

# tic\_tac\_toe 게임 만들기

빅데이터융합학과 222400 김서연

## 1. 서론

### 1. 프로젝트 목적 및 배경:

지금까지 배운 내용을 바탕으로 실습하기 위해 프로젝트를 진행하였다.

### 2. 목표:

Tic Tac Toe 게임 구현

## 2. 요구사항

### 1. 사용자 요구사항 :

1) 두 명의 사용자가 번갈아가면서 O와 X를 놓기

### 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) 빙고시 승자 출력후 종료
- 7) 모든 칸이 찼으면 종료

## 3. 설계 및 구현

### 1. 기능 별 구현 사항:

1) 누구의 차례인지 출력

<코드블록 스크린샷>

```
//1. switch 문을 이용해서 누구 차례인지 출력
switch (k % 2) {
case 0:
    cout << k % 2 + 1 << "번 유저 (X)의 차례입니다 -> ";
    currentUser = 'X';
    break;
case 1:
    cout << k % 2 + 1 << "번 유저 (O)의 차례입니다 -> ";
    currentUser = 'O';
    break;
}
```

<결과>

- 누구 차례인지 출력
- 출력 후 while문 초반으로 이동

<설명>

1. 나머지가 0이면 유저 X차례 , 나머지가 1이면 유저 O차례
2. switch 문을 이용한 조건문 사용

## 2) 좌표 입력 받기

<코드블록 스크린샷>

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

<입력>

- 사용자가 좌표 입력(x,y)

<결과>

- 좌표 출력
- 출력후 while문 초반으로 이동

<설명>

1. cout 함수를 이용해 좌표 입력하라는 말 작성
2. cin 함수를 이용해 x,y 값(좌표) 받기

## 3) 입력 받은 좌표 유효성 체크

<코드블록 스크린샷>

```
//3. 입력받은 좌표의 유효성 체크
// x , y 둘 중 하나가 3이상을 입력할 경우 벗어난다는 말 출력
if (x >= numCell || y >= numCell) {
    cout << x << ", " << y << ": ";
    cout << "x와 y 둘 중 하나가 칸을 벗어납니다." << endl;
    continue;
}

//놓은 돌의 위치가 공백이 아닌경우 돌이 차있다고 출력
if (board[x][y] != ' ') {
    cout << x << ", " << y << ": 이미 돌이 차있습니다." << endl;
    continue;
}
```

<입력>

- x= 좌표 x값
- y= 좌표 y값
- numCell = 가로/세로 칸 개수

<결과>

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

<설명>

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

4) 좌표에 O/X 놓기

<코드블록 스크린샷>

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

<입력>

- x= 좌표 x값
- y= 좌표 y값
- currentUser = o,x 사용자

<결과>

- 게임 판 좌표에 O,X 놓기
- 출력 후 while문 초반으로 이동

<설명>

1. 2차원 배열을 이용해서 현재 유저의 돌 놓기

5) 현재 보드판 출력

<코드블록 스크린샷>

```
//5. 현재 보드판 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>

- numCell = 가로/세로 칸 개수

<결과>

- 보드판 출력
- 출력 후 while문 초반으로 이동

<설명>

1. i와j를 이용한 이중 반복문 사용(i가 행, j 가 열)
2. i가 numCell보다 작을 때까지 ---|---|--- 출력
3. j가 numCell -1 이 되면 멈추고, 그 전까진 | 출력

## 6) 빙고시 승자 출력후 종료

<코드블록 스크린샷>

```
//6. 승자 체크하기
bool isWin = false;

// 6.1 가로/세로 둘 체크하기
for (int i = 0; i < numCell; i++) {
    // 내가 입력한 좌표와 board[i][0], board[i][1], board[i][2] 와 같으면 가로 빙고
    if (board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser) {
        cout << "가로에 모두 돌이 놓였습니다 !: ";
        isWin = true;
    }
    // 똑같은 방식
    if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser) {
        cout << "세로에 모두 돌이 놓였습니다 !: ";
        isWin = true;
    }
}

//6.2 대각선 둘 체크하기
if (board[0][0] == currentUser && board[1][1] == currentUser && board[2][2] == currentUser) {
    cout << "왼쪽 위에서 오른쪽 아래 대각선으로 모두 돌이 놓였습니다!:";
    isWin = true;
}

if (board[0][2] == currentUser && board[1][1] == currentUser && board[2][0] == currentUser) {
    cout << "오른쪽 위에서 왼쪽 아래 대각선으로 모두 돌이 놓였습니다!:";
    isWin = true;
}

if (isWin == true) {
    cout << k % 2 + 1 << "번 유저(" << currentUser << ")의 승리입니다! " << endl;
    cout << "종료합니다" << endl;
    break;
}
```

<입력>

- numCell = 가로/세로 칸 개수

- currentUser = board[x][y] (내가 입력한 좌표)

<결과>

- 승자가 누군지 출력
- 출력 후 while문 초반으로 이동

<설명>

1. 가로/세로에 있는 돌이 내가 입력한 좌표와 같은지 if 문과 배열을 이용해서 확인
2. 대각선에 있는 돌이 내가 입력한 좌표와 같은지 if 문과 배열을 이용해서 확인
3. 내가 입력한 좌표의 돌(currentUser)이 한줄로 있다면 if 와 break를 사용해서 승자임을 알려주고 종료

7) 모든 칸이 찼으면 종료

<코드블록 스크린샷>

```
// 7. 모든 칸이 찼으면 종료
int checked = 0;
// i와 j 의 반복문을 이용해 board[i][j]가 공백이면 checked 에 1씩 더해주기
for (int i = 0; i < numCell; i++) {
    for (int j = 0; j < numCell; j++) {
        if (board[i][j] == ' ') {
            checked++;
        }
    }
}

//checked가 0이면 종료
if (checked == 0) {
    cout << "모든 칸이 다 찹습니다. 종료합니다." << endl;
    break;
}

return 0;
```

<입력>

-numCell = 가로/세로 칸 개수

<결과>

- 모든 칸이 찼으면 게임 종료

<설명>

1. i와 j의 이중 반복문 이용해서 board 2차원 배열에 사용
2. board[i][j]가 공백이면 checked에 1씩 더해주기
3. 공간이 없어서 0이되면 게임 종료

#### 4. 테스트

##### 1. 기능 별 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: 0 2  
---|---|---
```

###### 3) 입력 받은 좌표 유효성 체크

```
1번 유저(x)의 차례입니다 -> (x,y)좌표를 입력하세요: 0 2  
0, 2: 이미 둘이 차있습니다.
```

```
1번 유저(x)의 차례입니다 -> (x,y)좌표를 입력하세요: 3 2  
3, 2: x와 y 둘 중 하나가 칸을 벗어납니다.
```

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

###### 4) 좌표에 O/X 놓기(5번이랑 같이)

###### 5) 현재 보드판 출력

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

```
---|---|---
```

```
---|---|---
```

```
---|---|---
```

```
---|---|X
```

```
---|---|---
```

```
---|---|---
```

```
---|---|---
```

```
2번 유저(o)의 차례입니다 -> (x,y)좌표를 입력하세요: 0 2
```

```
---|---|---
```

```
---|---|O
```

```
---|---|---
```

```
---|---|X
```

```
---|---|---
```

```
---|---|---
```

```
---|---|---
```

## 6) 빙고시 승자 출력 후 종료

```
1번 유저(X)의 차례입니다 -> (x,y)좌표를 입력하세요: 0 1
---|---|---
0  |X  |0
---|---|---
0  |X  |X
---|---|---
   |X  |
---|---|---
세로에 모두 둘이 놓였습니다 ! : 2번 유저(X)의 승리입니다!
종료합니다
```

## 7) 모든 칸이 찼으면 종료

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

## 2. 최종 테스트 스크린 샷:

```
---|---|---
1번 유저(X)의 차례입니다 -> (x,y)좌표를 입력하세요: 1 1
---|---|---
X  |O  |
---|---|---
   |X  |
---|---|---
   |   |
---|---|---
2번 유저(O)의 차례입니다 -> (x,y)좌표를 입력하세요: 1 2
---|---|---
X  |O  |
---|---|---
   |X  |O
---|---|---
   |   |
---|---|---
1번 유저(X)의 차례입니다 -> (x,y)좌표를 입력하세요: 2 2
---|---|---
X  |O  |
---|---|---
   |X  |O
---|---|---
   |X  |
---|---|---
왼쪽 위에서 오른쪽 아래 대각선으로 모두 둘이 놓였습니다! : 2번 유저(X)의 승리입니다!
종료합니다
```

## 5. 결과 및 결론

### 1. 프로젝트 결과 :

Tic Tac Toe 게임을 만들었다.

### 2. 느낀점:

c++도 처음 배워보고, 이렇게 코드를 많이 쳐본 적도 처음이라 시작하기 전에 어떻게 해야 될지 몰라서 너무 막막했었다. 그래도 교수님이 나중에 올려주신 코드를 보면서 하나하나씩 이해해 보니, 이제는 어떤 식으로 접근해야 할지 조금 알 거 같다. 아직은 이런 내용들이 너무 어렵고 낯설지만 여러 개의 문제들을 보고, 프로젝트를 하면서 많이 배우는 거 같다. 이번 프로젝트에선 내가 짠 코드가 하나도 없어서 스스로가 부끄러웠다. 그래서 이번 프로젝트를 계기로 나 혼자 저 게임 하나를 만들 수 있는 실력까지 만들어보고 싶다는 욕심이 생겼고, 앞으로 더 열심히 공부해야겠다는 생각이 들었다.