

# 214766 고혜림

## 1. 서론

1. 프로젝트의 목적 및 배경: 4주차까지 배운 내용을 이용한 실습
2. 목표: tic tac toe 게임 구현

## 2. 요구사항

1. 사용자 요구사항: 두명의 사용자가 번갈아 가며 X와 O를 놓아서 게임을 한다.
2. 기능 요구사항:
  1. 누구의 차례인지 출력
  2. 좌표 입력 받기
  3. 입력 받은 좌표 유효성 체크
  4. 좌표에 O, X 놓기
  5. 현재 보드 판 출력
  6. 빙고 시 승자 출력
  7. 모든 칸이 찼으면 종료

### 3. 설계 및 구현

#### 1. 기능별 구현 사항

##### ① 누구의 차례인지 출력

##### 1) 스크린샷

```
20
21 // 1. 누구의 차례인지 출력
22 switch (k % 2) {
23     case 0:
24         cout << "첫번째 유저(X)의 차례입니다. -> ";
25         currentUser = 'X';
26         break;
27     case 1:
28         cout << "두번째 유저(O)의 차례입니다. -> ";
29         currentUser = 'O';
30         break;
31 }
32
33 64 k++;
```

##### 2) 입력

- k = 누구 차례인지 체크하기위한 변수 (int), 한 차례가 지날 때마다 1씩 증가
- currentUser = 현재 유저의 돌을 저장하기 위한 문자 변수 (char)

##### 3) 결과

- 어떤 유저가 돌을 놓을 차례인지 출력 후 switch문을 나감

##### 4) 설명

- k를 2로 나눈 나머지가 0일 경우 첫번째 유저, 1일경우 두번째 유저
- switch로 구현

## ② 좌표 입력 받기

## ③ 입력 받은 좌표 유효성 체크

### 1) 스크린샷

```
33 // 2. 좌표입력받기
34 cout << "좌표를 입력하세요 (0 ~ 2) (0 ~ 2): ";
35 cin >> x >> y;
36
37 // 3. 입력받은 좌표의 유효성 체크
38 if (x >= numCell || y >= numCell) {
39     cout << x << ", " << y << ": ";
40     cout << "좌표가 칸을 벗어납니다." << endl;
41     continue;
42 }
43 if (board[x][y] != ' ') {
44     cout << x << ", " << y << ": 이미 돌이 차있습니다." << endl;
45     continue;
46 }
```

### 2) 입력

- x = 행, y = 열
- numCell = 가로/세로 칸 수
- board[][] = 틱택토 게임을 하는 사용자가 입력한 값이 저장되는 2차원 배열

### 3) 결과

- 좌표를 입력하라는 문구 출력
- 입력 받은 좌표가 칸에 들어갈 수 없는 이유 출력
- 출력 후 while문 초반으로 이동(continue)

### 4) 설명

1. 사용자가 좌표를 입력
2. 입력한 좌표를 x, y 에 저장
3. 입력한 좌표가 보드를 벗어나는지 if로 확인
4. 입력한 좌표에 이미 돌이 있는지 if로 확인

#### ④ 좌표에 O, X 놓기

##### 1) 스크린샷

```
47  
48 // 4. 입력받은 좌표에 현재 유저의 돌 놓기  
49 board[x][y] = currentUser;  
50
```

##### 2) 입력

- x = 행, y = 열 사용자가 입력한 좌표가 저장되어 있음
- board[][] = 틱택토 게임을 하는 사용자가 입력한 값이 저장되는 2차원 배열
- currentUser = 현재 유저의 돌을 저장하기 위한 문자 변수 (char), 'X' 또는 'O'

##### 3) 결과 및 설명

- 사용자가 입력한 좌표가 배열에 저장

#### ⑤ 현재 보드 판 출력

##### 1) 스크린샷

```
51 // 5. 현재 보드판 출력  
52 for (int i = 0; i < numCell; i++) {  
53     cout << "---|---|---" << endl;  
54     for (int j = 0; j < numCell; j++) {  
55         cout << board[i][j];  
56         if (j == numCell - 1) {  
57             break;  
58         }  
59         cout << " |";  
60     }  
61     cout << endl;  
62 }  
63 cout << "---|---|---" << endl;  
64 k++;  
65
```

##### 2) 입력

- numCell = 보드의 가로/세로 칸 수
- board[][] = 틱택토 게임을 하는 사용자가 입력한 값이 저장되는 2차원 배열
- k = 누구 차례인지 체크하기 위한 변수 (int), 한 차례가 지날 때마다 1씩 증가

### 3) 결과

- 3 X 3 보드 판을 출력

### 4) 설명

1. for문을 사용해서 보드 판을 출력
2. k를 증가시켜서 다음 차례로 넘김

## ⑥ 빙고 시 승자 출력

### 1) 스크린샷

```
66 // 6. 빙고 시 승자 출력 후 종료
67 /*
68 가로 세로 대각선
69 가로: (00, 01, 02)(10, 11, 12)(20, 21, 22)
70 세로: (00, 10, 20)(01, 11, 21)(02, 12, 22)
71 대각선: 왼쪽 위에서 부터(00, 11, 22)
72 | | 오른쪽 위에서 부터(02, 11, 20)
73 */
74
75 // 승자출력
76 for (int i = 0; i < numCell; i++) {
77     // 가로로 승리할 경우
78     if (board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser) {
79         cout << "가로에 모든 돌이 놓였습니다. " << currentUser << "의 승리입니다. ";
80         return 0;
81     }
82     // 세로로 승리할 경우
83     if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser) {
84         cout << "세로에 모든 돌이 놓였습니다. " << currentUser << "의 승리입니다. ";
85         return 0;
86     }
87
88     // 대각선으로 승리할 경우
89     if (board[0][0] == currentUser && board[1][1] == currentUser && board[2][2] == currentUser) {
90         cout << " 대각선 왼쪽 위부터 오른쪽 아래까지 모든 돌이 놓였습니다. " << currentUser << "의 승리입니다. ";
91         return 0;
92     }
93     if (board[0][2] == currentUser && board[1][1] == currentUser && board[2][0] == currentUser) {
94         cout << " 대각선 오른쪽 위부터 왼쪽 아래까지 모든 돌이 놓였습니다. " << currentUser << "의 승리입니다. ";
95         return 0;
96     }
97 }
```

### 2) 입력

- numCell = 보드의 가로/세로 칸 수
- board[][] = 틱택토 게임을 하는 사용자가 입력한 값이 저장되는 2차원 배열
- currentUser = 현재 유저의 돌을 저장하기 위한 문자 변수 (char), 'X' 또는 'O'

### 3) 결과

- 누군가 가로/세로/대각선으로 3돌을 채워 승리했을 경우 승리 메시지를 출력
- 출력 후 코드를 종료

### 4) 설명

1. for문과 if문을 이용해 첫번째 인덱스 혹은 두번째 인덱스가 모두 일치하는 경우 확인 후 코드 종료
2. if문을 이용해 대각선으로 승리하는 경우를 확인 후 코드 종료

## ⑦ 모든 칸이 찼으면 종료

### 1) 스크린샷

```
99      // 7. 모든칸이 찼으면 종료
100     int checked = 0;
101     for (int i = 0; i < numCell; i++) {
102         for (int j = 0; j < numCell; j++) {
103             if (board[i][j] == ' ') {
104                 checked++;
105             }
106         }
107     }
108     if (checked == 0) {
109         cout << "칸이 다 찼습니다. 종료합니다." << endl;
110         break;
111     }
112 }
```

### 2) 입력

- checked = 보드에 비어있는 칸 수
- numCell = 보드의 가로/세로 칸 수
- board[][] = 틱택토 게임을 하는 사용자가 입력한 값이 저장되는 2차원 배열

### 3) 결과

- 보드에 칸이 다 찼다면 종료

### 4) 과정

1. for문을 이용해 checked에 비어있는 칸의 수를 저장
2. checked가 0이 되면 비어있는 칸이 없는 것이므로 칸이 다 찼다는 말 출력 후 종료

## 4. 테스트

### 1. 기능 별 테스트 결과

#### ① 누구의 차례인지 출력

첫번째 유저(x)의 차례입니다. -> 좌표를 입력하세요 (0 ~ 2) (0 ~ 2):

두번째 유저(o)의 차례입니다. -> 좌표를 입력하세요 (0 ~ 2) (0 ~ 2):

#### ② 좌표 입력 받기

첫번째 유저(x)의 차례입니다. -> 좌표를 입력하세요 (0 ~ 2) (0 ~ 2): 0 0

#### ③ 입력 받은 좌표 유효성 체크

##### A. 좌표가 칸을 벗어날 경우

두번째 유저(o)의 차례입니다. -> 좌표를 입력하세요 (0 ~ 2) (0 ~ 2): 4 4  
4, 4: 좌표가 칸을 벗어납니다.  
두번째 유저(o)의 차례입니다. -> 좌표를 입력하세요 (0 ~ 2) (0 ~ 2):

##### B. 이미 돌이 차있을 경우

두번째 유저(o)의 차례입니다. -> 좌표를 입력하세요 (0 ~ 2) (0 ~ 2): 0 0  
0, 0: 이미 돌이 차있습니다.  
두번째 유저(o)의 차례입니다. -> 좌표를 입력하세요 (0 ~ 2) (0 ~ 2):

#### ④ 좌표에 O, X 놓기

#### ⑤ 현재 보드 판 출력

첫번째 유저(x)의 차례입니다. -> 좌표를 입력하세요 (0 ~ 2) (0 ~ 2): 0 0

---	---	---
X		
---	---	---
---	---	---

두번째 유저(o)의 차례입니다. -> 좌표를 입력하세요 (0 ~ 2) (0 ~ 2): 0 1

---	---	---
X	O	
---	---	---
---	---	---

⑥ 빙고 시 승자 출력

A. 가로로 승리할 경우

```
---|---|---
X |X |X
---|---|---
  |O |
---|---|---
  |  |O
---|---|---
가로에 모든 돌이 놓였습니다. x의 승리입니다.
```

B. 세로로 승리할 경우

```
---|---|---
X |O |
---|---|---
X |O |
---|---|---
X |  |
---|---|---
세로에 모든 돌이 놓였습니다. x의 승리입니다.
```

C. 대각선으로 승리할 경우

```
---|---|---
X |O |
---|---|---
  |X |
---|---|---
  |O |X
---|---|---
대각선 왼쪽 위부터 오른쪽 아래까지 모든 돌이 놓였습니다. x의 승리입니다.
```

```
---|---|---
O |  |X
---|---|---
  |X |
---|---|---
X |  |O
---|---|---
대각선 오른쪽 위부터 왼쪽 아래까지 모든 돌이 놓였습니다. x의 승리입니다.
```

⑦ 모든 칸이 찼으면 종료

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



## 2. 최종 테스트 스크린 샷

```
첫번째 유저(x)의 차례입니다. -> 좌표를 입력하세요 (0 ~ 2) (0 ~ 2): 0 0
---|---|---
X  |   |
---|---|---
   |   |
---|---|---
   |   |
---|---|---
두번째 유저(o)의 차례입니다. -> 좌표를 입력하세요 (0 ~ 2) (0 ~ 2): 0 1
---|---|---
X  |0  |
---|---|---
   |   |
---|---|---
   |   |
---|---|---
첫번째 유저(x)의 차례입니다. -> 좌표를 입력하세요 (0 ~ 2) (0 ~ 2): 1 0
---|---|---
X  |0  |
---|---|---
X  |   |
---|---|---
   |   |
---|---|---
두번째 유저(o)의 차례입니다. -> 좌표를 입력하세요 (0 ~ 2) (0 ~ 2): 1 1
---|---|---
X  |0  |
---|---|---
X  |0  |
---|---|---
   |   |
---|---|---
첫번째 유저(x)의 차례입니다. -> 좌표를 입력하세요 (0 ~ 2) (0 ~ 2): 2 0
---|---|---
X  |0  |
---|---|---
X  |0  |
---|---|---
X  |   |
---|---|---
세로에 모든 돌이 놓였습니다. x의 승리입니다.
```

## 5. 결과 및 결론

### 1. 프로젝트 결과

tic tac toe 게임을 만들어 봄

## 2. 느낀 점

수업시간 안에 완성하기에 시간이 너무 부족했다고 느꼈습니다. 배운 내용을 응용해서 해결할 수 있는 과제였습니다. 하지만 조금 어려웠습니다. 시간이 넉넉히 있었다면 교수님이 올려 주신 답을 보지 않고 해결할 수 있을 것 같습니다.