

# C프로그래밍 과제#3

2014726048

김정규

## -프로그램 전체 구성

- 선언한 헤더

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
```

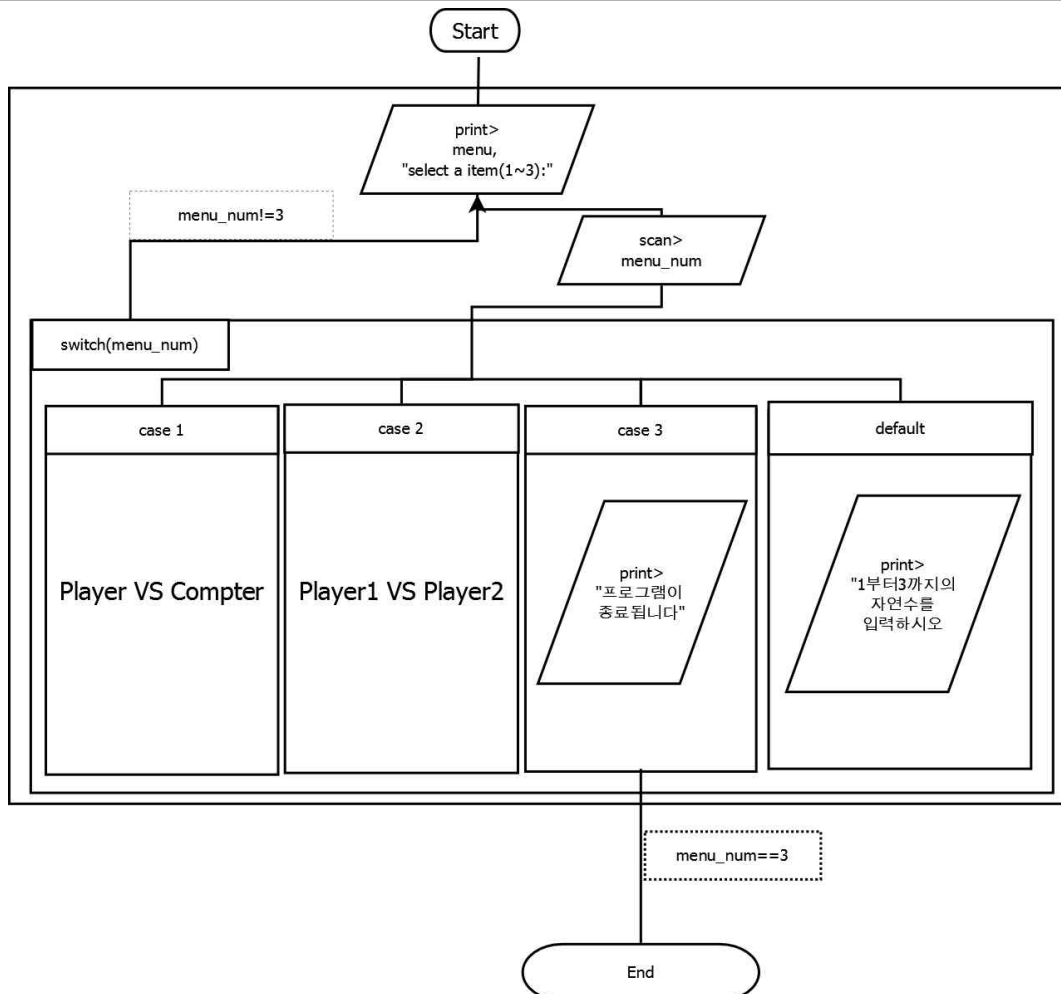
- 함수, 전역변수

```
void printBoard(); //게임판 출력
void clearBoard(); //게임판, 플레이어 순서 초기화
void Scan_Board(); //게임판 스캔(예를 들어 한 라인에 같은 말이 몇개있는가)
int winlose(); //승무패를 구분한다
void print_winlose(int* i, int menu_num); //승무패를 구분한걸 토대로 출력한다
void AI_dfs(int* x, int* y); //한 라인에 상대의 말이 2개 있는걸 감지
void AI_atk(int* x, int* y); //한 라인에 본인의 말이 2개 있는걸 감지
void blankfind(int i, int* x, int* y); //한 라인에 본인혹은 상대의 말이 두개 있으면 다른 나머지
하나 빈칸의 좌표를 찾는다
void AI(int i, int* x, int* y); //좌표를 받아서 게임판에 말을 놓는다
void Put_the_ox(int i, int* x, int* y); //좌표를 입력받아서 말을놓는다
int isPlaceable(int x, int y); //입력하거나 AI에서 받은 좌표가 말이 놓여진 좌표인지
(0~2)범위가 넘어가는지 판별해준다.
int num0[8], num1[8], num2[8]; //num0은 [ ], num1은 [O], num2는 [X], 0~2는 가로라인 세줄,
3~5는 세로라인 세줄, 6은\라인, 7은/라인
int board[3][3] = {0}; //게임판초기화
int random_start;
```

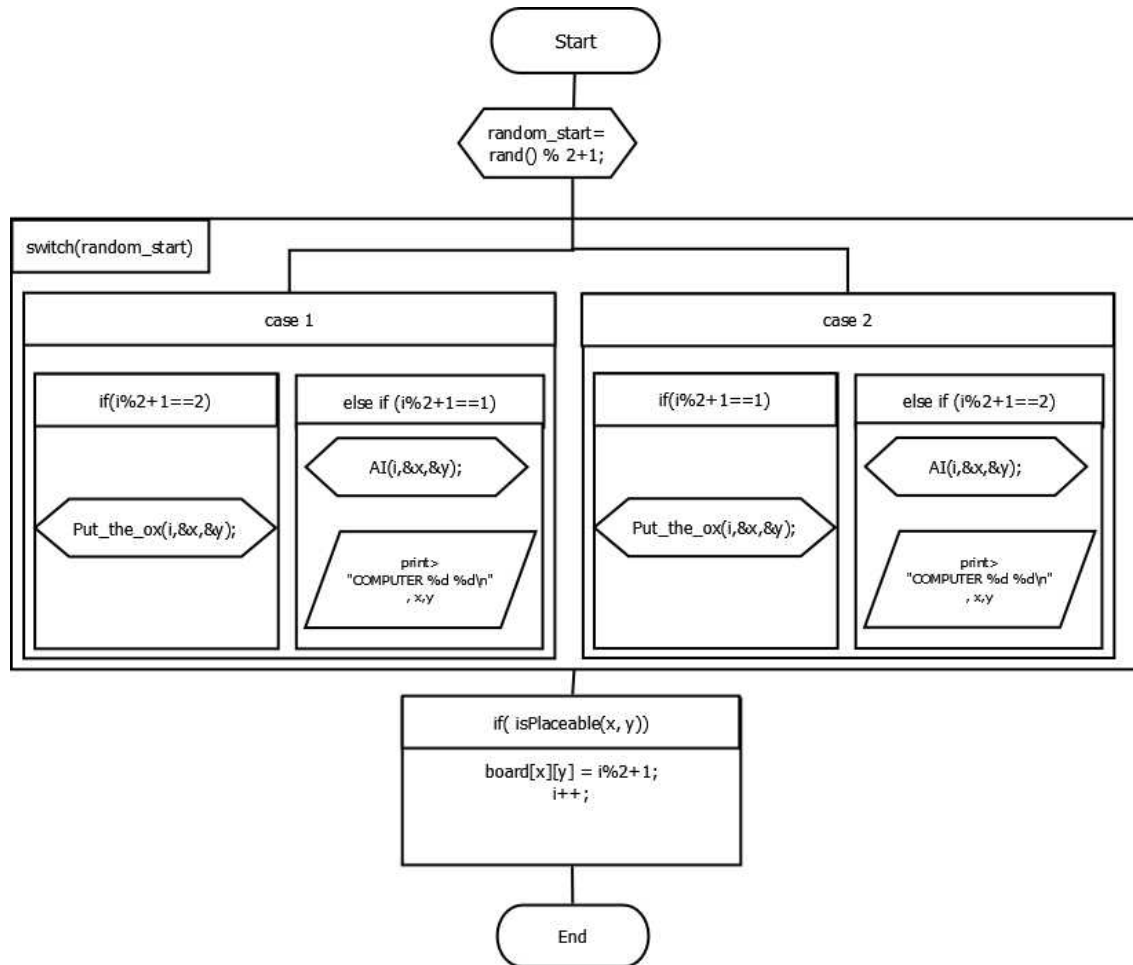
- 메인함수의 지역함수

```
int menu_num; //메뉴번호
```

```
int i, x, y; //i가 짝수면 player1, 홀수면 player2, x, y는 게임판 말의 좌표
```



## -컴퓨터 플레이어의 알고리즘



random\_start = rand()%2+1; 으로 1이나 2를 생성해서 random\_start에 대입한 후, random\_start가 1일 경우, Player먼저 턴을 시작하고, random\_start가 2일 경우, Computer가 먼저 시작을 한다.

Put\_the\_ox는 사용자에게 좌표를 입력받아서, 게임 판에 말을 놓는다. AI는 세 가지 경우에 의해 좌표를 찍는다. 한 가지 경우는 위험(한 라인에 상대팀의 말이 2개 있는 경우)이 감지되었을 때, 다른 한 가지 경우는 기회(한 라인에 본인의 말이 2개 있는 경우)가 감지되었을 때, 나머지 경우는 위 두 가지가 감지되지 않았을 때, 빈 좌표를 찾아 말을 놓는다.

그리고 행어나 그 좌표가 빈 좌표가 아니면 다시 좌표를 입력받는다.

## -각 함수의 역할

`void printBoard()`

이중 for문을 이용해서 board[3][3]의 값이 0이면 [ ], 1이면 [O], 2이면 [X]를 출력한다.

`void clearBoard()`

이중 for문을 이용해서 board[3][3]의 값을 0으로 초기화 시킨다.

`int isPlaceable( int x, int y )`

입력하거나 AI에서 받은 좌표가 말이 놓여진 좌표인지 (0~2)범위가 넘어가는지 판별해준다.

`void Scan_Board();`

게임판 스캔

num0[8]([ ]의 개수를 파악하는 변수), num1[8]([O]의 개수를 파악하는 변수), num2[8]([X]의 개수를 파악하는 변수),를 이용해서 아래 표처럼 가로 세로 대각선 각 라인의 경우일 때, [ ], [O], [X]의 개수를 파악한다.

0 ---->	3 ↓	4 ↓	5 ↓	6 ↘	7 ↙
1 ---->	↓	↓	↓		
2 ---->	↓	↓	↓		

`int winlose();`

승무패를 구분한다

num1[i]==3일 경우는 O가 한 라인에 세 개가 있는 것이다. 이럴 경우 선공한 플레이어 혹은 AI의 승리가 되고, 1을 리턴해준다.

num2[i]==3일 경우는 X가 한 라인에 세 개가 있는 것이다. 이럴 경우 후공한 플레이어 혹은 AI의 승리가 되고, 2를 리턴해준다.

게임판에 [ ]가 없으면 무승부이고, 0을 리턴해준다.

위 경우에 해당되지 않으면 3을 리턴해준다.

`void print_winlose(int* i,int menu_num);`

승무패를 구분한걸 토대로 출력한다

winlose에서 1을 받았으면 선공자의 승리라는 것을 출력해주고, 2를 받았으면 후공자의 승리라는 것을 출력해주고, 0을 받으면 무승부라고 출력해준다.

`void AI_dfs(int* x, int* y);`

선공자의 기준으로 num2[i]==2이고, num0[i]==0이면 한 라인에 상대의 말이 두 개 있다는 것이다. 이처럼 한 라인에 상대의 말이 2개 있는걸 감지한다.

`void AI_atk(int* x, int* y)`

선공자의 기준으로 num1[i]==2이고, num0[i]==0이면 한 라인에 본인의 말이 두 개 있다는 것이다. 이처럼 한 라인에 본인의 말이 2개 있는걸 감지한다.

`void blankfind(int i,int* x, int* y);`

한 라인에 본인혹은 상대의 말이 두개 있다는 것을 AI\_dfs 혹은 AI\_atk을 통해 감지를 하면 그 라인의 다른 나머지 하나 빈칸의 좌표를 for문을 이용해 찾는다

`void AI(int i,int* x, int* y);`

AI\_dfs, AI\_atk, blankfind를 이용해 좌표를 받아와 게임판에 말을 놓는다.

`void Put_the_ox(int i,int* x,int* y);`

좌표를 사용자에게 입력받아서 게임판에 말을 놓는다

## -구현의 특이사항 및 발견된 문제점

AI\_dfs, AI\_atk의 함수에 약간 문제가 있는 것 같았다. 문제의 유형은 두 가지 이다.

1. 같은 말이 한 대각선에 있을 때 위험을 인식하지 못하고 0,0자리에 말을 놓지 않는다.

```
[ ][ ][ ]
```

```
[ ][O][X]
```

```
[ ][ ][O]
```

2. 같은 말이 같은 라인에 있는 경우 중 가운데가 빈칸일 경우 가끔 인식을 하지 못하는 경우가 있어 그 자리에 말을 놓지 않는다.

```
[ ][ ][O]
```

```
[ ][X][ ]
```

```
[ ][ ][O]
```

이 두 가지의 문제 때문에 엄청 똑똑한 AI가 될 수가 없었다.

## - 실행 화면

```
=====
=====
=== TicTacToe<틱택토>게임 ===
=====
=== 게임의 모드를 선택하시오 ===
=====
=== 1. 1인용(User VS Computer) ===
=== 2. 2인용(Player1 VS Player2) ===
=== 3. TicTacToe 종료 =====
Select an item <1-3>:
1
[ ][ ][ ]
[ ][ ][ ]
[ ][ ][ ]
COMPUTER 1 1
[ ][ ][ ]
[ ][O][ ]
[ ][ ][ ]
Player 2:2
2
[ ][ ][ ]
[ ][O][ ]
[ ][ ][X]
COMPUTER 2 1
[ ][ ][ ]
[ ][O][ ]
[ ][O][X]
Player 2:0
1
[ ][X][ ]
[ ][O][ ]
[ ][O][X]
COMPUTER 2 0
[ ][X][ ]
[ ][O][ ]
[ ][O][X]
Player 2:0
2
[ ][X][X]
[ ][O][ ]
[ ][O][X]
=====
=====
COMPUTER 0 0
[ ][X][X]
[ ][O][ ]
[ ][O][X]
Player 2:1
0
[ ][X][X]
[ ][O][ ]
[ ][O][X]
COMPUTER 1 2
[ ][X][X]
[ ][O][O]
[ ][O][X]
무승부입니다.
=====
=====
=== TicTacToe<틱택토>게임 ===
=====
=== 게임의 모드를 선택하시오 ===
=====
=== 1. 1인용(User VS Computer) ===
=== 2. 2인용(Player1 VS Player2) ===
=== 3. TicTacToe 종료 =====
Select an item <1-3>:
2
[ ][ ][ ]
[ ][ ][ ]
[ ][ ][ ]
Player 1:1
1
[ ][ ][ ]
[ ][O][ ]
[ ][ ][ ]
Player 2:2
0
[ ][ ][ ]
[ ][O][ ]
[ ][ ][ ]
Player 1:1 2
[ ][ ][ ]
[ ][O][O]
[ ][ ][ ]
=====
=====
Player 2:1 0
[ ][ ][ ]
[ ][O][O]
[ ][ ][ ]
Player 1:0 0
[ ][ ][ ]
[ ][O][O]
[ ][ ][ ]
Player 2:2 2
[ ][ ][ ]
[ ][O][O]
[ ][ ][X]
Player 1:2 1
[ ][ ][ ]
[ ][O][O]
[ ][ ][X]
Player 2:0 1
[ ][ ][ ]
[ ][O][O]
[ ][ ][X]
Player 1:0
2
[ ][X][O]
[ ][O][O]
[ ][ ][X]
무승부입니다.
=====
=====
=== TicTacToe<틱택토>게임 ===
=====
=== 게임의 모드를 선택하시오 ===
=====
=== 1. 1인용(User VS Computer) ===
=== 2. 2인용(Player1 VS Player2) ===
=== 3. TicTacToe 종료 =====
Select an item <1-3>:
3
프로그래밍이 종료됩니다.
계속하려면 아무 키나 누르십시오 . . .
```