

No. 01

2024 CPP Project

Tic-Tac-Toe Game

Student ID: 214930

Name: 범지성

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. 제약 조건: 보드판은 2차원 배열 사용

3. 설계 및 구현

1) 누구의 차례인지 출력

```
20 // 1. Who's Turn
21 switch(k % 2){
22     case 0:
23         cout << "첫번째 유저(X)의 차례입니다 -> ";
24         currentUser = 'X';
25         break;
26     case 1:
27         cout << "두번째 유저(O)의 차례입니다 -> ";
28         currentUser = 'O';
29         break;
30 }
```

-입력

- k = 현재 유저 구분자

-결과

- 어떤 유저의 입력을 받을지 출력
- 문장 출력 후 좌표 입력으로 넘어감

-설명

- k를 2로 나눈 나머지가 0이면 첫 번째 유저, 1이면 두 번째 유저임을 알림
- 각 조건에 대응하는 유저의 기호를 currentUser에 대입

2) 좌표 입력 받기

```
32 // 2. Get Position
33 cout << "(x, y) 좌표를 입력하세요: ";
34 cin >> x >> y;
```

-입력

- x = 좌표 x 값
- y = 좌표 y 값

-결과

- 좌표 입력하라는 지시문 출력 후 유저 입력 대기

-설명

- cin으로 x와 y에 유저 입력을 받는다.

3) 입력 받은 좌표 유효성 체크

```

36 // 3. Check Validity
37 if(x >= numCell || y >= numCell){
38     cout << x << ", " << y << ": ";
39     cout << " x와 y 둘 중 하나가 칸을 벗어납니다.\n";
40     continue;
41 }
42 if(board[y][x] != ' '){
43     cout << x << ", " << y << ": 이미 둘이 차있습니다.\n";
44     continue;
45 }

```

-입력

- x = 좌표 x 값
- y = 좌표 y 값
- numCell = 가로/세로 칸 개수

-결과

- 칸을 놓을 수 없는 이유를 출력
- 출력 후 while문 초반으로 이동

-설명

- 사용자가 입력한 좌표가 게임 판을 벗어나는지 if로 체크
- 사용자가 입력한 좌표에 이미 둘이 있는지 if로 체크

4) 좌표에 O/X 놓기

```

47 // 4. Set Marker on Position
48 board[y][x] = currentUser;
49 count++; // Increase when Turn ends

```

-입력

- x = 좌표 x 값
- y = 좌표 y 값
- board = 게임판

- currentUser = 현재 유저의 돌 기호
- count = 유저들이 돌을 놓은 총 횟수

-결과

- 게임판의 (x, y)칸에 currentUser의 값이 저장된다

-설명

- 이후 기능을 위해 유저들의 돌 놓은 횟수를 추가

5) 현재 보드판 출력

```

51 // 5. Print Board
52 for(int i = 0; i < numCell; i++){
53     cout << "---|---|---\n";
54     for(int j = 0; j < numCell; j++){
55         cout << board[i][j];
56         if(j == numCell - 1) break;
57         cout << " |";
58     }
59     cout << endl;
60 }
61 cout << "---|---|---\n";
62 k++;

```

-입력

- numCell = 가로/세로 칸 개수
- board = 게임판
- k = 현재 유저 구분자

-결과

- 현재 게임판에 저장된 정보가 그대로 출력된다

-설명

- 2중 반복문으로 2차원 배열로 저장된 board의 정보를 2차원으로 출력한다.
- 한 차례가 지났기 때문에 구분자를 전환하기 위해 k에 1을 더한다.

6) 빙고 시 승자 출력 후 종료

```

64 // 6-1. Winner (Winner can appear at least count reach 5)
65 if(count >= 5){
66     // For every case check Vertical and Horizontal Lines
67
68     // Check Vertical Line
69     int lineCheck[numCell] = {0, };
70     for(int i = 0; i < numCell; i++){
71         if(board[i][x] == currentUser)
72             lineCheck[i] = 1;
73     }
74     if ((lineCheck[0] == lineCheck[1]) && (lineCheck[0] == lineCheck[2]) && lineCheck[0]){
75         cout << "vert\n";
76         winnerAppear = 1;
77         break;
78     }
79     // Check Horizontal Line
80     for(int i = 0; i < numCell; i++) lineCheck[i] = 0; // lineChecker initialize
81
82     for(int i = 0; i < numCell; i++){
83         if(board[y][i] == currentUser)
84             lineCheck[i] = 1;
85     }
86     if ((lineCheck[0] == lineCheck[1]) && (lineCheck[0] == lineCheck[2]) && lineCheck[0]){
87         cout << "hori\n";
88         winnerAppear = 1;
89         break;
90     }

```

```

92 // When mark is on Top to Bottom Tilted Line
93 if(x == y){
94     for(int i = 0; i < numCell; i++) lineCheck[i] = 0; // lineChecker initialize
95
96     for(int i = 0; i < numCell; i++){
97         if(board[i][i] == currentUser)
98             lineCheck[i] = 1;
99     }
100     if ((lineCheck[0] == lineCheck[1]) && (lineCheck[0] == lineCheck[2]) && lineCheck[0]){
101         winnerAppear = 1;
102         cout << "ttb\n";
103         break;
104     }
105 }
106
107 // When mark is on Bottom to Top Tilted Line
108 if((x + y) == 2){
109     for(int i = 0; i < numCell; i++) lineCheck[i] = 0; // lineChecker initialize
110
111     for(int i = 0; i < numCell; i++){
112         if(board[2 - i][i] == currentUser)
113             lineCheck[i] = 1;
114     }
115     if ((lineCheck[0] == lineCheck[1]) && (lineCheck[0] == lineCheck[2]) && lineCheck[0]){
116         winnerAppear = 1;
117         cout << "btt\n";
118         break;
119     }
120 }
121 }

```

```

128 // If Winner Exist or Not
129 if(winnerAppear == 1)
130     cout << "유저(" << currentUser << ")가 이겼습니다!\n";

```

-입력

- numCell = 가로/세로 칸 개수
- board = 게임판
- lineCheck = 해당 줄 빙고가 완성됐는지 확인하는 배열
- x = 좌표 x 값

- $y = \text{좌표 } y \text{ 값}$
- `winnerAppear` = 승자 등장 여부 확인 플래그

-결과

- 현재 게임판에 저장된 정보가 그대로 출력된다

-설명

- 2중 반복문으로 2차원 배열로 저장된 `board`의 정보를 2차원으로 출력한다.
- 한 차례가 지났기 때문에 구분자를 전환하기 위해 `k`에 1을 더한다.

7) 모든 칸이 찼으면 종료