

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. 모든칸이 찼으면 종료
 7. 빙고시 승자 출력후 종료

3. 설계 및 구현

1. 기능별 구현 사항:

- 1) 변수 k를 이용 매 차례마다 k를 늘려 누구 차례인지 나머지로 확인

```
int k = 0; // 누구 차례인지 확인 변수
char currentUser = 'X'; // 현재유저의 둘을 저장하는 문자 변수
while (1)
{ // 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;
    }
}
```

2)

2.좌표를 입력 받음 이때 코드상의 좌표를 받음

3. 코드상의 좌표를 받지만 통상적으로 1부터 시작하기에 -1을 하여 코드가 인식하도록 하여
둘 수 없으면 그에 대한 이유를 출력

4.좌표에 -1을 하여 코드상의 좌표에 돌을 놓음

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

// 3. 입력받은 좌표 유효성 체크
if (x - 1 >= numCell || y - 1 >= numCell)
{
    cout << x << ", " << y << ": ";
    cout << "x와 y 둘 중 하나가 칸을 벗어납니다." << endl;
    continue;
}
if (board[x - 1][y - 1] != ' ')
{
    cout << x << ", " << y << ": 이미 돌이 차있습니다." << endl;
    continue;
}
// 4.입력받은 좌표에 돌 놓기
board[x - 1][y - 1] = currentUser;
```

3)

5. 보드판격좌를 ---|---|---로 출력하여 출력

6. 보드판 빈 칸이 있는지 확인하여 차면 끝냄

```

// 5.현재 보드판 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++;

// 6.보드판이 찾는지 확인
int A = 0; // 보드판 찾는지 확인 변수
for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        if (board[i][j] != ' ')
        {
            A++;
        }
    }
}
if (A == 9)
{
    cout << "보드판이 전부 찾습니다. 무승부 입니다.";
    return 0;
}
}

```

4. 테스트

1. 기능별 테스트 결과:
2. 최종 테스트 스크린 샷: