

# 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:두명의 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. 빙고 시 승자 출력 후 종료
  7. 모든 칸이 찼으면 종료

## 3.설계 및 구현

```
// 짝수 홀수로 누구 차례인지 판단함
int k = 0;
char currentUser = 'X';
while (true)
{
    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;
    }
}
```

2. 입력

아직 없음

3. 결과

-누구의 차례인지  
출력

출력후 while문 뒤

로 이동

#### 4. 설명

1. k=0으로 초기화를 시킨다음 인원수가 2명이므로 2로 나누어서 나머지로 순서를 판단함
2. 나머지는 0,1이므로 순서를 보기 좋게 하기위해 1를 각각 더해줌

```
//좌표 입력을 받음
cout << "(x,y)좌표를 입력하세요: ";
cin >> x >> y;
```

2.입력

currentUser가 X인자 O인지

#### 3. 결과

-사용자로부터 x,y좌표를 입력받음

```
//입력받은 좌표가 이미 돌이 차있는곳이거나 범위를 벗어날 경우 오류메세지 표시
if (x >= numCell || y >= numCell)
{
    cout << x << ", " << y << ": ";
    cout << "x와 y중 하나가 칸을 벗어납니다." << endl;
    continue;
}

if (board[x][y] != ' ')
{
    cout << x << ", " << y << ": 이미 돌이 차있습니다" << endl;
    continue;
}

//입력받은 좌표에 유저 돌 놓기
board[x][y] = currentUser;
```

입력

X=좌표x값

Y=좌표y값

numCell=가로/세로 칸 갯수

#### 결과

-칸을 놓을 수 없는 이유를 출력

-출력후 while문 초반으로 감

-칸을 놓을수있다면 보드판에 좌표에 유저의 돌을 저장시킴

## 설명

- 사용자가 입력한 좌표가 게임 판을 벗어나는지 if문으로 체크
- 사용자가 입력한 좌표에 돌이 이미 있는지 if문으로 체크
- 만약 오류가 없다면 board 배열에 인덱스를 활용해서 유저(O or X)의 돌을 저장 시킨다!

```
//게임판 표시시킴
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=보드판 가로 세로 크기
- board[][]=보드판 기록 저장
- k=실행 횟수

## 결과

- 보드판의 형식을 만들어줌 3\*3
- 보드판의 게임 진행 상황을 표시 시킴

-실행횟수 1회 증가 시킴

## 설명

-for문을 활용하여서 "--- | --- | --- |"를 가로에 출력하고 세로에는 " |"를 for문을 활용하여 출력하였는데 마지막 행 " |"를 표시하지 않도록 if조건을 만든 후 break문을 사용함. for문이 끝난후 실행횟수를 1증가 시켜서 다음 사람 턴으로 넘어갈 수 있게 한다.

```
//보드판에 빈칸이 없는지 확인
int checked = 0;
for (int i = 0; i < numCell; i++)
{
    for (int j = 0; j < numCell; j++) {
        if (board[i][j] == ' ') {
            checked++;
        }
    }
}
//checked가 0이라면 빈칸이 없다는 의미이므로 프로그램 종료
if (checked == 0)
{
    cout << "모든 칸이 다 찼습니다. 종료합니다" << endl;
    break;
}
```

## 입력

-checked=빈칸갯수 확인

-board[i][j]="O or X or 공백"

-numCell=보드판 크기

## 결과

-보드판에 빈칸이 없으면 다음 코드로 이동

,

-보드판에 빈칸이 있다면 종료하겠다는 문구 출력후 종료

## 설명

-이중 for문을 통해서 board 배열에 빈칸이 있는지 확인을 한다 그후 if문을 통해 빈칸이 있으면 while문을 탈출하여서 프로그램을 종료시킨다.

```
bool isWin = false;

//가로로 이기는 경우
for (int i = 0; i < numCell; i++) {
    if (board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser)
    {
        cout << "가로에 모두 돌이 놓였습니다! : ";
        isWin = true;
    }
    //세로로 이기는 경우
    if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser)
    {
        cout << "세로에 모두 돌이 놓였습니다! : ";
        isWin = true;
    }
}

//대각선이 꼭차서 이기는 경우
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;
}

// isWin는 기본적으로 false이지만 각 게임 조건에 맞으면 true되므로 if문을 만족시켜서 종료시킴
if (isWin == true) {
    cout << k % 2 + 1 << "번 유저(" << currentUser << ")의 승리입니다!" << endl;
    cout << "종료합니다" << endl;
    break;
}
```

## 입력

isWin=게임 승리

borad=게임판 저장 값

currentUser="X or O"

k=게임 진행 횟수

,

## 결과

가로로 이기는 경우, 세로로 이기는 경우 왼쪽 대각선으로 이기는 경우 오른쪽 대각선으로 이기는 경우에 관해서 각 케이스별로 분류 후 1or2번째 사용자가 이겼다는 문구 출력후 프로그램을 종료시킴

## 설명

가로 나 세로로 이기는 경우는 for문과 if문을 통해서 승리 조건을 판단함 대각선으로 이기는 경우에는 if문을 통해서 승리조건을 판단하는데 조건을 입력할 때 인덱스를 직접 접근함 또한 모든 if 문에는 isWin을 true로 바꿔줌 isWin이 true 라는 것은 여러 if문중 1개의 if문이라도 들어갔다는 의미 즉 어떤 승리조건을 만족시켰으므로 또 다시 if문을 통해 어떤유저가 이겼다고 출력 후 프로그램을 종료시킴

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

## 4.테스트

### 1.누구의 차례인지 출력

```
1번 유저(X)의 차례입니다 -> (x,y)좌표를 입력하세요: 0 0
```

## 2.좌표 입력 받기

```
C:\Users\User\source\repos\Project4\Project4\Debug\Project4.exe
```

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

## 3.입력 받은 좌표 유효성 체크

```
C:\Users\User\source\repos\Project4\Project4\Debug\Project4.exe
```

```
1번 유저(X)의 차례입니다 -> (x,y)좌표를 입력하세요: 0 3
0, 3: x와 y중 하나가 칸을 벗어납니다.
1번 유저(X)의 차례입니다 -> (x,y)좌표를 입력하세요: _
```

## 4&5.좌표에 O X 두기 & 현재 보드판 출력

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

## 6.빙고 승리시 출력후 종료

```
---|---|---
X  |X  |X
---|---|---
   |   |
---|---|---
   |0  |0
---|---|---
가로에 모두 돌이 놓였습니다!: 2번 유저(X)의 승리입니다!
종료합니다
```

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

```

---|---|---
0 |   |IX
---|---|---
   |IX |O
---|---|---
X |   |
---|---|---
오른쪽 위에서 왼쪽 아래 대각선으로 모두 돌이 놓였습니다! :2번 유저(X)의 승리입니다!
종료합니다

```

```

---|---|---
X |   |
---|---|---
   |IX |O
---|---|---
O |   |IX
---|---|---
왼쪽 위에서 오른쪽 아래 대각선으로 모두 돌이 놓였습니다! :2번 유저(X)의 승리입니다!
종료합니다

```

+

## 7.모든 칸이 찼으면 종료

```

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

```



## 8. 최종 실행 결과

```
Microsoft Visual Studio 디버그 콘솔
1번 유저(X)의 차례입니다 -> (x,y)좌표를 입력하세요: 0 3
0, 3: x와 y중 하나가 칸을 벗어납니다.
1번 유저(X)의 차례입니다 -> (x,y)좌표를 입력하세요: 0 0
---|---|---
X  |  | 
---|---|---
   |  | 
---|---|---
   |  | 
---|---|---
   |  | 
---|---|---
2번 유저(O)의 차례입니다 -> (x,y)좌표를 입력하세요: 0 1
---|---|---
X  |O  | 
---|---|---
   |  | 
---|---|---
   |  | 
---|---|---
1번 유저(X)의 차례입니다 -> (x,y)좌표를 입력하세요: 0 2
---|---|---
X  |O  |X
---|---|---
   |  | 
---|---|---
   |  | 
---|---|---
2번 유저(O)의 차례입니다 -> (x,y)좌표를 입력하세요: 1 0
---|---|---
X  |O  |X
---|---|---
```

```
---|---|---
1번 유저(X)의 차례입니다 -> (x,y)좌표를 입력하세요: 1 1
---|---|---
X  |O  |X
---|---|---
O  |X  | 
---|---|---
   |  | 
---|---|---
2번 유저(O)의 차례입니다 -> (x,y)좌표를 입력하세요: 1 2
---|---|---
X  |O  |X
---|---|---
O  |X  |O
---|---|---
   |  | 
---|---|---
1번 유저(X)의 차례입니다 -> (x,y)좌표를 입력하세요: 2 2
---|---|---
X  |O  |X
---|---|---
O  |X  |O
---|---|---
   |X  | 
---|---|---
왼쪽 위에서 오른쪽 아래 대각선으로 모두 둘이 놓였습니다! 2번 유저(X)의 승리입니다!
종료합니다
D:\#class0920\#x64\#Debug\#class0920.exe(프로세스 23884개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용
```

## 5.테스트

- 1.프로젝트 결과:2인용 Tic Tac Toe 게임을 만들었음
- 2.칸이 다 차면 종료가 아니라 무승부가 출력되도록 만드는게 사용자의 편의성이 높아질 것 같다.
- 3.느낀점: 단순한 코드들은 사용하는 간단한 문제들만 경험해봤는데 이번에 여러 가지 코드들이 합쳐지는 코딩이 생각보다 어렵지만 얻어가는게 많은 것 같다. 물론 어렵지만 실력이 좀 더 늘어나는 느낌이 있다.