



# MINI-XBOX

빅데이터학과 3학년 최민정  
빅데이터학과 3학년 이승희  
빅데이터학과 2학년 최서원

# 목차

- 담당 파트
- 실행 환경
- 다이어그램
- mini-xbox 설명
  - chatting
  - sipping
  - avoid char
  - bakin robbins
  - rock sissor paper
  - rush dog
- 활용방안

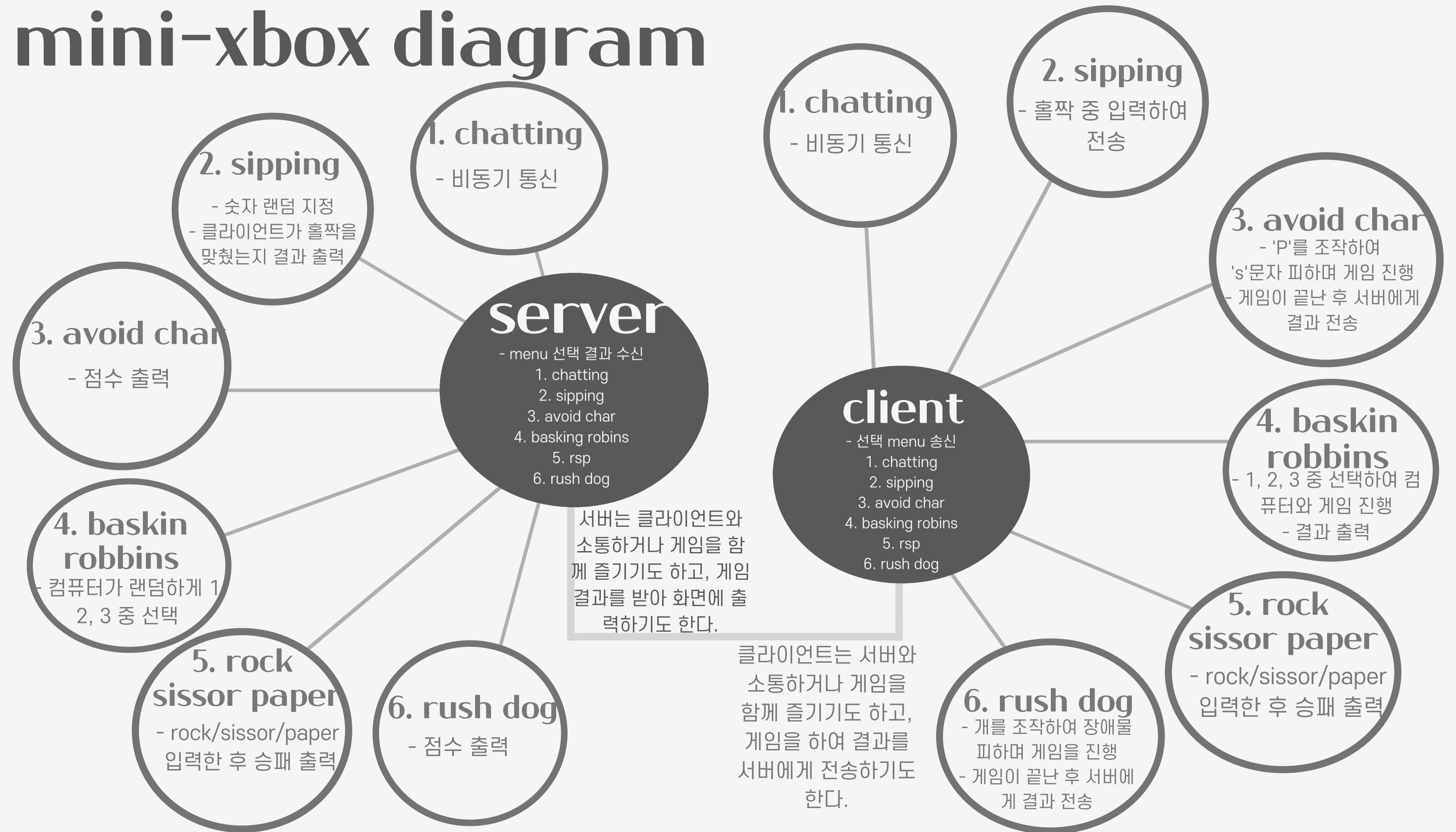
# 담당 파트

이승희 - menu 구현  
sipping 게임 구현  
avoid char 게임 구현  
발표 자료 제작

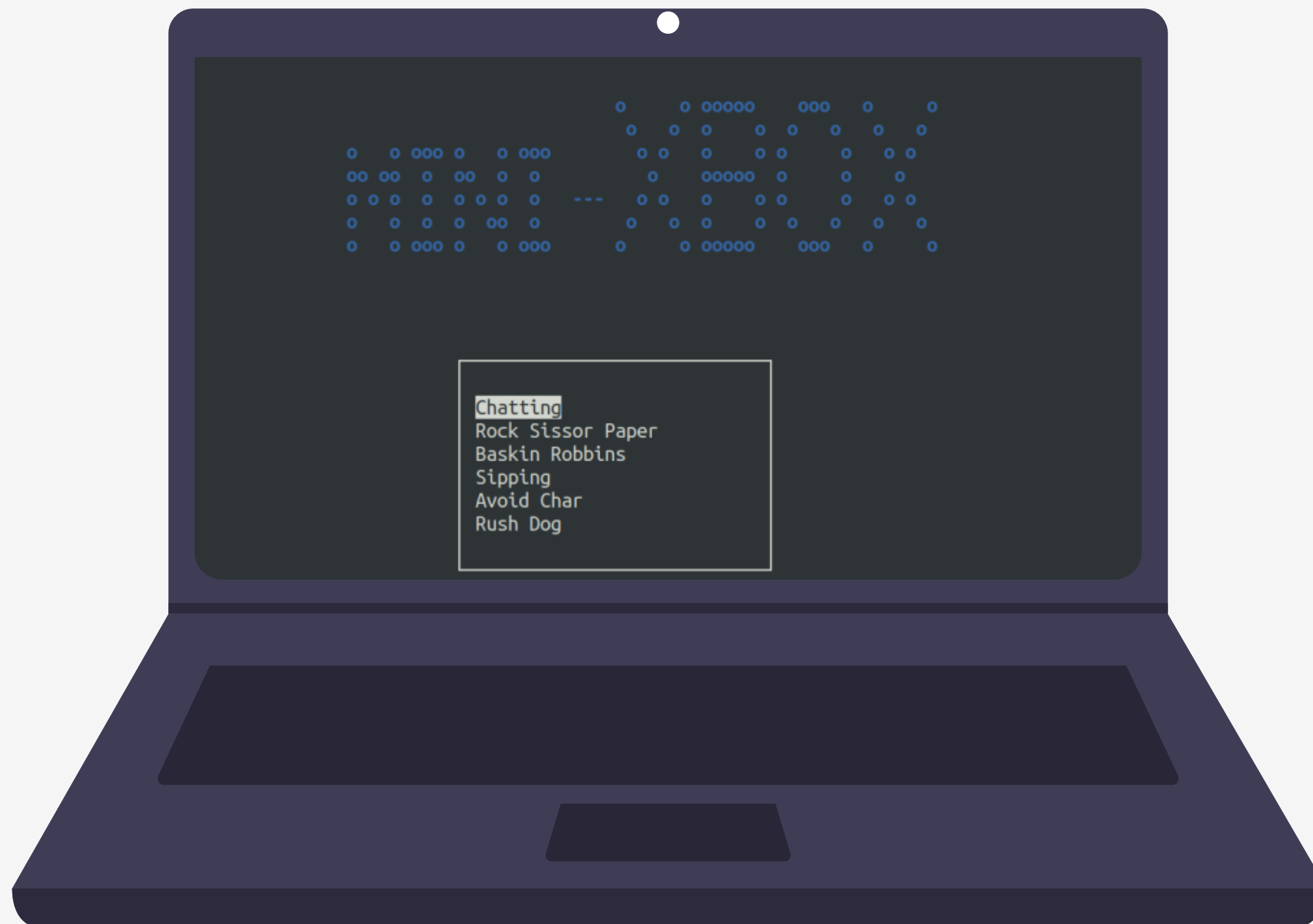
최서원 - basking robbers 게임 구현  
발표 자료 제작

최민정 - 다이어그램 제작  
rock sissor paper 게임 구현  
rush dog 게임 구현  
발표 자료 제작

# mini-xbox diagram



# 실행 환경



+

# Chatting

```

if(choice == 1){ //start choice 1
    strcpy(menu, "1");
    send(sd, menu, strlen(menu)+1, 0);
    printf("====chatting start====\n");

    pthread_create(&pth1, NULL, send_func, NULL);
    pthread_create(&pth2, NULL, recv_func, NULL);

    pthread_join(pth1, NULL);
    pthread_join(pth2, NULL);
}

```

client

```

if(!strcmp(menu, "1", 1)){ //chatting program
    printf("====chatting start====\n");

    pthread_create(&pth1, NULL, send_func, NULL);
    pthread_create(&pth2, NULL, recv_func, NULL);

    pthread_join(pth1, NULL);
    pthread_join(pth2, NULL);
}

```

server

```

void *recv_func(void *arg){
    while(1){
        memset(rbuf, '\0', sizeof(rbuf));
        rlen = recv(ns, rbuf, sizeof(rbuf), 0);
        if(strncmp(rbuf, "exit", 4) == 0)
            exit(0);
        if(rlen > 0)
            printf("client: %s\n", rbuf);
    }
}

```

```

void *send_func(void *arg){
    while(1){
        memset(sbuf, '\0', sizeof(sbuf));
        printf("Enter message: %s\n", fgets(sbuf, sizeof(sbuf), stdin));
        sbuf[strlen(sbuf)] = '\0';
        if(send(ns, sbuf, strlen(sbuf)+1, 0) == -1){
            perror("send");
            exit(1);
        }
        if(strncmp(sbuf, "exit", 4) == 0)
            exit(0);
    }
}

```

# Shipping

```
=====Sipping Game=====
I received a number from the server.
Is this an odd number or an even number?(odd, even)
Enter(odd or even):
```

server

```
leeseunghui@ubuntu:~/sp_proj/t$ ./server.exe
client choose service 4
=====Sipping Game=====
random number: 1
recv: even, send: odd
=====CLIENT LOSE!=====
```

client

```
=====Sipping Game=====
I received a number from the server.
Is this an odd number or an even number?(odd, even)
Enter(odd or even): even

=====CLIENT LOSE!=====
```



server

```
if(!strcmp(menu, "4", 1)){ //Sipping Game
    printf("====Sipping Game====\n");
    int random = 0;

    srand(time(NULL));
    random = rand() % 2;

    printf("random number: %d\n", random);

    memset(sbuf, '\0', sizeof(sbuf));
    if(random == 0){ //even number
        strcpy(sbuf, "even");
    }else{
        strcpy(sbuf, "odd");
    }
    send(ns, sbuf, strlen(sbuf)+1, 0);

    memset(rbuf, '\0', sizeof(rbuf));
    recv(ns, rbuf, sizeof(rbuf), 0);

    printf("recv: %s, send: %s\n", rbuf, sbuf);

    if(strcmp(rbuf, sbuf) == 0){
        printf("====CLIENT WIN!====\n");
    }else{
        printf("====CLIENT LOSE!====\n");
    }

    sleep(1);
    close(ns);
    close(sd);
    return(0);
}
```

client

```
if(choice == 4){
    initscr();
    clear();
    echo();
    curs_set(TRUE);

    strcpy(menu, "4");
    send(sd, menu, strlen(menu)+1, 0);
    mvprintw(0,0,"====Sipping Game====\n");

    //Receive random number from server
    memset(rbuf, '\0', sizeof(rbuf));
    recv(sd, rbuf, sizeof(rbuf), 0);

    mvprintw(1,0,"I received a number from the server.\n");
    mvprintw(2,0,"Is this an odd number or an even number?(odd, even)\n");

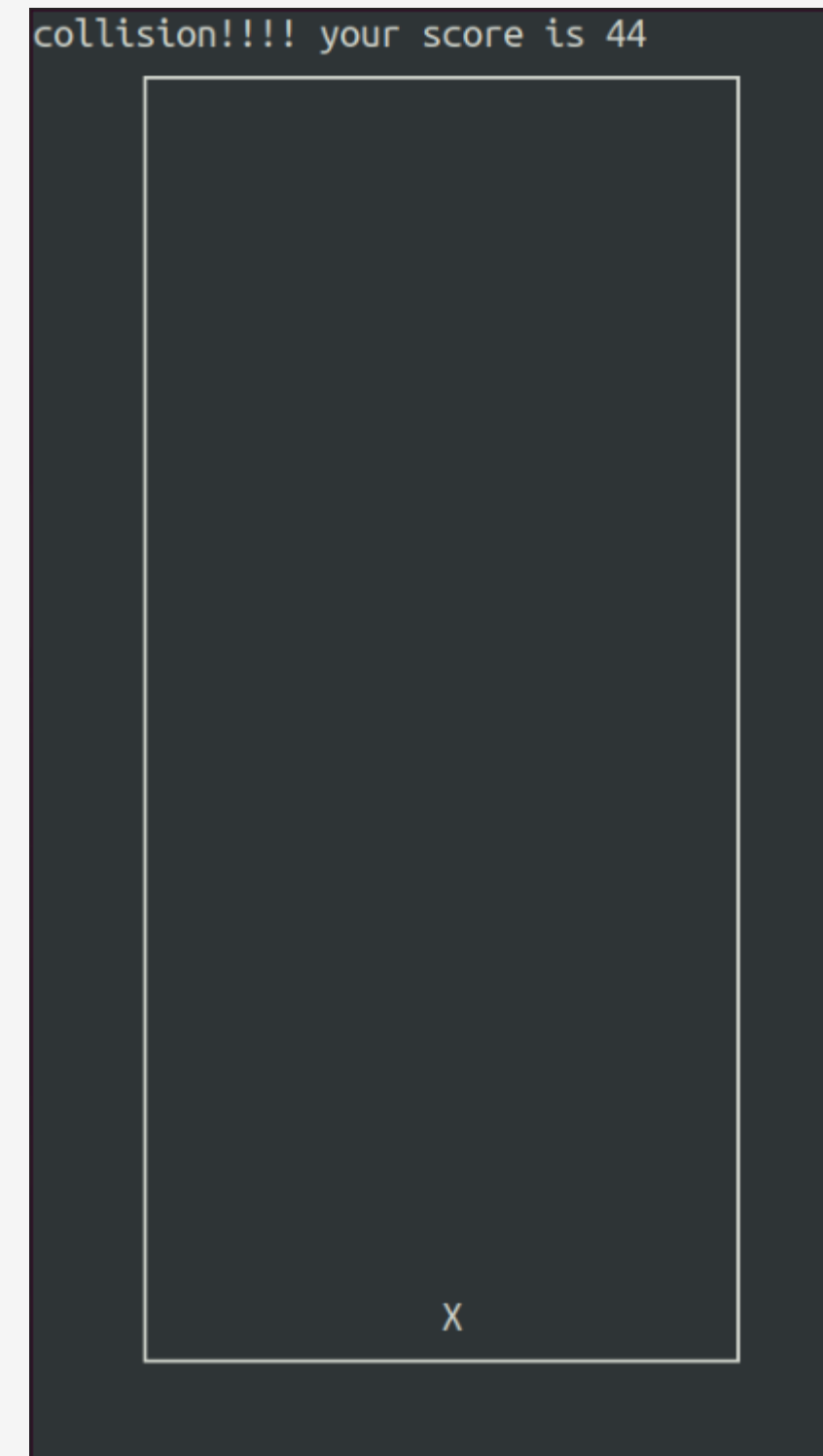
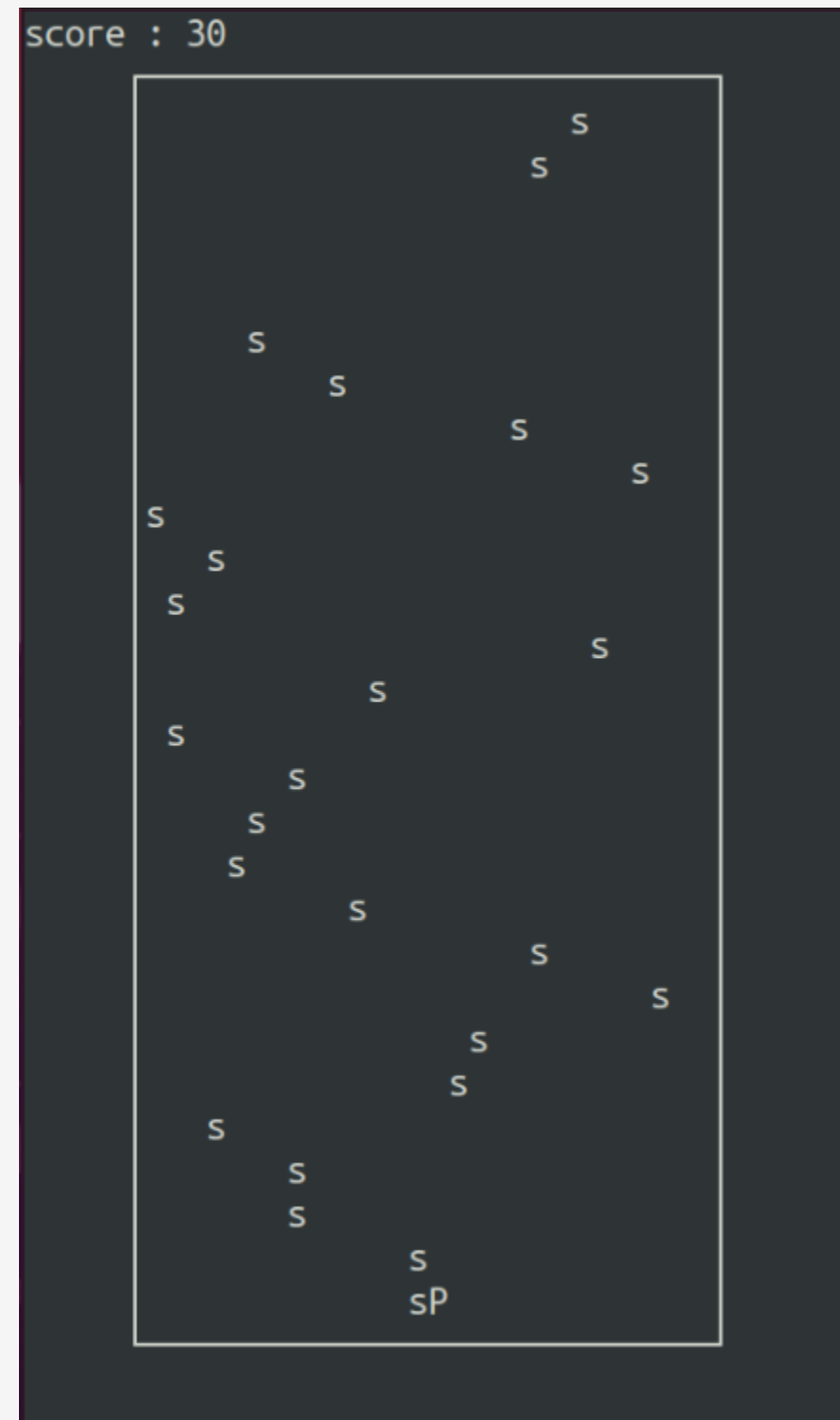
    memset(sbuf, '\0', sizeof(sbuf));
    mvprintw(3,0,"Enter(odd or even): ");
    scanw("%s", sbuf);
    send(sd, sbuf, sizeof(sbuf), 0);

    if(strcmp(rbuf, sbuf) == 0){
        mvprintw(6,0,"====CLIENT WIN!====\n");
    }else{
        mvprintw(6,0,"====CLIENT LOSE!====\n");
    }
    refresh();

    sleep(5);
    close(sd);
    clear();
    endwin();
    return 0;
}
```



# Avoid Char Game



게임화면  
가리키는 변수

별의 개수 30

플레이어(P)와 S  
충돌체크 하는 변수

게임화면 좌표와  
크기 지정

문자 S의 출력 좌표

```
#define MAX_CNT 30

WINDOW *game_win;

//player
char *player = "P";
int startx = 0, starty = 0;
int height = 30, width = 30;

int flag_P = 1;
int u_score = 0; //score

//star
int flag[MAX_CNT] = {0,};
int starx[MAX_CNT], stary[MAX_CNT];
int c = 0;

int ac_game_start();
void print_playground(WINDOW *game_win);
void init_setting();

void create_star();
void input_process();
void move_star();
void user_score();
```

```

void init_setting(){
    initscr();
    clear();
    noecho();
    curs_set(FALSE);
    cbreak();

    //game playground start position
    startx = 5;
    starty = 1;

    //random
    srand(time(NULL));

    game_win = newwin(height, width, starty, startx);
    keypad(game_win, TRUE);
    nodelay(game_win, TRUE);

    //print playground
    print_playground(game_win);
}

```

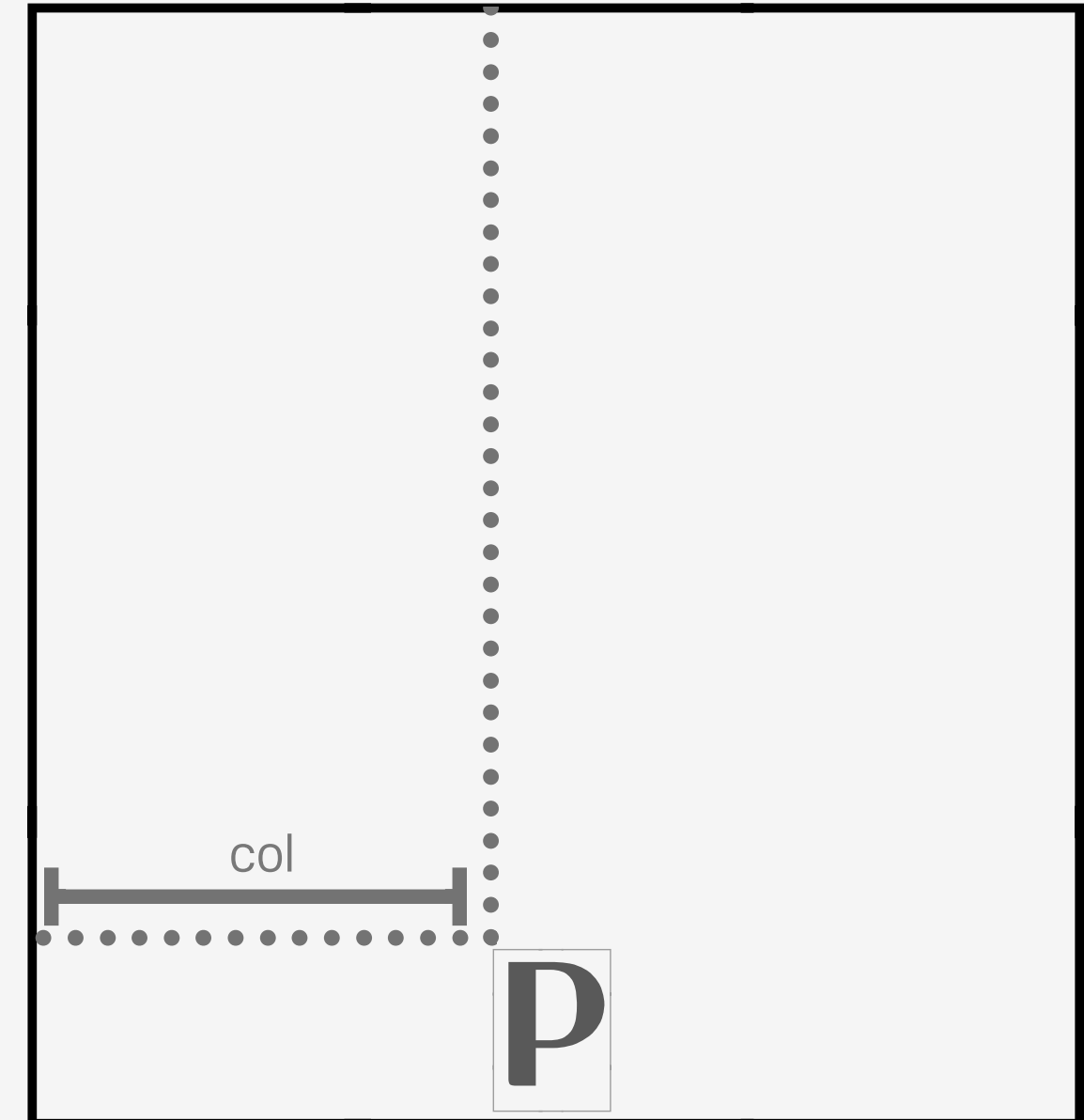
```

int ac_game_start(){
    int sc;
    init_setting();
    while(1) {
        if(flag_P){
            create_star();
            input_process();
        }
        move_star();
        print_playground(game_win);
        if(flag_P == 0){
            mvprintw(0, 0, "collision!!!! your score is %d", u_score);
            refresh();
            sleep(4);
            break;
        }
    }
    sc = u_score;
    endwin();
    return u_score;
}

```

충돌 시 점수 출력 후 종료

```
void input_process(){
    c = wgetch(game_win);
    switch(c)
    {
        case KEY_LEFT:
            --col;
            break;
        case KEY_RIGHT:
            ++col;
            break;
        default:
            break;
    }
    if(c == 'q'){
        end_setting();
        exit(0);
    }
}
```



```

void print_playground(WINDOW *game_win)
{
    wclear(game_win);
    box(game_win, 0, 0);
    if(col < 1) col = 1;
    if(col > 28) col = 28;
    mvwprintw(game_win, height - 2, col, player);
    wrefresh(game_win);
    user_score();
}

void create_star(){
    for(int i = 0 ; i < MAX_CNT ; i++){
        if(!flag[i]){
            starx[i] = (rand() % 27) + 1;
            stary[i] = 1;
            flag[i] = 1;
            break;
        }
    }
}

```

플레이어가 게임화면에서 벗어나지 않도록 설정

```

void user_score(){
    if(flag_P){
        u_score++;
        mvprintw(0,0,"score : %d",u_score);
        refresh();
    }
}

```

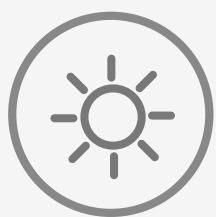
```

void move_star(){
    for(int i = 0 ; i < MAX_CNT ; i++){
        if(flag[i]){
            mvwprintw(game_win, stary[i], starx[i], "s");
            if((stary[i] == (height - 2)) && (starx[i] == col)){
                //collision
                player = "X";
                flag_P = 0;
            }
            stary[i]++;
            if(stary[i] > (height - 2)){
                flag[i] = 0;
            }
        }
    }
    wrefresh(game_win);
    usleep(10000);
}

```

# 베스킨라빈스 31

## 색상 변경하기 - ncurses 이용



글자색 변경

- `initscr()`;
- `start_color()`;
- `init_pair(글자, 글자색, 배경색)`;



문자열색 변경

- `attron()`;
- `attroff()`;

-> 문자열 전체 색생변경 할때 , 함수 사용하여  
출력의 색상 변경

```

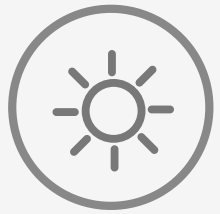
138     if(choice == 3){ //start choice 3 baskin robbins
139         strcpy(menu, "3");
140         send(sd, menu, strlen(menu)+1, 0);
141         initscr();
142         curs_set(TRUE);
143         echo();
144
145     start_color();
146         init_pair(1,COLOR_RED,COLOR_YELLOW);
147         init_pair(2,COLOR_BLUE,COLOR_BLACK);
148         init_pair(4,COLOR_YELLOW,COLOR_BLACK);
149         init_pair(5,COLOR_WHITE,COLOR_BLACK);
150
151     int player=0;int computer=0;
152     int num=0;
153
154     srand((unsigned)time(NULL));
155
156         clear();
157         attron(COLOR_PAIR(1));
158     mvprintw(0,0,"-----baskinrobbins31 game-----\n");
159         attroff(COLOR_PAIR(1));
160         init_pair(3,COLOR_GREEN,COLOR_BLACK);
161         attron(COLOR_PAIR(3));
162     mvprintw(1,0,"GAME START");
163         attroff(COLOR_PAIR(3));
164         attron(COLOR_PAIR(2));
165     printf("**Enter the number from 1 to 3**\n");
166         attroff(COLOR_PAIR(2));
167         int j = 2;
168     while(1){
169         int xx = 50;
170         refresh();
171         attron(COLOR_PAIR(5));
172         move(j, 0);
173         printf("[player] how many numbers will you enter? : ");
174         attroff(COLOR_PAIR(5));
175         scanw("%d",&player);

```



# 베스킨라빈스 31

## 게임방법 구현



### if 문 사용

- 사용자가 입력한 숫자의 개수가 0개보다 작거나 3개보다 클 때
- 입력한 숫자가 31보다 클 때와 작을 때



### for 문 사용

- 사용자 기준일 경우
- 컴퓨터 기준일 경우

```

176     if(player>3 || player<=0){
177         printf("Only enter numbers from 1 to 3\n\n");
178         continue;
179     }
180     for(int i=1;i<=player;i++){
181         num++;
182         if(num<=31){
183             move(j, xx+=4);
184             printf("%d",num);
185             refresh();
186         }
187     }
188     if(num>31){
189         move(32, 0);
190         printf("*** computer win! ***");
191         refresh();
192         break;
193     }
194     computer = (rand()%3+1);
195     move(j+1, 0);
196     printf("[computer]: %d\n",computer);
197     refresh();
198
199     xx = 50;
200     for(int i=0;i<computer;i++){
201         num++;
202         if(num<=31){
203             move(j+1, xx+=4);
204             printf("%d",num);
205             refresh();
206         }
207     }
208     if(num>31){
209         move(32,0);
210         printf("*** player win! ***\n");
211         refresh();
212         break;
213     }
214     j+=2;
215 }
216 sleep(10);
217 endwin();

```

# 출력 화면



```
-----baskinrobbins31 game-----
GAME START**Enter the number from 1 to 3**
[player] how many numbers will you enter? : 1      1
[computer]: 1                                     2
[player] how many numbers will you enter? : 2      3      4
[computer]: 1                                     5
[player] how many numbers will you enter? : 3      6      7      8
[computer]: 2                                     9     10
[player] how many numbers will you enter? : 1     11
[computer]: 3                                     12     13     14
[player] how many numbers will you enter? : 2     15     16
[computer]: 3                                     17     18     19
[player] how many numbers will you enter? : 3     20     21     22
[computer]: 2                                     23     24
[player] how many numbers will you enter? : 1     25
[computer]: 1                                     26
[player] how many numbers will you enter? : 2     27     28
[computer]: 2                                     29     30
[player] how many numbers will you enter? : 1     31
[computer]: 2

*** player win! ***
```

사용자가 시작하여 사용자가 이기고 있습니다.



# Rock Sissor Paper

가위바위보 게임

server

```
minjeong@minjeong-VirtualBox:~/sp2021-2/final$ ./server.exe
client choose service 2
====RSP game====
CLIENT: paper
rock, sissor, paper: sissor
====SERVER WIN!====
```

client

```
minjeong@minjeong-VirtualBox:~/sp2021-2/final$ ./client.exe
2
====RSP game====
rock, sissor, paper: paper
SERVER: sissor
====CLIENT LOSE!====
```

# Rock Sissor Paper 서버

## 가위바위보 게임

클라이언트에게 소켓을 받으면 서버는 무엇을 낼지 입력하여 클라이언트에게 보낸다. 클라이언트와 서버가 낸 것을 비교하여 승패 결과를 출력한다.

- `!(strcmp(" ", " ")) && !(strcmp(" ", " "))`  
: rock, sissor, paper 비교

```
minjeong@minjeong-VirtualBox:~/sp2021-2/final$ ./server.exe
client choose service 2
====RSP game====
CLIENT: paper
rock, sissor, paper: sissor
====SERVER WIN!====
```

```
if(choice == 2){ //start choice 2
    strcpy(menu, "2");
    printf("%s\n", menu);
    send(sd, menu, strlen(menu)+1, 0);
    printf("====RSP game====\n");
    while(1){ //start while
        memset(sbuf, '\0', sizeof(sbuf));
        printf("rock, sissor, paper: ");
        scanf("%s", sbuf);

        if(send(sd, sbuf, strlen(sbuf)+1, 0) == -1){
            perror("send");
            exit(1);
        }

        memset(rbuf, '\0', sizeof(rbuf));
        rlen = recv(sd, rbuf, sizeof(rbuf), 0);
        printf("SERVER: %s\n", rbuf);

        if((!(strcmp(sbuf, "rock")) && !(strcmp(rbuf, "sissor"))) || \
            (!(strcmp(sbuf, "sissor")) && !(strcmp(rbuf, "paper"))) || \
            (!(strcmp(sbuf, "paper")) && !(strcmp(rbuf, "rock")))){
            printf("====CLIENT WIN!====\n");
            sleep(1);
            exit(1);
        }else if((!(strcmp(sbuf, "rock")) && !(strcmp(rbuf, "paper"))) || \
            (!(strcmp(sbuf, "sissor")) && !(strcmp(rbuf, "rock"))) || \
            (!(strcmp(sbuf, "paper")) && !(strcmp(rbuf, "sissor")))){
            printf("====CLIENT LOSE!====\n");
            sleep(1);
            exit(1);
        }else if((!(strcmp(sbuf, "rock")) && !(strcmp(rbuf, "rock"))) || \
            (!(strcmp(sbuf, "sissor")) && !(strcmp(rbuf, "sissor"))) || \
            (!(strcmp(sbuf, "paper")) && !(strcmp(rbuf, "paper")))){
            printf("====EVEN: PLAY AGAIN!====\n");
            sleep(1);
        }else{
            printf("====THAT IS NOT CORRECT RSP!====\n");
            sleep(1);
        }
    } //end while
    sleep(1);
    close(sd);
    return 0;
} //end choice 2
```

# Rock Sissor Paper 클라이언트

## 가위바위보 게임

무엇을 낼지 입력하여 서버에게 보내고 서버가 무엇을 냈는지 소켓을 받는다. 클라이언트와 서버가 낸 것을 비교하여 승패 결과를 출력한다.

```
minjeong@minjeong-VirtualBox:~/sp2021-2/final$ ./client.exe
2
=====RSP game=====
rock, sissor, paper: paper
SERVER: sissor
=====CLIENT LOSE!=====
```

```
if(!strcmp(menu, "2", 1)){ //RSP
    printf("=====RSP game=====\n");
    while(1){
        memset(rbuf, '\0', sizeof(rbuf));
        rlen = recv(ns, rbuf, sizeof(rbuf), 0);
        printf("CLIENT: %s\n", rbuf);

        memset(sbuf, '\0', sizeof(sbuf));
        printf("rock, sissor, paper: ");
        scanf("%s", sbuf);

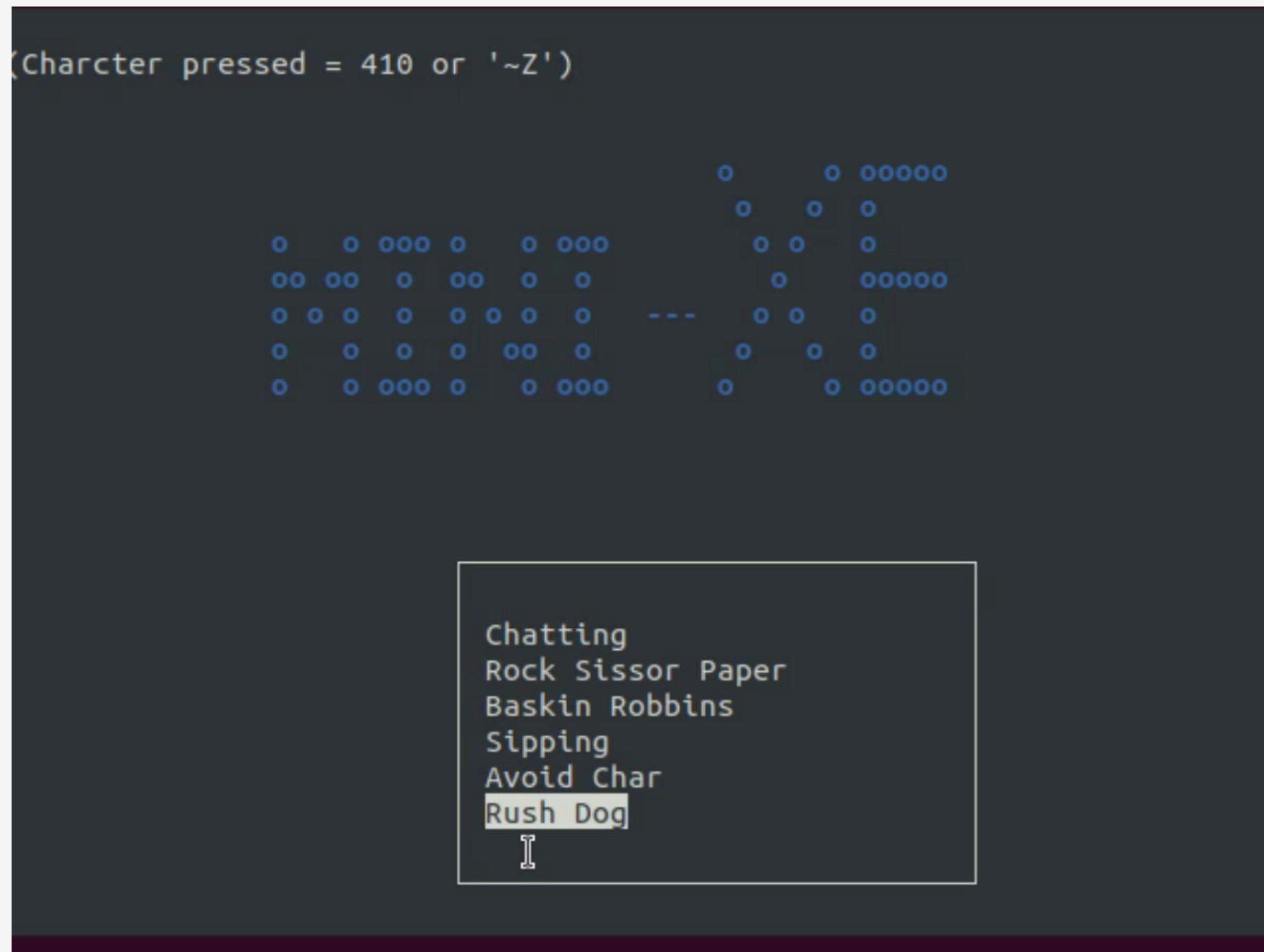
        if(send(ns, sbuf, strlen(sbuf)+1, 0) == -1){
            perror("send");
            exit(1);
        }

        if(((!(strcmp(sbuf, "rock")) && !(strcmp(rbuf, "sissor")))) || \
        (!(strcmp(sbuf, "sissor")) && !(strcmp(rbuf, "paper")))) || \
        (!(strcmp(sbuf, "paper")) && !(strcmp(rbuf, "rock"))))){
            printf("=====SERVER WIN!=====\n");
            sleep(1);
            close(ns);
            close(sd);
            exit(1);
        }
        else if(((!(strcmp(sbuf, "rock")) && !(strcmp(rbuf, "sissor")))) || \
        (!(strcmp(sbuf, "sissor")) && !(strcmp(rbuf, "paper")))) || \
        (!(strcmp(sbuf, "paper")) && !(strcmp(rbuf, "rock"))))){
            printf("=====SERVER LOSE!=====\n");
            sleep(1);
            close(ns);
            close(sd);
            exit(1);
        }
        else if(((!(strcmp(sbuf, "rock")) && !(strcmp(rbuf, "sissor")))) || \
        (!(strcmp(sbuf, "sissor")) && !(strcmp(rbuf, "paper")))) || \
        (!(strcmp(sbuf, "paper")) && !(strcmp(rbuf, "rock"))))){
            printf("=====EVEN: PLAY AGAIN!=====\n");
            sleep(1);
        }
        else{
            printf("=====THAT IS NOT CORRECT RSP!=====\n");
            sleep(1);
        }
    }
    sleep(1); //bind error prevention
    close(ns);
    close(sd);
    return(0);
}
```

# Rush Dog

## 장애물 뛰어넘기 게임

크롬 오프라인 게임인 공룡 게임을 구상하여 ncurses를 이용해 만든 게임이다. 클라이언트가 게임을 하고, 게임 종료 시, 서버는 결과를 받아 출력한다. 개가 점프를 해서 장애물을 피하고, 피하지 못해 장애물과 닿으면 게임이 종료된다. 끝이 없고, 점프하는 박자를 손에 익히기만 하면 되기 때문에 머리를 식힐 때 간단하게 할 수 있다.



# Rush Dog 서버

## 장애물 뛰어넘기 게임

서버는 클라이언트에게서 소켓을 통해 점수를 받아 화면에 출력한다.  
클라이언트에게 "q"라는 문자열을 받으면 서버가 종료된다.

```
if(!strcmp(menu, "6", 1)){ //Rush Dog
    printf("====Rush Dog Game====\n");
    while(1){
        memset(rbuf, '\0', sizeof(rbuf));
        recv(ns, rbuf, sizeof(rbuf), 0);
        if(!strcmp(rbuf, "q")){
            break;
        }
        printf("client's score is %s", rbuf);
    }
    sleep(1); //bind error prevention
    close(ns);
    close(sd);
    return(0);
}
```

```
minjeong@minjeong-VirtualBox:~/sp2021-2/final$ ./server.exe
client choose service 6
====Rush Dog Game====
client's score is 9
client's score is 2
client's score is 1
client's score is 4
minjeong@minjeong-VirtualBox:~/sp2021-2/final$
```

# Rush Dog 클라이언트

## 장애물 뛰어넘기 게임

DOG\_Y: 개 y좌표

DOG\_X: 개 x좌표

POOP\_Y: 장애물 x좌표

POOP\_X: 장애물 x좌표

dog\_shape(): 개의 모양을 그리는 함수

=> jump: 개의 점프 여부

점프했을 경우, if문에 해당하는 그림

점프하지 않았을 경우, else문에 해당하는 그림

poop\_shape(): 장애물을 그리는 함수

```
//rush dog
#define DOG_Y 15
#define DOG_X 20
#define POOP_Y 20
#define POOP_X 40
bool jump;
```

```
void dog_shape(int x, int y){
    move(y, x);
    if(!jump){
        printf("o      oooo\n");
        printf("oo     oooooo\n");
        printf("oooooooooooo\n");
        printf(" oooooooo\n");
        printf("ooo   ooo\n");
        printf("  ooo oo\n");
        printf("   oo oo\n");
    }else{
        printf("  o      oooo\n");
        printf(" oo     oooooo\n");
        printf("oooooooooooo\n");
        printf(" oooooooo\n");
        printf("ooo   ooo\n");
        printf("oo      oo\n");
        printf("oo      oo\n");
    }
}
```

```
void poop_shape(int x, int y){
    mvprintw(y++, x, " o \n");
    mvprintw(y++, x, "ooo \n");
    mvprintw(y++, x, "ooooo\n");
}
```

# Rush Dog 클라이언트

## 장애물 뛰어넘기 게임

- nodelay(stdscr, TRUE): 키 입력 대기(비동기)
- bottom: 개가 바닥에 있는지 여부

개가 바닥에 있는 상태에서 점프를 했을 경우,  
jump = true; bottom = false; 로 변경한다.

```
if(choice == 6){
    //rush dog game
    strcpy(menu, "6");
    send(sd, menu, strlen(menu)+1, 0);

    initscr();
    int ch;

    while(1){
        int dog_y = DOG_Y;
        int dog_x = DOG_X;
        int poop_y = POOP_Y;
        int poop_x = POOP_X;
        int score = 0;
        jump = false;
        bool bottom = true;
        time_t ttime = time(NULL);
        int jump_y = dog_y-5;

        while(1){
            nodelay(stdscr, TRUE);
            ch = getch();
            if(ch == ' ' && bottom){
                jump = true;
                bottom = false;
            }
        }
    }
}
```

# Rush Dog 클라이언트

## 장애물 뛰어넘기 게임

장애물은 왼쪽으로 이동하다가 x좌표가 0에 다다르면,  
다시 처음에 설정한 위치로 돌아간다.

점프했을 경우, 개의 y좌표는 줄어든다(점프하는 과정)  
점프하지 않았을 경우, 개의 y좌표는 늘어난다(바닥에 내려가는 과정).

개가 설정된 y좌표보다 크거나 같으면 개는 바닥에 있다.

개가 이미 점프를 한 상태일 경우, 더 이상 jump할 수  
없기 때문에 jump = false; 로 바꾼다.

```
}  
poop_x -= 2;  
if(poop_x <= 0){  
    poop_x = POOP_X;  
}  
clear();
```

```
if(jump){  
    dog_y--;  
}else{  
    dog_y++;  
}
```

```
if(dog_y >= 16){  
    dog_y = 16;  
    bottom = true;  
}
```

```
if(dog_y <= 11){  
    jump = false;  
}
```

```
dog_shape(0, dog_y);  
poop_shape(poop_x, POOP_Y);
```



# Rush Dog 클라이언트

## 장애물 뛰어넘기 게임

장애물과 개의 x좌표가 겹칠 때  
점프를 하지 않았다면, 게임은 끝난다.

```
if(poop_x <= 8 && poop_x >= 7){  
    if(!jump){  
        break;  
    }  
}
```

개와 장애물의 x좌표가 겹치는 순간 점수 증가

```
if(poop_x == dog_x){  
    ttime = time(NULL);  
    score++;  
}
```

## +

```

mvprintw(0, 0, "-----P
PRESENT SCORE: %d-----", score);
mvprintw(23, 0, "-----
");
mvprintw(2, 0, "+      +      +      +
+      +");
mvprintw(3, 0, "      +      +      +      +
      +      +      +");
mvprintw(4, 0, "+      +      +      +
      +      +      +");
mvprintw(5, 0, "      +      +      +      +
      +      +      +");
mvprintw(6, 0, "+      +      +      +
      +      +      +");
mvprintw(7, 0, "      +      +      +      +
      +      +      +");
mvprintw(8, 0, "+      +      +      +
      +      +      +");
mvprintw(9, 0, "      +      +      +      +
      +      +      +");
mvprintw(10, 0, "+      +      +      +
      +      +      +");
mvprintw(11, 0, "      +      +      +      +
      +      +      +");
mvprintw(2, 40, "+++JumpKey: <space>+++");
mvprintw(3, 40, "|      Jump      |");
mvprintw(4, 40, "|      and      |");
mvprintw(5, 40, "|      avoid      |");
mvprintw(6, 40, "|      poop      |");
mvprintw(7, 40, "+++++");
refresh();
napms(1000/15);
}

```

```
-----PRESENT SCORE: 1-----
```

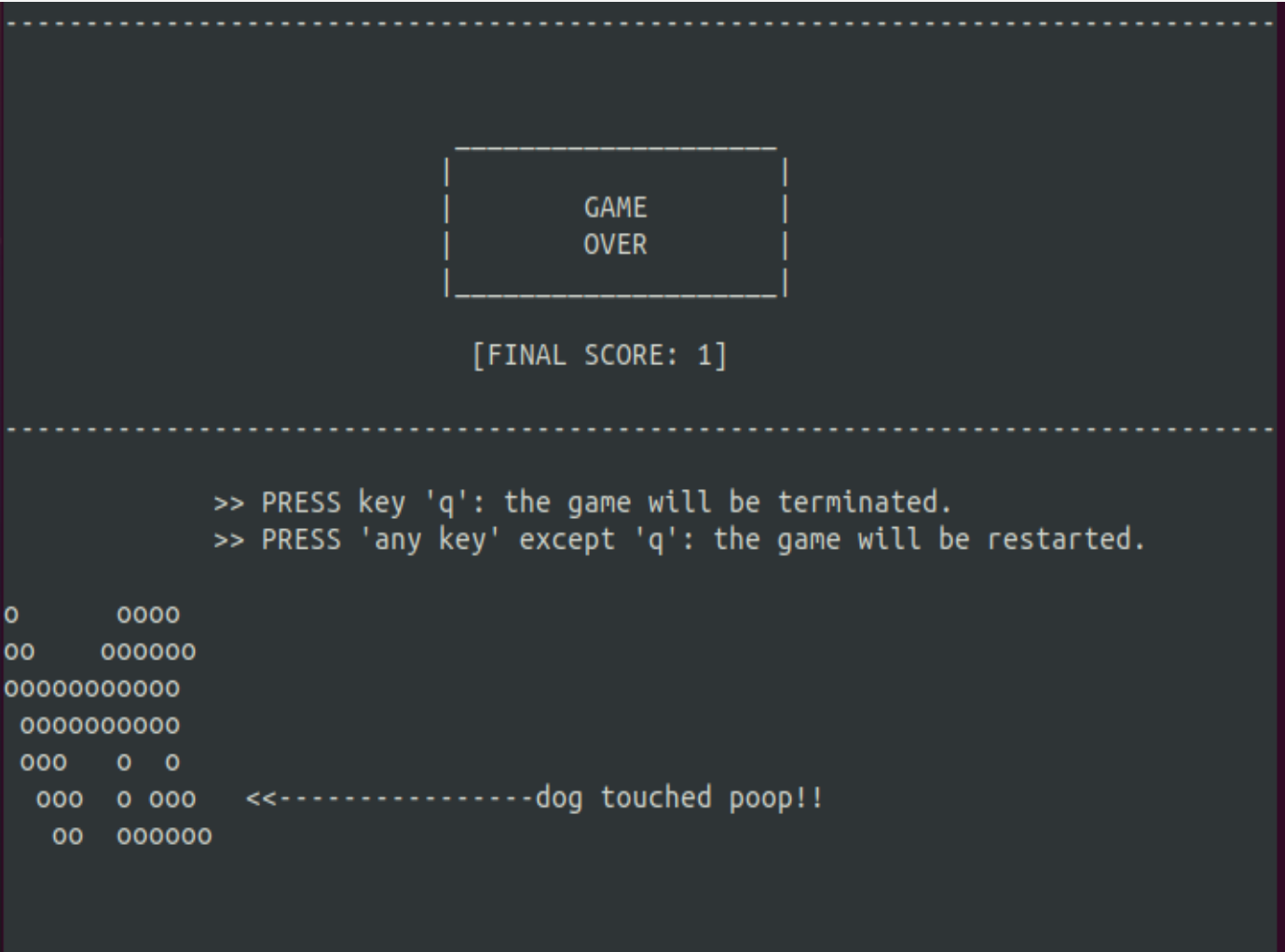
```
+      +      +      +      +      +      +++JumpKey: <space>+++  
+    +      +      +      +      +      |   Jump           |       +      +  
+      +      +      +      +      +      |     and         |       +      +  
+      +      +      +      +      +      |        avoid      |       +      +  
+      +      +      +      +      +      |          poop      |       +      +  
+      +      +      +      +      +      +++++++              +      +  
+      +      +      +      +      +      +      +      +      +      +      +  
+      +      +      +      +      +      +      +      +      +      +      +  
+      +      +      +      +      +      +      +      +      +      +      +  
+      +      +      +      +      +      +      +      +      +      +      +  
+      +      +      +      +      +      +      +      +      +      +      +  
  
0      0000  
00     000000  
000000000000  
  00000000000  
000  000      0  
  000  00      000  
   00  00      00000  
-----
```

# Rush Dog 클라이언트

## 장애물 뛰어넘기 게임



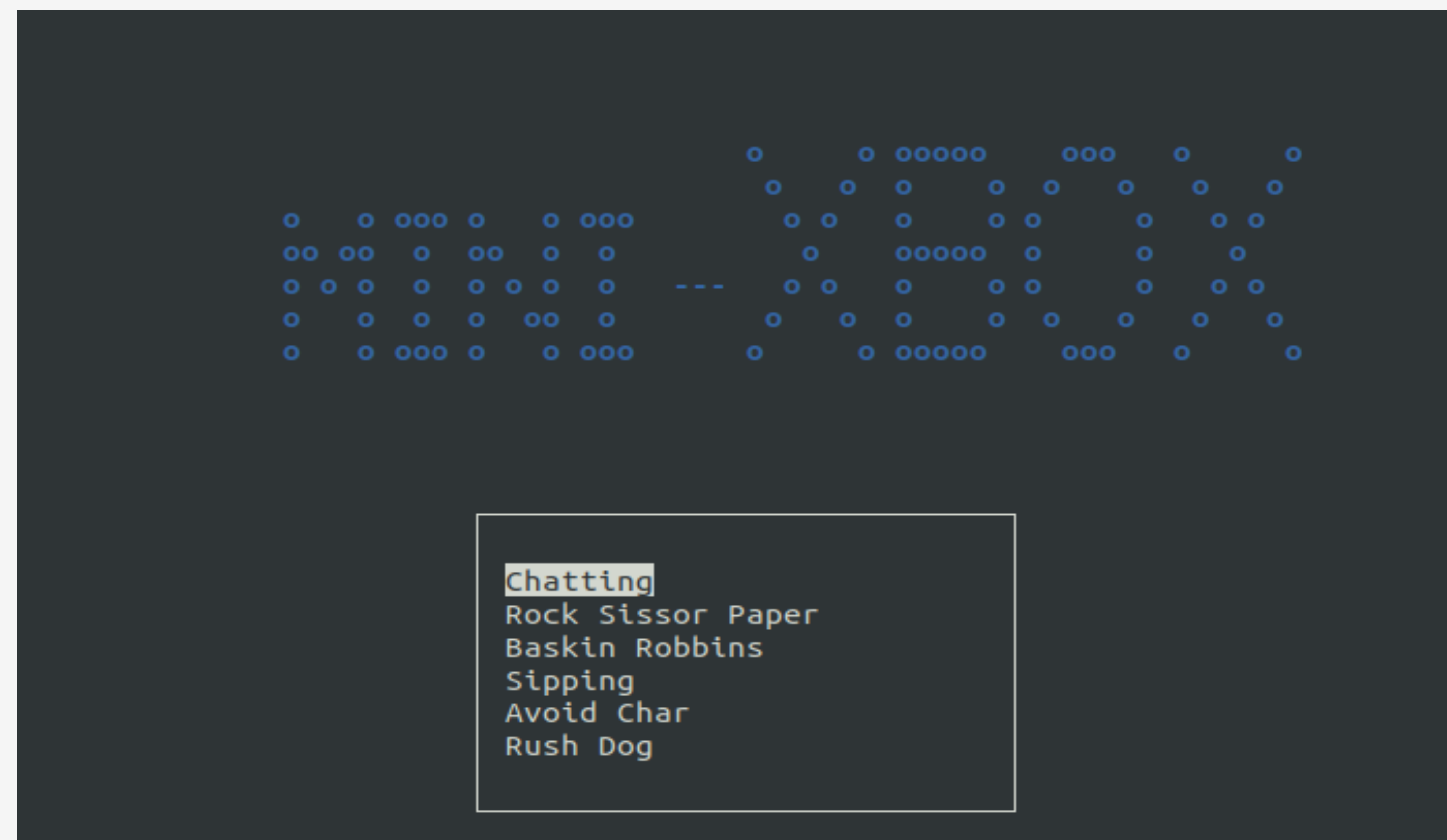
```
mvprintw(0, 0, "-----");
mvprintw(3, 27, " \n");
mvprintw(4, 27, "| \n");
mvprintw(5, 27, "| GAME \n");
mvprintw(6, 27, "| OVER \n");
mvprintw(7, 27, "| \n");
mvprintw(9, 29, "[FINAL SCORE: %d]", score);
mvprintw(11, 0, "-----");
mvprintw(13, 13, ">> PRESS key 'q': the game will be te
rminated.");
mvprintw(14, 13, ">> PRESS 'any key' except 'q': the game will
be restarted.");
mvprintw(21, 15, "<<-----dog touched poop!!"
);
memset(sbuf, '\0', sizeof(sbuf));
sprintf(sbuf, "%d", score);
if(send(sd, sbuf, strlen(sbuf)+1, 0) == -1){
    perror("send");
    exit(1);
}
refresh();
nodelay(stdscr, FALSE);
ch = getch();
if(ch == 'q'){
    memset(sbuf, '\0', sizeof(sbuf));
    strcat(sbuf, "q");
    send(sd, sbuf, strlen(sbuf), 0);
    break;
}
}
endwin();
sleep(1);
close(sd);
return 0;
}
```



게임 종료 시, 종료 화면이 나타나며, q를 누르면 게임에서 나가고 그 외 다른 키를 누르면 게임을 재개한다.

# mini-xbox 활용방안

바쁜 삶 속에서도 많은 사람들이 부담없이 할 수 있는 프로그램을 목적으로 mini-xbox를 만들었다. mini-xbox는 단순한 미니게임들로 구성되어 있기 때문에, 무료하지만 머리를 쓰고 싶지 않을 때 손쉽게 할 수 있고, 쉬는 시간처럼 짧은 시간을 보낼 때에도 할 수 있는 프로그램이다. 오랜 시간을 투자할 필요없고, 중간에 끊어도 문제가 없기 때문에 다음 스케줄이 있거나 이동할 때 하면 유용하다. 또, 자극적인 게임이 전혀 없기 때문에 모든 연령이 이용 할 수 있어, 낮은 연령의 아이들과 놀아줄 때 같이 할 수 있다. 다양한 게임이 존재하기 때문에, 저녁 식사처럼 간단한 내기에 이용하는 것도 좋은 방법이다.





발표를 들어주셔서  
감사합니다 :) )