

Tic tac toc 보고서

소프트웨어공학과/ 213958/ 윤상권

목차

1. 서론
2. 요구사항
3. 설계 및 구현
4. 테스트
5. 결과 및 결론

1. 서론

- 1) 프로젝트 목적 및 배경: 4주차까지 배운 내용에 대한 실습을 위해 진행
- 2) 목표: Tic Tac Toc 게임 구현

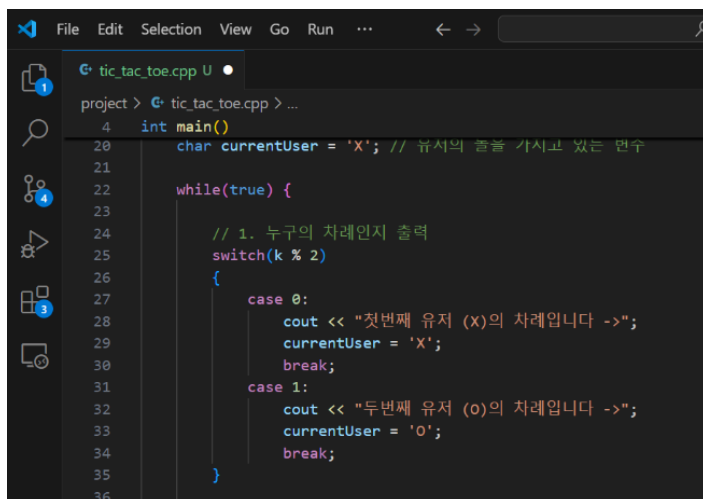
2. 요구사항

- 1) 사용자 요구사항: 두명의 사용자가 번갈아가며 O와 X를 놓기
- 2) 기능 요구사항 :
 - 누구의 차례인지 출력
 - 좌표 입력받기
 - 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) 기능 별 구현 사항

1- 누구의 차례인지 출력



```
1  tic_tac_toe.cpp U
2  project > C:\tic_tac_toe.cpp > ...
3
4  int main()
5
6      char currentUser = 'X'; // 유저의 줄을 가지고 있는 변수
7
8      while(true) {
9
10         // 1. 누구의 차례인지 출력
11         switch(k % 2)
12         {
13             case 0:
14                 cout << "첫번째 유저 (X)의 차례입니다 ->";
15                 currentUser = 'X';
16                 break;
17             case 1:
18                 cout << "두번째 유저 (O)의 차례입니다 ->";
19                 currentUser = 'O';
20                 break;
21         }
22     }
23
24
25
26
27
28
29
30
31
32
33
34
35
36
```

- 입력: k의 입력을 받는데, k는 누구의 차례인지 표시하는 함수로 짝수와 홀수에 따라 입력 받는 사용자를 달리한다.
- 결과 : 게임을 처음 진행하는 경우 k는 0인 짝수이므로 "첫번째 유저 (X)의 차례입니다 ->"를 출력하고 switch문을 벗어 난다.

- 설명 : 입력 값 k에 따라 출력되는 결과물이 달라지는 코드이다. k값이 짝수이면 첫번째 사용자의 입력 값을 'cout'으로 받고 k값이 홀수이면 두번째 사용자의 입력값을 받는다.

2- 좌표 입력 받기

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

- 입력 : 위 코드에서부터 전달되는 입력값은 없다. 다만 cin으로부터 x, y에 각각 정수를 입력받는다.
- 결과 : x, y좌표를 입력하라는 메시지가 입력되고 사용자로부터 x, y 좌표의 입력을 받는다.
- 설명 : 사용자로부터 cin을 사용하여 x,y좌표 입력을 받는다.

3- 입력받은 좌표의 유효성 체크

```
// 3. 입력 받은 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 둘이 차 있습니다." << endl;
    continue;
}
```

- 입력 : x = 좌표 x의 값, y = 좌표 y의 값이 들어 있고 'numCell = 3' 으로 위에서 선언을 했었다. board배열은 3X3 배열인 2차원 배열로 처음엔 초기값 ' '인 공백이 들어 있다.
- 결과 : 입력받은 x, y 좌표가 배열의 인덱스 안에 속하는지 유효성을 검사하고, 인덱스 밖을 벗어나면 "x와 y 둘 중 하나가 칸을 벗어납니다."를 출력한다. 또, 입력 받은 칸에 이미 다른 값이 들어 있다면 이미 둘이 차 있다는 메시지를 출력한다.
- 설명 : if문을 사용해, x와 y의 값 둘중 하나가 3보다 크거나 같으면 다음으로 오는 코드를 실행한다. board배열이 3X3 배열로 x, y의 최대 인덱스는 2이다. 그보다 넘어가면 벗어난다는 코드를 출력시키고 다시 입력을 받게 한다. 그 다음 if문은 배열 board안에

처음 초기값인 ' '이 아닌 다른 값이 들어 있으면, 다른 사용자가 그곳에 수를 두었다는 뜻이므로 두지 못하게 하는 코드이다.

4- 입력 받은 좌표에 현재 유저의 돌 놓기

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

- 입력 : board는 지금 사용자들이 수를 두는 게임 판이다. x와 y는 사용자가 둘 곳이며 currentUser은 두는 사용자에 따라 돌이 char 형식인 'X', 'O'가 순서에 맞게 저장되어 입력된다.
- 결과 : 보드 판에 돌을 놓을 곳이 정해져 위치에 돌이 입력된다.
- 설명 : 보드 배열 인덱스에 x, y 값이 입력되고, 현재 사용자에 따라 'X', 'O'가 입력된다.

5-

```
// 5. 보드 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;
```

- 입력: i, j는 numCell의 크기인 3만큼 각각 3번 반복된다.
- 결과: 사용자가 둔 돌들의 위치와 함께 바둑판이 출력된다.
- 설명: for 반복문의 이중 반복문 사용으로 3X3배열인 board의 내용을 출력한다. 'cout'을 사용해서 board의 값이 출력되는데 i와 j에 따라 인덱스 0부터 2까지의 내용들이 출력된다. 그리고 중간 중간 바둑판 모양을 나타내줄 기호들을 출력하여 구분하기 쉽게 한다.

6- 빙고 시 승자 출력 후 종료

```
// 6. 빙고 시 승자 출력 후 종료
for (int i = 0; i < 3; i++) {
    if ((board[i][0] == board[i][1] && board[i][0] == board[i][2] && board[i][0] != ' ') ||
        (board[0][i] == board[1][i] && board[0][i] == board[2][i] && board[0][i] != ' '))
    {
        cout << "축하합니다. 승리자는 " << ((k % 2 == 0) ? "1번 유저(x)" : "2번 유저(o)") << " 입니다." << endl;
        return 0;
    }
}

if ((board[0][0] == board[1][1] && board[0][0] == board[2][2] && board[0][0] != ' ') ||
    (board[0][2] == board[1][1] && board[0][2] == board[2][0] && board[0][2] != ' '))
{
    cout << "축하합니다. 승리자는 " << ((k % 2 == 0) ? "1번 유저(x)" : "2번 유저(o)") << " 입니다." << endl;
    return 0;
}
```

-입력: i엔 빙고 조건이 3칸이므로 3번 반복되기 위해 0, 1, 2의 값들이 입력된다. board에는 이전에까지 진행중이던 경기가 저장되어 있고, k에는 현재 어느 사용자의 차례인지에 대한 정보가 저장되어 있다.

-결과: 만약 조건문을 만족하여 빙고에 해당한다면 어느 사용자인지에 따라 승리자를 명시하여 출력해주고 게임을 종료한다.

-설명: 빙고 조건이 만족되기 위해서는 가로, 세로, 대각선의 경우의 수가 있다. 가로의 빙고의 경우의 수는 3개, 세로의 빙고의 경우의 수도 3개이다. 그러므로 3번 반복을 한다. 이어지는 첫번째 if문은 '||'을 기준으로 가로인 경우와 세로인 경우로 구분 지을 수 있다. 'i'의 변화되는 값을 이용하여 첫번째 줄부터 세번째 줄까지 검사를 실시한다. Board에 서로 같은 값들이 저장되어 있다면 참인데, 이때 ' '은 아닌 값들이어야 한다.

두번째 if문은 대각선 값들을 검사하는 내용으로 '||'을 기준으로 왼쪽은 좌하향의 대각선 오른쪽은 우상향하는 대각선인지 검사한다.

두개의 if문중 하나라도 참이 되면 승리자를 축하하며 게임이 종료되는데. K의 값이 짝수인지 홀수인지에 따라 빙고 조건을 달성한 사람이 누구인지 구분할 수 있다.

7- 모든 칸이 찼으면 종료

```
// 7. 모든 칸이 찼으면 종료
if (k == 8) {
    cout << "칸이 모두 찹습니다. 게임을 종료합니다." << endl;
    return 0;
}
k++;
```

-입력: k는 어느 사용자인지에 대한 차례를 구분할 수도 있지만 몇번째 수인지 구분할 수

도 있다.

-결과 : 칸이 모두 차면 게임을 종료시키고 아직 남은 칸이 있다면 게임을 계속 진행 시킨다.

-설명 : 수를 둘 때, 중복해서 둘 수 없으므로 k의 증가 값은 수를 둔 칸수에 비례한다. k의 값은 0~8까지 증가되면 총 9번으로 board의 칸이 다 차게 되어 게임이 종료된다. 만약 아직 남은 칸수가 있다면 k값을 1증가시켜 다음 순서가 오게 한다.

4. 설계 및 구현

1) 기능 별 테스트 결과

(1) 누구의 차례인지 출력

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

(2) 좌표 입력 받기

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

(3) 입력 받은 좌표의 유효성 체크

```
두번째 유저 (o)의 차례입니다 ->(x, y) 좌표를 입력하세요: 4 4
4, 4: x와 y 둘 중 하나가 칸을 벗어납니다.
두번째 유저 (o)의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 0
0, 0: 이미 돌이 차 있습니다.
두번째 유저 (o)의 차례입니다 ->(x, y) 좌표를 입력하세요: 
```

(4) 입력받은 좌표에 현재 유저의 돌 놓기

```
두번째 유저 (o)의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 1
---|---|---
x  |o  |
---|---|---
   |   |
---|---|---
   |   |
---|---|---
```

(5) 보드 출력

```

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

```

(6) 빙고 시 승자 출력 후 종료

```

---|---|---
X | O | X
---|---|---
X |   | X
---|---|---
O | O | O
---|---|---
축하합니다. 승리자는 2번 유저(o) 입니다.

```

(7) 모든 칸이 찼으면 종료

```

---|---|---
X | O | X
---|---|---
X | O | O
---|---|---
O | X | X
---|---|---
칸이 모두 찼습니다. 게임을 종료합니다.

```

2) 최종 테스트 스크린샷

(1) 가로 일 경우

```

첫번째 유저 (x)의 차례입니다 ->(x, y) 좌표를 입력하세요: 1 0
---|---|---
|   |   |
---|---|---
X  |   |
---|---|---
|   |   |
---|---|---
두번째 유저 (o)의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 0
---|---|---
O  |   |
---|---|---
X  |   |
---|---|---
|   |   |
---|---|---
첫번째 유저 (x)의 차례입니다 ->(x, y) 좌표를 입력하세요: 1 1
---|---|---
O  |   |
---|---|---
X  |X  |
---|---|---
|   |   |
---|---|---
두번째 유저 (o)의 차례입니다 ->(x, y) 좌표를 입력하세요: 2 2
---|---|---
O  |   |
---|---|---
X  |X  |
---|---|---
|   |O  |
---|---|---
첫번째 유저 (x)의 차례입니다 ->(x, y) 좌표를 입력하세요: 1 2
---|---|---
O  |   |
---|---|---
X  |X  |X
---|---|---
|   |O  |
---|---|---
축하합니다. 승리자는 1번 유저(x) 입니다.
PS C:\CPP2409>
PS C:\CPP2409> █

```

(2) 세로일 경우

```

첫번째 유저 (x)의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 2
---|---|---
|   |X  |
---|---|---
|   |   |
---|---|---
|   |   |
---|---|---
두번째 유저 (o)의 차례입니다 ->(x, y) 좌표를 입력하세요: 1 0
---|---|---
|   |X  |
---|---|---
O  |   |
---|---|---
|   |   |
---|---|---
첫번째 유저 (x)의 차례입니다 ->(x, y) 좌표를 입력하세요: 1 2
---|---|---
|   |X  |
---|---|---
O  |X  |
---|---|---
|   |   |
---|---|---
두번째 유저 (o)의 차례입니다 ->(x, y) 좌표를 입력하세요: 2 1
---|---|---
|   |X  |
---|---|---
O  |X  |
---|---|---
|   |   |
---|---|---
|O  |   |
---|---|---
첫번째 유저 (x)의 차례입니다 ->(x, y) 좌표를 입력하세요: 2 2
---|---|---
|   |X  |
---|---|---
O  |X  |
---|---|---
|   |   |
---|---|---
|O  |X  |
---|---|---
축하합니다. 승리자는 1번 유저(x) 입니다.

```


(3) 대각선 일 경우

```
첫번째 유저 (X)의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 0
---|---|---
X  |   |   |
---|---|---
   |   |   |
---|---|---
   |   |   |
---|---|---
두번째 유저 (O)의 차례입니다 ->(x, y) 좌표를 입력하세요: 0 2
---|---|---
X  |  O  |   |
---|---|---
   |   |   |
---|---|---
   |   |   |
---|---|---
첫번째 유저 (X)의 차례입니다 ->(x, y) 좌표를 입력하세요: 1 0
---|---|---
X  |  O  |   |
---|---|---
X  |   |   |
---|---|---
   |   |   |
---|---|---
두번째 유저 (O)의 차례입니다 ->(x, y) 좌표를 입력하세요: 1 1
---|---|---
X  |  O  |   |
---|---|---
X  | O  |   |
---|---|---
   |   |   |
---|---|---
첫번째 유저 (X)의 차례입니다 ->(x, y) 좌표를 입력하세요: 1 2
---|---|---
X  |  O  |   |
---|---|---
X  | O  | X  |
---|---|---
   |   |   |
---|---|---
두번째 유저 (O)의 차례입니다 ->(x, y) 좌표를 입력하세요: 2 0
---|---|---
X  |  O  |   |
---|---|---
X  | O  | X  |
---|---|---
O  |   |   |
---|---|---
축하합니다, 승리자는 2번 유저(O) 입니다.
```

(4) 승부가 나지 않는 경우

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

5. 결과 및 결론

1) 프로젝트 결과 :

Tic Tac Toc 게임을 만들며 배열과 조건문에 관한 능력을 더 키울 수 있었습니다.

2) 느낀점:

조건문을 만들 때 내가 예상치 못한 결과는 항상 있을 수 있다는 사실을 기억해야 함을 느꼈습니다. 처음 빙고 시 승자 출력 후 결과를 짤 때, 공백도 해당되어 if문이 참 값이 되어 첫번째 입력인데도 빙고라고 뜨는 문제가 발생했습니다. 이 버그를 찾는데, 다행이 오랜시간이 걸리진 않았지만, 예상치 못한 오류는 언제든지 생길 수 있다는 사실을 뼈저리게 느낄 수 있었습니다.