

1, 프로젝트 목적 및 배경

C++ 4주차 까지 내용 복습 및 실습

목표 : Tic Tac Toe 게임 구현

2. 요구사항

두 명의 사용자가 O, X돌을 놓으며 Tic Tac Toe 게임 진행 // 보드판은 2차원 배열 사용

- 누구의 차례인지 출력
- 좌표 입력받기 (x, y)
- 입력 받은 좌표 유효성 체크
- 좌표에 O, X 놓기
- 현재 보드판 출력
- 빙고 시 승자 출력 후 종료
- 모든 칸이 찼으면 종료

3. 설계 및 구현

1. 누구의 차례인지 출력

입력

k: 플레이어 차례를 나타내는 변수

currentUser: 현재 차례인 사용자의 돌

결과

```
while (true) {  
    // 1. 플레이어 차례 출력  
    switch (k % 2) {  
        case 0:  
            cout << "첫번째 유저(X)의 차례입니다 -> ";  
            currentUser = 'X';  
            break;  
        case 1:  
            cout << "두번째 유저(O)의 차례입니다 -> ";  
            currentUser = 'O';  
            break;  
    }  
}
```

- 첫 번째 플레이어는 'X', 두 번째 플레이어는 'O'가 차례를 가지며 출력

설명

- $k \% 2$ 를 이용하여 짝수일 때 첫 번째 유저의 차례, 홀수일 때 두 번째 유저의 차례를 나타냄.
- 플레이어 차례에 맞게 "첫번째 유저(X)의 차례입니다" 혹은 "두번째 유저(O)의 차례입니다"를 출력.

2. 좌표 입력받기 (x, y)

입력

- 사용자로부터 x와 y 좌표를 입력받음.

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

결과

- 좌표가 x와 y 변수에 저장됨.

설명

- 콘솔에서 cin을 통해 사용자가 입력한 좌표를 x, y 변수에 저장.

3. 입력 받은 좌표 유효성 체크

입력

- x: 좌표 x값
- y: 좌표 y값
- numCell: 게임 판 크기 (3)

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

결과

- 유효하지 않은 좌표일 경우, 적절한 오류 메시지를 출력하고 다음 차례로 넘어감.

설명

- 두 가지 유효성 검사를 수행:

1. 입력한 좌표가 게임 판 크기를 초과했는지 검사. 초과하면 오류 메시지 출력.
2. 입력한 좌표에 이미 돌이 있는지 검사. 이미 차있으면 오류 메시지 출력.

4. 좌표에 O, X 놓기

입력

- x: 좌표 x값
- y: 좌표 y값
- currentUser: 현재 유저 ('X' 또는 'O')

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

결과

- 입력받은 좌표에 현재 유저의 돌을 놓음.

설명

- 유효성 검사를 통과한 후 해당 좌표에 현재 유저의 돌을 놓음.

5. 현재 보드판 출력

입력

- board: 현재 보드 배열
- numCell: 게임 판 크기

결과

- 현재 상태의 보드판을 출력

설명

- 중첩 for문을 통해 board 배열을 순회하며 현재 보드 상태를 출력.
- 각 칸 사이에 | 기호를 사용하고, 가로줄 끝에 ---|---|---를 출력하여 가독성을 높임.

```
// 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) {  
            cout << " |";  
        }  
    }  
    cout << endl;  
}  
cout << "---|---|---" << endl;
```

6. 빙고 시 승자 출력 후 종료

입력

- board: 현재 보드 배열
- currentUser: 현재 유저 ('X' 또는 'O')

결과

- 빙고가 완성되면 승자를 출력하고 게임 종료

설명

- 빙고가 완성되었는지 가로, 세로, 대각선을 체크:
- 빙고가 완성되면 "플레이어 X 승리!" 또는 "플레이어 O 승리!"를 출력하고 break로 게임을 종료.

```
// 6. 빙고 체크 (가로, 세로, 대각선)
int bingo = 0;

// 가로 체크
for (int i = 0; i < numCell; i++) {
    if (board[i][0] == currentUser && board[i][1] == currentUser && board[i][2] == currentUser) {
        bingo = 1;
        break;
    }
}

// 세로 체크
for (int i = 0; i < numCell; i++) {
    if (board[0][i] == currentUser && board[1][i] == currentUser && board[2][i] == currentUser) {
        bingo = 1;
        break;
    }
}

// 대각선 체크
if (board[0][0] == currentUser && board[1][1] == currentUser && board[2][2] == currentUser) {
    bingo = 1;
}
if (board[0][2] == currentUser && board[1][1] == currentUser && board[2][0] == currentUser) {
    bingo = 1;
}

if (bingo == 1) {
    cout << "플레이어 " << currentUser << " 승리!" << endl;
    break; // 게임 종료
}
```

7. 모든 칸이 찼으면 종료

입력

- board: 현재 보드 배열
- numCell: 게임 판 크기

결과

- 모든 칸이 찼다면 무승부 메시지를 출력하고 게임 종료

설명

- 이중 for문을 사용하여 보드를 순회하며 빈 칸이 있는지 체크.
- 빈 칸이 없으면 "무승부입니다. 게임이 종료됩니다." 메시지를 출력하고 break로

```
// 7. 보드가 꽉 찼는지 확인
int full = 1;
for (int i = 0; i < numCell; i++) {
    for (int j = 0; j < numCell; j++) {
        if (board[i][j] == ' ') {
            full = 0;
            break;
        }
    }
    if (full == 0) break;
}

if (full == 1) {
    cout << "무승부입니다. 게임이 종료됩니다." << endl;
    break; // 게임 종료
}
```

게임을 종료.

4. 테스트

- 누구의 차례인지 출력

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

- 좌표 입력 받기

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

- 입력 받은 좌표 유효성 체크

두번째 유저 (0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 4 1
4, 1: x와 y 둘 중 하나가 칸을 벗어납니다.

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

- 입력 좌표에 O, X 놓기, 현재 보드판 출력

---	---	---
	0	X
---	---	---
---	---	---
---	---	---

- 빙고 시 승자 출력 후 종료

---	---	---
	0	X
---	---	---
	0	X
---	---	---
		X
---	---	---

플레이어 X 승리!

- 모든 칸이 찼으면 종료

```
---|---|---
X  |O  |X
---|---|---
X  |O  |O
---|---|---
O  |X  |X
---|---|---
무승부입니다. 게임이 종료됩니다.
```

최종 테스트 스크린샷

```
첫번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
---|---|---
X  |O  |X
---|---|---
O  |X  |O
---|---|---
O  |X  |X
---|---|---
플레이어 X 승리!
```

결과 및 결론

Tic Tac Toe 게임 완성!

어렵지만 할만합니다!