

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 구현 사항:

- ① 보드판 초기화

```
int main() {
    const int numCell = 3; // 보드판의 가로, 세로 칸 수
    char board[numCell][numCell]{}; // 배열의 모든 칸을 0이나 null로 초기화
    int x, y; // 사용자에게 입력받는 (x,y) 좌표를 저장할 변수

    // 보드판을 빈칸으로 초기화
    for (x = 0; x < numCell; x++) {
        for (y = 0; y < numCell; y++) {
            board[x][y] = ' ';
        }
    }
}
```

입력

- numCell = 가로/세로 칸 개수

결과

- 가로/세로가 각각 3칸인 보드(2차원 배열) 생성
- 보드판의 모든 칸을 '.'(빈칸)으로 초기화

설명

- 중첩 for문을 이용해 2차원 배열 보드판의 각 칸들에 접근 & 초기화
- 바깥 for문에서 x는 board의 행을, 안쪽 for문에서 y는 board의 열을 나타냄

② 누구의 차례인지 출력

```
// 게임 코드
int k = 0; // 누구의 차례인지 체크하기 위한 변수
char currentUser = 'X'; // 현재 유저의 돌을 저장하기 위한 문자 변수
while (true) {
    // 1. 누구의 차례인지 출력
    switch (k % 2) {
        case 0:
            cout << k % 2 + 1 << "번 유저(X)의 차례입니다 -> ";
            currentUser = 'X';
            break;
        case 1:
            cout << k % 2 + 1 << "번 유저(O)의 차례입니다 -> ";
            currentUser = 'O';
            break;
    }
}
```

입력

- k = 누구의 차례인지 체크하기 위한 변수
- currentUser = 현재 유저의 돌(O/X)

결과

- 현재 누구의 차례인지 출력
- 현재 유저의 돌(O/X)이 currentUser에 저장

설명

- switch문을 이용해 누구의 차례인지 출력
- k의 값이 0 또는 짝수일 때 1번 유저(X), 홀수일 때 2번 유저(O)의 차례

(while의 하위문장의 맨 끝에 k++; 존재 -> 루프를 돌 때마다 k값 증가

-> 두 유저가 번갈아가며 게임 진행)

③ 좌표 입력 받기

```
// 2. (x, y) 좌표를 입력받음
cout << "(x, y) 좌표를 입력하세요: ";
cin >> x >> y;
```

입력

· x = 좌표 x값

· y = 좌표 y값

결과

· 돌을 놓을 좌표 (x, y)를 입력받음

④ 입력 받은 좌표 유효성 체크

```
// 3. 입력받은 좌표의 유효성 체크
if (x >= numCell || y >= numCell) {
    cout << x << ", " << y << ": ";
    cout << "x와 y 둘 중 하나가 칸을 벗어납니다." << endl;
    cout << endl;
    continue;
}
if (board[x][y] != ' ') {
    cout << x << ", " << y << ": 이미 돌이 차있습니다." << endl;
    cout << endl;
    continue;
}
```

입력

· x = 좌표 x값

· y = 좌표 y값

· numCell = 가로/세로 칸 개수

결과

· 칸에 돌을 놓을 수 없는 이유 출력

· 출력 후 while문 초반으로 이동, 다시 좌표 설정

설명

- if문을 사용해 사용자가 입력한 좌표가 보드판을 벗어나는지 체크
- if문을 사용해 사용자가 입력한 좌표에 돌이 이미 놓여있는지 체크
- 두 조건 중 어느 하나라도 해당할 시, continue;를 통해 while문 초반으로 이동하고 좌표 다시 설정

⑤ 좌표에 O / X 놓기

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

입력

- board[x][y] = 사용자가 입력한 보드판의 좌표 칸
- currentUser = 현재 유저의 돌(O/X)

결과

- 사용자가 입력한 보드판의 좌표 칸에 유저의 돌을 넣음(대입)

⑥ 현재 보드판 출력

```
// 5. 현재 보드 판 출력
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;
cout << endl;
```

입력

- i = board의 행(가로)
- j = board의 열(세로)
- numCell = 가로/세로 칸 개수

결과

- 현재 보드판 출력

설명

- 중첩 for문을 이용해 이차원 배열 board의 각 칸에 접근하여 내용물 출력
- 바깥 for문은 board의 행을 나타내고, 안쪽 for문은 board의 열을 나타냄
- 바깥 for문과 안쪽 for문 사이에 "---|---|---"을 출력하게 하여, 각 행을 구별함
- if문을 사용해 출력 형식을 정함 -> board의 마지막 열들을 제외한 모든 칸은 내용물 출력 후, " |"을 출력하여 다음 칸(열)과 구분을 줌

⑦ 빙고 시 승자 출력 후 종료

```
// 6. 빙고 시 승자 출력 후 종료
for (int i = 0; i < numCell; i++) {
    // (1) 가로에 모두 같은 돌이 놓임
    if (board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser) {
        cout << "가로에 모두 같은 돌이 놓였습니다! ";
        cout << k % 2 + 1 << "번 유저(" << currentUser << ")의 승리입니다!" << endl;
        cout << "게임을 종료합니다\n";
        break;
    }

    // (2) 세로에 모두 같은 돌이 놓임
    if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser) {
        cout << "세로에 모두 같은 돌이 놓였습니다! ";
        cout << k % 2 + 1 << "번 유저(" << currentUser << ")의 승리입니다!" << endl;
        cout << "게임을 종료합니다\n";
        break;
    }
}

// (3) 왼쪽 위에서 오른쪽 아래 대각선으로 모두 같은 돌이 놓임
if (board[0][0] == currentUser && board[1][1] == currentUser && board[2][2] == currentUser) {
    cout << "왼쪽 위에서 오른쪽 아래 대각선으로 모두 같은 돌이 놓였습니다! ";
    cout << k % 2 + 1 << "번 유저(" << currentUser << ")의 승리입니다!" << endl;
    cout << "게임을 종료합니다\n";
    break;
}

// (4) 오른쪽 위에서 왼쪽 아래 대각선으로 모두 같은 돌이 놓임
if (board[2][0] == currentUser && board[1][1] == currentUser && board[0][2] == currentUser) {
    cout << "오른쪽 위에서 왼쪽 아래 대각선으로 모두 같은 돌이 놓였습니다! ";
    cout << k % 2 + 1 << "번 유저(" << currentUser << ")의 승리입니다!" << endl;
    cout << "게임을 종료합니다\n";
    break;
}
}
```

입력

- i = board의 행/열
- numCell = 가로/세로 칸 개수

- currentUser = 현재 유저의 돌(O/X)
- $k \% 2 + 1$ = 현재 유저의 순번

결과

- 가로, 세로, 대각선 중 하나라도 모두 같은 돌이 놓였을 경우 승자 출력 후 while문을 빠져나옴

설명

- for문을 이용하여 board의 각 행에 접근하고, if문을 이용해 같은 행에 있는 모든 원소가 currentUser와 같은지 체크
- for문을 이용하여 board의 각 열에 접근하고, if문을 이용해 같은 열에 있는 모든 원소가 currentUser와 같은지 체크
- if문을 이용해 보드의 대각선(왼쪽 위 -> 오른쪽 아래) 칸(0, 0), (1, 1), (2, 2)의 모든 원소가 currentUser와 같은지 체크
- if문을 이용해 보드의 대각선(오른쪽 위 -> 왼쪽 아래) 칸(0, 2), (1, 1), (2, 0)의 모든 원소가 currentUser와 같은지 체크
- 만약 빙고가 만들어질 시, 현재 유저가 승리했음을 출력하여 알리고, break;을 통해 while문을 빠져나옴

⑧ 모든 칸이 찼으면 종료

```
// 7. 모든 칸이 다 찼으면 게임 종료
int blank = 0; // 보드판에서 빈칸의 수를 나타내는 변수
for (int i = 0; i < numCell; i++) {
    for (int j = 0; j < numCell; j++) {
        if (board[i][j] == ' ') {
            blank++;
        }
    }
}
if (blank == 0) { // 빈칸이 없을 때 게임 종료
    cout << "모든 칸이 다 찼습니다. 게임을 종료합니다\n";
    break;
}

k++; // 다음 유저에게 차례를 넘김
}
return 0;
```

입력

- blank = 2차원 배열 board의 원소 중 ' '(빈칸)의 개수
- i = board의 행(가로)
- j = board의 열(세로)
- numCell = 가로/세로 칸 개수

결과

- 보드판의 모든 칸이 다 찼다 출력 후 게임 종료, while문을 빠져나옴

설명

- for문을 이용하여 board의 각 원소에 접근하고, if문을 통해 원소가 ' '(빈칸)일 경우 blank 값 1 증가
- if문을 이용해 blank 값이 0일 때, 모든 칸이 다 찼다는 코멘트 출력 후 게임 종료, break;을 통해 while문을 빠져나옴

4. 테스트

(1). 기능 별 테스트 결과:

① 누구의 차례인지 출력

1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요:

2번 유저(O)의 차례입니다 -> (x, y) 좌표를 입력하세요:

② 좌표 입력 받기

1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 1

2번 유저(O)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 0

③ 입력 받은 좌표 유효성 체크

1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 3 3
3, 3: x와 y 둘 중 하나가 칸을 벗어납니다.

1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 0
0, 0: 이미 돌이 차있습니다.

1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요:

④ 좌표에 O / X 놓기 & 현재 보드판 출력

1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 1

---	---	---
	X	

2번 유저(O)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 0

---	---	---
O		
	X	

⑤ 빙고 시 승자 출력 후 종료

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

가로에 모두 같은 돌이 놓였습니다!: 1번 유저(X)의 승리입니다!
게임을 종료합니다

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

세로에 모두 같은 돌이 놓였습니다!: 2번 유저(O)의 승리입니다!
게임을 종료합니다

0		X
X	0	
X		0

왼쪽 위에서 오른쪽 아래 대각선으로 모두 같은 돌이 놓였습니다!: 2번 유저(0)의 승리입니다!
게임을 종료합니다

0		X
0	X	
X		

오른쪽 위에서 왼쪽 아래 대각선으로 모두 같은 돌이 놓였습니다!: 1번 유저(X)의 승리입니다!
게임을 종료합니다

⑥ 모든 칸이 찼으면 종료

X	0	X
X	X	0
0	X	0

모든 칸이 다 찹습니다. 게임을 종료합니다

(2). 최종 테스트 스크린샷:

①

1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 0

X		

2번 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 1

X	0	

1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 0

X	0	
X		

2번 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 0

0		
X	0	
X		

1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2

0		
X	0	
X		X

2번 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 1

0		
X	0	
X	0	X

1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 1

0	X	
X	0	
X	0	X

2번 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
2, 2: 이미 들어 있습니다.

2번 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 2

0	X	
X	0	0
X	0	X

1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 2

0	X	X
X	0	0
X	0	X

모든 칸이 다 찼습니다. 게임을 종료합니다

②

1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 1

	X	

2번 유저(O)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 0

	X	
O		

1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 0

X		
	X	
O		

2번 유저(O)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 3
1, 3: x와 y 둘 중 하나가 칸을 벗어납니다.

2번 유저(O)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 1

X		
	X	
O	O	

1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2

X		
	X	
O	O	X

왼쪽 위에서 오른쪽 아래 대각선으로 모두 같은 돌이 놓였습니다!: 1번 유저(X)의 승리입니다!