

C++프로그래밍및실습 프로젝트형 실습

지역바이오시스템공학과 생물산업기계공학전공 184128 박지환

2024년 10월 2일 | `project/week5/tic_tac_toe.cpp`

1. 서론

1-1. 프로젝트 목적 및 배경

- 본 강의에서 4주차까지 배운 내용에 대한 실습을 위해 진행하였습니다.

1-2. 프로젝트 목표

- Tic Tac Toe 게임의 구현을 목표로 하였습니다.

2. 요구사항

2-1. 사용자 요구사항

- 두 명의 사용자가 번갈아가며 O와 X를 놓는 방식의 게임입니다.

2-2. 기능 요구사항

0. 보드판은 2차원 배열을 사용합니다.
1. 누구의 차례인지 출력합니다.
2. 좌표를 입력받습니다.
3. 입력받은 좌표의 유효성을 체크합니다.
4. 좌표에 O 또는 X를 놓습니다.
5. 현재 보드판을 출력합니다.
- 6-1. 빙고 시, 승자를 출력하고 종료합니다.
- 6-2. 모든 칸이 찼다면 종료합니다.

3. 설계 및 구현

3-1. 기능 별 구현 사항

해당 기능의 코드 블록, 블록 내 입력 및 결과, 블록에 대한 설명을 나타낸 섹션입니다.

1. 2차원 배열을 사용하여 보드판을 제공하고, 사용자에게 입력받은 X, Y 좌표를 저장합니다.

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

2. X, Y 좌표를 기반으로 보드판을 초기화합니다.

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

3. 누구의 차례인지 체크하고, 현재 유저의 돌을 저장합니다.

```
int k = 0;
char currentUser = 'X';
while (true)
{
    currentUser = (k % 2 == 0) ? 'X' : 'O';
    cout << (currentUser == 'X' ? "첫 번째 유저(X)" : "두 번째 유저(O)");
    k++;
}
```

4. 좌표를 입력받습니다.

```
cout << "(X, Y) 좌표를 입력하세요: ";  
cin >> x >> y;
```

5. 입력받은 좌표의 유효성을 체크합니다.

```
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 돌이 차있습니다." << endl;  
    continue;  
}
```

6. 입력받은 좌표에 현재 유저의 돌을 놓습니다.

```
board[x][y] = currentUser;
```

7. 현재 보드판을 출력합니다.

```
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;

```

8. 게임의 승자를 확인합니다.

```
bool Winner = false;
```

8-1. 2차원 배열에서 가로에 빙고가 있는지 체크합니다.

```

for (int i = 0; i < numCell; i++)
{
    if (board[i][0] == currentUser && board[i][1] == currentUser)
    {
        Winner = true;
        break;
    }
}

```

8-2. 2차원 배열에서 세로에 빙고가 있는지 체크합니다.

```

{
    if (board[0][i] == currentUser && board[1][i] == currentUser)
    {
        Winner = true;
        break;
    }
}

```

8-3. 2차원 배열에서 대각선에 빙고가 있는지 체크합니다.

```
if (board[0][0] == currentUser && board[1][1] == currentUser)
{
    Winner = true;
}
if (board[0][2] == currentUser && board[1][1] == currentUser)
{
    Winner = true;
}
```

8-4. 승자가 결정된 경우 승자가 누구인지 출력합니다.

```
if (Winner)
{
    cout << "승자: " << currentUser << "!" << endl;
    break;
}
```

9. 모든 칸이 찼는지 확인합니다.

```
bool isFull = true;
for (int i = 0; i < numCell; i++)
{
    for (int j = 0; j < numCell; j++)
    {
        if (board[i][j] == ' ')
        {
            isFull = false;
            break;
        }
    }
}
```

9-1. 모든 칸이 찼다면 다음 내용을 출력하고 프로그램을 종료합니다.

- "모든 칸이 찼습니다. 게임 종료!"

```
if (isFull)
{
    cout << "모든 칸이 찼습니다. 게임 종료!" << endl;
    break;
}

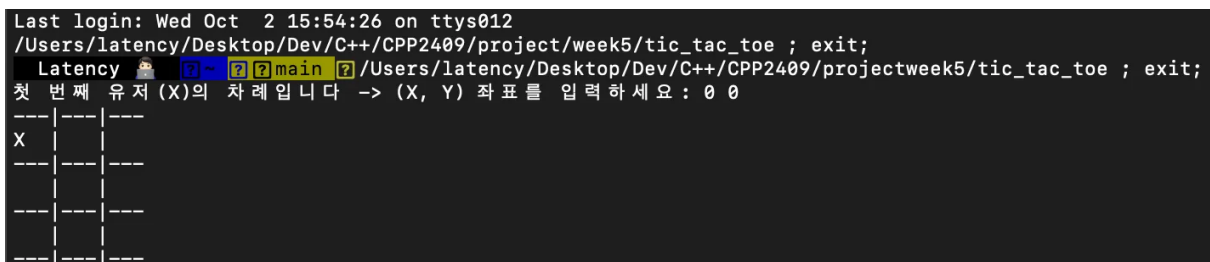
k++;
```

4. 테스트

4-1. 기능 별 테스트 결과

| *요구사항 별 스크린샷을 제공하는 섹션입니다.*

1. 2차원 배열을 사용하여 보드판을 제공하고, 사용자에게 입력받은 X, Y 좌표를 저장합니다.
2. X, Y 좌표를 기반으로 보드판을 초기화합니다.



```
Last login: Wed Oct  2 15:54:26 on ttys012
/Users/latency/Desktop/Dev/C++/CPP2409/project/week5/tic_tac_toe ; exit;
Latency ~ ? main ? /Users/latency/Desktop/Dev/C++/CPP2409/projectweek5/tic_tac_toe ; exit;
첫 번째 유저 (X)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 0 0
---|---|---
X  |   |   |
---|---|---
---|---|---
---|---|---
---|---|---
```

3. 누구의 차례인지 체크하고, 현재 유저의 돌을 저장합니다.

첫 번째 유저 (X)의 차례입니다

두 번째 유저 (O)의 차례입니다

4. 좌표를 입력받습니다.

(X, Y) 좌표를 입력하세요 :

5. 입력받은 좌표의 유효성을 체크합니다.

- 입력받은 좌표가 칸을 벗어났을 때

첫 번째 유저 (X)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 3 3
3, 3: X와 Y 둘 중 하나가 칸을 벗어납니다.

- 입력받은 좌표에 이미 돌이 있을 때

첫 번째 유저 (X)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 0 0
0, 0: 이미 돌이 차 있습니다.

6. 입력받은 좌표에 현재 유저의 돌을 놓습니다.

- 입력한 좌표인 (0, 2)에 돌을 배치합니다.

첫 번째 유저 (X)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 0 2

7. 현재 보드판을 출력합니다.

- (0, 2)에 돌이 배치된 현재 보드판을 출력합니다.

```
---|---|---  
X  |O  |X  
---|---|---  
---|---|---  
---|---|---  
---|---|---
```

8. 게임의 승자를 확인합니다.

- 2차원 배열에서 가로에 빙고가 있는지 체크합니다.

```
첫 번째 유저 (X)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 0 2
---|---|---
X  |X  |X
---|---|---
   |   |
---|---|---
0  |0  |
---|---|---
승자 : X!

Saving session...completed.
```

- 2차원 배열에서 세로에 빙고가 있는지 체크합니다.

```
첫 번째 유저 (X)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 2 0
---|---|---
X  |   |0
---|---|---
X  |   |0
---|---|---
X  |   |
---|---|---
승자 : X!

Saving session...completed.
```

- 2차원 배열에서 대각선에 빙고가 있는지 체크합니다.


```

첫 번째 유저 (X)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 2 2
---|---|---
X  |O  |O
---|---|---
   |X  |
---|---|---
   |   |X
---|---|---
승자 : X!

Saving session...completed.

```

- 승자가 결정된 경우 승자가 누구인지 출력합니다.

```

승자 : X!

Saving session...completed.

```

9. 모든 칸이 찼는지 확인합니다.

```

---|---|---
X  |O  |X
---|---|---
O  |O  |X
---|---|---
X  |X  |O
---|---|---

```

9-1. 모든 칸이 찼다면 다음 내용을 출력하고 프로그램을 종료합니다.

- "모든 칸이 찼습니다. 게임 종료!"

```

첫 번째 유저 (X)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 1 2
---|---|---
X  |O  |X
---|---|---
O  |O  |X
---|---|---
X  |X  |O
---|---|---
모든 칸이 찻습니다 . 게임 종료 !

Saving session...completed.

[Process completed]

```

4-2. 최종 테스트 스크린샷

| 프로그램 전체 동작에 대한 스크린샷을 제공하는 섹션입니다.

```

첫 번째 유저 (X)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 0 0
---|---|---
X  |   |
---|---|---
   |   |
---|---|---
   |   |
---|---|---
두 번째 유저 (O)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 0 1
---|---|---
X  | 0  |
---|---|---
   |   |
---|---|---
   |   |
---|---|---
첫 번째 유저 (X)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 0 2
---|---|---
X  | 0  | X
---|---|---
   |   |
---|---|---
   |   |
---|---|---
두 번째 유저 (O)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 1 0
---|---|---
X  | 0  | X
---|---|---
O  |   |
---|---|---
   |   |
---|---|---
   |   |
---|---|---
첫 번째 유저 (X)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 2 1
---|---|---
X  | 0  | X
---|---|---
O  |   |
---|---|---
   | X  |
---|---|---
두 번째 유저 (O)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 1 1
---|---|---
X  | 0  | X
---|---|---
O  | 0  |
---|---|---
   | X  |
---|---|---

```

```

첫 번째 유저 (X)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 2 0
---|---|---
X  | 0 | X
---|---|---
0  | 0 | 
---|---|---
X  | X | 
---|---|---
두 번째 유저 (O)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 2 2
---|---|---
X  | 0 | X
---|---|---
0  | 0 | 
---|---|---
X  | X | O
---|---|---
첫 번째 유저 (X)의 차례입니다 -> (X, Y) 좌표를 입력하세요 : 1 2
---|---|---
X  | 0 | X
---|---|---
0  | 0 | X
---|---|---
X  | X | O
---|---|---
모든 칸이 찼습니다 . 게임 종료 !

Saving session...completed.

[Process completed]

```

5. 결과 및 결론

5-1. 프로젝트 결과

- Tic Tac Toe 게임을 주어진 요구사항을 바탕으로 성공적으로 만들었습니다.

5-2. 느낀 점

- 예시로 주어진 코드에서 승자가 나왔을 때 혹은 모든 칸이 채워졌을 때 프로그램을 종료하는 코드가 주어지지 않았지만, 4주차 강의에서 배운 조건문을 사용하여 해당 조건을 충족하였을 때 프로그램을 종료하는 코드를 작성해보며 C++ 공부에 흥미가 생겼습니다.