

5주체 Tic Tac Toe 프로젝트 보고서

213961 김태균

1. 서론

- A. 프로젝트 목적 및 배경: 4주차까지 배운 내용에 대한 실습을 위해 진행
- B. 목표: Tic Tac Toe 게임 구현

2. 요구사항

- A. 사용자 요구사항: 두 명의 사용자가 번갈아가며 O와 X를 놓기
- B. 기능 요구사항
 - i. 보드판 초기화
 - ii. 누구의 차례인지 출력
 - iii. 좌표 입력 받기
 - iv. 입력 받은 좌표 유효성 체크
 - v. 좌표에 O / X 놓기
 - vi. 현재 보드판 출력
 - vii. 빙고 시 승자 출력 후 종료
 - viii. 모든 칸이 다 찼으면 종료
 - ix. 게임이 끝나지 않았을 경우 계속 진행

3. 설계 및 구현

- A. 보드판 초기화

```

4  int main(){
5      const int numCell = 3;
6      char board[numCell][numCell]{};
7      int x, y;
8
9      // 보드판 초기화
10     // 보드의 인덱스를 예시와 반대로 [y][x]로 해 주어야 예시와 같은 오류가 안 생긴다고 판단했습니다
11     for(y = 0; y < numCell; y++)
12         for(x = 0; x < numCell; x++)
13             board[y][x] = ' ';
14

```

B. 누구의 차례인지 출력

```

16     // 게임 진행 코드
17     int k = 0; // 유저 차례 체크 위한 변수
18
19     // 유저가 바뀔 때마다 돌모양을 지정해주는 것보다 돌을 놓을 때 k값에 맞는 돌을 놓는 것이 더 좋다고 판단했습니다
20     char markers[2] = {'X', 'O'}; // 현재 유저의 돌을 저장하기 위한 변수
21
22     while(true){
23         // 1. 누구 차례인지 출력
24         switch(k % 2) {
25             case 0:
26                 cout << "첫번째 유저(X)의 차례입니다 -> ";
27                 break;
28             case 1:
29                 cout << "두번째 유저(O)의 차례입니다 -> ";
30                 break;
31         }
32

```

C. 좌표 입력 받기

```

33     // 2. 좌표 입력 받기
34     cout << "(x, y) 좌표를 입력하세요: ";
35     cin >> x >> y;
36

```

D. 입력 받은 좌표의 유효성 체크

```

37     // 3. 입력받은 좌표의 유효성 체크
38     // 음수 값을 입력받은 경우의 예외처리가 되지 않아 추가했습니다
39     if((x >= numCell || x < 0) || (y >= numCell || y < 0)){
40         cout << x << ", " << y << ": ";
41         cout << "x 와 y 둘 중 하나가 칸을 벗어납니다." << endl;
42         continue;
43     }
44     if (board[y][x] != ' '){
45         cout << x << ", " << y << ": 이미 돌이 차있습니다." << endl;
46         continue;
47     }
48

```

E. 입력 받은 좌표에 현재 유저의 돌 놓기

```

49     // 4. 입력받은 좌표에 현재 유저의 돌 놓기
50     // k값에 맞는 돌을 찾아 보드에 놓습니다
51     board[y][x] = markers[k % 2];
52

```

F. 현재 보드 판 출력

```
53 // 5. 현재 보드 판 출력
54 // x, y 좌표 축에 맞게 보이게 설정했습니다
55 for(int i = numCell - 1; i >= 0; i--){
56     cout << "---|---|---" << endl;
57     for(int j = 0; j < numCell; j++){
58         cout << board[i][j];
59         if(j == numCell - 1){
60             break;
61         }
62         cout << "  |";
63     }
64     cout << endl;
65 }
66 cout << "---|---|---" << endl;
67
```

G. 빙고 시 승자 출력 후 종료

```
68 // 6. 빙고 시 승자 출력 후 종료
69
70 // 가로줄 빙고 조건
71 if((board[0][0] != ' ' ) && (board[0][0] == board[0][1]) && (board[0][1] == board[0][2])){
72     cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
73     break;
74 }
75 if((board[1][0] != ' ' ) && (board[1][0] == board[1][1]) && (board[1][1] == board[1][2])){
76     cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
77     break;
78 }
79 if((board[2][0] != ' ' ) && (board[2][0] == board[2][1]) && (board[2][1] == board[2][2])){
80     cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
81     break;
82 }
83
84 // 세로줄 빙고 조건
85 if((board[0][0] != ' ' ) && (board[0][0] == board[1][0]) && (board[1][0] == board[2][0])){
86     cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
87     break;
88 }
89 if((board[0][1] != ' ' ) && (board[0][1] == board[1][1]) && (board[1][1] == board[2][1])){
90     cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
91     break;
92 }
93 if((board[0][2] != ' ' ) && (board[0][2] == board[1][2]) && (board[1][2] == board[2][2])){
94     cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
95     break;
96 }
97
98 // 대각선 빙고 조건
99 if((board[0][0] != ' ' ) && (board[0][0] == board[1][1]) && (board[1][1] == board[2][2])){
100     cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
101     break;
102 }
103 if((board[2][0] != ' ' ) && (board[2][0] == board[1][1]) && (board[1][1] == board[0][2])){
104     cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
105     break;
106 }
```

H. 모든 칸이 찼으면 종료

```

109
110 // 7. 모든 칸이 찼으면 종료
111 for(int i = 0; i < numCell; i++){
112     for(int j = 0; j < numCell; j++){
113         if(board[i][j] == ' ')
114             isFull = false;
115     }
116 }
117
118 if(isFull == true){
119     cout << "무승부입니다!" << endl;
120     break;
121 }

```

I. 게임이 끝나지 않았을 경우 계속 진행

```

123     k++;
124 }
125
126 }

```

k값이 1증가하면서 while문에 의해 다음 유저의 차례로 넘어가게 된다

4. 테스트

A. 기능별 테스트 결과

B. 최종 테스트 스크린샷

i. 가로줄