은 while 루프 밖에 선언 되어있기 때문에 사용자가 입력한 좌표의 배열 변화가 누적됩니다.
③	``` //3. 좌표 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; } ```		
	입력	결과	설명
	x : 좌표 x 값 y : 좌표 y 값 numCell : 가로/세로 칸 개수	칸을 놓을 수 없는 이유를 출력 후 continue에 의해 while문 초반으로 이동합니다.	1. 사용자가 입력한 좌표가 게임판을 벗어나는지 if로 체크합니다. 2. 사용자가 입력한 좌표에 돌이 이미 있는지 if로 체크합니다.

④

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

입력	결과	설명
x : 좌표 x 값 y : 좌표 y 값	사용자가 입력한 좌표에 currentUser에 해당하는 돌을 놓습니다.	1. 1번 요구사항에서 currentUser에 배정한 기호의 돌을 보드 위의 사용자가 지정한 좌표에 놓습니다.

⑤

```
//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++;
```

입력	결과	설명
x : 좌표 x 값 y : 좌표 y 값 numCell : 가로/세로 칸 개수	현재의 보드 배열의 상태를 보드판과 함께 나타냅니다.	1. 이중 for문을 통해서 numCell만큼의 가로와 세로 칸을 출력하고 마지막에 보드의 아래 모서리를 닫습니다. 2. 사용자 차례를 나타내는 지표인 k를 1 증가시켜서 다음 사람에게 차례를 넘깁니다.

⑥

```
//6. 승리조건 판별
bool isWin = false;
char symbol[3] = {'X', 'O'};//모든 사용자에 대해 승리 조건 충족 확인 및 승리시 루프 탈출

for (char n : symbol){
    for(int i = 0; i < numCell; i++){
    if(board[i][0]== n && board[i][1]== n && board[i][2]== n){ // 가로 승리
    cout<< "user "<< n <<" 이 승리했습니다!";
    isWin = true;
    break;
    }
    else if(board[0][i]== n && board[1][i]== n && board[2][i]== n){// 세로 승리
    cout<< "user "<< n <<" 이 승리했습니다!";
    isWin = true;
    break;
    }
    }
    if(board[0][0]== n && board[1][1]== n && board[2][2]== n){ // \대각선 승리
    cout<< "user "<< n <<" 이 승리했습니다!";
    isWin = true;
    break;
    }
    else if(board[0][2]== n && board[1][1]== n && board[2][0]== n){// /대각선 승리
    cout<< "user "<< n <<" 이 승리했습니다!";
    isWin = true;
    break;
    }
}
if (isWin){// 승리 후 종료
    break;
}
```

입력	결과	설명
x : 좌표 x 값 y : 좌표 y 값 numCell : 가로/세로 칸 개수 isWin: 승리 조건 만족 시 true로 바뀌는 지표.	Symbol 배열에 있는 원소에 대해서 승리 조건을 검사합니다. 추후 3명으로 확장하기 위해 배열의 크기를 3으로 설정했습니다.	1. 만일 승리 조건을 만족한다면 isWin에 true를 대입하고 그대로 for문을 탈출합니다. 2. 만족하지 못할 시 다음 승리 조건을 만족하는지 검사합니다. 3. 만일 isWin이 true라면 그대로 while문을 탈출해 종료합니다.

⑦

```
//7. 다 찼을 경우 종료 판별
int endcount = 0;

for (int i = 0; i < numCell; i++){
    for(int j = 0; j < numCell; j++){
        if(board[i][j] != ' '){
            endcount++;
        }
    }
    }

if(endcount == (numCell)*(numCell)){
    cout << "모든칸이 다 찼습니다. 종료합니다" << endl;
    break;
}

}
```

입력	결과	설명
endcount: 종료 조건 판별 지표 numCell = 가로/세로 칸 개수	for문을 돌며 공백이 아닌 원소의 수를 측정합니다. 모두 공백이 아니라면 메시지를 출력하고 종료합니다.	1. 매 루프마다 endcount를 0으로 초기화하고, board 배열을 순회하며 원소가 공백이 아닐 경우 endcount를 1씩 증가시킵니다. 2. 만일 모든 원소가 공백이 아니어서 endcount가 numCell의 제곱이 되었을 경우 종료합니다.

IV 테스트

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

<table>
<tr><th></th><th>코드블록 스크린샷</th></tr>
<tr><td>①</td><td>

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

</td></tr>
<tr><td>②</td><td>

```
두번째 유저(Y)의 차례입니다(x, y) 좌표를 입력하세요: 2
2
첫번째 유저(X)의 차례입니다(x, y) 좌표를 입력하세요: 0
0
```

</td></tr>
<tr><td>③</td><td>

```
첫번째 유저(X)의 차례입니다(x, y) 좌표를 입력하세요: 3
3
3, 3:x와 y 둘 중 하나가 칸을 벗어납니다
```

칸 밖의 좌표를 입력한 경우

```
1
---|---|---
X  |   |
---|---|---
X  |O  |
---|---|---
   |   |O
---|---|---
첫번째 유저(X)의 차례입니다(x, y) 좌표를 입력하세요: 2
2
2, 2: 이미 돌이 차있습니다
```

이미 돌을 둔 곳의 좌표를 입력한 경우

</td></tr>
<tr><td>④, ⑤</td><td>

```
첫번째 유저(X)의 차례입니다(x, y) 좌표를 입력하세요: 0
0
---|---|---
X  |   |
---|---|---
   |   |
---|---|---
   |   |
---|---|---
두번째 유저(Y)의 차례입니다(x, y) 좌표를 입력하세요: 2
2
---|---|---
X  |   |
---|---|---
   |   |
---|---|---
   |   |O
---|---|---
```

</td></tr>
</table>

⑥

```
첫번째 유저(X)의 차례입니다(x, y) 좌표를 입력하세요: 0
2
---|---|---
X  |X  |X
---|---|---
   |O  |
---|---|---
   |   |O
---|---|---
userX 이 승리했습니다!
PS C:\CPP2409> ^C
```

가로 승리

```
첫번째 유저(X)의 차례입니다(x, y) 좌표를 입력하세요: 2
0
---|---|---
X  |   |
---|---|---
X  |O  |
---|---|---
X  |   |O
---|---|---
user X 이 승리했습니다!
PS C:\CPP2409>
```

세로 승리

```
첫번째 유저(X)의 차례입니다(x, y) 좌표를 입력하세요: 2
2
---|---|---
X  |O  |O
---|---|---
   |X  |
---|---|---
   |   |X
---|---|---
user X 이 승리했습니다!
PS C:\CPP2409>
```

대각선 승리

```
첫번째 유저(X)의 차례입니다(x, y) 좌표를 입력하세요: 2
0
---|---|---
O  |O  |X
---|---|---
   |X  |
---|---|---
X  |   |
---|---|---
user X 이 승리했습니다!
PS C:\CPP2409>
```

대각선 승리

⑦

```
첫번째 유저(X)의 차례입니다(x, y) 좌표를 입력하세요: 2
2
---|---|---
X  |X  |O
---|---|---
O  |O  |X
---|---|---
X  |O  |X
---|---|---
모든칸이 다 찼습니다. 종료합니다
PS C:\CPP2409>
```

⑧

```
첫번째 유저(X)의 차례입니다(x, y) 좌표를 입력하세요: 2
0
---|---|---
X  |X  |O
---|---|---
   |O  |
---|---|---
X  |   |
---|---|---
두번째 유저(Y)의 차례입니다(x, y) 좌표를 입력하세요: 1
0
---|---|---
X  |X  |O
---|---|---
O  |O  |
---|---|---
X  |   |
---|---|---
첫번째 유저(X)의 차례입니다(x, y) 좌표를 입력하세요: 1
2
---|---|---
X  |X  |O
---|---|---
O  |O  |X
---|---|---
X  |   |
---|---|---
두번째 유저(Y)의 차례입니다(x, y) 좌표를 입력하세요: 2
1
---|---|---
X  |X  |O
---|---|---
O  |O  |X
---|---|---
X  |O  |
---|---|---
첫번째 유저(X)의 차례입니다(x, y) 좌표를 입력하세요: 2
2
---|---|---
X  |X  |O
---|---|---
O  |O  |X
---|---|---
X  |O  |X
---|---|---
모든칸이 다 찼습니다. 종료합니다
PS C:\CPP2409>
```

프로그램 전체 동작 스크린샷

Ⅴ 결과 및 결론

1. 프로젝트 결과

Tic Tac Toe 게임을 만들었습니다. 결과적으로 강의 시간 내에서는 종료조건을 맞추어도 계속 루프를 돌아서 괄호를 세다가 시간이 넘었습니다. 집에 와서 확인해보니 조건문 안에서 isWin에 true를 대입하는 과정에서 isWin의 앞에 bool이 붙어 지역변수가 되어 for문 안에서만 유효했다는 사실을 알았습니다. 시간을 줄이기 위해 복사 붙여넣기를 활용했지만 그것도 주의 깊게 써야 한다는 사실을 간과한 결과였습니다. 또한 while문의 몸체가 커지다 보니 변수가 어떤 루프를 시작할 때 초기화되는지 확인하기가 어려웠습니다. While문 안의 요소들을 따로 정의해서 몸체의 크기를 줄이거나, 변수를 한데 모아 초기화하는 방식을 사용한다면 어떨지 궁금했습니다.

2. 느낀 점

비록 저는 늘 늦게까지 남아서 하고, 객관적으로 쉬운 문제도 어려울 때가 있지만 그렇다고 이 강의의 난이도가 더 쉽길 바라지 않습니다. 저는 성장을 원하며 그것을 위해 시간을 쏟는 건 당연하다고 생각합니다. 하지 못했던 걸 할 수 있어야 성장할 수 있다고 생각합니다. 교수님의 시각에서 필요한 부분이 있다면 얼마든지 배우길 희망합니다. 좋은 강의와 함께 배울 수 있음에 감사함을 느끼고 있습니다.