

서론

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

요구사항

1. 사용자 요구사항: 두명의 사용자가 번갈아가며 O와 X를 놓기
2. 기능 요구사항: 누구의 차례인지 출력, 좌표 입력 받기, 입력 받은 좌표 유효성 체크, 좌표에 O / X 놓기, 현재 보드판 출력, 빙고 시 승자 출력 후 종료, 모든 칸이 찼으면 종료

설계 및 구현

1. 기능 별 구현 사항:

누구의 차례인지 출력

1. 코드블록 스크린샷

```
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

K = 게임의 차례가 지나갈때마다 0에서부터 1씩 증가하는 변수 k

currentUser = 현재 차례 사용자의 돌 모양

3. 결과

유저의 차례를 출력

현재 유저로 currentUser의 값을 'X'와 'O'로 설정

4. 설명

1. K의 나머지를 switch문의 조건으로 두어 0과 1이 반복되도록 해 유저의

차례를 설정

좌표 입력 받기

1. 코드블록 스크린샷

```
cout << "(x y) 좌표를 입력하세요: "; // 입력되는 x, y좌표 사이에 심표가 있다면 제대로 동작하지 않음
cin >> y >> x; // 표시될 x, y좌표에 맞춰서 입력 순서 변경
```

2. 입력

X= 좌표 x값

Y= 좌표 y값

3. 결과

좌표를 입력하라는 문구 출력

X, Y에 사용자의 입력 받기

4. 설명

1. Cin을 이용해 X, Y에 사용자의 입력 받기

입력 받은 좌표 유효성 체크

1. 코드블록 스크린샷

```
if(x >= numCell || y >= numCell
|| x < 0 || y < 0) { // 유효성 검사 조건 보강
    cout << y << ", " << x << ": ";
    cout << "x와 y들 중 하나가 칸을 벗어납니다." << endl;
    continue;
}

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

2. 입력

X= 좌표 x값

Y= 좌표 y값

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

좌표에 O / X 놓기

1. 코드블록 스크린샷

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

2. 입력

X= 좌표 x값

Y= 좌표 y값

currentUser = 현재 차례 사용자의 돌 모양

3. 결과

사용자가 입력한 좌표의 배열 위치에 사용자의 돌을 저장

4. 설명

1. 사용자가 입력한 좌표의 배열 위치에 사용자의 돌을 저장

현재 보드판 출력

1. 코드블록 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;
}

```

2. 입력

numCell= 가로/세로 칸 개수

3. 결과

현재 보드판의 상황을 출력

4. 설명

1. 중첩된 두 개의 for문으로 현재 보드판의 정보를 담고있는 배열 board의 내용을 출력
2. 보드판의 모양은 "---|---|---"과 배열의 원소 출력 후 원소 사이를 " |"로 채워 넣어 만든다

빙고 시 승자 출력 후 종료

1. 코드블록 스크린샷

```

//가로
for(int i = 0; i < numCell; i++){
    int l = 0;
    for(int j = 0; j < numCell; j++){
        if(board[i][j] == board[i][0]
            && board[i][0] != ' '){
            l++;
        }
        if(l == 3){
            cout << board[i][0] <<"의 승리입니다. 게임을 종료합니다" << endl;
            return 0;
        }
    }
}

//세로
for(int i = 0; i < numCell; i++){
    int l = 0;
    for(int j = 0; j < numCell; j++){
        if(board[j][i] == board[0][i]
            && board[0][i] != ' '){
            l++;
        }
        if(l == 3){
            cout << board[0][i] <<"의 승리입니다. 게임을 종료합니다" << endl;
            return 0;
        }
    }
}

//대각선
if(board[0][0] == board[1][1]
    && board[0][0] == board[2][2]
    && board[0][0] != ' '){
    cout << board[0][0] <<"의 승리입니다. 게임을 종료합니다" << endl;
    break;
}

if(board[2][0] == board[1][1]
    && board[2][0] == board[0][2]
    && board[2][0] != ' '){
    cout << board[2][0] <<"의 승리입니다. 게임을 종료합니다" << endl;
    break;
}

```

2. 입력

numCell= 가로/세로 칸 개수

3. 결과

가로, 세로, 대각선으로 한 유저의 돌이 놓이면 승리한 유저의 돌 모양과 함

게 승리 문구 출력

4. 설명

1. 가로, 세로의 경우는 해당하는 일직선의 보드판의 내용물을 그 일직선의 첫번째 원소와 비교하여 모두 동일한지, 아무것도 들어가지 않은 경우는 아닌지 if문을 이용해 확인. 원소가 같으면 전역변수 l이 1씩 증가하고 l이 3인 경우 승리 문구를 출력하고 return문으로 프로그램 종료
2. 대각선의 경우는 해당하는 좌표를 if문으로 확인

모든 칸이 찼으면 종료

1. 코드블록 스크린샷

```
// 7. 모든 칸이 찼으면 종료
if(k == numCell * numCell){
    cout << "모든 칸이 다 찼습니다. 게임을 종료합니다" << endl;
    break;
}
```

2. 입력

K = 게임의 차례가 지나갈때마다 0에서부터 1씩 증가하는 변수 k

numCell = 가로/세로 칸 개수

3. 결과

K가 numCell의 제곱이 될 경우에 알림 문구 출력과 함께 프로그램 종료

4. 설명

1. 모든 칸이 찼 경우의 유저들이 둘을 둔 횟수는 보드판의 크기와 같으므로 이 경우를 if문으로 확인

테스트

1. 기능 별 테스트 결과:

누구의 차례인지 출력

첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요:

두번째 유저(o)의 차례입니다 -> (x y) 좌표를 입력하세요: 0 1

좌표 입력 받기

두번째 유저(0)의 차례입니다 -> (x y) 좌표를 입력하세요: 0 1

입력 받은 좌표 유효성 체크

```
첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: -1 0
-1, 0: x와 y들 중 하나가 칸을 벗어납니다.
첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 0 -1
0, -1: x와 y들 중 하나가 칸을 벗어납니다.
첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 3 0
3, 0: x와 y들 중 하나가 칸을 벗어납니다.
첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 0 3
0, 3: x와 y들 중 하나가 칸을 벗어납니다.
첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: -1 -1
-1, -1: x와 y들 중 하나가 칸을 벗어납니다.
첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 3 3
3, 3: x와 y들 중 하나가 칸을 벗어납니다.
첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 0 0
0, 0: 이미 돌이 차있습니다.
```

좌표에 O / X 놓기

```
첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 0 0
---|---|---
X  |   |
---|---|---
   |   |
---|---|---
   |   |
---|---|---
두번째 유저(0)의 차례입니다 -> (x y) 좌표를 입력하세요: 0 1
---|---|---
X  |   |
---|---|---
O  |   |
---|---|---
   |   |
---|---|---
```

첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 1 0

---|---|---

X |X |

---|---|---

O | |

---|---|---

| |

---|---|---

두번째 유저(o)의 차례입니다 -> (x y) 좌표를 입력하세요: 2 0

---|---|---

X |X |O

---|---|---

O | |

---|---|---

| |

---|---|---

첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 1 1

---|---|---

X |X |O

---|---|---

O |X |

---|---|---

| |

---|---|---

두번째 유저(o)의 차례입니다 -> (x y) 좌표를 입력하세요: 2 1

---|---|---

X |X |O

---|---|---

O |X |O

---|---|---

| |

---|---|---

첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 0 2

---|---|---

X |X |O

---|---|---

O |X |O

---|---|---

X | |

---|---|---


```

두번째 유저(0)의 차례입니다 -> (x y) 좌표를 입력하세요: 1 2
---|---|---
X  |X  |O
---|---|---
O  |X  |O
---|---|---
X  |O  |
---|---|---
첫번째 유저(X)의 차례입니다 -> (x y) 좌표를 입력하세요: 2 2
---|---|---
X  |X  |O
---|---|---
O  |X  |O
---|---|---
X  |O  |X
---|---|---

```

현재 보드판 출력

```

---|---|---
X  |X  |O
---|---|---
O  |X  |O
---|---|---
X  |O  |X
---|---|---

```

빙고 시 승자 출력 후 종료

```

첫번째 유저(X)의 차례입니다 -> (x y) 좌표를 입력하세요: 2 0
---|---|---
X  |X  |X
---|---|---
O  |O  |
---|---|---
  |  |
---|---|---
X의 승리입니다. 게임을 종료합니다

```

```

두번째 유저(0)의 차례입니다 -> (x y) 좌표를 입력하세요: 2 1
---|---|---
X  |X  |
---|---|---
O  |O  |O
---|---|---
X  |  |
---|---|---
0의 승리입니다. 게임을 종료합니다

```

첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 2 2

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

```
0  |0  |
```

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

```
   |   |
```

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

```
x  |x  |x
```

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

x의 승리입니다. 게임을 종료합니다

첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 0 2

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

```
x  |0  |
```

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

```
x  |   |
```

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

```
x  |   |0
```

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

x의 승리입니다. 게임을 종료합니다

두번째 유저(o)의 차례입니다 -> (x y) 좌표를 입력하세요: 1 2

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

```
   |0  |x
```

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

```
x  |0  |
```

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

```
x  |0  |
```

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

o의 승리입니다. 게임을 종료합니다

두번째 유저(o)의 차례입니다 -> (x y) 좌표를 입력하세요: 2 2

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

```
x  |x  |0
```

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

```
x  |x  |0
```

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

```
   |0  |0
```

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

o의 승리입니다. 게임을 종료합니다

첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 2 2

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

```
x  |0  |0
```

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

```
   |x  |
```

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

```
   |   |x
```

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

x의 승리입니다. 게임을 종료합니다

```

두번째 유저(0)의 차례입니다 -> (x y) 좌표를 입력하세요: 0 2
---|---|---
X  |X  |O
---|---|---
X  |O  |
---|---|---
O  |   |
---|---|---
0의 승리입니다. 게임을 종료합니다

```

모든 칸이 찼으면 종료

```

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

```

2. 최종 테스트 스크린샷:

첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 1 0

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

두번째 유저(o)의 차례입니다 -> (x y) 좌표를 입력하세요: 1 1

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

첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 2 0

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

두번째 유저(o)의 차례입니다 -> (x y) 좌표를 입력하세요: 0 0

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

첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 2 2

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

두번째 유저(o)의 차례입니다 -> (x y) 좌표를 입력하세요: 0 1

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

```
첫번째 유저(x)의 차례입니다 -> (x y) 좌표를 입력하세요: 2 1
---|---|---
0  |X  |X
---|---|---
0  |0  |X
---|---|---
   |   |X
---|---|---
x의 승리입니다. 게임을 종료합니다
```

결과 및 결론

1. 프로젝트 결과

Tic Tac Toe 게임을 완성했음

2. 느낀 점

조건을 설정하는 것이 항상 어려운 것 같음