

1. 서론

-반복문 조건문 배열 등 4주차까지 배운 내용에 대한 실습을 위한 프로그램 코드 작성

-목표: Tic Tac Toe 게임 구현

2. 요구사항

-사용자 요구사항: 두명의 사용자가 번갈아가며 O와 X를 놓기

-기능 요구사항: 누구의 차례인지 출력, 좌표 입력 받기, 입력 받은 좌표 유효성 체크, 좌표에 O/X 놓기, 현재 보드판(보드판은 2차원 배열 사용) 출력, 빙고 시 승자 출력 후 종료, 모든 칸이 찼으면 종료 등

3. 설계 및 구현

-2차원 배열 생성 및 초기화

```
int main(){
    const int numCell = 3;
    char board[numCell][numCell]{};
    int x, y; // 사용자에게 입력받는 x, y 좌표를 저장할 변수

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

-번갈아 가며 놓기 위해 k를 선언하고 증가시켜서 나머지를 통해 번갈아가며 차례를 출력

```
// 게임하는 코드
int k = 0; // 누구 차례인지 체크하기 위한 변수
char currentUser = 'x'; // 현재 유저의 돌을 저장하기 위한 문자 변수
while (true) {
    //1. 누구 차례인지 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;
    }
}
```

-좌표 입력 받기

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

-배열 크기를 넘거나 이미 차있는 곳에는 놀 수 없도록 설정

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

-좌표와 유저에 따른 돌 설정

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

-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;
k++;
```

-조건(한 줄이 만들어졌는지) 확인 후 승리자 출력 및 종료

```
// 빙고 시 승자 출력 후 종료 (가로 세로 대각선 비교 but 빈칸일 때도 작동하는 오류)
/*00 10 20
01 11 21
02 12 22*/
if(k>5){
    if(((currentUser==board[0][0]) && (board[0][0]==board[0][1])) && (board[0][1]==board[0][2])){
        cout << "가로로 첫째줄이 완성되었습니다. 유저("<< currentUser << ")의 승리입니다. 게임을 종료합니다." << endl;
        break;
    }else if(((currentUser==board[1][0]) && (board[1][0]==board[1][1])) && (board[1][1]==board[1][2])){
        cout << "가로로 둘째줄이 완성되었습니다. 유저("<< currentUser << ")의 승리입니다. 게임을 종료합니다." << endl;
        break;
    }else if(((currentUser==board[2][0]) && (board[2][0]==board[2][1])) && (board[2][1]==board[2][2])){
        cout << "가로로 셋째줄이 완성되었습니다. 유저("<< currentUser << ")의 승리입니다. 게임을 종료합니다." << endl;
        break;
    }else if(((currentUser==board[0][0]) && (board[0][0]==board[1][0])) && (board[1][0]==board[2][0])){
        cout << "세로로 첫째줄이 완성되었습니다. 유저("<< currentUser << ")의 승리입니다. 게임을 종료합니다." << endl;
        break;
    }else if(((currentUser==board[0][1]) && (board[0][1]==board[1][1])) && (board[1][1]==board[2][1])){
        cout << "세로로 둘째줄이 완성되었습니다. 유저("<< currentUser << ")의 승리입니다. 게임을 종료합니다." << endl;
        break;
    }else if(((currentUser==board[0][2]) && (board[0][2]==board[1][2])) && (board[1][2]==board[2][2])){
        cout << "세로로 셋째줄이 완성되었습니다. 유저("<< currentUser << ")의 승리입니다. 게임을 종료합니다." << endl;
        break;
    }else if(((currentUser==board[0][0]) && (board[0][0]==board[1][1])) && (board[1][1]==board[2][2])){
        cout << "대각선으로 한줄이 완성되었습니다. 유저("<< currentUser << ")의 승리입니다. 게임을 종료합니다." << endl;
        break;
    }else if(((currentUser==board[2][0]) && (board[2][0]==board[1][1])) && (board[1][1]==board[0][2])){
        cout << "대각선으로 한줄이 완성되었습니다. 유저("<< currentUser << ")의 승리입니다. 게임을 종료합니다." << endl;
        break;
    }
}
```

-입력횟수를 확인해서 칸이 모두 찼으면 종료.

```
// 모든 칸이 찼으면 종료
if(k==9){
    cout << "모든 칸이 찼습니다. 게임을 종료합니다." << endl;
    break;
}
```

4. 테스트

1) 기능 별 구현 사항

-누구의 차례인지 출력

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

-좌표 입력 받기

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

-입력한 좌표에 O/X 놓기

x		

-유효성 체크

두번째 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 3 3
3, 3: x와 y둘 중 하나가 칸을 벗어납니다.

두번째 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 0
0, 0: 이미 둘이 차있습니다.

-한 줄 완성 시 승리자 출력 및 게임 종료

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

x		
x	0	
x		0

세로로 첫째줄이 완성되었습니다. 유저(x)의 승리입니다. 게임을 종료합니다.

- 모두 다 차면 종료

```
첫번째 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 2
---|---|---
x | 0 | x
---|---|---
x | 0 | 0
---|---|---
0 | x | x
---|---|---
모든 칸이 찼습니다. 게임을 종료합니다.
```

2) 최종 테스트 스크린샷

```
첫번째 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 0
---|---|---
x |   | 
---|---|---
   |   | 
---|---|---
   |   | 
---|---|---
두번째 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 1
---|---|---
x |   | 
---|---|---
   | 0 | 
---|---|---
   |   | 
---|---|---
첫번째 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 1
---|---|---
x |   | 
---|---|---
x | 0 | 
---|---|---
   |   | 
---|---|---
두번째 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
---|---|---
x |   | 
---|---|---
x | 0 | 
---|---|---
   | 0 | 
---|---|---
첫번째 유저(x)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 2
---|---|---
x |   | 
---|---|---
x | 0 | 
---|---|---
x |   | 0
---|---|---
세로로 첫째줄이 완성되었습니다. 유저(x)의 승리입니다. 게임을 종료합니다.
```

5. 결과 및 결론

프로젝트 결과: tic tac toe 게임을 만들었다.

느낀점: 일단 주어진 코드를 따라치는데 시간이 너무 오래 걸려서 힘들었다. '빙고 시 승자 출력'과 '모든 칸이 찼으면 종료'라는 기능을 추가하기 위해서 조건문을 사용하면 좋겠다고 생각했다. 그래서 &&와 ||를 이용해 조건문의 조건을 설정하고 실행을 했는데 빈칸도 입력한 것으로 인식해서 이상하게 작동했다. 조금 지치고 속상했지만 ||대신 else if문을 여러개로 추가하고 조건을 조금 수정한 후 작동시켰더니 제대로 작동했다. 다행히 생각보다 오래 걸리지 않아 다행이라 생각했고 조건문의 조건들이 조금 복잡해서 아쉬웠지만 해냈다는 성취감을 느꼈다.