

C++ 프로그래밍 및 실습 프로젝트 보고서

Tic Tac Toe project

214935 이승진

2024/10/02

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
- 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

1. 기능 별 구현사항

- ① 누구의 차례인지 출력

1. 코드블록 스크린샷

```
while(true){
    switch(k % 2){
        case 0:
            cout << "첫 번째 유저(x)의 차례입니다 -> ";
            currentUser = 'X';
            break;
        case 1:
            cout << "두 번째 유저(o)의 차례입니다 -> ";
            currentUser = 'O';
            break;
    }
}
```

2. 입력

- A. K값

3. 결과

- A. 누구의 차례인지 출력한다.
- B. $K \% 2 == 0$: 첫 번째 유저(X)의 차례
- C. $K \% 2 == 1$: 두 번째 유저(O)의 차례

4. 설명

K값을 2로 나눈 나머지를 기준으로 유저의 차례를 결정한다.

- ② 좌표 입력 받기

1. 코드블록 스크린샷

```
cout << "(x,y) 좌표를 입력하세요(최대범위 x:2, y:2): "; //제한된 입력범위 안내
cin >> y >> x; //x좌표부터 입력되도록 순서 변경
```

2. 입력

A. x, y 좌표를 입력받는다.

3. 결과

A. 사용자가 입력한 좌표를 프로그램이 인식한다.

4. 설명

x가 열, y가 행을 나타내므로, y부터 입력받도록 하여 일반적인 좌표처럼 입력받을 수 있도록 입력순서 할당

③ 입력 받은 좌표 유효성 체크

1. 코드블록 스크린샷

```
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 차 있습니다." << endl;
    continue;
}
```

2. 입력

A. x = x좌표 x 값

B. y = y좌표 y 값

C. numCell = 가로/세로 칸 개수

3. 결과

A. 칸을 놓을 수 없는 이유를 출력

B. 출력 후 while문 초반으로 이동

4. 설명

A. 사용자가 입력한 좌표가 게임 판을 벗어나는지 if로 체크

B. 사용자가 입력한 좌표에 돌이 이미 있는지 if로 체크

④ 좌표에 O / X 놓기

1. 코드블록 스크린샷

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

2. 입력

A. 입력받은 유효한 좌표

3. 결과

A. 입력된 좌표에 현재 차례 사용자의 돌(O or X)을 놓는다.

4. 설명

currentUser는 이전에 결정된 유저의 차례에 따라 설정된다.

⑤ 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;
}
cout << "---|---|---" << endl;
k++;
```

2. 입력

A. 사용자들이 입력한 값들이 반영 된 보드의 상태

3. 결과

A. 현재 상태의 게임 보드 출력

4. 설명

행을 구분하기 위해 ---|---|---와 같은 구분선을 출력한다.

각 열의 칸 사이에 구분선(|)을 출력하고, 이를 통해 현재까지의 게임 진행 상황을 확인 할 수 있다.

⑥ 빙고 시 승자 출력 후 종료

1. 코드블록 스크린샷

```
//승자 체크하기
bool isWin = false;
//가로/세로들 체크하기
for (int i = 0; i < numCell; i++){
    if(board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser){
        cout << "가로에 모두 들어 놓았습니다! ";
        isWin = true;
    }
    if(board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser){
        cout << "세로에 모두 들어 놓았습니다! ";
        isWin = true;
    }
}
//대각선들 체크하기
if (board[0][0] == currentUser && board[1][1] == currentUser && board [2][2] == currentUser){
    cout << "왼쪽 위에서 오른쪽 아래 대각선으로 모두 들어 놓았습니다! ";
    isWin = true;
}
if (board[0][2] == currentUser && board[1][1] == currentUser && board [2][0] == currentUser){
    cout << "오른쪽 위에서 왼쪽쪽 아래 대각선으로 모두 들어 놓았습니다! ";
    isWin = true;
}

if (isWin == true){
    cout << k % 2 + 1 << "번 유저(" << currentUser << ")의 승리입니다!" << endl;
    cout << "종료합니다." << endl;
    break;
}
```

2. 입력

A. isWin = false값

B. 현재 보드의 상태

C. 현재 게임을 진행중인 유저

3. 결과

가로, 세로, 대각선으로 같은 돌이 3개 연속으로 놓였는지 체크하고, 승리 시 해당 유저를 출력하고 게임 종료

4. 설명

조건을 만족할 시 isWin을 true로 설정하고 승리 메시지를 출력 한 후 break로 게임을 종료한다.

7. 모든 칸이 찼으면 종료

1. 코드블록 스크린샷

```
int checked = 0;
for (int i = 0; i < numCell; i++){
    for (int j = 0; j < numCell; j++){
        if (board[i][j] == ' '){
            checked++;
        }
    }
}
if (checked == 0){
    cout << "모든 칸이 다 찹습니다. 종료합니다." << endl;
}
```

2. 입력

- A. 초기 checked 값 = 0
- B. 현재 보드의 상태

3. 결과

현재 보드의 모든 칸이 다 찼는지 확인하고 비어있는 칸이 없으면 "모든 칸이 다 찹습니다. 종료합니다." 라는 메시지를 출력하고 게임을 종료한다.

4. 설명

보드의 모든 칸을 순회하며 빈 칸이 있는지 확인하고, 빈 칸이 하나라도 있을 경우, 게임을 계속 진행하며 빈 칸이 없다면, 게임을 종료한다.

4. 테스트

1. 기능 별 테스트 결과

1. 누구의 차례인지 출력

첫 번째 유저 (x)의 차례입니다 -> (x,y) 좌표를 입력하세요 (최대범위 x:2, y:2):

2. 좌표 입력 받기

```
첫 번째 유저 (x)의 차례입니다 -> (x,y) 좌표를 입력하세요 (최대범위 x:2, y:2): 1
1
```

3. 입력 받은 좌표 유효성 체크

```
첫 번째 유저 (x)의 차례입니다 -> (x,y) 좌표를 입력하세요 (최대범위 x:2, y:2): 4
4
4, 4: x와 y 둘 중 하나가 칸을 벗어납니다 .
```

4. 좌표에 O/X 놓기

5. 현재 보드판 출력

```
---|---|---
|   |   |
---|---|---
|X  |   |
---|---|---
|   |   |
---|---|---
```

6. 빙고 시 승자 출력 후 종료

```
---|---|---
|X  |   |
---|---|---
O  |X  |
---|---|---
|X  |O  |
---|---|---
세로에 모두 둘이 놓였습니다!: 2번 유저 (X)의 승리입니다!
종료합니다 .
```

```
---|---|---
O  |X  |X
---|---|---
O  |X  |
---|---|---
X  |   |O
---|---|---
오른쪽 위에서 왼쪽쪽 아래 대각선으로 모두 둘이 놓였습니다!: 2번 유저 (X)의 승리입
니다!
종료합니다 .
```

7. 모든 칸이 찼으면 종료

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

2. 최종 테스트 스크린샷

```
첫 번째 유저 (x)의 차례입니다 -> (x,y) 좌표를 입력하세요 (최대범위 x:2, y:2): 1
1
---|---|---
---|---|---
---|X|---
---|---|---
---|---|---

두 번째 유저 (O)의 차례입니다 -> (x,y) 좌표를 입력하세요 (최대범위 x:2, y:2): 4
4, 4: x와 y 둘 중 하나가 칸을 벗어납니다.
두 번째 유저 (O)의 차례입니다 -> (x,y) 좌표를 입력하세요 (최대범위 x:2, y:2): 0
1
---|---|---
---|---|---
0|X|---
---|---|---
---|---|---

첫 번째 유저 (x)의 차례입니다 -> (x,y) 좌표를 입력하세요 (최대범위 x:2, y:2): 0
0
X|---|---
---|---|---
0|X|---
---|---|---
---|---|---

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

첫 번째 유저 (x)의 차례입니다 -> (x,y) 좌표를 입력하세요 (최대범위 x:2, y:2): 2
1
X|---|---
---|---|---
0|X|X|
---|---|---
---|O|---

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

첫 번째 유저 (x)의 차례입니다 -> (x,y) 좌표를 입력하세요 (최대범위 x:2, y:2): 1
2
X|---|---
---|---|---
0|X|X|
---|---|---
0|X|O|

두 번째 유저 (O)의 차례입니다 -> (x,y) 좌표를 입력하세요 (최대범위 x:2, y:2): 1
0
X|O|---
---|---|---
0|X|X|
---|---|---
0|X|O|

---|---|---
첫 번째 유저 (x)의 차례입니다 -> (x,y) 좌표를 입력하세요 (최대범위 x:2, y:2): 2
0
X|O|X|
---|---|---
0|X|X|
---|---|---
0|X|O|
---|---|---
모든 칸이 다 찹습니다. 종료합니다.
```

5. 결과 및 결론

1. 프로젝트 결과 : 게임 진행 및 차례 표시, 유효한 입력을 처리, 현재 게임판을 출력, 승리 조건을 검증, 무승부 처리, 게임 종료 기능이 구현된 Tic Tac Toe 게임을 만들었음
2. 느낀점 : 본인의 C++프로그래밍 실력에 대한 부족함을 깨닫고, 특히 반복문과 조건식 설정에서 가장 힘들어 하는 것을 알게 되었다. 이에 약점을 보완할 수 있도록 노력해야겠다는 생각이 들었다.