# 5주차 프로젝트 보고서

# Tic Tac Toe 게임

213961 김태균

# 1. 서론

- A. 프로젝트 목적 및 배경: 4주차까지 배운 내용에 대한 실습을 위해 진행
- B. 목표: Tic Tac Toe 게임 구현

## 2. 요구사항

- A. 사용자 요구사항: 두 명의 사용자가 번갈아가며 O와 X를 놓기
- B. 기능 요구사항
  - i. 보드판 초기화
  - ii. 누구의 차례인지 출력
  - iii. 좌표 입력 받기
  - iv. 입력 받은 좌표 유효성 체크
  - v. 좌표에 O / X 놓기
  - vi. 현재 보드판 출력
  - vii. 빙고 시 승자 출력 후 종료
  - viii. 모든 칸이 다 찼으면 종료
  - ix. 게임이 끝나지 않았을 경우 계속 진행

## 3. 설계 및 구현

A. 보드판 초기화

```
int main(){
const int numCell = 3;
char board[numCell][numCell]{};
int x, y;

// 보드판 초기화
// 보드만 초기화
// 보드의 인덱스를 예시와 반대로 [y][x]로 해 주어야 예시와 같은 오류가 안 생긴다고 판단했습니다
for(y = 0; y < numCell; y++)
for(x = 0; x < numCell; x++)
board[y][x] = ' ';
```

- 입력

```
      x = x 좌표 값

      y = y 좌표 값

      numCell = 가로/세로 칸 개수

      board = 보드의 값을 저장하는 2차원 배열
```

- 결과

배열의 모든 셀에 ''값을 대입한다

- 설명

보드판을 빈 칸으로 초기화한다

B. 누구의 차례인지 출력

- 입력

k = 유저 차례를 나타내는 변수 markers = 각 유저의 돌을 저장하는 1차원 배열

- 출력

현재 어떤 유저의 차례인지를 출력

- 설명

k가 짝수일 경우 첫 번째 유저, 홀수일 경우 두 번째 유저의 차례이다 switch문에서 k의 홀짝을 판별해 각 케이스 적용 k는 while문의 마지막에 1씩 증가하며 while문이 반복될 때마다 홀짝이 바뀐다

C. 좌표 입력 받기

- 입력

x = x 좌표 값 y = y 좌표 값

- 출력

입력 받은 x, y좌표 값을 출력

- 설명

입력 받은 좌표 값을 x, y에 대입한다

D. 입력 받은 좌표의 유효성 체크

- 입력

x = x 좌표 값

y = y 좌표 값

numCell = 가로/세로 칸 개수

- 출력

칸을 놓을 수 없는 이유를 출력

출력 후 while문 초반으로 이동

- 설명

사용자가 입력한 좌표가 보드판을 벗어니는지 if로 체크

사용자가 입력한 좌표에 돌이 이미 있는지 if로 체크

E. 좌표에 O / X 놓기

```
      49
      // 4. 입력받은 좌표에 현재 유저의 돌 놓기

      50
      // k값에 맞는 돌을 찾아 보드에 놓습니다

      51
      board[y][x] = markers[k % 2];
```

- 입력

x = x 좌표 값

y = y 좌표 값

k = 유저 차례를 나타내는 변수

board = 보드의 값을 저장하는 2차원 배열

- 출력: X
- 설명

k값에 맞춰 해당 차례의 유저의 돌을 좌표 값에 해당하는 board의 셀에 대입

F. 현재 보드 판 출력

- 입력

i, j = 보드판의 모든 값을 탐색하기 위한 인덱스 board = 보드의 값을 저장하는 2차원 배열

- 출력

보드판과 놓여진 각 돌을 출력

- 설명

for문을 통해 보드판의 격자무늬 배열과 놓여있는 돌을 화면에 출력한다

#### G. 빙고 시 승자 출력 후 종료

```
// 6. 빙고 시 승자 출력 후 종료
            // 가로줄 빙고 조건
             if((board[0][0] != ' ') \& (board[0][0] == board[0][1]) \& (board[0][1] == board[0][2])){
                cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
             if((board[1][0] != ' ' ) && (board[1][0] == board[1][1]) && (board[1][1] == board[1][2])){
                cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
             if((board[2][0] != ' ' ) && (board[2][0] == board[2][1]) && (board[2][1] == board[2][2])){
                cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
                break:
             // 세로줄 빙고 조건
             if((board[0][0] != ' ' ) && (board[0][0] == board[1][0]) && (board[1][0] == board[2][0])){
                cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
                break;
             if((board[0][1] != ' ') \&\& (board[0][1] == board[1][1]) \&\& (board[1][1] == board[2][1])){}
                cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
                break;
             if((board[0][2] != ' ' ) && (board[0][2] == board[1][2]) && (board[1][2] == board[2][2])){
                cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
94
                break;
             // 대각선 빙고 조건
             if((board[0][0] != ' ') && (board[0][0] == board[1][1]) && (board[1][1] == board[2][2])){
                cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
             if((board[2][0] != ' ') && (board[2][0] == board[1][1]) && (board[1][1] == board[0][2])){
                cout << k % 2 + 1 << "번째 유저의 승리입니다! " << endl;
                break;
```

- 입력

k = 유저 차례를 나타내는 변수 board = 보드의 값을 저장하는 2차원 배열

- 출력

승리한 유저를 출력한다

- 설명

if문을 통해 유저가 승리한 경우를 체크

승리한 유저가 있을 경우 k값에 해당하는 유저의 승리를 출력 후 while문 종료

H. 모든 칸이 찼으면 종료

- 입력

i, j = 보드판의 모든 값을 탐색하기 위한 인덱스

board = 보드의 값을 저장하는 2차원 배열

isFull = 보드판의 모든 칸이 다 찼는지 판별하는 bool 변수

- 출력

무승부임을 출력

- 설명

보드판의 모든 셀을 검사하며 빈 곳이 나올 경우 isFull값이 false가 된다 isFull 값이 그대로 true일 경우 모든 칸이 찬 경우이므로 while문 종료

I. 게임이 끝나지 않았을 경우 계속 진행

```
123 k++;
124 b
125 126 }
```

- 입력

k = 유저 차례를 나타내는 변수

- 출력: X
- 설명

k값이 1증가하면서 while문에 의해 다음 유저의 차례로 넘어가게 된다

## 4. 테스트

- A. 기능별 테스트 결과
  - i. 누구의 차례인지 출력

첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요: ▮

두 번째 유저 (O)의 차례입니다 -> (x, y) 좌표를 입력하세요: ▮

ii. 좌표 입력 받기

「두번째 유저 (O)의 차례입니다 →> (x, y) 좌표를 입력하세요: 1 0

iii. 입력받은 좌표 유효성 체크

첫 번째 유 저 (X)의 차 례 입 니 다 -> (x, y) 좌 표 를 입력 하 세 요 : 3 1 3, 1: x 와 y 둘 중 하 나 가 칸을 벗 어 납 니 다 .

첫 번째 유저 (X)의 차례입니다  $\rightarrow$  (x, y) 좌표를 입력하세요:  $0\ 3$ 

0, 3: x 와 y 둘 중 하나가 칸을 벗어납니다.

첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요: -1 0

-1, 0: x 와 y 둘 중 하나가 칸을 벗어납니다. 처번째 오저(Y)의 하례입니다 -> (x y) 자프를 인력하세요 0 A 1

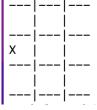
첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 1

두 번째 유저 (0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 1 0, 1: 이미 돌이 차있습니다.

두 번째 유저 (O)의 차례입니다 →> (x, y) 좌표를 입력하세요: ■

iv. 좌표에 O / X 놓기

┃첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 1



두 번째 유저 (O)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 2

0 | | ---|---|---X | | ---|--|--- v. 현재 보드판 출력

-		
10 1	ΧI	
10 17	Х	
j-	<u> </u>	

vi. 빙고 시 승자 출력 후 종료

첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 2 ---|---|---|X | ---|---|---| 0 |X | ---|---|---| 0 |X | | ---|---|---

두 번 째 유저 (0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2 ---|---|---0 |0 |0 ---|---|X | X | X | ---|---|---2번째 유저의 승리입니다!

vii. 모든 칸이 찼으면 종료

Χ	0	0
		<b> </b>
0	Х	X
Х	Х	0
무 승	부 입	니다!

### B. 최종 테스트 스크린샷

i. 가로줄 빙고 케이스

```
X | |
--- | ---- | ----
x ix i
--- | --- | ---
0 | 0
---|---|
두 번째 유저 (0)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 0
--- | ---- | ----
--- | --- | ---
x ix i
--- | --- | ---
o jo jo
---|---|
2번째 유저의 승리입니다!
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
[프로세스 완료됨]
```

```
| |0
--- i --- i ---
x ix i
--- | ---- | ----
i jo
--- | --- | ---
첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 1
---|---|
 | |0
--- | --- | ---
X \mid X \mid X
--- | --- | ---
 | |0
---|---|
1번째 유저의 승리입니다!
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
[프로세스 완료됨]
```

```
두 번째 유 저 (0)의 차 례 입 니 다 -> (x, y) 좌 표 를 입력 하 세 요 : 0 1
---|---|
X \mid X \mid
--- | --- | ---
0 0 1
--- | --- | ---
 ---|---|
첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
---|---|
x \mid x \mid x
--- | --- | ---
o jo j
--- | --- | ---
---|---|
1번째 유저의 승리입니다!
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
```

### ii. 세로줄 빙고 케이스

```
X | |
 |0 |
--- | --- | ---
X |0 |
첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 1
--- | ---- | ----
X | |
--- | --- | ---
x jo j
---|---
x jo j
---|---|
1번째 유저의 승리입니다!
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
[프로세스 완료됨]
```

```
두 번째 유저 (O)의 차례입니다 -> (x, y) 좌표를 입력하세요: 0 1
--- | ---- | ----
  i i
o ix i
--- | --- | ---
0 |X |
--- | --- |
첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 1 2
--- | ---- | ----
 |x |
--- | --- | ---
0 |X |
--- | ---- | ----
0 |X |
---|---|
1번째 유저의 승리입니다!
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
 |0 |
--- | ---- | ----
  j jx
--- | --- | ---
  jo įx
---|---|
첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
--- | ---- | ----
 jo jx
--- | --- | ---
 į įx
--- | --- | ---
 0 X
--- | --- | ---
1번째 유저의 승리입니다!
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
[프로세스 완료됨]
```

### iii. 대각선 빙고 케이스

```
0 | |
0 |X |
---|---
X | |
---|---|
첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 2
---|---|
o j jx
---|---
0 |X |
--- | ---- | ----
x | |
---|---|
1번째 유저의 승리입니다!
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
[프로세스 완료됨]
```

```
X | |
--- | ---- | ----
0 |X |
--- | --- | ---
0 | |
첫 번째 유저 (X)의 차례입니다 -> (x, y) 좌표를 입력하세요: 2 0
---|---|
X | |
--- | --- | ---
0 |X |
--- | ---- | ----
0 | |X
---|---|
1번째 유저의 승리입니다!
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
[프로세스 완료됨]
```

```
X |X |0
--- | --- |
0 |X |
--- | --- | ---
X |0 |0
--- | ---- | ----
첫 번째 유 저 (X)의 차 례 입 니 다 -> (x, y) 좌 표 를 입력 하 세 요 : 2 1
--- | ---- | ----
X |X |0
--- | ---- | ----
0 |X |X
--- | --- | ---
X |0 |0
--- | --- | ---
무 승 부 입 니 다 !
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
[프로세스 완료됨]
```

## 5. 결과 및 결론

- A. 프로젝트 결과: Tic Tac Toe 게임을 만들었음
- B. 느낀 점: 실습 과정에선 시간이 없어 깊게 고민해보지 못 했지만 정답을 보고 빙고 조건을 더 효율적으로 작성할 수 있는 방법을 알 수 있었다.