

tic_tac_toe 보고서

213235 소프트웨어공학과 이재륜

1. 서론

-프로젝트 목적 및 배경 : 4주차까지 배운 내용을 종합적으로 사용하여 tic_tac_toe 게임을 만들어보는 토이 프로젝트를 진행하여 스스로 코드를 사용하는 사고력을 높이기 위해 진행함.

-목표 : Tic Tac Toe 게임의 추가 기능을 구현한다.

2. 요구사항

-사용자 요구사항 : 두 명의 사용자가 번갈아가며 O와 X를 놓고 빙고시 승자 출력 후 종료하고 모든 칸이 찼으면 종료한다. 이후 한 명의 사용자가 추가되어 세 명의 사용자가 게임을 진행할 경우도 구현한다.

-기능 요구사항 : 1. 누구의 차례인지 출력, 2. 좌표 입력받기, 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

-기능별 구현사항

1. 누구의 차례인지 출력

```
//1. 누구 차례인지 출력
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=누구 차례인지 체크하기 위한 변수

3. 결과

- 현재 어떤 유저 차례인지 출력

4. 설명

- switch문의 조건을 사용자가 입력한 k값을 전체 유저수로 나눈 것으로 정의하여 유저가 차례대로 게임을 반복할 수 있도록 하였음

1. 좌표 입력 받기

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

2. 입력

- x=좌표 x값

- y=좌표 y값

3. 결과

- x,y 좌표를 입력하라는 안내문 출력

4. 설명

- 사용자가 입력한 x y좌표를 입력받음

1. 입력받은 좌표의 유효성 체크

```
//3. 입력받은 좌표의 유효성 체크  
if (x>=numCell || y>=numCell){  
    cout << x << ", " << y << ": ";  
    cout << " x 와 y 둘 중 하나가 칸을 벗어납니다." << endl;  
    continue;  
}  
if (board[x][y] != ' '){  
    cout << x << ", " << y << ": 이미 돌이 차있습니다." << endl;  
    continue;  
}
```

2. 입력

- x=좌표 x값

- y=좌표 y값

- numCell=가로/세로 칸 개수

3. 결과

- 이미 차있거나 칸을 벗어난 x,y좌표 및 안내문이 출력된다

4. 설명

- 칸을 벗어난 x,y 좌표와 안내문을 출력하고 continue를 통해 다시 입력 받음
- 이미 차있는 x,y 좌표와 안내문을 출력하고 continue를 통해 다시 입력 받음

1. 입력받은 좌표에 현재 유저의 돌 놓기

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

2. 입력

- x=좌표 x값
- y=좌표 y값
- currentUser=현재유저

3. 결과

- 현재유저가 입력받은 좌표에 저장된다

4. 설명

- 현재유저 currentUser의 값이 입력받은 좌표인 board[x][y]에 저장된다

1. 현재 보드 판 출력

```
//5. 현재 보드 판 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++;
```

2. 입력

- numCell=가로/세로 칸 개수

3. 결과

- 현재 보드 판이 전부 출력된다
- k가 1증가 된다

4. 설명

- for문을 활용한 이중 반복문과 2차원 배열을 통해 현재 보드 판이 전부 출력된다
- k를 1 증가시켜 다음 차례의 유저가 게임을 진행하도록 한다
- if문을 통하여 j가 numCell-1이면 |를 출력하는걸 멈춰 예쁜모양의 틱택토가 완성 되도록 돕는다

1. 빙고시 승자 출력 후 종료

```
//6. 빙고시 승자 출력 후 종료
bool bingo = false;

//가로 체크
for (int i=0; i<numCell; i++){
    if (board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser){
        bingo = true;
    }
}
//세로체크
for (int i=0; i<numCell; i++){
    if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser){
        bingo = true;
    }
}
//대각선체크
if (board[0][0] == currentUser && board[1][1] == currentUser && board[2][2] == currentUser){
    bingo = true;
}
if (board[0][2] == currentUser && board[1][1] == currentUser && board[2][0] == currentUser){
    bingo = true;
}
//승자가 나오면 게임 종료
if (bingo){
    cout << currentUser << "유저가 승리하였습니다." << endl;
    break;
}
```

2. 입력

- numCell=가로/세로 칸 개수
- bingo = 빙고를 채웠는지 안채웠는지의 유무

3. 결과

- 가로, 세로, 대각선을 체크하여 빙고이면 bingo값을 true로 완성한다.

4. 설명

-

4. 테스트

- 기능별 테스트 결과
- 최종 테스트 스크린샷

5. 결과 및 결론

- 프로젝트 결과
- 느낀 점