

Tic Tac Toe week6 보고서

175989 백성준

1. 서론

1-1.프로젝트 목적 및 배경

지금까지 배운 내용에 대한 실습, 보고서 작성법에 대한 실습

1-2. 목표

Tic Tac Toe 게임의 구현, 보고서 작성 해보기

2. 요구사항

2-1. 사용자 요구사항 :

두 명이 할 수 있는 tic tac toe 게임 만들기.

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 찼으면 종료

3. 설계 및 구현

3-1. 기능 별 구현 사항

1) 누구의 차례인지 출력

1.

```
//1. 차례를 출력
switch (k % 2)//k를 2로 나눈값이
{
case 0://k가 짝수일 때
    cout << k % 2 + 1 << "번 유저(x)의 차례입니다 -> ";
    currentUser = 'X';//x로 돌 모양을 바꿈
    break;
case 1://k가 홀수일 때
    currentUser = 'O';//O로 돌 모양을 바꿈
    cout << k % 2 + 1 << "번 유저(o)의 차례입니다 -> ";
}
```

```
k++; //차례를 확인을 위해서 1을 더해줌
```

2.입력

- k=한번 돌을 놓을 때 마다 1증가 ,0부터 시작

-currentUser=현재 사용자의 돌 모양 x or y

3. 결과

-x의 차례 또는 y의 차례임을 알려주는 메시지 출력

4. 설명

K가 짝수이면 x의 차례인 메시지를 띄우고 반대면 y의 차례임을 메시지로 출력

2) 좌표 입력 받기

1.

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

2.입력

-없음

3.결과

-입력을 받은 값이 x와 y에 저장됨

4.설명

-x와 y값을 초기화 함. 이 값은 각각 x,y 좌표가 됨.

3) 입력 받은 좌표 유효성 체크

```
//3. 받은 좌표 값이 유효성을 벗어났을 때
if (x >= numCell || y >= numCell)//x나 y가 칸을 벗어나서 입력받으면
{
    cout << x << ", " << y << ": ";
    cout << "x 와 y 둘 중 하나가 칸을 벗어납니다." << endl;
    continue;//루프 맨 앞으로
}
if (board[x][y] != ' ')//x나 y가 빈칸(' ')이 아닐 때, 이미 돌이 놓였을 때
{
    cout << x << ", " << y << ": 이미 돌이 차있습니다." << endl;
    continue;//루프 맨 앞으로
}
1. }
```

2.입력

-x=좌표 x의 값

-y=좌표 y의 값

3. 결과

-x나 y가 칸의 수를 초과했음을 알리는 메시지 출력

-또는 지정 좌표에 이미 돌이 놓아졌음을 알리는 메시지 출

력

4. 설명

-좌표의 값이 칸의 수를 넘었는지를 확인하고 넘으면 반복문의 초기로 돌아감

-또는 중복된 자리의 좌표 값이 불러지면 메시지 출력하고 반복문의 초기로 돌아감

4) 좌표에 O/X 놓기

1.

```
//4. 돌을 놓기 위해서 배열을 초기화  
board[x][y] = currentUser;
```

2. 입력

-x=좌표 x의 값

-y=좌표y의 값

-currentUser=현재 돌을 쥔 유저의 돌 모양

3. 결과

-x와 y의 값에 해당하는 배열을 현재 유저의 돌 모양으로 초기화됨

4. 설명

-입력한 좌표 값에 해당하는 배열을 돌 모양으로 초기화하여 돌을 놓는 것을 구현

5) 현재 보드판 출력

```
//5. 게임의 틀이 되는 부분 출력 및 돌도 함께 출력
for (int i = 0; i < numCell; i++)//칸 수 만큼 반복
{
    cout << "---|---|---" << endl;//직선 칸 출력
    for (int j = 0; j < numCell; j++)//각 줄에 왔을 때 칸 수만큼 반복
    {
        cout << board[i][j];//돌 모양을 출력
        if (j == numCell - 1)//칸의 끝에 오면 탈출하기 바로 위 for문을 탈출
        {
            break;
        }
        cout << " |";// 중간 |를 출력
    }
    cout << endl;//다음 칸으로 넘어감
}
1. cout << "---|---|---" << endl;//맨 아랫 칸 출력
```

2. 입력

-board배열의 모든 각 배열

-numCell=게임 판의 수

-i, j =for문에서 배열의 호출을 위한 변수들

3. 결과

---|---|---이것이 칸의 수 만큼 출력됨. 그 때 이것이 하나 출력될 때, 각 칸에 맞는 board배열이 출력되고 |가 출력됨. 칸의 마지막에 다다라서는 break를 통해 상위 for문을 탈출함. 이 과정의 반복(가장 상위의 for문이 끝날 때까지).

- 마지막에 게임판을 닫기 위한 아래 칸 출력

4. 설명

-게임의 판을 만들고 각 위치에 맞는 배열을 출력.

-공란이 들어간 경우는 돌이 놓이지 않은 상황

6) 빙고 시 승자 출력 후 종료

```
//6. 빙고 시 승자 출력 후 종료
bool isWin = false;
//6. 1. 직선인 경우 체크
for (int i = 0; i < numCell; i++)//각 가로줄이나 세로줄 이 모두 같은지를 확인
{
    if (board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser)
    {
        cout << "가로에 모두 돌이 놓였습니다! ";
        isWin = true;
    }
    if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser)
    {
        cout << "세로에 모두 돌이 놓였습니다! ";
        isWin = true;
    }
}
//6.2. 대각선 체크
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;
}
//종료를 위한 코드
if (isWin == true)//종료 조건인 isWin이 트루가되면 종료
{
    cout << k % 2 + 1 << "번 유저(" << currentUser << ")의 승리입니다!" << endl;
    cout << "종료합니다" << endl;
    break;
}
```

1.

2. 입력

-isWin=게임판을 종료 할지를 결정하기 위해 사용하는 변수

-board 배열=각 돌이 놓이는 위치를 나타내는 변수

3.결과

-가로 또는 세로가 모두 놓였음을 출력하고 게임을 종료

-또는 대각선으로 돌이 놓여서 이겼음을 출력하고 게임

종료

4. 설명

-가로 세로의 한 열에 해당하는 것은 for에 의해 아래 코드들이 실행될 때마다 각 열이 모두 같은 돌인지 확인. 맞다면 게임종료를 위한 값으로 초기화(true)후 게임종료

-대각선에 해당하는 모든 배열이 같으면 게임종료를 위한 값을 전달(true)후 게임종료

7) 모든 칸이 찼으면 종료

```
//7. 모든 칸 다 찼는지 확인
int checked = 0;
for (int i = 0; i < numCell; i++)
{
    for (int j = 0; j < numCell; j++)
    {
        if (board[i][j] == ' ')//공란일 경우 checked를 올림
        {
            checked++;
        }
    }
}

if (checked == 0)//모든 칸에 공란이 없는경우
{
    cout << "모든 칸이 다 찼습니다. 종료합니다" << endl;
    break;
}
```

1.

2. 입력 -l,j= for문에서 사용하여 각 배열들을 불러오기 위해 사용

-checked= 공란이 몇 개 있는지를 확인하기 위해 사용

3. 결과

-돌이 모두 차 있어서 checked가 선언초기화 이후 한번

도 초기화 되지 않았으면 게임종료

4. 설명

-돌의위치인 배열이 플레이어의 돌모양으로 초기화 되지 않았다면 checked가 1씩 올라가게 되고 최종적으로 모든 배열의 칸이 돌 모양으로 초기화되면 while 무한루프를 탈출.

4. 테스트

4-1.기능별 테스트

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

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

3) 입력 받은 좌표 유효성 체크

3,3: x 와 y 둘 중 하나가 칸을 벗어납니다.

0, 0: 이미 돌이 차있습니다.

4) 좌표에 O/X 놓기

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

	X	

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

O		
	X	

5) 현재 보드판 출력

	X	

O		
	X	

6) 빙고 시 승자 출력 후 종료

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

O	X	O
	X	
	X	

세로에 모두 돌이 놓였습니다!: 1번 유저(X)의 승리입니다!
종료합니다

1번 유저(X)의 차례입니다 -> (x,y) 좌표를 입력하세요: 0 2

X	X	X
O	O	

가로에 모두 돌이 놓였습니다!: 1번 유저(X)의 승리입니다!
종료합니다

```

1번 유저(X)의 차례입니다 -> (x,y) 좌표를 입력하세요: 2 0
---|---|---
0 | 0 | X
---|---|---
  | X | 
---|---|---
X |   | 
---|---|---
오른쪽 위에서 왼쪽 아래 대각선으로 모두 돌이 놓였습니다!: 1번 유저(X)의 승리입니다!
종료합니다
1번 유저(X)의 차례입니다 -> (x,y) 좌표를 입력하세요: 2 2
---|---|---
X | 0 | 0
---|---|---
  | X | 
---|---|---
  |   | X
---|---|---
왼쪽 위에서 오른쪽 아래 대각선으로 모두 돌이 놓였습니다!: 1번 유저(X)의 승리입니다!
종료합니다

```

모든 칸이 찼으면 종료

```

1번 유저(X)의 차례입니다 -> (x,y) 좌표를 입력하세요: 2 2
---|---|---
0 | X | X
---|---|---
X | X | 0
---|---|---
0 | 0 | X
---|---|---
모든 칸이 다 찼습니다. 종료합니다

```

4-2. 최종 테스트 스크린샷

1)모든 칸이 찬 경우

```

1번 유저(X)의 차례입니다 -> (x,y) 좌표를 입력하세요: 2 2
---|---|---
0 | X | X
---|---|---
X | X | 0
---|---|---
0 | 0 | X
---|---|---
모든 칸이 다 찼습니다. 종료합니다

```

2)가로 빙고로 종료된 경우

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

3)세로 빙고로 종료된 경우

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

4)대각선(좌하>우상) 빙고로 종료된 경우

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

5)대각선(좌상>우하) 빙고로 종료된 경우

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