

214014 김현규

1. 서론

1-1. 프로젝트 목적 및 배경 : 4주차까지 배운 내용에 대한 실습을 위해 진행

1-2. 목표 : TIC TAC TOE 게임 구현

2. 요구사항

2-1. 사용자 요구사항 : 두 명의 사용자가 번갈아가며 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 빙고 시 승자 출력 후 종료
- ⑦ 모든 칸이 차면 게임 종료

3. 설계 및 구현

3-1. 기능 별 구현 사항 :

①에 대한 구현

1. 스크린샷

```
int k = 0;
char currentUser = 'X';
while(true){
    switch (k % 2){
        case 0:
            cout << "첫 번째 유저(X)의 차례입니다. --> ";
            currentUser = 'X';
            break;
        case 1:
            cout << "두 번째 유저(O)의 차례입니다 --> ";
            currentUser = 'O';
            break;
    }
}
```

2. 입력

k = 0

currentUser = 'X'

3. 결과

처음엔 스크린샷에 있는 문장인 “첫 번째 유저(X)의 차례입니다. -->”가 출력
두 번째 유저의 차례에 “두 번째 유저(O)의 차례입니다 -->”가 출력

4. 설명

1. 정수 int k로 순서를 결정 --> $k \% 2$ 를 조건으로 하여 맨 마지막에 k++;이 있는데 이를 이용해 첫 번째 유저와 두 번째 유저가 번갈아가며 차례를 부여받음
2. 한 유저의 차례가 끝나면 currentUser의 값을 각 유저의 값으로 설정하여 둘을 두었음

②에 대한 구현

1. 스크린샷

```
cout << "(x, y) 좌표를 입력하세요 : ";  
cin >> y >> x; // x,y가 거꾸로 되어있어 변경
```

2. 결과

“(x,y) 좌표를 입력하세요 : ”가 출력
문장 출력 후 x,y 좌표값을 입력받음

3. 설명

1. x,y 좌표값을 입력받음

③에 대한 구현

1. 스크린샷

```
if(x >= numCell || y >= numCell){  
    cout << y << ", " << x << ": ";  
    cout << "정해진 범위를 벗어납니다. x, y 좌표를 0 ~ 2로 설정해주세요." << endl; // 구문 조정  
    continue;  
}  
if(board[x][y] != ' '){  
    cout << y << ", " << x << ": 이미 돌이 차있습니다." << endl;  
    continue;  
}
```

2. 입력

x = x 좌표 값
y = y 좌표 값
numCell = 가로/세로 칸 개수로 상수 3으로 고정

3. 결과

좌표를 벗어나거나 이미 놓았으면 칸에 놓을 수 없음을 출력
출력 후 while문 초반으로 이동하여 다시 입력받음

4. 설명

1. 사용자가 입력한 좌표가 게임 판을 벗어나는지 if문으로 체크
2. 사용자가 입력한 좌표에 이미 수를 놓았는지 if문으로 체크

④에 대한 구현

1. 스크린샷

```
board[x][y] = currentUser;
```

2. 입력

currentUser = 차례에 맞는 유저의 돌

3. 결과

입력받은 x,y 좌표 값에 차례에 맞는 유저의 돌을 놓음

4. 설명

board라는 2차원 배열을 사용해 좌표를 구현하고 이에 맞는 곳에 currentUser를 대입하여 돌을 넣음.

⑤에 대한 구현

1. 스크린샷

```
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;
```

2. 입력

numCell = 가로/세로 칸 개수로 상수 3으로 고정

3. 결과

네모난 칸을 9개 만들고, 입력받은 좌표값에 사용자에게 따른 돌을 입력

4. 설명

for문을 사용해 9개의 네모난 칸을 만들어서 보드판을 출력함.

board가 입력받지 않으면 빈칸이고, 입력받으면 그에 맞는 돌을 출력함.

⑥에 대한 구현

1. 스크린샷

```
// X 승리 시 출력
if(board[0][0]=='X'){
    if(board[0][1]=='X'){
        if(board[0][2]=='X'){
            cout << "첫 번째 유저(X)가 승리했습니다!";
            break;
        }
    }
    if(board[1][1]=='X'){
        if(board[2][2]=='X'){
            cout << "첫 번째 유저(X)가 승리했습니다!";
            break;
        }
    }
    if(board[1][0]=='X'){
        if(board[2][0]=='X'){
            cout << "첫 번째 유저(X)가 승리했습니다!";
            break;
        }
    }
}

if(board[2][0]=='X' && board[2][1]=='X' && board[2][2]=='X'){
    cout << "첫 번째 유저(X)가 승리했습니다!" << endl;
    break;
}

if(board[1][0]=='X' && board[1][1]=='X' && board[1][2]=='X'){
    cout << "첫 번째 유저(X)가 승리했습니다!" << endl;
    break;
}

if(board[0][2]=='X' && board[1][1]=='X' && board[2][0]=='X'){
    cout << "첫 번째 유저(X)가 승리했습니다!" << endl;
    break;
}

// O 승리 시 출력
if(board[0][0]=='O'){
    if(board[0][1]=='O'){
        if(board[0][2]=='O'){
            cout << "두 번째 유저(O)가 승리했습니다!";
            break;
        }
    }
    if(board[1][1]=='O'){
        if(board[2][2]=='O'){
            cout << "두 번째 유저(O)가 승리했습니다!";
            break;
        }
    }
    if(board[1][0]=='O'){
        if(board[2][0]=='O'){
            cout << "두 번째 유저(O)가 승리했습니다!";
            break;
        }
    }
}

if(board[2][0]=='O' && board[2][1]=='O' && board[2][2]=='O'){
    cout << "두 번째 유저(O)가 승리했습니다!" << endl;
    break;
}

if(board[1][0]=='O' && board[1][1]=='O' && board[1][2]=='O'){
    cout << "두 번째 유저(O)가 승리했습니다!" << endl;
    break;
}

if(board[0][2]=='O' && board[1][1]=='O' && board[2][0]=='O'){
    cout << "두 번째 유저(O)가 승리했습니다!" << endl;
    break;
}
```

2. 입력

board[n][n] = 입력받은 currentUser 값 or 입력 받지 못했다면 ' '

3. 결과

빙고가 될 시에 승리 멘트를 출력

4. 설명

- 좌표에 따라서 한 열이나 행의 값이 모두 X/O가 된다면 빙고가 되기에 이를 if문으로 체크
- 대각선의 값이 모두 X/O가 된다면 이도 빙고가 되기에 이를 if문으로 체크
- 승리 멘트 이후 break:로 while문을 빠져나가 게임을 종료시킴.

⑦에 대한 구현

1. 스크린샷

```
//모든 칸이 차면 종료
int checkNum = 0;
for(int i = 0; i < numCell; i++){
    for(int j = 0; j < numCell; j++){
        if(board[i][j]!=' '){
            checkNum++;
        }
    }
}

if(checkNum == 9){
    cout << "모든 칸이 가득 찼습니다! 게임이 종료되었습니다.";
    break;
}
```

2. 입력

checkNum = 0

numCell = 가로/세로 칸 개수로 상수 3으로 고정

board[n][n] = 입력받은 currentUser 값 or 입력 받지 못했다면 ' '

3. 결과

누군가의 승리 이전에 checkNum이 9에 도달하면 “모든 칸이 가득 찼습니다! 게임이 종료되었습니다.” 문구가 출력

4. 설명

1. checkNum을 정수로 설정하여 사용자의 입력을 받을때마다 1씩 증가시킴
2. 그렇게 증가시킨 checkNum이 9에 도달하게 된다면 누군가의 승리 이전에 모든 보드 칸이 가득차게 된다는 의미이므로 게임이 종료되게 만듦

4. 테스트

① 누구의 차례인지 출력, ② 좌표 입력 받기

```
첫 번째 유저(x)의 차례입니다. --> (x, y) 좌표를 입력하세요 : 
```

③ 입력 받은 좌표 유효성 체크, ④ 좌표에 O/X 놓기, ⑤ 현재 보드판 출력

```
첫 번째 유저(x)의 차례입니다. --> (x, y) 좌표를 입력하세요 : 2 3
2, 3: 정해진 범위를 벗어납니다. x, y 좌표를 0 ~ 2로 설정해주세요.
첫 번째 유저(x)의 차례입니다. --> (x, y) 좌표를 입력하세요 : 0 0
---|---|---
x  |   |   |
---|---|---
---|---|---
---|---|---
두 번째 유저(o)의 차례입니다 --> (x, y) 좌표를 입력하세요 : 0 0
0, 0: 이미 들어 있습니다.
두 번째 유저(o)의 차례입니다 --> (x, y) 좌표를 입력하세요 : 
```

⑥ 빙고 시 승자 출력 후 종료

6-1. 첫 번째 유저가 승리 시

6-1-1. 1열 승리 시

```
첫 번째 유저(x)의 차례입니다. --> (x, y) 좌표를 입력하세요 : 2 0
---|---|---
x  |x  |x  |
---|---|---
0  |   |   |
---|---|---
0  |   |   |
---|---|---
첫 번째 유저(x)가 승리했습니다!
PS C:\CPP\cpp2409> ^C
```

6-1-2. 2열 승리 시

```

첫 번째 유저(x)의 차례입니다. --> (x, y) 좌표를 입력하세요 : 2
1
---|---|---
0 |0 |
---|---|---
X |X |X
---|---|---
---|---|---
첫 번째 유저(x)가 승리했습니다!
PS C:\CPP\cpp2409> ^C

```

6-1-3. 3열 승리 시

```

첫 번째 유저(x)의 차례입니다. --> (x, y) 좌표를 입력하세요 : 2 2
---|---|---
0 | |
---|---|---
 |0 |
---|---|---
X |X |X
---|---|---
첫 번째 유저(x)가 승리했습니다!
PS C:\CPP\cpp2409> 

```

6-1-4. 1행 승리 시

```

첫 번째 유저(x)의 차례입니다. --> (x, y) 좌표를 입력하세요 : 0 2
---|---|---
X |0 |
---|---|---
X |0 |
---|---|---
X | |
---|---|---
첫 번째 유저(x)가 승리했습니다!
PS C:\CPP\cpp2409> 

```

6-1-5. 2행 승리 시

```

첫 번째 유저(x)의 차례입니다. --> (x, y) 좌표를 입력하세요 : 1 2
---|---|---
0 |X |
---|---|---
0 |X |
---|---|---
 |X |
---|---|---
두 번째 유저(o)의 차례입니다 --> (x, y) 좌표를 입력하세요 : 

```

6-1-6. 3행 승리 시

```

첫 번째 유저(x)의 차례입니다. --> (x, y) 좌표를 입력하세요 : 2 2
---|---|---
0  |  |X
---|---|---
0  |  |X
---|---|---
   |  |X
---|---|---
두 번째 유저(o)의 차례입니다 --> (x, y) 좌표를 입력하세요 :

```

--> 이 부분에 대해서는 실수로 2행, 3행 승리 시 게임 종료 기능을 추가하지 않았습니다. 앞서 했던 것과 마찬가지로

```

if(board[0][1] == 'X' && board[1][1] == 'X' && board[2][1] == 'X'){
    cout << "첫 번째 유저(X)가 승리했습니다!" << endl;
    break;
}

```

와

```

if(board[0][2] == 'X' && board[1][2] == 'X' && board[2][2] == 'X'){
    cout << "첫 번째 유저(X)가 승리했습니다!" << endl;
    break;
}

```

를 추가한다면 이 또한 작동할 것이라고 생각합니다.

6-1-7. 우하향 대각선 승리 시

```

첫 번째 유저(x)의 차례입니다. --> (x, y) 좌표를 입력하세요 : 2 2
---|---|---
X  |  |
---|---|---
0  |X  |
---|---|---
0  |  |X
---|---|---
첫 번째 유저(x)가 승리했습니다!

```

6-1-8. 우상향 대각선 승리 시

```

첫 번째 유저(x)의 차례입니다. --> (x, y) 좌표를 입력하세요 : 0 2
---|---|---
0  |  |X
---|---|---
0  |X  |
---|---|---
X  |  |
---|---|---
첫 번째 유저(x)가 승리했습니다!

```

6-2. 두 번째 유저가 승리 시

6-2-1. 1열 승리 시

```
두 번째 유저(o)의 차례입니다 --> (x, y) 좌표를 입력하세요 : 2 0
---|---|---
0  |0  |0
---|---|---
   |x  |x
---|---|---
   |   |x
---|---|---
두 번째 유저(o)가 승리했습니다!
PS C:\CPP\cpp2409>
```

6-2-2. 2열 승리 시

```
두 번째 유저(o)의 차례입니다 --> (x, y) 좌표를 입력하세요 : 2 1
---|---|---
x  |   |x
---|---|---
0  |0  |0
---|---|---
   |   |x
---|---|---
두 번째 유저(o)가 승리했습니다!
PS C:\CPP\cpp2409>
```

6-2-3. 3열 승리 시

```
두 번째 유저(o)의 차례입니다 --> (x, y) 좌표를 입력하세요 : 2 2
---|---|---
x  |   |
---|---|---
x  |x  |
---|---|---
0  |0  |0
---|---|---
두 번째 유저(o)가 승리했습니다!
PS C:\CPP\cpp2409>
```

6-2-4. 1행 승리 시

```
두 번째 유저(o)의 차례입니다 --> (x, y) 좌표를 입력하세요 : 0 2
---|---|---
0  |   |
---|---|---
0  |x  |x
---|---|---
0  |   |x
---|---|---
두 번째 유저(o)가 승리했습니다!
PS C:\CPP\cpp2409>
```

6-2-5. 2행 승리 시


```

두 번째 유저(0)의 차례입니다 --> (x, y) 좌표를 입력하세요 : 1 2
---|---|---
X |O |
---|---|---
X |O |X
---|---|---
   |O |
---|---|---
첫 번째 유저(X)의 차례입니다. --> (x, y) 좌표를 입력하세요 : 

```

6-2-6. 3행 승리 시

```

두 번째 유저(0)의 차례입니다 --> (x, y) 좌표를 입력하세요 : 2 2
---|---|---
X |  |O
---|---|---
X |X |O
---|---|---
   |O
---|---|---
첫 번째 유저(X)의 차례입니다. --> (x, y) 좌표를 입력하세요 : 

```

--> 여기도 또한 앞서 했던 코드를 복사해 O로만 교체했기에 2행과 3행을 빠뜨렸습니다.

6-2-7. 우하향 승리 시

```

두 번째 유저(0)의 차례입니다 --> (x, y) 좌표를 입력하세요 : 2 2
---|---|---
O |X |X
---|---|---
X |O |
---|---|---
   |O
---|---|---
두 번째 유저(0)가 승리했습니다!
PS C:\CPP\cpp2409> 

```

6-2-8. 우상향 승리 시

```

두 번째 유저(0)의 차례입니다 --> (x, y) 좌표를 입력하세요 : 0 2
---|---|---
X |X |O
---|---|---
X |O |
---|---|---
O |  |
---|---|---
두 번째 유저(0)가 승리했습니다!
PS C:\CPP\cpp2409> 

```

⑦ 모든 칸이 차면 게임 종료

```
첫 번째 유저(x)의 차례입니다. --> (x, y) 좌표를 입력하세요 : 1 2
---|---|---
X  |O  |X
---|---|---
O  |X  |X
---|---|---
O  |X  |O
---|---|---
모든 칸이 가득 찼습니다! 게임이 종료되었습니다.
PS C:\CPP\cpp2409>
```

5. 결과 및 결론

1. 프로젝트 결과 : Tic Tac Toe 게임을 만들었음.

2. 느낀 점 :

2차원 배열의 개념에 대해 헷갈린 부분이 있었습니다. 저는 x, y 좌표를 그대로 행과 열로 생각하고 코드를 작성했는데, 그게 아니어서 많은 시간을 허비하게 되었습니다. 다행히 교수님께서 뒤에서 유심히 보시다가 진행되지 않는 제 코드의 문제를 알려주셔서 해결할 수 있었습니다. 하지만 2차원 배열의 개념을 정확히 이해하지 못해 시간을 낭비한 탓에 제대로 테스트를 진행하지 못했고, 보고서 또한 제 시간 안에 작성하지 못했습니다. 또한 2행과 3행 승리시 멘트를 작성하고 게임을 종료하는 코드를 빼먹는 큰 실수를 해서 매우 아쉬웠습니다.

수업 시간에 챌린지 문제를 풀며 마치 시험을 보는 듯한 기분이 들었지만, 오히려 이런 점이 좋았습니다. 집에서 충분한 시간을 가지고 과제를 할 때는 ChatGPT의 도움을 많이 받곤 했지만, 수업 시간 내에 해결하려 하다 보니 스스로 문제를 풀기 위해 더 노력하는 제 모습을 발견할 수 있었습니다. 개인적으로 실수를 한 부분이 매우 아쉬웠지만, 주어진 시간 내에 문제를 해결하고 결과를 올리는 수업 방식은 굉장히 좋다고 생각합니다 !