

tic_tac_toe 게임구현

234124 손명현

2024-10-2

1. 서론

- 1) 프로젝트 목적 및 배경 : 반복문, if 문, switch 문 등 이전에 배운 것들을 활용하여 요구하는 기능을 실현시키기 위해 진행하였음.
- 2) 목표 : Tic Tac Toe 게임을 구현

2. 요구사항

- 1) 사용자 요구사항 : 두명의 사용자가 3x3 의 판에 0 X 를 번갈아 두며 가로 세로 대각선으로 연속되게 자기 문양을 채워 승자를 가리는 게임을 만들기
- 2) 기능 요구사항 : 누구의 차례인지 출력, 좌표 입력 받기, 입력 받은 좌표 유효성 체크, 좌표에 0/X 놓기, 현재 보드판 출력, 빙고 시 승자 출력 후 종료, 모든 칸이 찼으면 종료.

3. 설계 및 구현

- 1) 기능 별 구현 사항

```
2)      // 1. 누가 차례인지 출력
3)      switch (k % 2) {
4)          case 0:
5)              cout << "첫 번째 유저(X)의 차례입니다 ~" << endl;
6)              currentUser = 'X';
7)              break;
8)          case 1:
9)              cout << "두 번째 유저(O)의 차례입니다 ~" << endl;
10)             currentUser = 'O';
11)             break;
12)     }
13)
14)     // 2. 좌표 입력 받기
15)     cout << "(x, y) 좌표를 입력하세요: ";
16)     cin >> x >> y;
17)
18)     // 3. 입력받은 좌표의 유효성 체크
19)     if (x >= numCell || y >= numCell) {
20)         cout << "범위를 벗어났습니다." << endl;
21)         cout << "x 와 y 둘 중 하나가 값을 벗어났습니다." << endl;
22)         continue;
23)     }
24)
25)     if (board[x][y] != ' ') {
26)         cout << "(x, y) : 이미 돌이 차있습니다." << endl;
27)         continue;
28)     }
```

```

29)
30) // 4. 입력받은 좌표에 현재 유저의 돌 놓기
31) board[x][y] = currentUser;
32)
33) // 5. 현재 보드판 출력
34) for (int i = 0; i < numCell; i++) {
35)     cout << "---|---|---" << endl;
36)     for (int j = 0; j < numCell; j++) {
37)         cout << " " << board[i][j];
38)         if (j == numCell - 1) {
39)             break;
40)         }
41)         cout << " |";
42)     }
43)     cout << endl;
44) }
45) cout << "---|---|---" << endl;
46)
47) k++;
48)
49) //6. 빙고 시 승자 출력 후 종료 (행, 열, 대각선)
50) int cntX = 0, cntO = 0;
51)
52) // 행 방향 체크
53) for (int j = 0; j < numCell; j++) {
54)     cntX = 0; cntO = 0;
55)     for (int i = 0; i < numCell; i++) {
56)         if (board[i][j] == 'X')
57)             cntX++;
58)         else if (board[i][j] == 'O')
59)             cntO++;
60)     }
61)
62)     if (cntX == numCell || cntO == numCell) {
63)         cout << "종료" << endl;
64)         switch (currentUser) {
65)             case 'X':
66)                 cout << "첫 번째 유저(X) 승리!" << endl;
67)                 break;
68)             case 'O':
69)                 cout << "두 번째 유저(O) 승리!" << endl;
70)                 break;
71)         }
72)         return 0;
73)     }
74) }
75)
76) // 열 방향 체크

```

```

77)         for (int j = 0; j < numCell; j++) {
78)             cntX = 0; cntO = 0;
79)             for (int i = 0; i < numCell; i++) {
80)                 if (board[j][i] == 'X')
81)                     cntX++;
82)                 else if (board[j][i] == 'O')
83)                     cntO++;
84)             }
85)
86)             if (cntX == numCell || cntO == numCell) {
87)                 cout << "종료" << endl;
88)                 switch (currentUser) {
89)                     case 'X':
90)                         cout << "첫 번째 유저(X) 승리!" << endl;
91)                         break;
92)                     case 'O':
93)                         cout << "두 번째 유저(O) 승리!" << endl;
94)                         break;
95)                 }
96)                 return 0;
97)             }
98)         }
99)
100)        // 대각선 방향 체크
101)        cntX = 0; cntO = 0;
102)
103)        for (int i = 0; i < numCell; i++) {
104)            if (board[i][i] == 'X')
105)                cntX++;
106)            else if (board[i][i] == 'O')
107)                cntO++;
108)        }
109)
110)        if (cntX == numCell || cntO == numCell) {
111)            cout << "종료" << endl;
112)            switch (currentUser) {
113)                case 'X':
114)                    cout << "첫 번째 유저(X) 승리!" << endl;
115)                    break;
116)                case 'O':
117)                    cout << "두 번째 유저(O) 승리!" << endl;
118)                    break;
119)            }
120)            return 0;
121)        }
122)
123)        cntX = 0; cntO = 0;
124)

```

```

125)         for (int i = 0; i < numCell; i++) {
126)             if (board[i][numCell - 1 - i] == 'X')
127)                 cntX++;
128)             else if (board[i][numCell - 1 - i] == 'O')
129)                 cntO++;
130)         }
131)
132)         if (cntX == numCell || cntO == numCell) {
133)             cout << "종료" << endl;
134)             switch (currentUser) {
135)                 case 'X':
136)                     cout << "첫 번째 유저(X) 승리!" << endl;
137)                     break;
138)                 case 'O':
139)                     cout << "두 번째 유저(O) 승리!" << endl;
140)                     break;
141)             }
142)             return 0;
143)         }
144)
145)         // 7. 모든 칸이 찼으면 실행 종료
146)         bool isBoardFull = true;
147)         for (int i = 0; i < numCell; i++) {
148)             for (int j = 0; j < numCell; j++) {
149)                 if (board[i][j] == ' ') {
150)                     isBoardFull = false;
151)                     break;
152)                 }
153)             }
154)             if (!isBoardFull) break;
155)         }
156)
157)         if (isBoardFull) {
158)             cout << "모든 칸이 꽉 찼습니다. 무승부!" << endl;
159)             return 0;
160)         }

```

4. 테스트

1) 기능 별 테스트 결과

누구 차례

첫 번째 유저(X)의 차례입니다 ~

좌표 입력

(y, x) 좌표를 입력하세요:

좌표 유효성

```
(y, x) 좌표를 입력하세요: 3 0
범위를 벗어났습니다.
x와 y 둘 중 하나가 값을 벗어났습니다.
```

돌 놓기 및 현재 보드판 출력

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

빙고 시 승자 출력

```
(y, x) 좌표를 입력하세요: 0 2
---|---|---
X  | X | X
---|---|---
O  | O |
---|---|---
   |   |
---|---|---
종료
첫 번째 유저(X) 승리!
```

2) 최종 테스트 스크린 샷

첫 번째 유저(X)의 차례입니다 ~

(y, x) 좌표를 입력하세요: 0 0

---|---|---

X | |

---|---|---

| |

---|---|---

| |

---|---|---

두 번째 유저(O)의 차례입니다 ~

(y, x) 좌표를 입력하세요: 1 0

---|---|---

X | |

---|---|---

O | |

---|---|---

| |

---|---|---

첫 번째 유저(X)의 차례입니다 ~

(y, x) 좌표를 입력하세요: 0 1

---|---|---

X | X |

---|---|---

O | |

---|---|---

| |

---|---|---

두 번째 유저(O)의 차례입니다 ~

(y, x) 좌표를 입력하세요: 1 1

---|---|---

X | X |

---|---|---

O | O |

---|---|---

| |

---|---|---

첫 번째 유저(X)의 차례입니다 ~

(y, x) 좌표를 입력하세요: 0 2

---|---|---

X | X | X

---|---|---

O | O |

---|---|---

| |

---|---|---

종료
첫 번째 유저(X) 승리!

X	O	O
	X	
		X

종료
첫 번째 유저(X) 승리!

X	O	
X	O	
X		

종료
첫 번째 유저(X) 승리!

O		X
O	X	
X		

종료
첫 번째 유저(X) 승리!

5. 결과 및 결론

- 1) 프로젝트 결과 : 사용자의 요구 사항과 교수님이 주신 코드에 x y 좌표가 바뀌어 있는 부분을 찾아 고쳐 tic tac toe 게임을 만들었음
- 2) 느낀점 : 코드를 작성할 때에 행 열 대각선이 빙고 시 어떻게 체크 할지에 대하여 막막함을 느꼈으나 바로 이전에 배운 배열을 이용하여 해결 하였다. 코드 작성과 보고서를 작성하는 시간이 생각보다 오래 걸려서 쉽지 않았다.