

# 게임 프로그래밍

날짜 : 2024 - 10 - 10

학번 : 2020575004

이름 : 강창우




# 소스분석 - main

---

```
int main(void)
{
    game_control();
    return 0;
}
```

game\_control 함수를 호출합니다.






# 소스분석 - gotoxy

```
void gotoxy(int x, int y)
{
    COORD Pos = {x - 1, y - 1};
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), Pos);
}
```

콘솔 창에서 커서를 지정  
한 좌표 (x, y)로 이동시키  
는 역할을 합니다.



# 소스분석 - draw\_check01

```
void draw_check01(int c, int r)
{
    int i, j;
    unsigned char a=0xa6;
    unsigned char b[12];
    for(i=1;i<12;i++)
        b[i]=0xa0+i;
    printf("%c%c",a, b[3]);
    for(i=0;i<c-1;i++)
        printf("%c%c", a, b[8]);
    printf("%c%c", a, b[4]);
    printf("\n");
    for(i=0;i<r-1;i++)
    {
        printf("%c%c", a, b[7]);
        for(j=0;j<c-1;j++)
            printf("%c%c", a, b[11]);
        printf("%c%c",a, b[9]);
        printf("\n");
    }
    printf("%c%c", a, b[6]);
    for(i=0;i<c-1;i++)
        printf("%c%c", a, b[10]);
    printf("%c%c", a, b[5]);
    printf("\n");
}
```

주어진 c(열 개수)와 r(행 개수)를 사용해 콘솔에 격자를 그리는 역할을 합니다.

# 소스분석 - game\_control

```
void game_control(void)
{
    int x=1, y=1, matrix[20][20]={0};
    char key;
    do
    {
        gotoxy(1,1);
        draw_check01(18, 18);
        gotoxy(x, y);
        printf("o");
        display_stone(matrix);
        gotoxy(1, 20);
        printf("방향키로 움직이고 ");
        printf("스페이스 키를 누르시오.");
        key=getch();
        if (key==27)
            exit(0);
        else if (key==32)
            matrix[(x+1)/2][y]=1;
        else if (key>=72)
            move_arrow_key(key, &x, &y, 37, 19);
    }while(1);
}
```

게임의 주요 로직을 담당하는 함수로,  
격자판에서 커서를 이동하거나 돌을 놓  
는 역할을 합니다.



# 소스분석 – display\_stone

```
void display_stone(int matrix[][20], int r, int stone_color, char stone_shape)
{
    int x, y;
    for (y = 1; y <= r; y++)
    {
        for (x = 1; x <= 20; x++)
        {
            if (matrix[y][x] == 1)
            {
                change_stone_color(stone_color);
                gotoxy(x * 2, y);
                printf("%c", stone_shape);
            }
        }
    }
    change_stone_color(7);
}
```

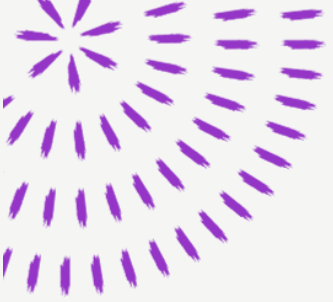
matrix 배열에 저장된 돌의 상태를 화면에 출력하는 역할을 합니다. 배열의 각 값이 돌이 놓여 있는지 여부를 나타내고, 그에 따라 문자를 해당 좌표에 출력합니다.

# 소스분석 - move\_arrow\_key

```
void move_arrow_key(char key, int *x1, int *y1, int x_b, int y_b)
{
    switch (key)
    {
        case 72:
            if (*y1 > 1) (*y1)--;
            break;
        case 75:
            if (*x1 > 1) (*x1)--;
            break;
        case 77:
            if (*x1 < x_b/2 + 1) (*x1)++;
            break;
        case 80:
            if (*y1 < y_b) (*y1)++;
            break;
        default:
            return;
    }
}
```

사용자가 방향키를 입력했을 때, 커서의 위치를 업데이트하는 역할을 합니다.

키보드의 화살표 키 입력에 따라 커서의 x와 y 좌표를 변경하고, 경계를 넘지 않도록 합니다.



# 추가기능





# 추가기능 - 크기 설정

main 함수에 추가

```
printf("열 크기를 입력하세요 (2-20): ");
scanf("%d", &c);
printf("행 크기를 입력하세요 (2-20): ");
scanf("%d", &r);
system("cls");

if (c < 2 || c > 20 || r < 2 || r > 20) {
    printf("유효하지 않은 크기입니다. 2와 20 사이의 값을 입력하세요.\n");
    return 1;
}
```

격자 박스의 크기를 설정해주는 기능

c와 r은 game\_control 함수의 전달인자

# 추가기능 - 크기 설정

열 크기를 입력하세요 (2-20): 6  
행 크기를 입력하세요 (2-20): 6



방향키로 움직이고 스페이스 키를 누르시오. (ESC: 종료, Enter: 초기화)

# 추가기능 - 돌 삭제

game\_control 함수에 추가

```
if (key == 27)
    exit(0);
else if (key == 32)
{
    int posX = (x + 1) / 2;
    if (matrix[y][posX] == 1)
        matrix[y][posX] = 0;
    else
        matrix[y][posX] = 1;
}
```

key == 27 / ESC

key == 32 / 스페이스바

ESC 입력 시 프로그램 종료

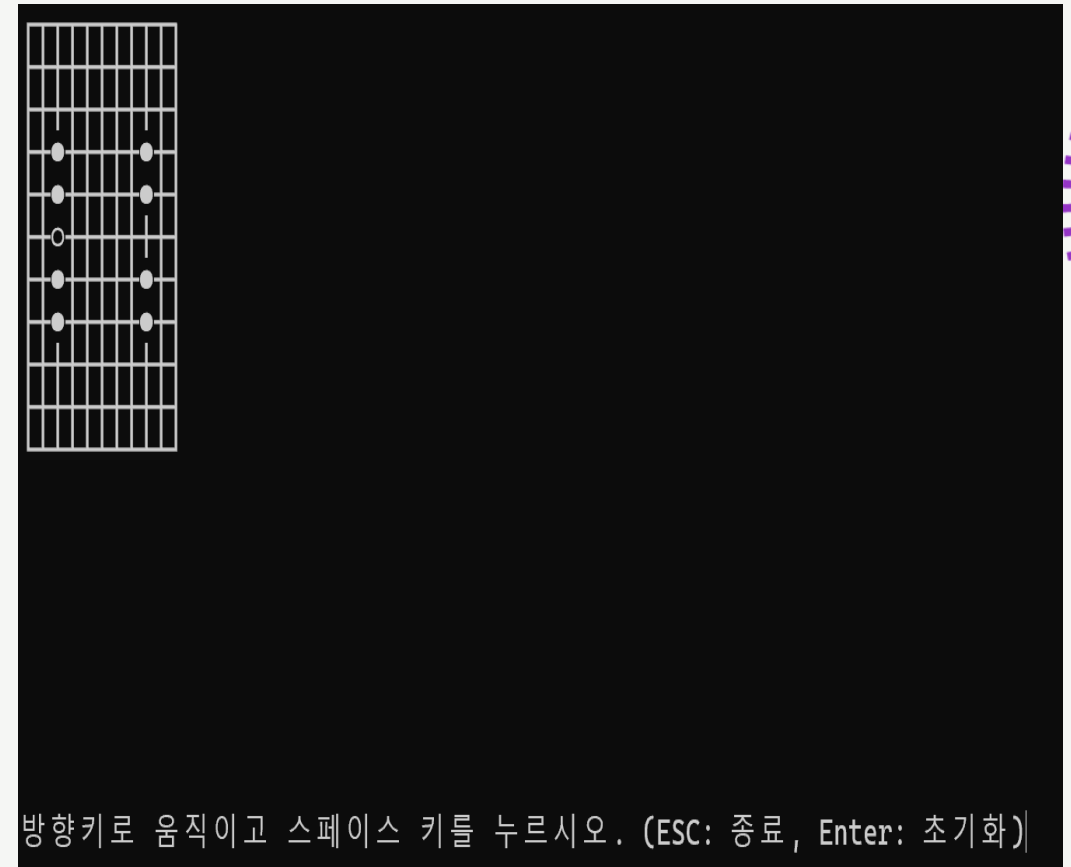
if (matrix[y][posX] == 1) // 돌이 있으면 삭제  
matrix[y][posX] = 0;

else matrix[y][posX] = 1;  
// 돌이 없으면 놓기

# 추가기능 - 돌 삭제



>  
>



# 추가기능 - 초기화

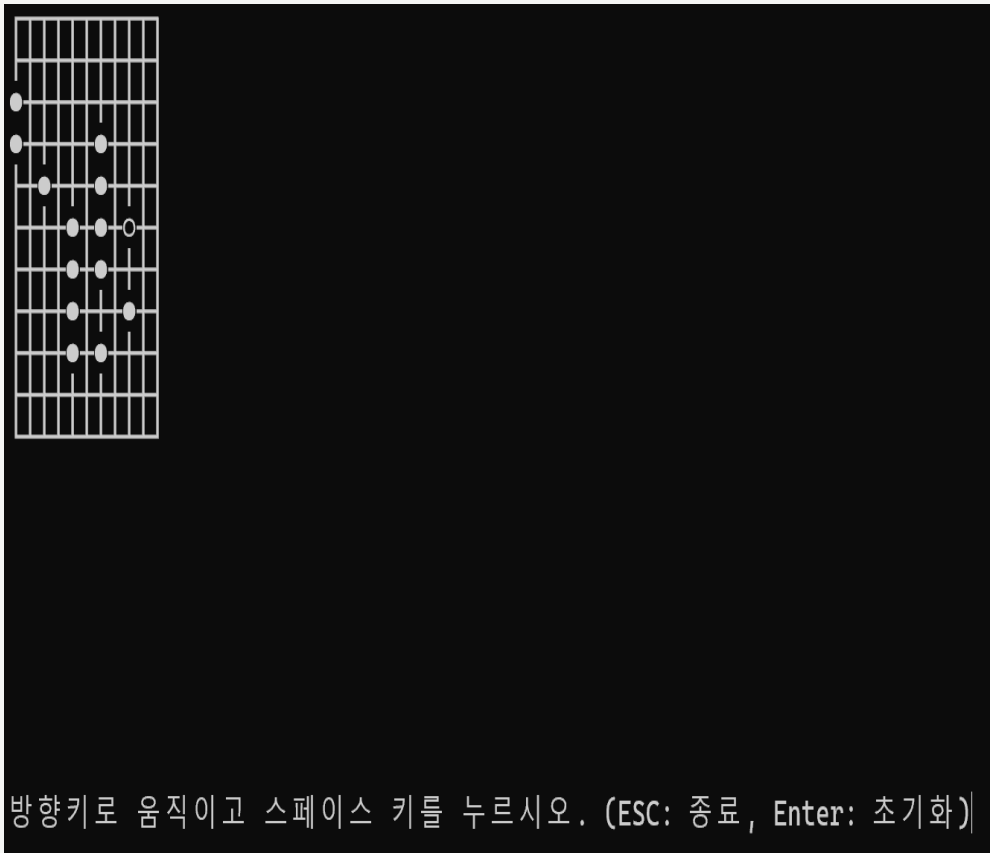
game\_control 함수에 추가

```
else if (key == 13)
{
    for (int i = 0; i < 20; i++)
        for (int j = 0; j < 20; j++)
            matrix[i][j] = 0;
    gotoxy(1, 1);
    draw_check01(c, r);
}
```

key == 13  
Enter키를 입력 시

놓여진 돌을 모두 삭제하는 초기화 기능

# 추가기능 - 초기화



>  
>



# 추가기능 - 돌 색상변경

game\_control 함수에 추가

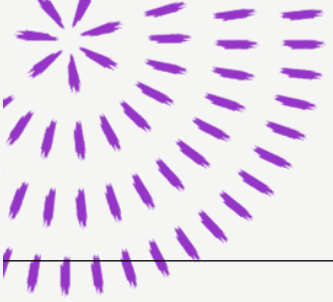
```
else if (key == 'c' || key == 'C')
{
    char color_choice;
    gotoxy(1, r + 4);
    printf("색상을 선택하세요 (1: 빨강, 2: 초록, 3: 파랑): ");
    color_choice = getch();
    stone_color = get_color_code(color_choice);
    gotoxy(1, r + 4);
    printf("색상이 변경되었습니다. ");
}
```

```
void change_stone_color(int color)
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), color);
}

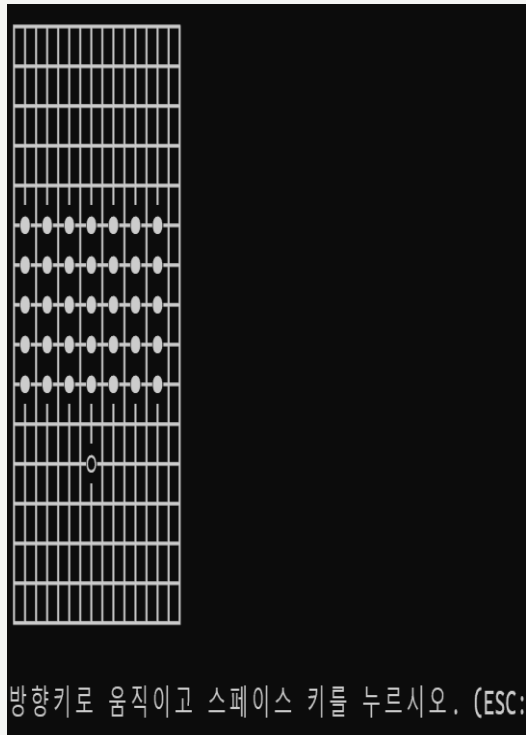
int get_color_code(char color_choice)
{
    switch (color_choice)
    {
        case '1': return 12; // 빨강
        case '2': return 10; // 초록
        case '3': return 9; // 파랑
        default: return 7; // 기본적으로 흰색
    }
}
```

윈도우 콘솔의 기능 중 하나인 색상 변경 기능을 사용하여 색상을 변경한다

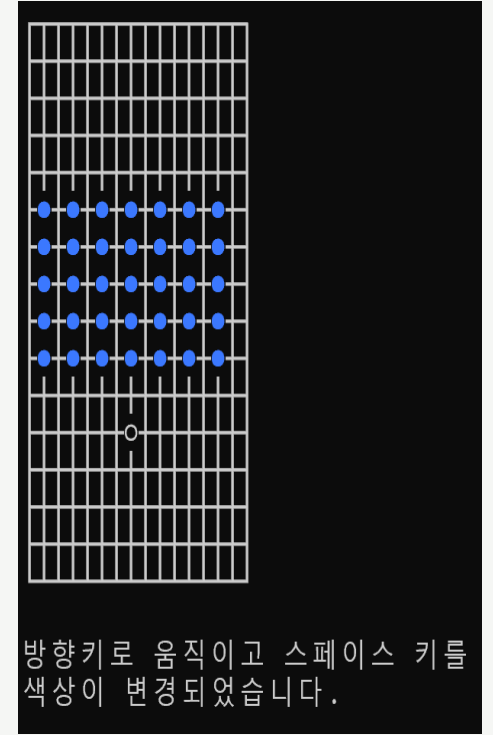
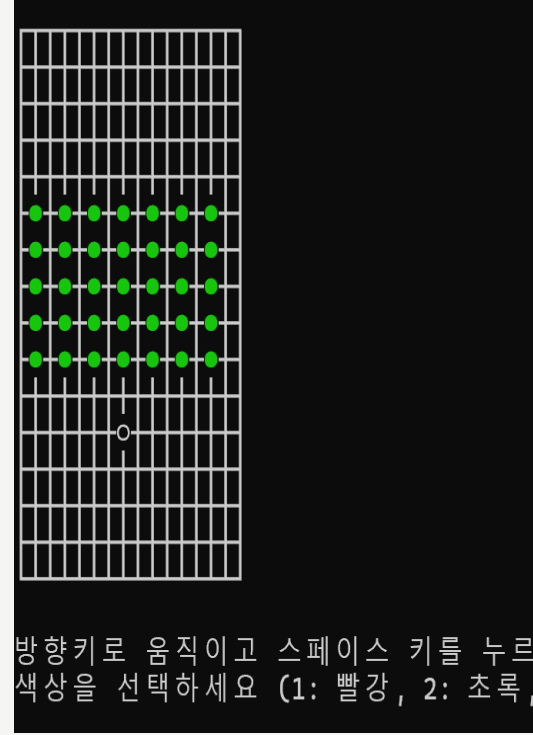
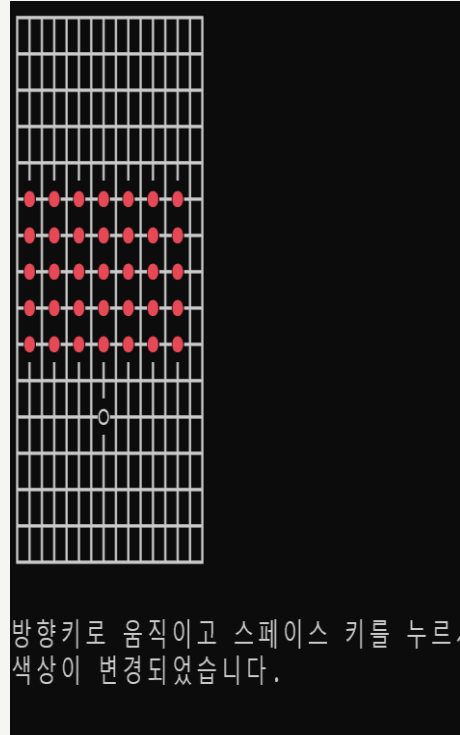




# 추가기능 - 돌 색상변경



>  
>



1,2,3을 입력하여 원하는 RGB 색깔로 바꾼다

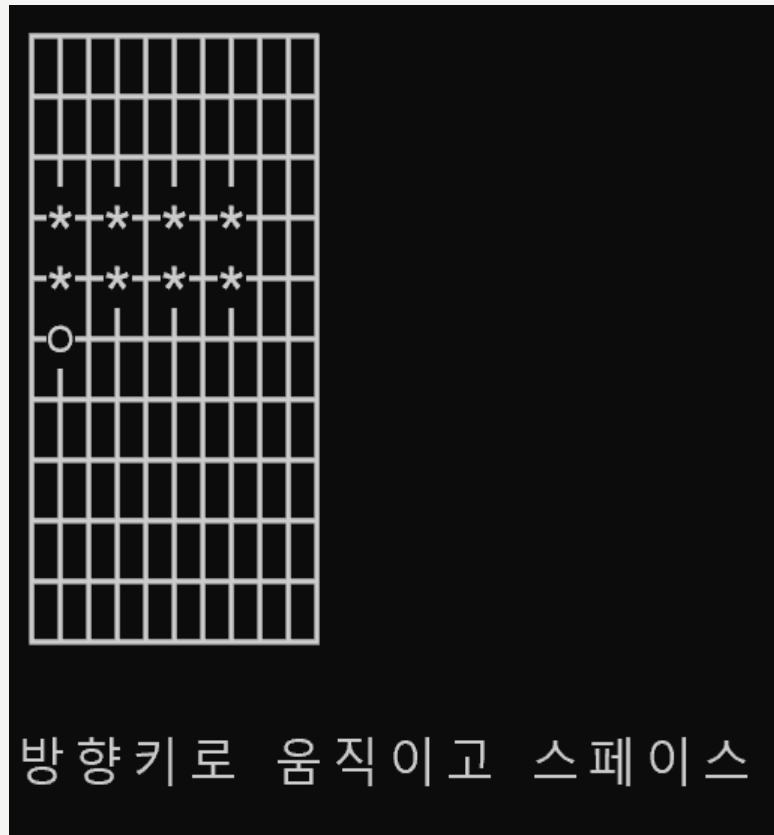
# 추가기능 - 모양변경

game\_control 함수에 추가

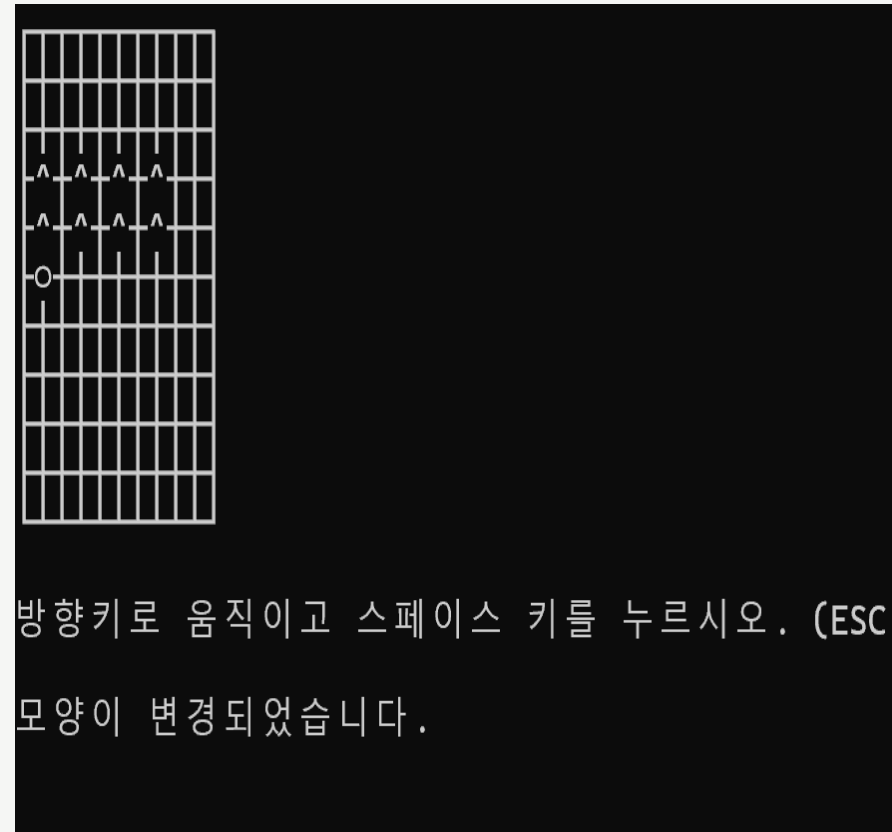
```
else if (key == 's' || key == 'S')
{
    char shape_choice;
    gotoxy(1, r + 5);
    printf("모양을 선택하세요 (1: *, 2: ^, 3: #): ");
    shape_choice = getch();
    switch (shape_choice)
    {
        case '1': stone_shape = '*'; break;
        case '2': stone_shape = '^'; break;
        case '3': stone_shape = '#'; break;
        default: stone_shape = '*'; break;
    }
    gotoxy(1, r + 5);
    printf("모양이 변경되었습니다. ");
}
```

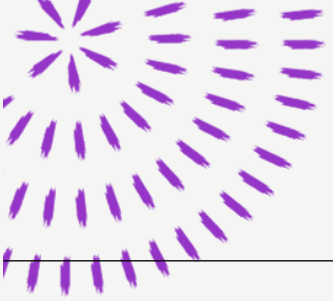
stone\_shape에 사용자가 입력한  
모양이 저장되어,  
display\_stone 함수에서 사용된다

# 추가기능 - 모양변경



>  
>





# 추가기능

1. 크기 설정
2. 돌 삭제
3. 초기화
4. 돌 색상 변경
5. 돌 모양 변경





# 참고자료

---

[cmd 글자색 및 배경색 바꾸는 법 : 네이버 블로그 \(naver.com\)](#)

[\[C언어/C++\] 원하는 좌표로 커서 이동하기 \(SetConsoleCursorPosition\) \(tistory.com\)](#)

[아스키 코드 - 나무위키 \(namu.wiki\)](#)