Tic Tac Toe 보고서

215118 독일지역학과 이규민

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
    - 보드판은 2차원 배열을 사용한다
    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. 누구의 차례인지 출력

```
// 게임하는 코드
  int k = 0;              // 누구 차례인지 체크하기
  char currentUser = 'X'; // 현재 유저의 돌을 저장하기 위한 문자 변수

  while (!gameWon && !boardFull) { // 빙고가 완성되거나, 모든 칸이 차지 않는 이상 계속 게임 코드 반복
    // 1. 누구 차례인지 출력
    switch (k % 2) {
    case 0:
      cout << k % 2 + 1 << "번 유저(X)의 차례입니다 -> "; // 홀수번 user = X
      currentUser = 'X';
      break;
    case 1:
      cout << k % 2 + 1 << "번 유저(O)의 차례입니다 -> "; // 짝수번 user = O
      currentUser = 'O';
```

```
      break;
    }
```

2. 좌표 입력 받기

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

3. 입력 받은 좌표 유효성 체크

```
// 3. 입력받은 좌표의 유효성 체크
    if (x >= numCell || y >= numCell) { // x혹은 y좌표가 3*3 배열보다 큰 값일 경우 유효성 검사
      cout << x << ", " << y << ": ";
      cout << " x와 y 둘 중 하나가 칸을 벗어납니다." << endl;
      continue;
    }
    if (board[x][y] != ' ') { // 지정한 x, y 좌표가 ' ' 빈 칸이 아닌 경우 (이미 돌이 차있는 경우) 유효성 검사
      cout << x << ", " << y << ": 이미 돌이 차있습니다.";
      continue;
    }
```

4. 좌표에 O / X 놓기

```
while (!gameWon && !boardFull) { // 빙고가 완성되거나, 모든 칸이 차지 않는 이상 계속 게임 코드 반복
    // 1. 누구 차례인지 출력
    switch (k % 2) {
    case 0:
      cout << k % 2 + 1 << "번 유저(X)의 차례입니다 -> "; // 홀수번 user = X
      currentUser = 'X';
      break;
    case 1:
      cout << k % 2 + 1 << "번 유저(O)의 차례입니다 -> "; // 짝수번 user = O
      currentUser = 'O';
      break;
    }

// 4. 입력받은 좌표에 현재 유저의 돌 놓기
    board[x][y] = currentUser; // X 혹은 O 돌 놓기(위의 swtich 문에 따라 결정됨)
```

5. 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;
```

6. 빙고 시 승자 출력 후 종료

```
bool gameWon = false; // 빙고 여부 체크 (초기상태는 flase 로 선언)

// 6. 빙고 확인 (가로, 세로, 대각선)
    // 가로 빙고 확인
    for (int i = 0; i < numCell; i++) {
      if (board[i][0] == currentUser && board[i][1] == currentUser &&
          board[i][2] == currentUser) {
        gameWon = true;
        break;
      }
    }

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

    // 대각선 빙고 확인
    if ((board[0][0] == currentUser && board[1][1] == currentUser &&
         board[2][2] == currentUser) ||
        (board[0][2] == currentUser && board[1][1] == currentUser &&
         board[2][0] == currentUser)) {
      gameWon = true;
    }

// 8. 빙고 시 승자 출력 후 종료 (가로, 세로, 대각선)
  if (gameWon) {
    cout << "빙고! " << currentUser << " 플레이어가 승리하였습니다." << endl;
  } else {
    cout << "무승부! 모든 칸이 찼습니다." << endl;
  }
```

7. 모든 칸이 찼으면 종료

```
bool boardFull = false; // 모든 칸이 찼는지 여부 체크 (초기상태는 flase 로 선언)

// 7. 모든 칸이 찼는지 확인
    boardFull = true;
    for (int i = 0; i < numCell; i++) {
      for (int j = 0; j < numCell; j++) {
        if (board[i][j] == ' ') {
          boardFull = false;
          break;
        }
      }
      if (!boardFull)
        break;
    }
```

4. 테스트

1. 기능 별 테스트 결과

1. 누구의 차례인지 출력

```
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 2
---|---|---
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |   |
---|---|---
2번 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
---|---|---
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |   |0
---|---|---
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 1
1, 1: 이미 돌이 차있습니다.1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 1
---|---|---
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |X  |0
---|---|---
```

2. 좌표 입력 받기

```
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 2
---|---|---
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |   |
---|---|---
2번 유저(0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
---|---|---
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |   |0
---|---|---
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 1
1, 1: 이미 돌이 차있습니다.1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 1
---|---|---
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |X  |0
---|---|---
```

3. **입력 받은 좌표 유효성 체크**

   a. **지정된 좌표를 넘어가는 경우**

```
X  |O  |
---|---|---
   |X  |
---|---|---
   |   |
---|---|---
2번 유저(O)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 0
---|---|---
X  |O  |
---|---|---
   |X  |
---|---|---
O  |   |
---|---|---
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 3 1
3, 1:  x와 y 둘 중 하나가 칸을 벗어납니다.
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
---|---|---
X  |O  |
---|---|---
   |X  |
---|---|---
O  |   |X
---|---|---
빙고! X 플레이어가 승리하였습니다.
```

   b. **이미 돌이 차있는 경우**

```
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 2
---|---|---
X  |O  |X
---|---|---
X  |O  |X
---|---|---
O  |   |
---|---|---
2번 유저(O)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
---|---|---
X  |O  |X
---|---|---
X  |O  |X
---|---|---
O  |   |O
---|---|---
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 1
1, 1: 이미 돌이 차있습니다.1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 1
---|---|---
X  |O  |X
---|---|---
X  |O  |X
---|---|---
O  |X  |O
---|---|---
```

4. **좌표에 O / X 놓기**

```
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 2
---|---|---
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |   |
---|---|---
2번 유저(O)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
---|---|---
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |   |0
---|---|---
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 1
1, 1: 이미 돌이 차있습니다.1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 1
---|---|---
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |X  |0
---|---|---
```

5. **현재 보드판 출력**

```
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 2
---|---|---
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |   |
---|---|---
2번 유저(O)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
---|---|---
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |   |0
---|---|---
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 1
1, 1: 이미 돌이 차있습니다.1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 1
---|---|---
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |X  |0
---|---|---
```

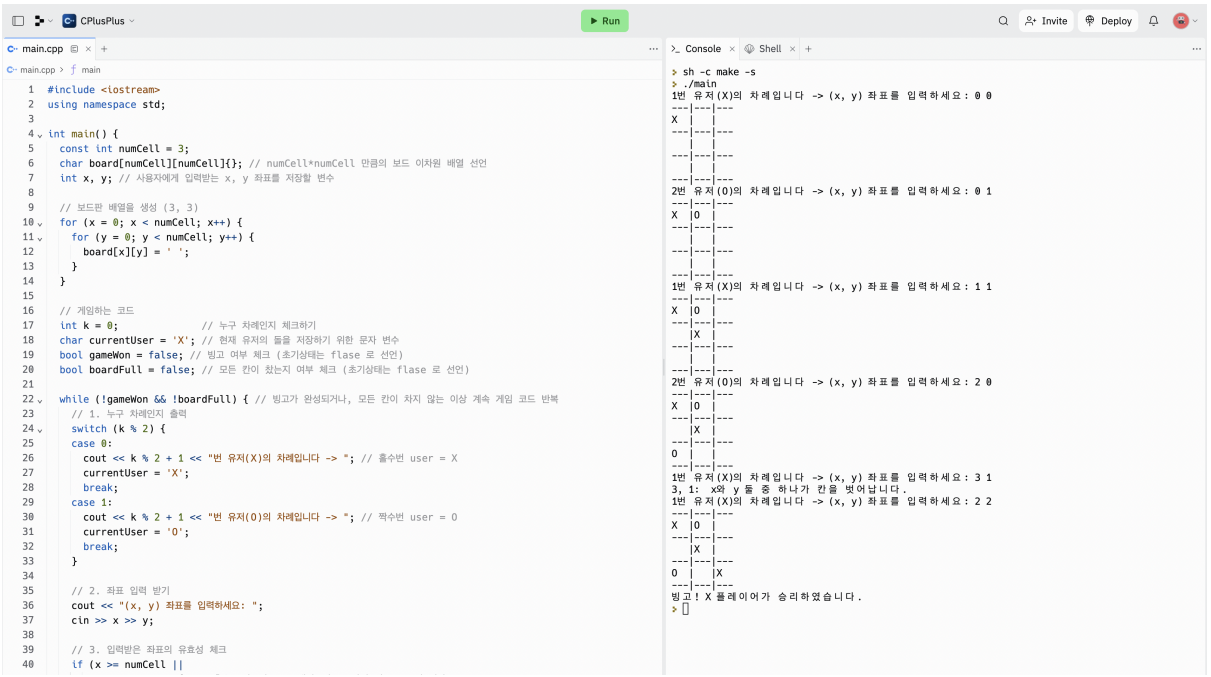
6. **빙고 시 승자 출력 후 종료**

```
X  |0  |
---|---|---
   |X  |
---|---|---
   |   |
---|---|---
2번 유저(O)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 0
---|---|---
X  |0  |
---|---|---
   |X  |
---|---|---
0  |   |
---|---|---
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 3 1
3, 1:  x와 y 둘 중 하나가 칸을 벗어납니다.
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
---|---|---
X  |0  |
---|---|---
   |X  |
---|---|---
0  |   |X
---|---|---
빙고! X 플레이어가 승리하였습니다.
```

7. **모든 칸이 찼으면 종료**

```
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |   |
---|---|---
2번 유저(O)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
---|---|---
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |   |0
---|---|---
1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 1
1, 1: 이미 돌이 차있습니다.1번 유저(X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 1
---|---|---
X  |0  |X
---|---|---
X  |0  |X
---|---|---
0  |X  |0
---|---|---
무승부! 모든 칸이 찼습니다.
```

2. 최종 테스트 스크린샷



- 기존 VSCode를 사용중인데, 빈 배열 초기화하는 문법이 C++ 11 버전 이후만 가능한 반면 VScode에서 실행시 오류를 발생시기는 이유로

  Replit으로 코드를 작성하고 실행하였습니다.

5. 결과 및 결론

1. 프로젝트 결과
   - Tic Tac Toe 게임을 요구사항에 따라 만들었고, 성공적으로 실행하였음
2. 느낀 점
   a. 입력 받은 좌표 유효성 체크
   b. 빙고 시 승자 출력 후 종료
   c. 모든 칸이 찼으면 종료
   - 위의 세 가지 내용이 특히 구현하는 데 어려움을 느꼈음
   - 최초의 '상태'를 지정해주는 것을 조금 더 머릿속에 정형화시킬 필요가 있다고 느낌
     - 예)

```
// 빙고 여부 체크 (초기상태는 flase 로 선언)
bool gameWon = false;

// 모든 칸이 찼는지 여부 체크 (초기상태는 flase 로 선언)
bool boardFull = false;
```

   - 시험이 두렵다…