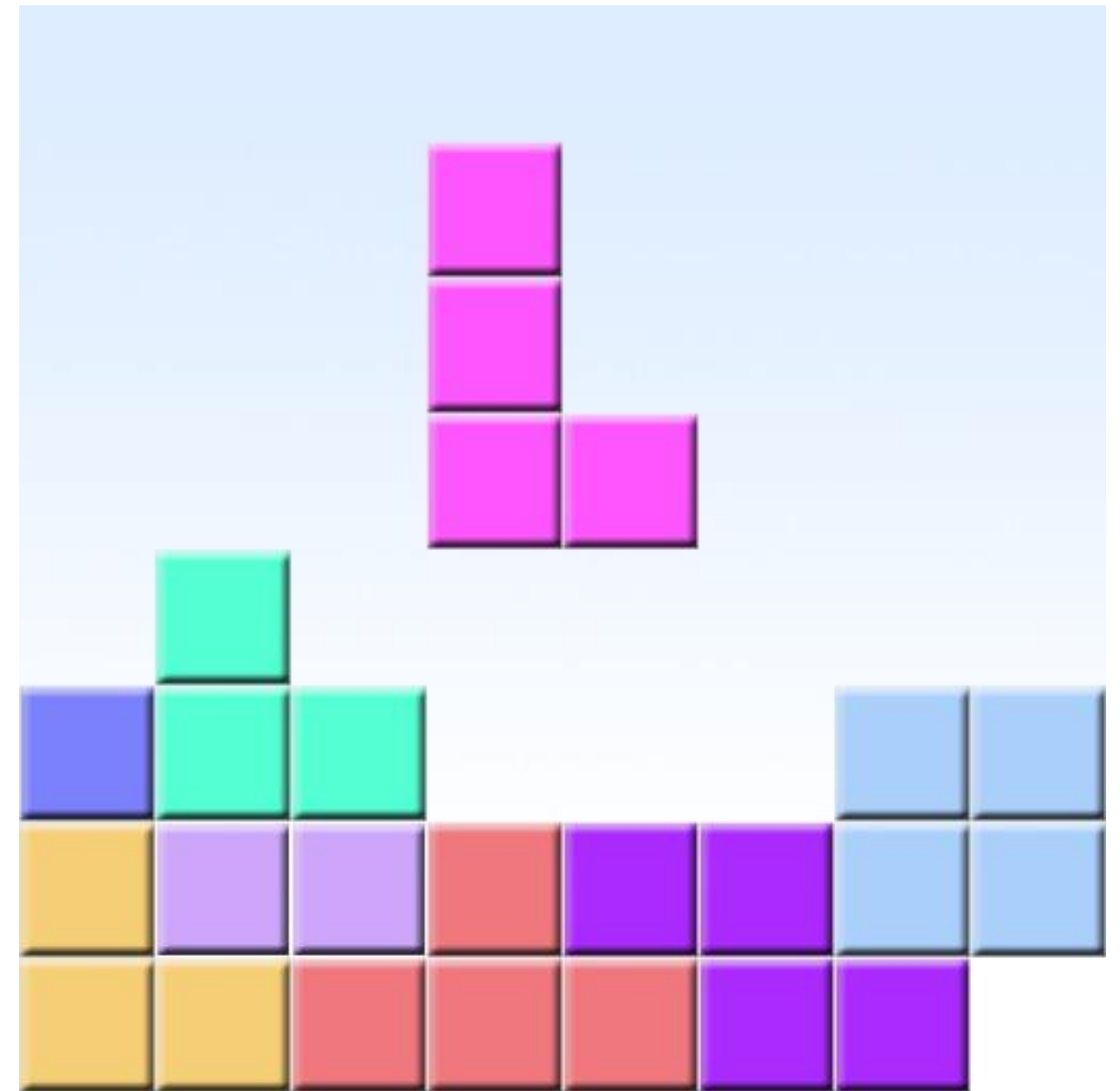


임베디드운영체제

테트리스 게임

전자공학부 임베디드시스템 전공
2019146027 이희준
2019146032 조성호
2019146037 홍석영



목 차

1. 계획 비교
2. 기능
3. 핵심 코드
4. 아쉬운점

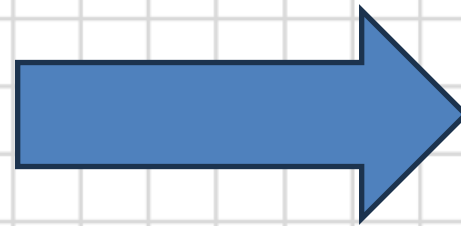




계획 비교

게임 로직은 완성했지만 합치면서
문제점 발생

서버 독창성(USER LOGIN + QUEST +
ROOM + CHAT) 로직은
완성했지만 합치면서 문제점 발생



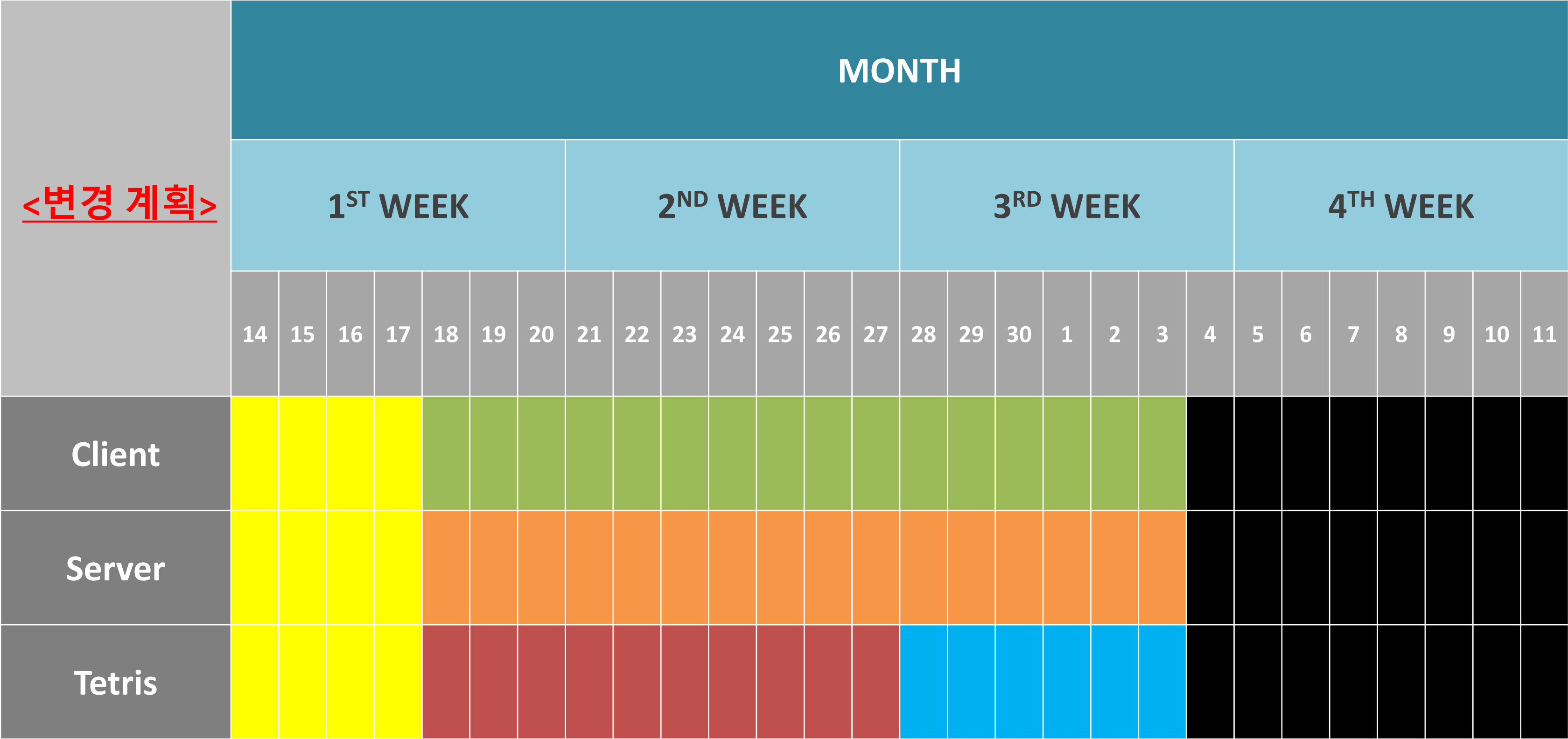
테트리스만 넣는걸로 목표 변경

서버 독창성에서 QUEST 부분을 포기 후
나머지 부분 합치는 것으로 목표 변경

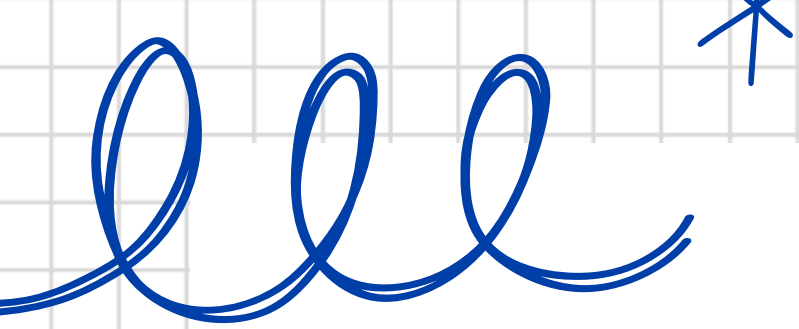


<초기 계획>	MONTH																											
	1 ST WEEK							2 ND WEEK							3 RD WEEK							4 TH WEEK						
	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11
Indian poker																												
BlackJack																												
Tetris																												
Concave																												
Server																												
Quest																												
Additional ideas																												

이희준
 조성호
 홍석영
 공동 개발
 디버깅 및 테스트
 최종 발표 준비
 주제 구상



이희준 조성호 홍석영 공동 개발 디버깅 및 테스트 최종 발표 준비 주제 구상

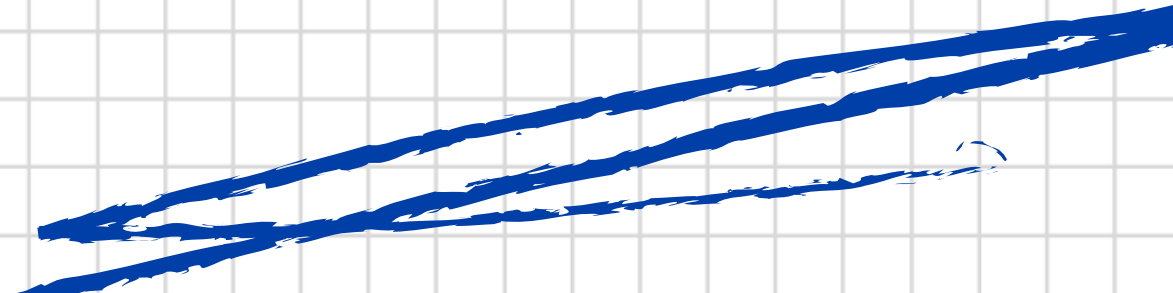
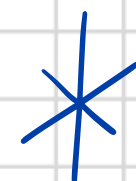


기능

클라이언트 각각 채팅방 입장

테트리스 게임 구현

클라이언트 닉네임 변경가능



핵심 코드

- 게임 룸(Server)

```
while (1) {
    int client_socket = accept(server_socket, (struct sockaddr *)&client_address, &client_address_len);

    // 클라이언트로부터 방 번호 수신
    char buffer[1024];
    int bytes_received = recv(client_socket, buffer, sizeof(buffer), 0);
    buffer[bytes_received] = '\0';

    int room_number;
    char message[1024];
    sscanf(buffer, "%d:%s", &room_number, message);

    // 방이 비어있는지 확인
    if (room_dict[room_number].client_count == 0) {
        send(client_socket, "방이 비어있습니다.", strlen("방이 비어있습니다."), 0);
    } else {
        // Send a message to all clients in the room
        for (int i = 0; i < room_dict[room_number].client_count; ++i) {
            send(room_dict[room_number].clients[i], message, strlen(message), 0);
        }
    }

    // 방에 클라이언트 추가
    room_dict[room_number].clients[room_dict[room_number].client_count++] = client_socket;

    // 새로운 클라이언트를 처리하는 스레드 생성
    pthread_t thread;
    int *client_socket_ptr = (int *)malloc(sizeof(int));
    *client_socket_ptr = client_socket;

    pthread_create(&thread, NULL, handle_client, (void *)client_socket_ptr);
    room_dict[room_number].thread = thread;

    printf("클라이언트가 방 %d에 참가했습니다.\n", room_number);
}

close(server_socket);
}
```

핵심 코드

- 채팅(Server)

```
void *handle_client(void *arg) {
    int client_socket = *((int *)arg);
    free(arg);

    // 클라이언트로부터 데이터 수신
    char buffer[1024];
    int bytes_received;
    while ((bytes_received = recv(client_socket, buffer, sizeof(buffer), 0)) > 0) {
        buffer[bytes_received] = '\0';

        int room_number;
        char message[1024];
        sscanf(buffer, "%d:%s", &room_number, message);

        for (int i = 0; i < MAX_CLIENTS; ++i) {
            if (room_dict[i].client_count > 0 && i == room_number) {
                for (int a = 0; a < MAX_CLIENTS; ++a) {
                    if (room_dict[i].clients[a] != client_socket) {
                        send(room_dict[i].clients[a], message, strlen(message), 0);
                    }
                }
            }
        }

        // 여기서 부터 if 문으로 방에 대한 로직 생성하기
        if (room_number == 1) {
            printf("1번방 입장\n");
        }
        else if (room_number == 2) {
            printf("2번방 입장\n");
        }
        else if (room_number == 3) {
            printf("3번방 입장\n");
        }
    }
}
```


핵심 코드

- 닉네임(Client)

```
def nick_change(self):
    new_nick = self.nick_entry.get()
    self.nickname = new_nick
    full_nick = f"{self.add}:e:{self.nickname}"
    self.client_socket.send(full_nick.encode())

def e_fun(self, add, nn):
    if add in self.nickroom[0]:
        for i, dres in enumerate(self.nickroom[0]):
            if dres == add:
                for j in range(2):
                    self.nickroom[j].pop(i)
                    break
    self.nickroom[0].append(add)
    self.nickroom[1].append(nn)

def __init__(self, master):
    self.nickname = ''
    self.nickroom = [[],[]]
```

핵심 코드

- 테스리스(Client)

```
class TetrisApp(object):
    def __init__(self):
        pygame.init()
        pygame.key.set_repeat(250, 25)
        self.width = config['cell_size'] * config['cols'] + 150 # 추가된 너비
        self.height = config['cell_size'] * config['rows']

        self.screen = pygame.display.set_mode((self.width, self.height))
        pygame.event.set_blocked(pygame.MOUSEMOTION)
        self.init_game()
        self.client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.client_socket.connect(("220.149.128.100", 4307))

        # 서버에 연결

        receive_thread = threading.Thread(target=self.receive_data)
        receive_thread.start()

    def send_message(self, message): # server에 message를 보냄
        try:
            if str(type(message)) != "<class 'str'>":
                message = str(message)
            self.client_socket.send(message.encode())
        except socket.error as e:
            print("Error sending message:", e)

            self.score += 10
            # 서버에 현재 게임 상태 전송
            self.send_message("score:"+str(self.score))
            break

    if self.gameover:
        self.center_msg(f"""Game Over! You loss
        Your Score: {self.score}
        Press space to continue""")
        self.send_message("game over")
        break
```

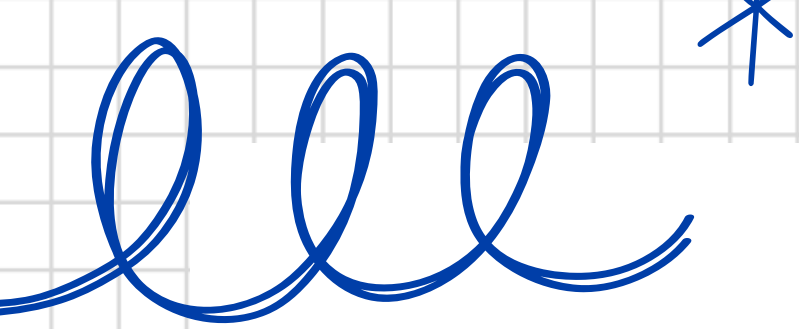
핵심 코드

- 테스트리스(Client)

```
def receive_data(self): #server에서 data 받음
    while True:
        try:
            data = self.client_socket.recv(1024)
            if not data:
                break

            received_data = data.decode('utf-8')
            print(received_data)
            if 'score' in received_data:
                score = received_data.split(":")
                self.display_other_score(score[1])
                self.other_score = score[1]
            elif 'over' in received_data:
                game_over = received_data.split(" ")
                print("You Win")
                self.center_msg("You Win!!")
                self.gameover = True
                pygame.display.update()
                sys.exit()
        except Exception as e:
            print("Error receiving data:", e)
            break

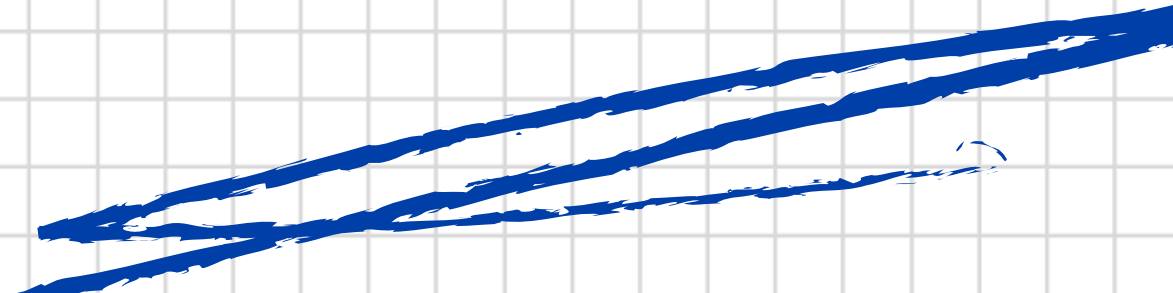
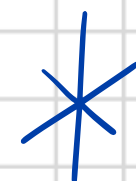
# 서버로부터 받은 점수를 화면에 표시하는 함수
def display_other_score(self, score):
    font = pygame.font.Font(None, 36)
    score_text = font.render(f"Score: {score}", True, (255, 255, 255))
    self.screen.blit(score_text, (self.width - 150, 90))
```



아쉬운 점

상대 테트리스 화면 확인 불가

과도한 계획 수립



시연



감사합니다

