

Tic Tac Toe 게임 프로젝트

소프트웨어공학과 214953 기도현

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: 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

3. 설계 및 구현

1. 기능별 구현 사항:

```

while(boo){
//1. 누구 차례인지 출력
switch(k%2){
    case 0:
        cout<<"첫번째 유저(x)의 차례입니다 ->";
        currentUser='x';
        break;
    case 1:"두번째 유저(o)의 차례입니다->";
        currentUser='o';
        break;
}
}

```

① 누구의 차례인지 출력

1. 입력:

- k= 루프가 돌 때마다 1씩 더해지는 값
- boo= 게임이 끝났을 때 0으로 바뀌어 while문을 컨트롤하는 변수
-

2. 결과:

- 누구의 차례인지 출력
- 누구의 차례인지에 따라currentUser를 O or X로 설정
- 좌표 입력하는 코드로 이동
-

3. 설명: k가 루프를 돌 때마다 1씩 증가함, 그러므로 2로 나눠 나머지를 보면 0,1이 반복되어 나옴, 그 값을 바탕으로 switch case문을 통해

누구의 차례인지 확정

```
//2. 좌표 입력받기
```

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

② 좌표 입력 받기

1. 입력:

2.

3. 결과:

- x,y에 x좌표값과 y좌표값 대입
- 입력받은 좌표의 유효성을 체크하는 코드로 이동
-

4. 설명: cin을 통해 x,y에 좌표 x,y값을 저장

```
//3. 입력받은 좌표의 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;  
}
```

③ 입력 받은 좌표 유효성 체크

1. 입력: numCell=가로/세로의 칸수

x,y=x,y좌표의 값

2. 결과:

- 돌을 놓을 수 없는 이유 출력
- 출력 후 while문 초반으로 이동

3. 설명:

- 사용자가 입력한 좌표가 게임판을 벗어나는지 if로 체크
- 사용자가 입력한 좌표에 이미 돌이 있는지 if로 체크

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

④ 좌표에 O / X 놓기

1. 입력: currentUser=차례에 따라 O or X

x,y= 좌표값 x,y

2. 결과:

- 칸에 돌 놓기
- 보드판 출력하는 코드로 이동
-

3. 설명:

- 2차원 배열을 이용해 만든 게임판에 currentUser에 해당하는 돌을 놓음

```
//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;
```

⑤ 현재 보드판 출력

1. 입력:

- numCell= 게임판 가로/세로의 크기

2. 결과:

- 현재까지 놓여진 돌을 포함한 보드판 출력
- 승자 확인 출력하는 코드로 이동

3. 설명:

- 2중 반복문을 활용하여 돌이 놓여진 바둑판을 출력

```

//6. 빙고시 승자 출력
//6-1. 가로 빙고 성공
for(int i=0;i<numCell;i++){
    if(board[i][0]!=' ' && board[i][0]==board[i][1]&&board[i][1]==board[i][2]){
        if(board[i][0]=='X'){
            cout<<"유저(x)가 승리하셨습니다."<<endl;
            boo=0;
            break;
        }
        else if(board[i][0]=='O'){
            cout<<"유저(o)가 승리하셨습니다."<<endl;
            boo=0;
            break;
        }
    }
}

//6-2 세로 빙고 성공
for(int i=0;i<numCell;i++){
    if(board[0][i]!=' ' && board[0][i]==board[1][i]&&board[1][i]==board[2][i]){
        if(board[0][i]=='X'){
            cout<<"유저(x)가 승리하셨습니다."<<endl;
            boo=0;
            break;
        }
        else if(board[0][i]=='O'){
            cout<<"유저(o)가 승리하셨습니다."<<endl;
            boo=0;
            break;
        }
    }
}

```

```

//6-3 대각선 빙고 성공
if((board[0][0]!=' ' && board[0][0]==board[1][1]&&board[1][1]==board[2][2]) || (board[0][2]!=' ' && board[0][2]==board[1][1]&&board[1][1]==board[2][0])){
    if(board[1][1]=='X'){
        cout<<"유저(x)가 승리하셨습니다."<<endl;
        boo=0;
        break;
    }
    else if(board[1][1]=='O'){
        cout<<"유저(o)가 승리하셨습니다."<<endl;
        boo=0;
        break;
    }
}

```

⑥ 빙고 시 승자 출력 후 종료

1. 입력:

Board[][]= 해당 칸에 있는 문자

Boo= while문을 빠져가기 위한 조건문

numCell= 바둑판의 가로/세로

2. 결과:

- 누군가 이기는 조건을 만족하면 게임을 끝내고 승리자를 출력
- 모든 칸이 찬 경우 게임을 끝내는 코드로 이동

3. 설명:

-가로 or 세로 or 대각선이 ' '이 아니고 같은 문자이면 if문을 실행

시켜 승리자를 출력하고 boo 값을 바꿔 while문 빠져나오기

```
// 7. 모든 칸이 찼으면 종료
int check = 0; // 빈칸이 하나라도 있으면 1
for(x=0; x<numCell; x++){
    for(y=0; y<numCell; y++){
        if (board[x][y] != ' '){
            check=1;
            break;
        }
    }
}
// 빈칸이 없어 check 값이 0이 나오면 종료
if(check==0){
    cout<<"모든 칸이 찼으므로 게임을 종료하겠습니다."<<endl;
    boo=0;
}
```

⑦ 모든 칸이 찼으면 종료

1. 입력:

Check= 빈칸이 있는지 확인하기 위한 변수

numCell= 바둑판의 가로/세로

2. 결과:

- 바둑판에 빈칸이 없다면 게임을 종료

3. 설명:

-check변수를 선언해 0으로 초기화하고 이중 for문을 통해 배열들을 확인한 후 ' '이 있으면 check 변수를 1로 저장하고 break, 반복문이 끝나고 if문을 통해 check가 0이면 ' '이 없는 걸로 판단하고 게임을 끝냄

4.테스트

4-1 기능 별 테스트

① 누구의 차례인지 출력

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

② 좌표 입력 받기

```
(x,y)좌표를 입력하세요: 
```

③ 입력 받은 좌표 유효성 체크

```
(x,y)좌표를 입력하세요: 0 0  
0, 0: 이미 돌이 차있습니다.  
(x,y)좌표를 입력하세요: 0 4  
0, 4, x와 y 둘 중 하나가 칸을 벗어납니다.  
(x,y)좌표를 입력하세요: 
```


④ 좌표에 O / X 놓기

```
(x,y)좌표를 입력하세요: 0 1
---|---|---
X  |O  |
---|---|---
---|---|---
---|---|---
---|---|---
```

⑤ 현재 보드판 출력

```
(x,y)좌표를 입력하세요: 0 1
---|---|---
X  |O  |
---|---|---
---|---|---
---|---|---
```

⑥ 빙고 시 승자 출력 후 종료

```
---|---|---
X  |O  |
---|---|---
X  |O  |
---|---|---
X  |   |
---|---|---
유저(x)가 승리하셨습니다
```

```
---|---|---
X  |X  |X
---|---|---
O  |O  |
---|---|---
---|---|---
유저(x)가 승리하셨습니다
PS C:\Users\user2\Desktop\CPP2409>
```

```

---|---|---
X  |O  |O
---|---|---
   |X  |
---|---|---
   |   |X
---|---|---
유저(x)가 승리하셨습니다

```

⑦ 모든 칸이 찼으면 종료

```

---|---|---
X  |O  |X
---|---|---
O  |O  |X
---|---|---
X  |X  |O
---|---|---
모든 칸이 찼으므로 게임을 종료하겠습니다.

```

4-2 최종 테스트 스크린샷 (프로그램 전체 동작 스크린샷)

```

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

```

5. 결과 및 결론

5-1. 프로젝트 결과: Tic Tac Toe 게임을 만들었음

5-2. 느낀 점: 지금까지 배운 것을 가지고 그럴듯한 프로그램이 만들어져 놀랐다. 처음에 긴 코드를 보고 당황했지만 하나하나 뜯어보니 다 수업시간에 배운 내용이었고 해석할 수 있었다. 코드를 짜는 과정에서는 이전에 실습했던 practice들이 많이 도움됐던 거 같다. 모두 완성된 코드를 보고 뿌듯함도 들었지만 분명 더 효율적인 코드가 있을 거란 생각이 들어(특히 플레이어가 이긴 상황을 판단하는 if 문) 더 노력해야겠다는 생각을 했다.