

목차

1.서론

- 프로젝트 목적 및 배경
- 목표 : Tic Tac Toe 게임 구현

2.요구사항

- 사용자 요구사항
- 기능 요구사항

3.설계 및 구현

- 기능 별 구현 사항 (요구사항 별 코드)

4.테스트

- 기능 별 테스트 결과 (요구사항 별 스크린 샷)
- 최종 테스트 스크린 샷 (프로그램 전체 동작 스크린 샷)

5.결과 및 결론

- 프로젝트 결과
- 느낀 점

1.서론

-프로젝트 목적 및 배경

이번 시간 이전까지 수업 시간에 배웠던 c++의 프로그램 기본 구조, 자료 형과 변수(선언, 초기화), 상수, 출력함수 cout, 입력함수 cin, 연산자와 표현식, 조건문(if-else, switch 문), 반복문 (for 문, while 문, do-while 문), 배열, 2차원 배열 등을 배웠고 이 내용에 대한 실습을 위해 진행하게 되었다.

-목표

2명이서 진행하는 Tic Tac Toe 게임 구현

2.요구 사항

-사용자 요구사항

: 두 명의 사용자가 번갈아가며 O와 X를 놓기

-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

-제약 조건

: 보드 판은 2차원 배열 사용

3. 설계 및 구현

1)누구의 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;  
}
```

-입력

*변수 k 선언 및 초기화

*변수 currentUser 선언 및 초기화

*while문을 통해 반복

-결과

* “ ” 번 유저(X)의 차례입니다 -> 출력

*“ ” 번 유저(O)의 차례입니다 -> 출력

-설명

*처음 k가 0으로 선언 되었기 때문에 k%2 는 0에 해당하는 case로 가서 수행

2)좌표 입력 받기

```
cout << "(x,y) 좌표를 입력하세요:";  
cin >> x >> y;
```

-입력

*몇 번 유저의 차례인지 나오기만 한다

-결과

*입력된 값이 x.y가 된다.

-설명

*입력 받아서 각각의 변수에 들어간다.

3)입력 받은 좌표 유효성 체크

```
if (x >= numCell || y >= numCell) {  
    cout << x << ", " << y << ": ";  
    cout << "x와 y 둘 중 하나가 칸을 벗어납니다." << endl;  
    continue;  
}  
  
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

- *사용자가 입력한 좌표가 게임 판을 벗어나는 지 if로 체크
- *사용자가 입력한 좌표에 돌이 이미 있는지 if로 체크

4)좌표에 O / X 놓기

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

-입력

- * x=좌표 x값
- * y=좌표 y값

-결과

- *해당되는 차례의 돌이 놓여진다.

-설명

- *해당되는 배열에 알맞은 차례의 돌 x나 o를 변수에 입력

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

* x=좌표 x값

* y=좌표 y값

-결과

* 3-3 보드 판이 출력된다.

-설명

*x,y값이 입력되면 그 차례에 해당하는 돌과 함께 보드 판이 출력된다.

6)빙고 시 승자 출력 후 종료

```
bool isWin = false;

//현재 유저의 돌을 놓은 칸이 가로 or 세로가 같을 경우를 찾을 경우
//현재 유저의 돌을 놓은 칸이 대각으로 같은 경우를 조건문을 통해 찾을 경우
//위의 두가지 경우를 만족하면 빙고로 어떤 경우에 누가 승리인지 출력한다.
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;
    }
    if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser) {
        cout << "세로에 모두 돌이 놓였습니다! ";
        isWin = true;
    }
    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

* x=좌표 x값

* y=좌표 y값

-결과

*가로, 세로, 대각선 중에 어느 방식으로 돌이 놓여져서 빙고인지 출력

*누구의 승리인지 출력

-설명

*돌이 놓여졌을 때 빙고일 경우 어느 방식과 누가 이겼는지 출력된다.

*특정 조건이 만족 될 때까지 반복되면서 해당하는 조건문에 만족하는 경우 그에 해당하는 문장을 출력

*그 차례가 누구였는지 확인하여 누구의 승리인지 출력

7)모든 칸이 찼으면 종료

```
int full = 0;
for (int i = 0; i < numCell; i++) {
    for (int j = 0; j < numCell; j++) {
        if (board[i][j] == ' ') {
            full++;
        }
    }
}

if (full == 0) {
    cout << "모든 칸이 다 찼습니다. 종료합니다" << endl;
    break;
}
```

-입력

* x=좌표 x값

* y=좌표 y값

-결과

*마지막 한 칸에 돌이 놓이며 종료 이유를 출력

-설명

*i,j 값이 반복되면서 변수 full 에 1씩 값이 추가된다.

*과정이 반복되다가 1이 추가되지 않는 경우 즉, 0에 있을 경우 해당 문장 출력

4.테스트

1)누구의 차례인지 출력

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

2)좌표 입력 받기

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

3)입력 받은 좌표 유효성 체크

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

4)좌표에 O / X 놓기

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

		X

5)현재 보드판 출력

		X

6)빙고 시 승자 출력 후 종료

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

7)모든 칸이 찼으면 종료

모든 칸이 다 찼습니다. 종료합니다

****최종 테스트 스크린 샷**

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

5.결과 및 결론

1)프로젝트 결과 :tic tac toe 게임을 만듦

2)느낀 점 : 수업 때 배운 여러 내용들을 한 프로젝트에 사용하는 경험이어서 좋았음