

# 소켓 프로그래밍

# Tic Tac Toe

박지아

• • •

빅데이터·IoT사물인터넷  
2차 프로젝트





# 목차

## Contents

• • •

• • •

### 01 프로젝트 개요

개발동기 / 기획의도 / Tic Tac Toe

### 03 프로젝트 시연

시연 영상

### 02 프로젝트 상세

구조도 / 기능소개 / 코드설명

### 04 진행과정

개발일정 / 개발환경 / 소감

# 01

## 프로젝트 개요

개발동기 / 기획의도

· · ·



빅데이터·IoT사물인터넷  
2차 프로젝트

### ♡ 심심풀이

짧은 시간 동안 무엇을 하면 좋을까?  
색다른 게 필요해

### ♡ 소켓 프로그래밍

소켓 프로그래밍을 어떻게 이용하면 좋을까?  
통신 - 채팅 - 게임

### ♡ 채팅을 이용한 보드 게임

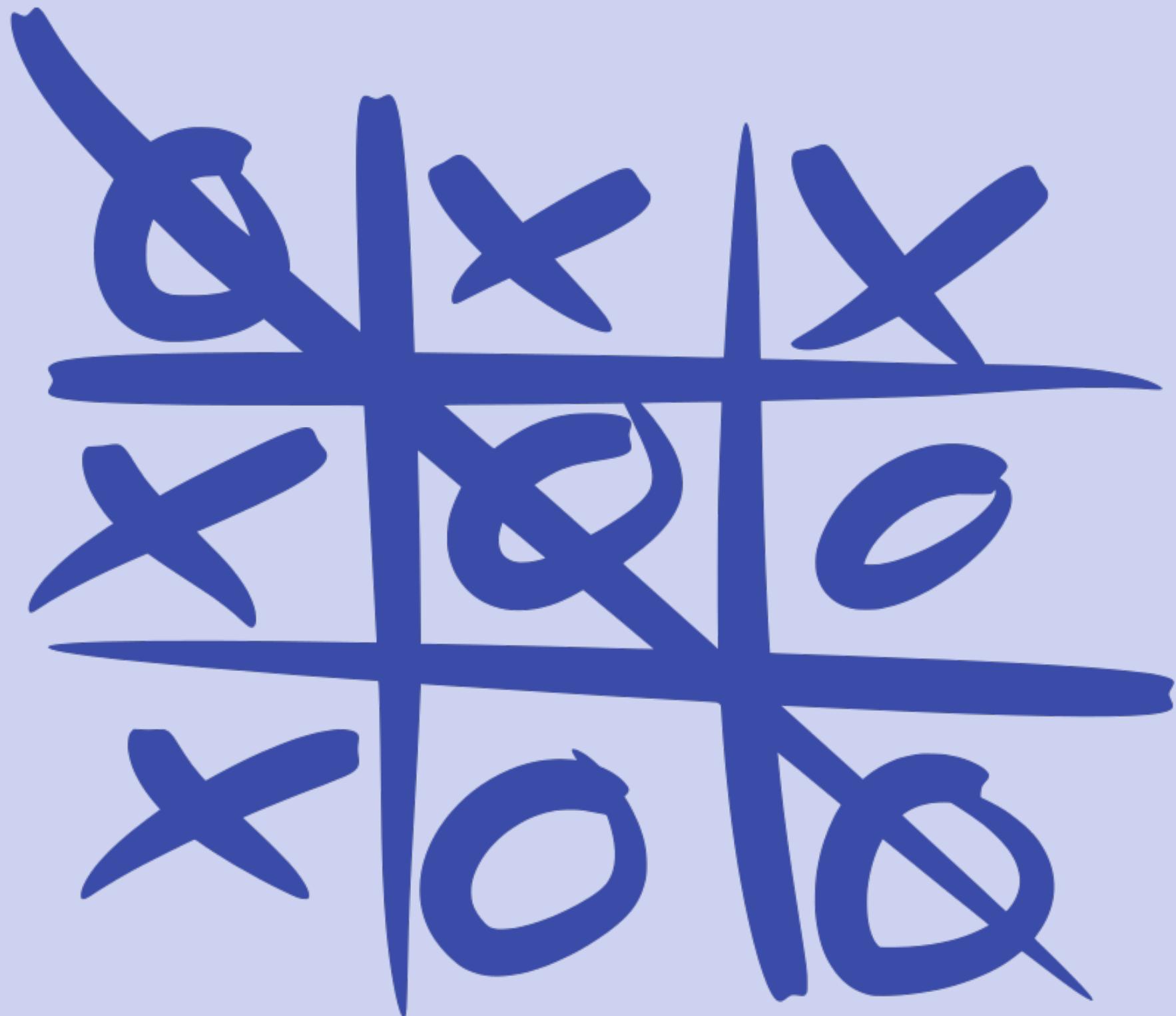
2인용  
간단한 규칙 및 조작  
짧은 시간 안에 시작하고 결과를 볼 수 있음

# Tic Tac Toe

:

2인용 게임으로,  $3 \times 3$  보드에서 가로, 세로,  
대각선 연속 세 칸을 차지하면 이기는 게임

01 프로젝트 개요



02

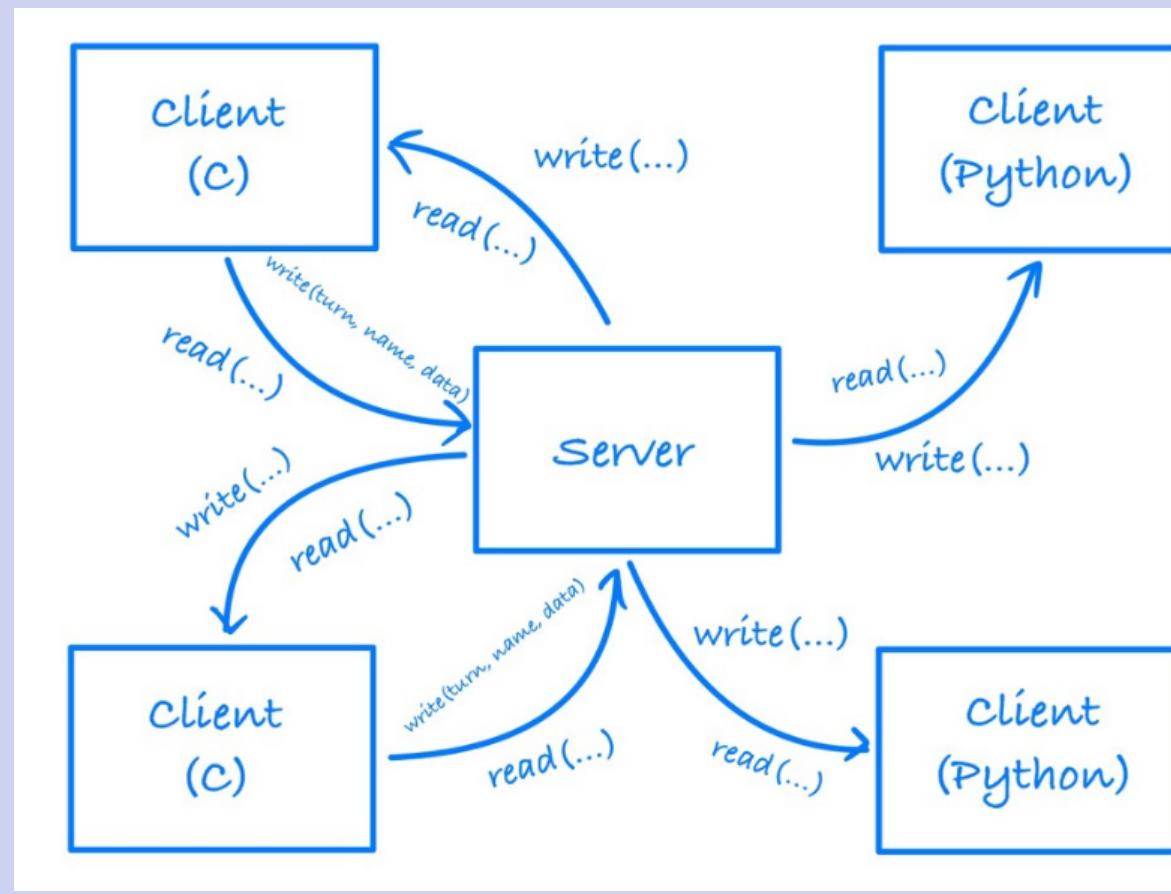
## 프로젝트 상세

구조도 / 기능소개 / 코드설명



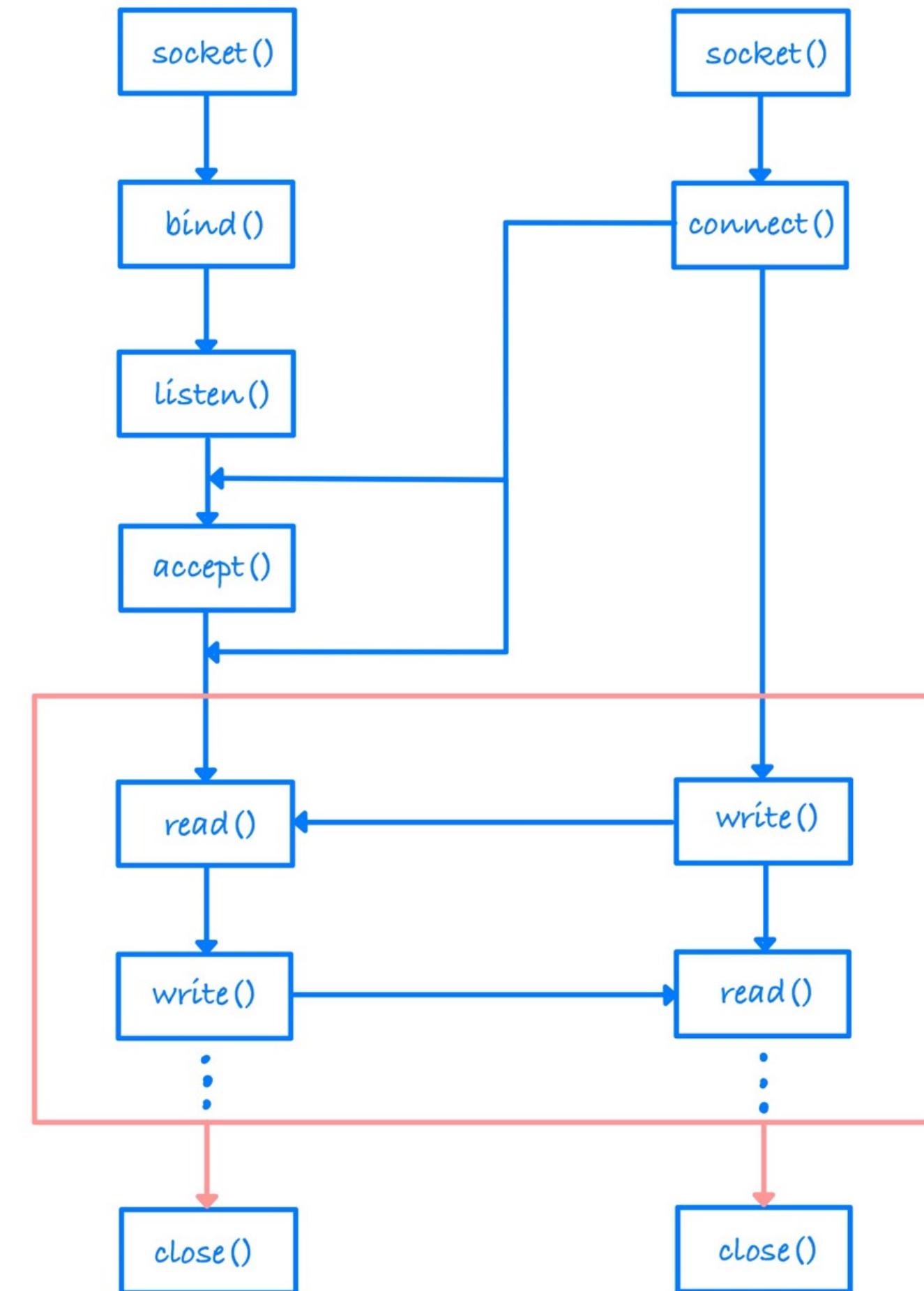
# 구조도

•  
•  
•  
TCP / IP  
1 Server  
4 Clients



02 프로젝트 상세

Server      Client

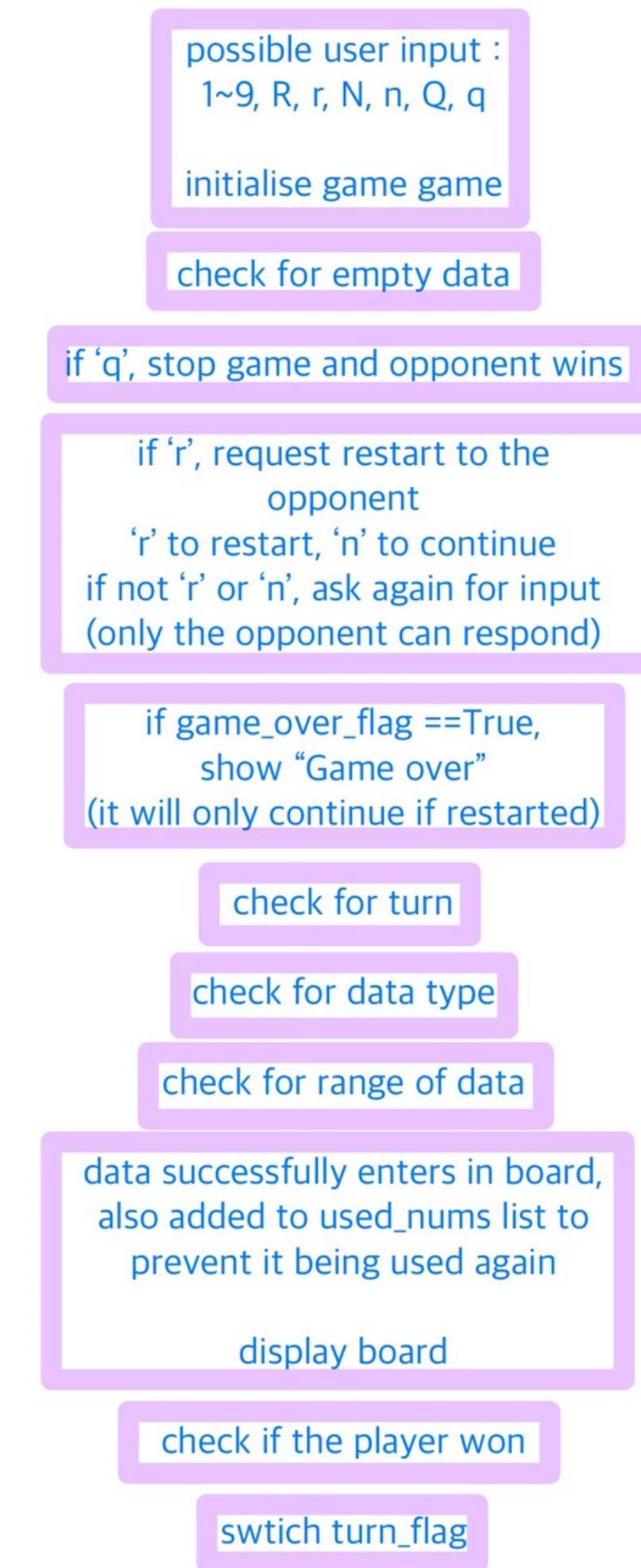


# 기능 소개

- 
- 
- 

입력 값에 따라 채우기, 재시작, 종료 가능  
재시작의 경우 양측 유저의 합의가 필요  
강제 종료시 상대측 유저 승리

02 프로젝트 상세



↑ : where continue may occur

# 기능 소개

- 
- 
- 

가로, 세로, 대각선이 완성되었는지 각각 확인

가능 결과: 1줄 완성, 2줄 완성, 무승부

종료 결과에 따라 LED 점등 여부 및 시간 결정

check for horizontal

check for vertical

check for diagonal

check for the combinations of  
hori, vert, diag

if none occurred and the  
board is full, a tie

if win or tie, game over  
LEDs blink  
blink longer if won by  
completing two lines at once



# 02

## TCTACTOE\_CLIENT.C

### 코드 설명

통신 클라이언트 서비스 제공 C 파일

# TICTACTOE\_CLIENT.C

- 
- 
- 

소켓, 주소, 스레드 초기화

인자 수, 인자값 확인

02 프로젝트 상세

```
25 int main(int argc, const char* argv[])
26 {
27     int sock = 0;
28     struct sockaddr_in serv_addr;
29     pthread_t send_thread = 0ul;
30     pthread_t receive_thread = 0ul;
31     void* thread_return = NULL; // NULL again when threads terminate
32
33     if (argc != 5) error_handling("./TICTACTOE_CLIENT5 serverIP 9999 nickname turn");
34     turn = atoi(argv[4]);
35     if (turn != 1 && turn != 2) error_handling("turn must be 1 or 2");
36     printf("Your turnID: %d\n\n", turn);
37
38     sprintf(name, "[%s]", argv[3]); // save nickname in [NAME] format in name var
```

```
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ ./TICTACTOE_CLIENT5
127.0.0.1 9999 user
./TICTACTOE_CLIENT5 serverIP 9999 nickname turn
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ ./TICTACTOE_CLIENT5
127.0.0.1 9999 user 3
turn must be 1 or 2
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ ./TICTACTOE_CLIENT5
127.0.0.1 9999 user j
turn must be 1 or 2
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ |
```

# TICTACTOE\_CLIENT.C

```
83 void* receive_msg(void* args)
84 {
85     int sock = *((int*)args);
86     char name_msg[NAME_SIZE + BUFSIZ] = {'\0', };
87     int str_len = 0;
88     int data = 0;
89
90     while (true) {
91         if ((str_len = read(sock, name_msg, NAME_SIZE + BUFSIZ - 1)) == -1) return NULL;
92
93         char* raw_msg = strtok(name_msg, " ");
94         printf("\nTurnID: %s\n", raw_msg); /* separate message : */
95
96         char* name = strtok(NULL, " ");
97         printf("%s", name);
98
99         raw_msg = strtok(NULL, " ");
100        //data = atoi(raw_msg);
101        printf(" %s\n\n", raw_msg);
102    }
}
```

# TICTACTOE\_CLIENT.C

```
62 void* send_msg(void* args)
63 {
64     int sock = *((int*)args);
65     char name_msg[NAME_SIZE + BUFSIZ] = {'\0', };
66     while(true) {
67         fgets(message, BUFSIZ, stdin);
68         sprintf(name_msg, "%d %s\t: %s", turn, name, message);
69         //fprintf(stdout, "name_msg: %s\n", name_msg);
70         write(sock, name_msg, strlen(name_msg));
71
72         if (!strcmp(message, "q\n") || !strcmp(message, "Q\n")) {
73             close(sock);
74             fputs("Bye\r\n", stdout);
75             exit(1);
76         }
    }
```



# 02

## tic-tac-toe.py

### 코드 설명

통신 클라이언트 게임 서비스 제공 PYTHON 파일

# TICTACTOE.PY

- 
- 
- 

게임 초기화

보드 초기화 및 랜덤 순서 지정

02 프로젝트 상세

```
34 def init():    # initialising a new game by star
35     global board
36     global turn_flag
37     global turn_cnt
38     global used_nums
39     global res_flag
40
41     turn_flag = randint(1, 2)      # randomly assign
42     board = [[0, 0, 0],           # init board with
43               [0, 0, 0],
44               [0, 0, 0]]
45     turn_cnt = 0
46     used_nums = []
47     res_flag = 0
48
49     print(f"Start with TurnID: {turn_flag}\n")
50
51     return 0
```

```
*****
*           *
*   O X -   *
*   O X -   *
*   - - -   *
*           *
*****
```

# TICTACTOE.PY - turn and input check

The screenshot shows two terminal windows side-by-side. The left window is titled 'TIC\_SERVER' and the right window is titled 'TIC\_CLIENT'. Both windows have a dark background and light-colored text.

**TIC\_SERVER Terminal:**

```
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ python tictactoe  
13.py  
=====
```

**Game Start**

```
Start with TurnID: 1  
  
** Not [user2]'s turn! **  
-> Your turnID: 2, This turn: 1  
  
** Empty input! (Enter R to RESTART the game) **  
** Enter an integer! (Enter R to RESTART the game) **  
-> Your input: l  
  
** Number must be between 1 and 9 **
```

**TIC\_CLIENT Terminal:**

```
l  
  
TurnID: 1  
[user1] : l  
  
54  
  
TurnID: 1  
[user1] : 54  
  
|
```

The TIC\_CLIENT terminal shows two messages from the server: one where it expects an integer input and one where it expects a turn ID. The user has responded with 'l' and '54' respectively. The TIC\_SERVER terminal shows the responses and validation messages from the client.

## 02 프로젝트 상세

# TICTACTOE.PY

```
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ python tictactoe  
13.py  
=====  
  
Game Start  
Start with TurnID: 1  
  
** Not [user2]'s turn! **  
-> Your turnID: 2, This turn: 1  
  
** Empty input! (Enter R to RESTART the game) **  
** Enter an integer! (Enter R to RESTART the game) **  
-> Your input: l  
  
** Number must be between 1 and 9 **
```

```
308     if over == 1:  
309         print("** The game is over **")  
310         continue  
311  
312     if turn_flag != turn:  
313         print(f"** Not {name}'s turn! **")  
314         print(f"-> Your turnID: {turn}, This turn: {turn_flag}\n")  
315         continue
```

## 02 프로젝트 상세

# TICTACTOE.PY

```
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ python tictactoe13.py
=====
Game Start
Start with TurnID: 1

** Not [user2]'s turn!
-> Your turnID: 2, This turn: 1

** Empty input! (Enter R to RESTART the game)
** Enter an integer! (Enter R to RESTART the game)
-> Your input: l

** Number must be between 1 and 9 **
```

```
246     while True:
247         raw_data = clnt.recv(1024).decode(encoding='utf-8') # read message
248         raw_data = raw_data.split() # split the message
249         turn = int(raw_data[0]) # save the turn
250         name = raw_data[1] # save the name
251
252         if turn not in players: # if it's not in the dictionary
253             players.update({turn : name}) # add the turn and name
254             player_num +=1 # increase the player number
255
256         if player_num == 2: # if both players have joined
257             op_name = players[switch_turn_flag(turn)] # save the opponent's name
258
259         if len(raw_data) < 4: # if the message is less than 4 bytes
260             print("** Empty input! (Enter R to RESTART the game) **") # all continue
261             continue
262
263         data = raw_data[3] # save data
```

## 02 프로젝트 상세

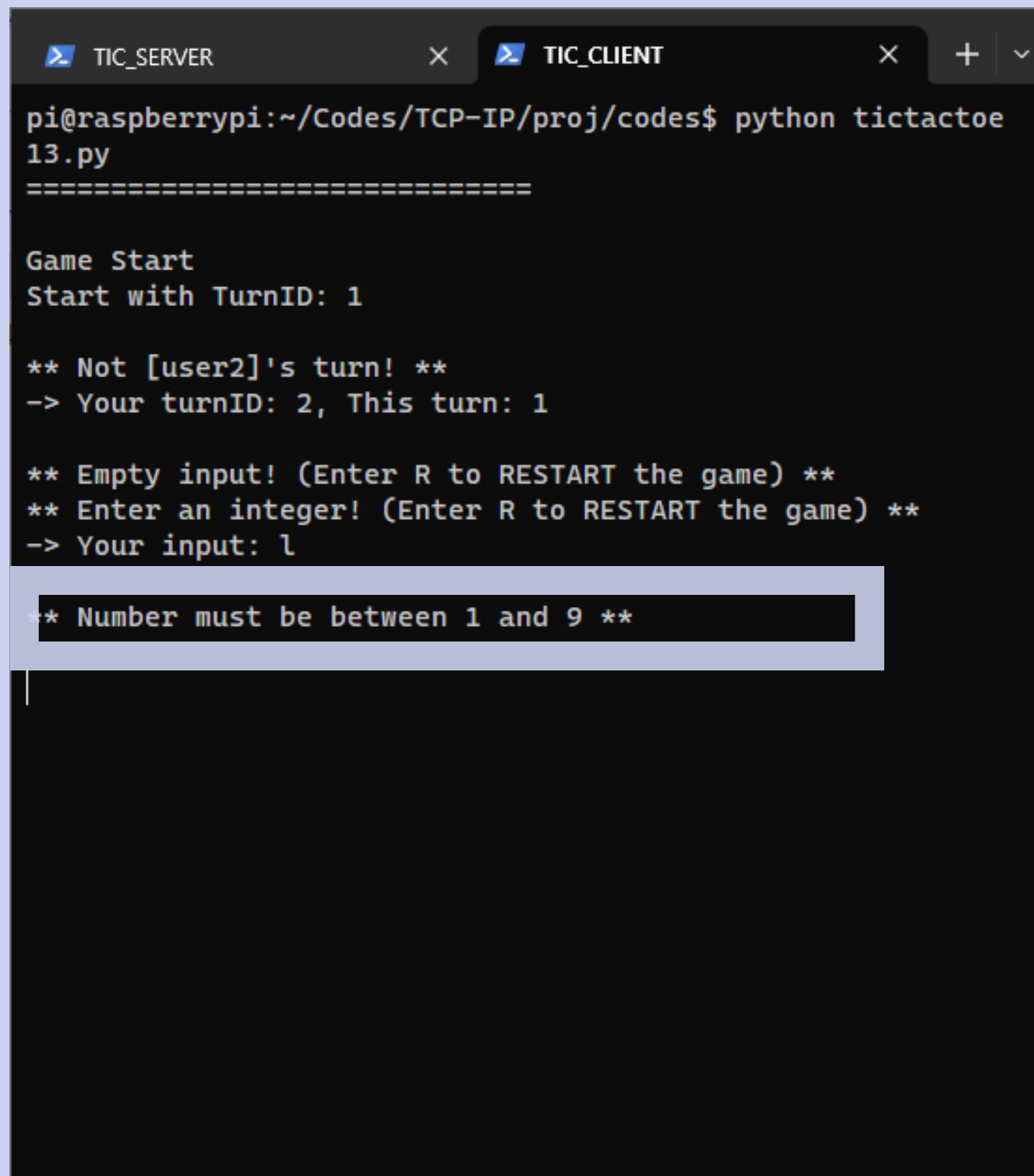
# TICTACTOE.PY

```
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ python tictactoe  
13.py  
=====  
  
Game Start  
Start with TurnID: 1  
  
** Not [user2]'s turn! **  
-> Your turnID: 2, This turn: 1  
  
** Empty input! (Enter R to RESTART the game) **  
** Enter an integer! (Enter R to RESTART the game) **  
-> Your input: l  
  
** Number must be between 1 and 9 **
```

```
323     try:                                         # check if the
324         data = int(data)
325     except ValueError:                         # it prevents f
326         print("** Enter an integer! (Enter R to RESTART the game) **")
327         print(f"--> Your input: {raw_data[3]}\n")      # shows why the
328         continue
329
330     if not 1 <= data <= 9:                      # checks if the
331         print("** Number must be between 1 and 9 **\n")
332         continue
333
334     if data in used_nums:                       # checks if the
335         print(f"** {data} is already used**\n")
336         continue
337
338     turn_cnt += 1                                # if the player
```

## 02 프로젝트 상세

# TICTACTOE.PY



A terminal window titled "TIC\_SERVER" and "TIC\_CLIENT" showing the execution of the tictactoe13.py script. The output indicates a game start, player turn IDs, and an invalid input validation step where the user is prompted to enter a number between 1 and 9.

```
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ python tictactoe13.py
=====
Game Start
Start with TurnID: 1
** Not [user2]'s turn!
-> Your turnID: 2, This turn: 1
** Empty input! (Enter R to RESTART the game)
** Enter an integer! (Enter R to RESTART the game)
-> Your input: l
* Number must be between 1 and 9 **
```

```
323     try:                                         # check if the
324         data = int(data)                         # it prevents f
325     except ValueError:                          print("** Enter an integer! (Enter R to RESTART the game) **")
326         print(f"-> Your input: {raw_data[3]}\n")      # shows why the
327         continue
328
329
330     if not 1 <= data <= 9:                      # checks if the
331         print("** Number must be between 1 and 9 **\n")  # checks if the
332         continue
333
334     if data in used_nums:                        # checks if the
335         print(f"** {data} is already used**\n")      # if the player
336         continue
337
338     turn_cnt += 1
```

## 02 프로젝트 상세

# TICTACTOE.PY

```
▶ TIC_SERVER X ▶ TIC_CLIENT X + | ▾

*   - - -   *
*           *
*****  
=====  
NEXT TurnID: 2  
=====  
TurnID : 2  
Name   : [user2]  
Number : 5  
  
*****  
*   O X -   *  
*   O X -   *  
*   - - -   *  
*           *  
*****  
=====  
NEXT TurnID: 1  
** 4 is already used**
```

```
323     try:                                     # check if the
324         data = int(data)                      # it prevents f
325     except ValueError:                      # prevents f
326         print("!! Enter an integer! (Enter R to RESTART the game) !!")
327         print(f"--> Your input: {raw_data[3]}\n")      # shows why the
328         continue
329
330     if not 1 <= data <= 9:                  # checks if the
331         print("!! Number must be between 1 and 9 **\n")# checks if the
332         continue
333
334     if data in used_nums:                   # checks if the
335         print(f"** {data} is already used**\n")# if the player
336         continue
337
338     turn_cnt += 1
```

# TICTACTOE.PY - check\_win() and game\_over()

```
  TIC_SERVER      X  TIC_CLIENT      X  + | - | X
=====
NEXT TurnID: 1
** 4 is already used**
=====
TurnID : 1
Name   : [user1]
Number : 7

*****
*      *
*  0 X -  *
*  0 X -  *
*  0 - -  *
*      *
*****
Congratulations! [user1] won by completing a vertical line!
Game Over
Enter R to start a new game
=====
```

TurnID: 1  
[user1] : 4

7

TurnID: 1  
[user1] : 7

TurnID: 1  
[user1] : 4

TurnID: 1  
[user1] : 7

## 02 프로젝트 상세

# TICTACTOE.PY

```
▶ TIC_SERVER × ▶ TIC_CLIENT × + ▾
=====
NEXT TurnID: 1
** 4 is already used**
=====
TurnID : 1
Name   : [user1]
Number : 7

*****
*      *
*  O X - *
*  O X - *
*  O - - *
*      *
*****
Congratulations! [user1] won by completing a vertical line !
Game Over

Enter R to start a new game
=====
```

```
359     # check for win
360     if turn_cnt >= 5:
361         win_msg = check_win(turn, name)
362         if win_msg != 0:
363             over = game_over(clnt, name, win_msg)
364         else:
365             print("=====\n")
366             print(f"NEXT TurnID: {switch_turn_flag(turn_flag)}\n")
367     else:
368         print("=====\n")
369         print(f"NEXT TurnID: {switch_turn_flag(turn_flag)}\n")
370
371     turn_flag = switch_turn_flag(turn_flag)      # switch
```

## 02 프로젝트 상세

# TICTACTOE.PY

```
▶ TIC_SERVER × ▶ TIC_CLIENT × + ▾
=====
NEXT TurnID: 1
** 4 is already used**

=====
TurnID : 1
Name   : [user1]
Number : 7

*****
* [ 0 | X | - | * |
* [ 0 | X | - | * |
* [ 0 | - | - | * |
* [ - | - | - | *
*****
Congratulations! [user1] won by completing a vertical line
!
Game Over
Enter R to start a new game
=====
```

```
359     # check for win
360     if turn_cnt >= 5:
361         win_msg = check_win(turn, name)
362         if win_msg != 0:
363             over = game_over(clnt, name, win_msg)
364         else:
365             print("=====\\n")
366             print(f"NEXT TurnID: {switch_turn_flag(turn_flag)}\n")

175 def check_win(turn, name):          # from turn
176     win_msg = "Congratulations! " + name + " won by completing "
177     hori = False
178     vert = False
179     diag_159 = False
180     diag_357 = False
181     # horizoantal win
182     if check_horizontal(turn):      # if a horizontal
183         hori = True
184
185     # vertical win
186     if check_vertical(turn):        # if a vertical
187         vert = True
188
189     # diagonal win
190     if check_diag_159(turn):       # if a \ line
191         diag_159 = True
192
193     if check_diag_357(turn):       # if a / line
194         diag_357 = True
```

## 02 프로젝트 상세

# TICTACTOE.PY

```
-----  
TIC_SERVER X TIC_CLIENT X + |  
  
=====  
NEXT TurnID: 1  
** 4 is already used**  
=====  
  
TurnID : 1  
Name   : [user1]  
Number : 7  
  
*****  
* 0 X - *  
* 0 X - *  
* 0 - - *  
* - - - *  
*****  
  
Congratulations! [user1] won by completing a vertical line!  
Game Over  
Enter R to start a new game  
=====
```

```
185     # vertical win  
186     if check_vertical(turn):  
187         vert = True  
188  
189  
190  
191  
192  
193  
194  
195  
196     if hori and vert:                      # if bot  
197         return win_msg+"horizontal and vertical lines!"  
198     elif diag_159 and diag_357:             # if bot  
199         return win_msg+"both diagonal lines!"  
200     elif diag_159 or diag_357:               # if onl  
201         if hori:                            # if dia  
202             return win_msg+"diagonal and horizontal lines!"  
203         elif vert:                          # if dia  
204             return win_msg+"diagonal and vertical lines!"  
205         elif diag_159:                      # if the  
206             return win_msg+"a diagonal line! ()"  
207         else:                             # the re  
208             return win_msg+"a diagonal line! ()"  
209     elif hori:                           # if onl  
210         return win_msg+"a horizontal line!"  
211     elif vert:                           # if onl  
212         return win_msg+"a vertical line!"  
213  
214     # tie  
215     if check_tie():  
216         return "It's a tie!"  
217  
218     return 0                                # if not
```

## 02 프로젝트 상세

# TICTACTOE.PY

```
▶ TIC_SERVER × ▶ TIC_CLIENT × + ▾  
=====  
NEXT TurnID: 1  
** 4 is already used**  
=====  
TurnID : 1  
Name   : [user1]  
Number : 7  
  
*****  
* [ 0 | X | - | *  
* [ 0 | X | - | *  
* [ 0 | - | - | *  
* [ - | - | - | *  
*****  
  
Congratulations! [user1] won by completing a vertical line!  
Game Over  
  
Enter R to start a new game  
=====
```

```
# check for win  
if turn_cnt >= 5:  
    win_msg = check_win(turn, name)  
    if win_msg != 0:  
        over = game_over(clnt, name, win_msg)  
  
elif vert:  
    return win_msg+"a vertical line!"  
  
# vertical win  
if check_vertical(turn):  
    vert = True  
  
137 def check_vertical(turn):  
138     global BOARD_LEN  
139  
140     for y in range(BOARD_LEN):  
141         win = True  
142         for x in range(BOARD_LEN):  
143             if board[x][y] != turn:  
144                 win = False  
145                 break  
146             if win:  
147                 return 1  
148  
return 0
```

## 02 프로젝트 상세

# TICTACTOE.PY

```
▶ TIC_SERVER × ▶ TIC_CLIENT × + | ▾

=====
NEXT TurnID: 1
** 4 is already used**

=====
TurnID : 1
Name   : [user1]
Number : 7

*****
* [ 0 | X | - ] * 
* [ 0 | X | - ] *
* [ 0 | - | - ] *
* [ - | - | - ] *
*****
Congratulations! [user1] won by completing a vertical line!
Game Over

Enter R to start a new game
=====
```

```
# check for win
if turn_cnt >= 5:
    win_msg = check_win(turn, name)
    if win_msg != 0:
        over = game_over(clnt, name, win_msg)

92 def game_over(clnt, name, win_msg):
93     led_times = 1 # multiple base for the number
94
95     if win_msg != 0:
96         print(win_msg)
97         print("Game Over\n")
98         print("Enter R to start a new game")
99         print("=====\n")
100
101    if 'both' in win_msg or 'and' in win_msg:
102        led_times = 2
103    elif 'tie' in win_msg:
104        led_times = 0
105    else:
106        print(f"{name} won by default")
107
108    for i in range(led_times):
109        for j in range(4):
110            GPIO.output(leds[j], GPIO.HIGH)
111            stop.wait(0.1)
112            GPIO.output(leds[j], GPIO.LOW)
113            stop.wait(0.1)
114    for i in range(led_times * 5):
115        for j in range(4):
116            GPIO.output(leds[j], GPIO.HIGH)
117            stop.wait(0.1)
118            for j in range(4):
119                GPIO.output(leds[j], GPIO.LOW)
120                stop.wait(0.1)
121
122    return 1
```

# TICTACTOE.PY

```
*****  
*          *  
*  0 - -  *  
*  X X X  *  
*  - - 0  *  
*          *  
*****  
  
Congratulations! [user2] won by completing a horizontal li  
ne!  
Game Over
```

```
124 def check_horizontal(turn):  
125     global BOARD_LEN  
126  
127     for y in range(BOARD_LEN):  
128         win = True  
129         for x in range(BOARD_LEN):  
130             if board[y][x] != turn:  
131                 win = False  
132                 break  
133             if win:  
134                 return 1  
135     return 0
```

# TICTACTOE.PY

```
*****
*   * 
*   X 0 - *
*   - X 0 * 
*   - - X * 
*   * 
*****
```

Congratulations! [user2] won by completing a diagonal line  
! ()  
Game Over

```
*****
*   * 
*   - - 0 * 
*   X 0 X * 
*   0 - - * 
*   * 
*****
```

Congratulations! [user1] won by completing a diagonal line  
! (/)  
Game Over

```
150 def check_diag_159(turn):
151     global BOARD_LEN
152
153     for i in range(BOARD_LEN):
154         if board[i][i] != turn:
155             return 0
156
157     return 1
158
159 def check_diag_357(turn):
160     global BOARD_LEN
161
162     for i in range(BOARD_LEN):
163         if board[i][BOARD_LEN - 1 - i] != turn:
164             return 0
165
166     return 1
```

# TICTACTOE.PY

```
*****
*   *   *
*   O X O   *
*   X X X   *
*   O X O   *
*           *
*****
```

Congratulations! [user2] won by completing horizontal and vertical lines!

Game Over

```
*****
*   *   *
*   O X O   *
*   X O X   *
*   O X O   *
*           *
*****
```

Congratulations! [user1] won by completing both diagonal lines!

Game Over

```

196     if hori and vert:                      # if bot
197         return win_msg+"horizontal and vertical lines!"
198     elif diag_159 and diag_357:             # if bot
199         return win_msg+"both diagonal lines!"
200     elif diag_159 or diag_357:               # if onl
201         if hori:                            # if dia
202             return win_msg+"diagonal and horizontal lines!"
203         elif vert:                          # if dia
204             return win_msg+"diagonal and vertical lines!"
205         elif diag_159:                     # if the
206             return win_msg+"a diagonal line! (\)"
207         else:                             # the re
208             return win_msg+"a diagonal line! (/)"
209     elif hori:                           # if onl
210         return win_msg+"a horizontal line!"
211     elif vert:                           # if onl
212         return win_msg+"a vertical line!"
213
214     # tie
215     if check_tie():                      # if nob
216         return "It's a tie!"
217
218     return 0                            # if not
```

## 02 프로젝트 상세

# TICTACTOE.PY

```
  TIC_SERVER   X  TIC_CLIENT   X  +  
  
NEXT TurnID: 2  
=====  
TurnID : 2  
Name   : [user2]  
Number : 8  
  
*****  
*      *  
*  X X 0  *  
*  0 0 X  *  
*  X X 0  *  
*      *  
*****  
  
It's a tie!  
Game Over  
  
Enter R to start a new game  
=====  
  
** The game is over **  
** [user2] requested for a RESTART. (Enter R to accept OR  
N to reject) **  
-> RESTART request rejected
```

```
166 def check_tie():  
167     global BOARD_LEN  
168  
169     for y in range(BOARD_LEN):  
170         for x in range(BOARD_LEN):  
171             if board[y][x] == 0:  
172                 return 0  
  
196     if hori and vert:                      # if bot  
197         return win_msg+"horizontal and vertical lines!"  
198     elif diag_159 and diag_357:            # if bot  
199         return win_msg+"both diagonal lines!"  
200     elif diag_159 or diag_357:              # if onl  
201         if hori:                            # if dia  
202             return win_msg+"diagonal and horizontal lines!"  
203         elif vert:                          # if dia  
204             return win_msg+"diagonal and vertical lines!"  
205         elif diag_159:                      # if the  
206             return win_msg+"a diagonal line! (\\""  
207         else:                             # the re  
208             return win_msg+"a diagonal line! (/)"  
209     elif hori:                           # if onl  
210         return win_msg+"a horizontal line!"  
211     elif vert:                           # if onl  
212         return win_msg+"a vertical line!"  
213  
214     # tie  
215     if check_tie():  
216         return "It's a tie!"  
217  
218     return 0                                # if not
```

# TICTACTOE.PY - print\_board()

```
  TIC_SERVER      X  TIC_CLIENT      X  +  -  □  X
NEXT TurnID: 2
=====
TurnID : 2
Name   : [user2]
Number : 8

*****
*   *
*   X X O   *
*   O O X   *
*   X X O   *
*           *
*****  
It's a tie!
Game Over

Enter R to start a new game
=====

** The game is over **
** [user2] requested for a RESTART. (Enter R to accept OR
N to reject) **
-> RESTART request rejected

TurnID: 2
[user2] : r

n

TurnID: 1
[user1] : n

|  
r

TurnID: 2
[user2] : r

TurnID: 1
[user1] : n
```

## 02 프로젝트 상세

# TICTACTOE.PY

```
☒ TIC_SERVER × ☒ TIC_CLIENT × + | ↴

=====
NEXT TurnID: 1
** 4 is already used**

=====
TurnID : 1
Name   : [user1]
Number : 7

    *****
    *   *
    * 0 X -
    * 0 X -
    * 0 - -
    *       *
    *****

Congratulations! [user1] won by completing a vertical line !
Game Over

Enter R to start a new game
=====
```

```
53 def print_board(frame):
54     print()
55     for y in range(frame):
56         print("\t", end='')
57         if y == 0 or y == frame-1:
58             print("*", end='')
59             for x in range(frame-1):
60                 print("**", end=" ")
61             elif y == 1 or y == frame-2:
62                 print("*", end=' ')
63                 for x in range(frame-1):
64                     if x == frame-2:
65                         print(" *", end=' ')
66                     else:
67                         print(" ", end=' ')
68             else:
69                 print(" * ", end=' ')
70                 for x in range(frame):
71                     if x == frame-1:
72                         print(" *", end=' ')
73                     elif x <= 1 or x == frame-2:
74                         print(" ", end=' ')
75                     else:
76                         item = board[y-2][x-2]
77                         if item == 1:
78                             print("O", end=' ')
79                         elif item == 2:
80                             print("X", end=' ')
81                         else:
82                             print("-", end=' ')
83             print()
84         print()
```

# TICTACTOE.PY - over

The screenshot shows a terminal window with two tabs: `TIC_SERVER` and `TIC_CLIENT`. The `TIC_CLIENT` tab contains the following text:

```

NEXT TurnID: 2
=====
TurnID : 2
Name   : [user2]
Number : 8
*****
*      *
*  X X O  *
*  O O X  *
*  X X O  *
*      *
*****
It's a tie!
Game Over

Enter R to start a new game
=====

** The game is over **
** [user2] requested for a RESTART. (Enter R to accept OR
N to reject) **
-> RESTART request rejected
|
```

The `TIC_SERVER` tab shows the source code for the game, with specific lines highlighted in a light gray box:

```

308     if over == 1:
309         print("** The game is over **")
310         continue
311
312     if turn_flag != turn:
313         print(f"** Not {name}'s turn! **")
314         print(f"-> Your turnID: {turn}, This turn: {turn_flag}\n")
315         continue
# if the ga
# if the ga
```

# TICTACTOE.PY - restart

The screenshot shows two terminal windows: TIC\_SERVER and TIC\_CLIENT.

**TIC\_SERVER Terminal:**

- Number : 2
- \*\*\*\*\*  
\* \* \* \* \*  
\* X O - \*  
\* - - - \*  
\* - - - \*  
\* \* \* \* \*  
=====
- NEXT TurnID: 2
- 1 \*\* [user2] requested for a RESTART. (Enter R to accept OR N to reject) \*\*  
-> RESTART request rejected  
-> TurnID: 2  
\*\* [user1] requested for a RESTART. (Enter R to accept OR N to reject) \*\*  
-> Please respond to the RESTART request (R or N)  
\*\* [user2] accepted! \*\*  
-> Restarting game...  
=====
- Start with TurnID: 1

**TIC\_CLIENT Terminal:**

- TurnID: 2  
[user2] : s
- TurnID: 2  
[user2] : r
- TurnID: 2  
[user2] : s
- r
- TurnID: 2  
[user2] : r

# TICTACTOE.PY - restart

The screenshot shows two terminal windows side-by-side, both titled "TIC\_SERVER".

**Left Terminal (TIC\_SERVER):**

- Number : 2
- \*\*\*\*\*  
\* \* \* \* \*  
\* X O - \*  
\* - - - \*  
\* - - - \*  
\* \* \* \* \*  
=====
- NEXT TurnID: 2
- \*\* [user2] requested for a RESTART. (Enter R to accept OR N to reject) \*\*  
-> RESTART request rejected  
-> TurnID: 2
- \*\* [user1] requested for a RESTART. (Enter R to accept OR N to reject) \*\*  
-> Please respond to the RESTART request (R or N)  
\*\* [user2] accepted! \*\*  
-> Restarting game...  
=====
- Start with TurnID: 1

**Right Terminal (TIC\_SERVER):**

- TurnID: 2  
[user2] : s
- TurnID: 2  
[user2] : r
- TurnID: 2  
[user2] : s
- r
- TurnID: 2  
[user2] : r

# TICTACTOE.PY - restart

The screenshot displays two terminal windows side-by-side, illustrating a session between a TIC\_SERVER and a TIC\_CLIENT.

**TIC\_SERVER Window:**

- Output:

```
Number : 2

*****
*      *
*  X  O  -
*  -  -  -
*  -  -  -
*      *
*****
```

---

```
=====
```

```
NEXT TurnID: 2

** [user2] requested for a RESTART. (Enter R to accept OR
N to reject) **
-> RESTART request rejected
-> TurnID: 2
** [user1] requested for a RESTART. (Enter R to accept OR
N to reject) **
-> Please respond to the RESTART request (R or N)
** [user2] accepted! **
-> Restarting game...

=====

Start with TurnID: 1
```
- Number 3 is displayed on the left side of the server window.

**TIC\_CLIENT Window:**

- Output:

```
TurnID: 2
[user2] : s
```

```
TurnID: 2
[user2] : r
```

---

```
TurnID: 2
[user2] : s
```

```
r
```

```
TurnID: 2
[user2] : r
```

# TICTACTOE.PY - restart

Number : 2

```
*****  
*      *  
*  X  O - *  
*  - - - *  
*  - - - *  
*      *  
*****
```

=====

NEXT TurnID: 2

\*\* [user2] requested for a RESTART. (Enter R to accept OR N to reject) \*\*  
-> RESTART request rejected  
-> TurnID: 2

\*\* [user1] requested for a RESTART. (Enter R to accept OR N to reject) \*\*  
-> Please respond to the RESTART request (R or N)  
\*\* [user2] accepted! \*\*  
-> Restarting game...

=====

Start with TurnID: 1

TurnID: 2  
[user2] : s

TurnID: 2  
[user2] : r

TurnID: 2  
[user2] : s

r

TurnID: 2  
[user2] : r

4

# TICTACTOE.PY

```

278     if res_flag == 1:                                # if somebody requested for a resta
279         if res_turn_flag != turn:                   # if a player sent a message when i
280             print("=> Please wait for the other player to respond") # ask to request and sh
281             print(f"=> Waiting for {players[res_turn_flag]}")
282             continue
283
284     if data == "r" or data == "R":                  # if the person responded with r, r
285         print(f"** {name} accepted! **")
286         print("=> Restarting game...\n")
287         print("=====\n")
288         over = init()
289         continue
290
291     if data == "n" or data == "N":                  # if the person resonded with n, it
292         print("=> RESTART request rejected")       # if the request was made during th
293         if over == 0:                            # if the request was made at the en
294             print(f"=> TurnID: {turn_flag}")        # change the flag back to 0, so tha
295         else:
296             print()
297             res_flag = 0
298             #res_turn_flag = switch_turn_flag(turn)
299             continue
300
301     else:                                         # if any other letter is entered, a
302         print("=> Please respond to the RESTART request (R or N)") # if entered r, request the other p
303         continue
304
305     if data == "r" or data == "R":                  # set the flag to 1 so that the pro
306         print(f"** {name} requested for a RESTART. (Enter R to accept OR N to reject) **") # set who has to respond
307         res_flag = 1
308         res_turn_flag = switch_turn_flag(turn)
309         continue

```

5

4

2

3

1

# TICTACTOE.PY - quit

```
*      *
*****
=====
NEXT TurnID: 1
=====
TurnID : 1
Name   : [user1]
Number  : 5
*****
*      *
*  X  -  -
*  -  O  -
*  -  -  -
*      *
*****
=====

NEXT TurnID: 2
=====
[User2] won by default
** [user1] has left the game. **
Bye!
```

```
TurnID: 2
[user2] : 1

5

TurnID: 1
[user1] : 5

q
Bye
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$
```

```
TurnID: 1
[user1] : 5

TurnID: 1
[user1] : q

q
Bye
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ |
```

## 02 프로젝트 상세

# TICTACTOE.PY

```
▶ TIC_SERVER X ▶ TIC_CLIENT X + | ▾

*          *
*****
=====
NEXT TurnID: 1
=====
TurnID : 1
Name   : [user1]
Number : 5
*****
*          *
*    X  - -  *
*    -  O  -  *
*    - - -  *
*          *
*****
=====

NEXT TurnID: 2
[User2] won by default
** [user1] has left the game. **
Bye!
```

```
265     if data == "q" or data == "Q":           # if a player enters
266         if player_num == 2:                  # if both players were
267             try:                            # it sometimes causes
268                 over = game_over(clnt, op_name, over)
269                 print(f"** {name} has left the game. **") # shows who left
270             except TypeError:
271                 print(f"** {name} has left the game. (but with error) **")
272             player_num -= 1                  # decrease the number
273         else:                           # if there is only one
274             print("Bye!")
275             player_num -= 1                  # decrease the number
276         continue
```

## 02 프로젝트 상세

# TICTACTOE.PY

The image shows a terminal window with two tabs: `TIC_SERVER` and `TIC_CLIENT`. The `TIC_SERVER` tab displays game logs and code snippets for handling player exits. The `TIC_CLIENT` tab shows a user interacting with the server, starting a game, and exiting.

**TIC\_SERVER Tab:**

```
Game Over
Enter R to start a new game
=====
1
Game Over
Enter R to start a new game
=====
** [user2] has left the game. (but with error) **
^CKeyboard interrupt
^CException ignored in: <module 'threading' from '/usr/lib/python3.9/threading.py'>
Traceback (most recent call last):
  File "/usr/lib/python3.9/threading.py", line 1428, in _shutdown
    lock.acquire()
KeyboardInterrupt:
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ python tictactoe13.py
=====
```

**TIC\_CLIENT Tab:**

```
265     if data == "q" or data == "Q": # if a player enter
266         if player_num == 2: # if both players w
267             try: # it sometimes caus
268                 over = game_over(clnt, op_name, over)
269                 print(f"** {name} has left the game. **") # shows who left
270             except TypeError:
271                 print(f"** {name} has left the game. (but with error) **")
272                 player_num -= 1 # decrease the numb
273             else: # if there is only
274                 print("Bye!")
275                 player_num -= 1 # decrease the numbe
276             continue
277
278 NT5 127.0.0.1 9999 user1 1
Your turnID: 1
q
Bye
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$
```

```
q
Bye
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ ./TICTACTOE_CLIE
NT5 127.0.0.1 9999 user2 2
Your turnID: 2
TurnID: 1
[user1] : q
q
Bye
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$
```

## 02 프로젝트 상세

# TICTACTOE.PY

```
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ python tictactoe13.py
=====
Game Start
Start with TurnID: 2

Bye!
Bye!
^CKeyboard interrupt
^CException ignored in: <module 'threading' from '/usr/lib/python3.9/threading.py'>
Traceback (most recent call last):
  File "/usr/lib/python3.9/threading.py", line 1428, in _shutdown
    lock.acquire()
KeyboardInterrupt:
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ python tictactoe13.py
=====

Game Start
Start with TurnID: 1

** Enter an integer! (Enter R to RESTART the game) **
-> Your input: g

[user1] won by default
** [user2] has left the game. **

265     if data == "q" or data == "Q": # if a player enter
266         if player_num == 2: # if both players w
267             try: # it sometimes caus
268                 over = game_over(clnt, op_name, over)
269                 print(f"** {name} has left the game. **") # shows who left
270             except TypeError:
271                 print(f"** {name} has left the game. (but with error) **")
272                 player_num -= 1 # decrease the numb
273             else: # if there is only
274                 print("Bye!")
275                 player_num -= 1 # decrease the numbe
276             continue
277
278     [user2] : q
279
280     q
281     Bye
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ |
```

```
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ ./TICTACTOE_CLIENT
NT5 127.0.0.1 9999 user2 2
Your turnID: 2

TurnID: 1
[user1] : g

q
Bye
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ |
```

## 02 프로젝트 상세

# TICTACTOE.PY

The screenshot shows a terminal window with two tabs: "TIC\_SERVER" and "TIC\_CLIENT".

**TIC\_SERVER Tab:**

```
=====
TurnID : 2
Name   : [user2]
Number : 7
*****
*      *
*  X  O  -
*  X  O  -
*  X  -  -
*      *
*****
Congratulations! [user2] won by completing a vertical line!
Game Over
Enter R to start a new game
=====
```

**TIC\_CLIENT Tab:**

```
265     if data == "q" or data == "Q":          # if a player enter
266         if player_num == 2:                 # if both players w
267             try:                           # it sometimes caus
268                 over = game_over(clnt, op_name, over)
269                 print(f"** {name} has left the game. **") # shows who left
270             except TypeError:
271                 print(f"** {name} has left the game. (but with error) **")
272                 player_num -= 1           # decrease the numb
273             else:
274                 print("Bye!")
275                 player_num -= 1           # decrease the numbe
276             continue
[User2] : q
q
Bye
pi@raspberrypi:~/Codes/TCP-IP/proj/codes$ |
```

**Bottom Terminal Window:**

```
1
Game Over <----?-----?
Enter R to start a new game
=====
** [user2] has left the game. (but with error) **
```

The terminal shows a game board state where player [user2] has completed a vertical line, winning the game. The client then exits with a "q" command. The server handles the exit and continues the loop for a new game.

# TICTACTOE.PY

## - board calculation

•  
•  
•

index는 0부터

x 예시:  $1 \% 3 = 1$

y 예시:  $1 / 3 = 0$

02 프로젝트 상세

```
345     x = int(data % BOARD_LEN - 1)
346     y = int(data / BOARD_LEN)
347
348     if data % BOARD_LEN == 0:
349         x += BOARD_LEN
350         y -= 1
351
352     board[y][x] = turn
353
354     used_nums.append(data)
355
356     # show game board
357     print_board(frame)
```

```
*****
*           *
*   O X -   *
*   O X -   *
*   - - -   *
*           *
*****
```

# TICTACTOE.PY

## - board calculation

### with multiples of 3

- 
- 
- 

목표: (2, 1)

x 예시:  $6 \% 3 - 1 = -1$

y 예시:  $6 / 3 = 2$

```
345     x = int(data % BOARD_LEN - 1)
346     y = int(data / BOARD_LEN)
347
348     if data % BOARD_LEN == 0:
349         x += BOARD_LEN
350         y -= 1
351
352     board[y][x] = turn
353
354     used_nums.append(data)
355
356     # show game board
357     print_board(frame)
```

```
*****
*           *
*   O X -   *
*   O X -   *
*   - - -   *
*           *
*****
```



03

## 프로젝트 시연

시연 영상

게임 실행 및 LED 영상

### 03 프로젝트 시연

The image shows three terminal windows side-by-side:

- TIC\_SERVER**: Displays a 3x3 Tic-Tac-Toe board with the following state:

```
* X O X *
*       *
*****
=====
```

NEXT TurnID: 1

```
=====
TurnID : 1
Name   : [user1]
Number : 6
```

```
*****
*       *
* O X O *
* X O O *
* X O X *
*       *
*****
```

It's a tie!  
Game Over

```
Enter R to start a new game
=====
```
- TIC\_CLIENT**: Shows the same board state as the server.
- TESTPY**: A Python script demonstrating turn handling. It prints:

```
TurnID: 2
[user2] : 4
```

(empty line)

```
6
```

```
TurnID: 1
[user1] : 6
```

(empty line)

```
4
```

```
TurnID: 2
[user2] : 4
```

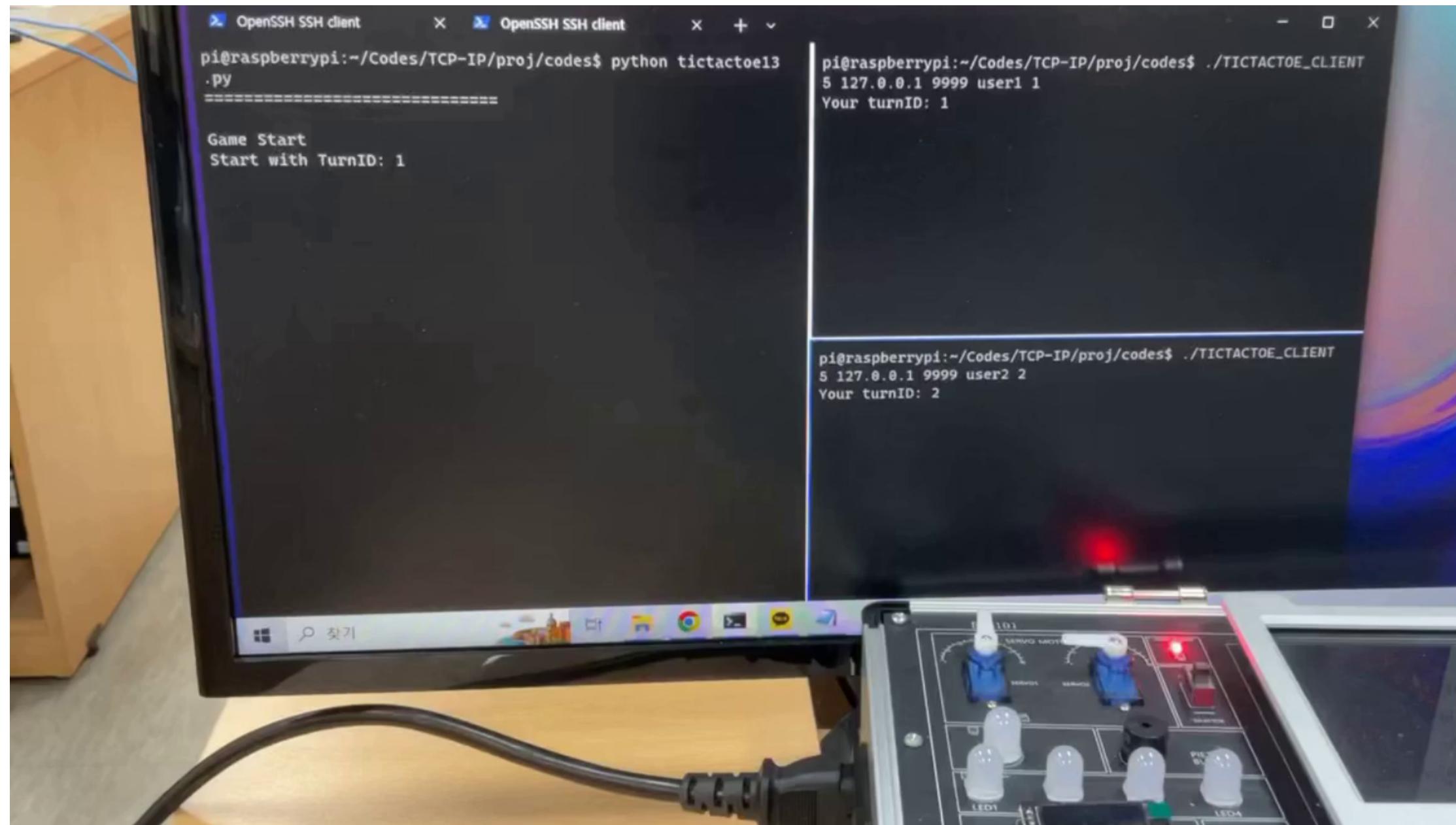
(empty line)

```
TurnID: 1
[user1] : 6
```

재시작 전: 게임 종료 후 입력, 재시작 거절, 재시작 거절 체크, 재시작 수락  
재시작 후: 재시작 요청, 요청 중 다른 키 입력, 재시작 거절, 재시작 수락



### 03 프로젝트 시연



승리 조건에 따른 LED 점등 여부 및 횟수





# 04

## 진행과정

개발일정 / 개발환경 / 소감

# 진행과정

## 개발일정



**04/19** D-2

아이디어 구상 및  
C 프로그램 개발

기존에 사용했던 server 및 client C  
소스코드를 수정하여 게임 방식에 맞  
게끔 메세지 전달 방법을 수정

**04/20** D-1

Python 프로그램 개발

Client c에서 server c를 통해 전달 받은  
메세지를 기반으로 tic tac toe 게임 서비  
스를 제공하는 Python 프로그램 개발

**04/21** D-day

Python 프로그램 개발  
마무리 및 발표 준비

개발중 지저분해진 소스코드를 정리  
하고 놓쳤던 버그를 수정  
발표 준비 시작

# 진행과정

## 개발환경



### IDE

- Windows powershell connecting to Linux environment
- vim for both C, Python

:

### Compiler

- C : GCC compiler
- Python: Default compiler by Python

:

### Libraries

- C : unistd.h, arpa/inet.h, sys/socket.h, netinet/in.h, pthread.h
- Python: socket, threading ,RPi, time, random

:

# 진행과정



## 프로젝트 진행 소감

### 배운점

- 소켓 프로그래밍을 완벽하게 제어할 수는 없지만 기본적인 구조와 기능 구현
- 서버와 클라이언트 구현을 탄탄히 해놓으면 다른 클라이언트 프로그램 개발시 수월
- 우선순위가 맞물려있는 알고리즘을 잘 풀어내는 사고력 (turn, int, q, r)

### 아쉬웠던 점

- 소켓 프로그래밍에 대한 지식이 부족. 서버 프로그램을 더 탄탄하게 만들고 싶음
- 소켓 프로그래밍 관련 코딩을 많이 하지 못함
- 시간 부족으로 원하는 기능들을 다 적용하지 못함

### 추후 개발

- Client c 프로그램 실행시 순번 1, 2 입력 제한
- 게임 종료 후 퇴장 error handling
- Client c 프로그램을 통해 Python 프로그램 디스플레이에 영향을 주지 않는 채팅 기능 구현
- 3판2선승제, 시간내 득점 등 규칙 도입
- 비슷한 규칙의 게임 - 오목, connect four

*Thank you*

감사합니다

Q&A



박지아