

# Tic Tac Toe 게임

## 1.서론

### 1-1. 프로젝트 목적 및 배경

-4주차까지 배운 내용에 대한 실습을 위해 진행

### 1-2. 목표

-Tic Tac Toe 게임 구현

## 2.요구사항

2-1. 사용자 요구사항: 두명의 사용자가 번갈아가며 O와 X를 놓기

2-2. 기능 요구사항: (앞에 있는 내용

## 3.설계 및 구현

<pre>//1. 누구 차례인지 출력 switch(k % 2){     case 0:         cout &lt;&lt; "첫번째 유저(x)의 차례입니다 -&gt; ";         currentUser = 'X';         break;     case 1:         cout &lt;&lt; "두번째 유저(o)의 차례입니다 -&gt; ";         currentUser = 'O';         break; }</pre>	<p>1. 입력 k = (반복 횟수 - 1) currentUser = 현재 유저의 돌의 모양 변수</p> <p>2. 결과 현재 차례를 출력 현재 차례의 유저의 돌의 모양 대입</p> <p>3. 설명 현재 차례에 대해 출력 현재 차례의 돌의 모양으로 대입</p>
<pre>//2. 좌표 입력 받기 cout &lt;&lt; "(x, y) 좌표를 입력하세요: "; cin &gt;&gt; x &gt;&gt; y;</pre>	<p>1. 입력 x = x좌표값 y = y좌표값</p> <p>2. 결과 x, y에 값 입력</p> <p>3. 설명 x, y의 값을 사용자에게 입력 받고 대입</p>
<pre>//3. 입력받은 좌표의 유효성 체크 if(x &gt;= numCell    y &gt;= numCell){     cout &lt;&lt; x &lt;&lt; ", " &lt;&lt; y &lt;&lt; ": ";     cout &lt;&lt; " x와 y 둘 중 하나가 칸을 벗어납니다." &lt;&lt; endl;     continue; } if(board[y][x] != ' '){     cout &lt;&lt; x &lt;&lt; ", " &lt;&lt; y &lt;&lt; ": 이미 돌이 차있습니다." &lt;&lt; endl;     continue; }</pre>	<p>1. 입력 x = x좌표값 y = y좌표값 numCell = 가로/세로 칸 개수 board[][] = 빙고의 칸과 각각 대응되는 배열</p> <p>2. 결과 칸을 놓을 수 없는 이유를 출력 출력 후 while문 초반으로 이동</p> <p>3. 설명 사용자가 입력한 좌표가 게임판을 벗어나는지 if로 체크 사용자가 입력한 좌표에 돌이 이미 있는지 if로 체크</p>
<pre>//4. 입력받은 좌표에 현재 유저의 돌 놓기 board[y][x] = currentUser;</pre>	<p>1. 입력 x = x좌표값 y = y좌표값 currentUser = 현재 유저의 돌의 모양 변수 board[][] = 빙고의 칸과 각각 대응되는 배열</p> <p>2. 결과 x, y값에 해당하는 위치에 돌 생성</p>

	3. 설명 x, y값에 해당하는
<pre>//5. 현재 보드 판 출력 for(int i = numCell - 1 ; i &gt;= 0; i--){     for(int j = 0; j &lt; numCell; j++){         cout &lt;&lt; board[i][j];         cout &lt;&lt; "   ";     }     cout &lt;&lt; endl; } cout &lt;&lt; "--- --- ---" &lt;&lt; endl; cout &lt;&lt; endl;</pre>	1. 입력
	2. 결과
	3. 설명
<pre>//6-1. 가로 줄이 끝난 후 열의 끝까지 열을 풀기 위한 if문 미포함 if(k &gt;= 4){ //열은 5개의 줄이 놓인 후부터 가능     //6-1. 가로     if(board[y][0] == board[y][1] &amp;&amp; board[y][0] == board[y][2]){         switch(k % 2){             case 0:                 cout &lt;&lt; "첫번째 유저 경고!";                 return 0;             case 1:                 cout &lt;&lt; "두번째 유저 경고!";                 return 0;         }     }      //6-2. 세로     if(board[0][x] == board[1][x] &amp;&amp; board[0][x] == board[2][x]){         switch(k % 2){             case 0:                 cout &lt;&lt; "첫번째 유저 경고!";                 return 0;             case 1:                 cout &lt;&lt; "두번째 유저 경고!";                 return 0;         }     }      //6-3. 대각선     if((x + y) % 2 == 0) {         if(board[1][1] != ' ' &amp;&amp; board[0][0] == board[1][1] &amp;&amp; board[1][1] == board[2][2]){             switch(k % 2){                 case 0:                     cout &lt;&lt; "첫번째 유저 경고!";                     return 0;                 case 1:                     cout &lt;&lt; "두번째 유저 경고!";                     return 0;             }         }          if(board[1][1] != ' ' &amp;&amp; board[0][2] == board[1][1] &amp;&amp; board[1][1] == board[2][0]){             switch(k % 2){                 case 0:                     cout &lt;&lt; "첫번째 유저 경고!";                     return 0;                 case 1:                     cout &lt;&lt; "두번째 유저 경고!";                     return 0;             }         }     }      //6-4. 모든 칸이 찼을 경우     if(k == 8){         cout &lt;&lt; "모든 칸이 찼습니다.";         return 0;     } }</pre>	1. 입력
	2. 결과
	3. 설명

#### 4. 테스트

4-1. 기능 별 테스트 결과

4-2. 최종 테스트 스크린샷

#### 5. 결과 및 결론

5-1. 프로젝트 결과

5-2. 느낀 점