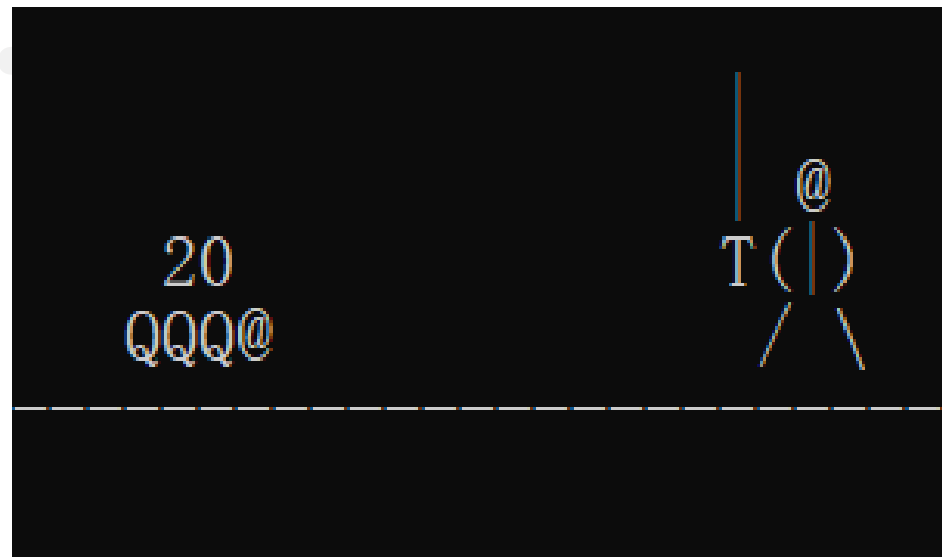


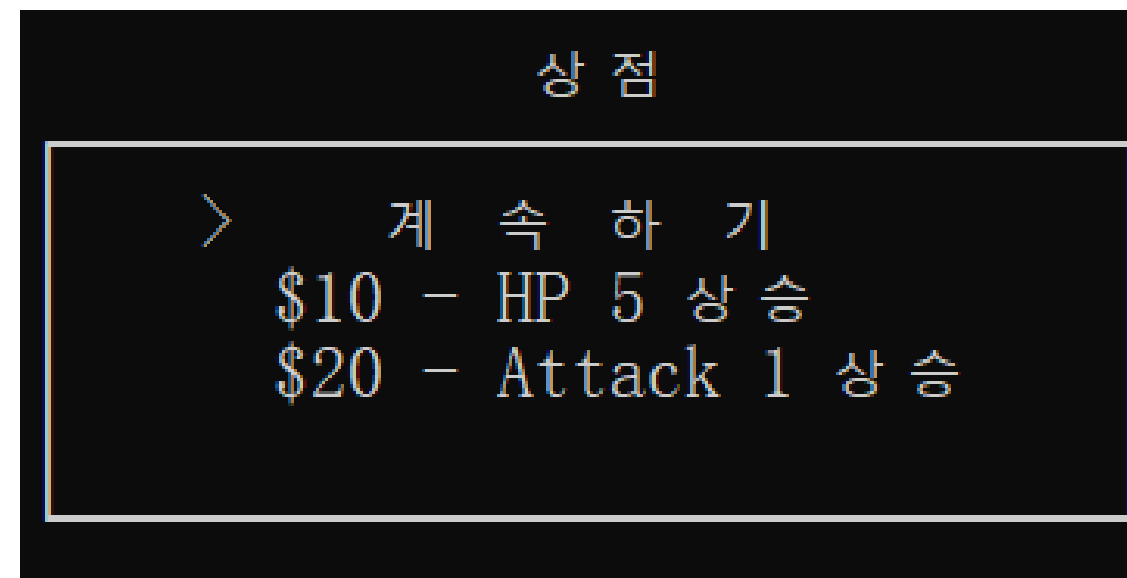
윤희준

# RPG GAME

# 01 캐릭터, 몬스터

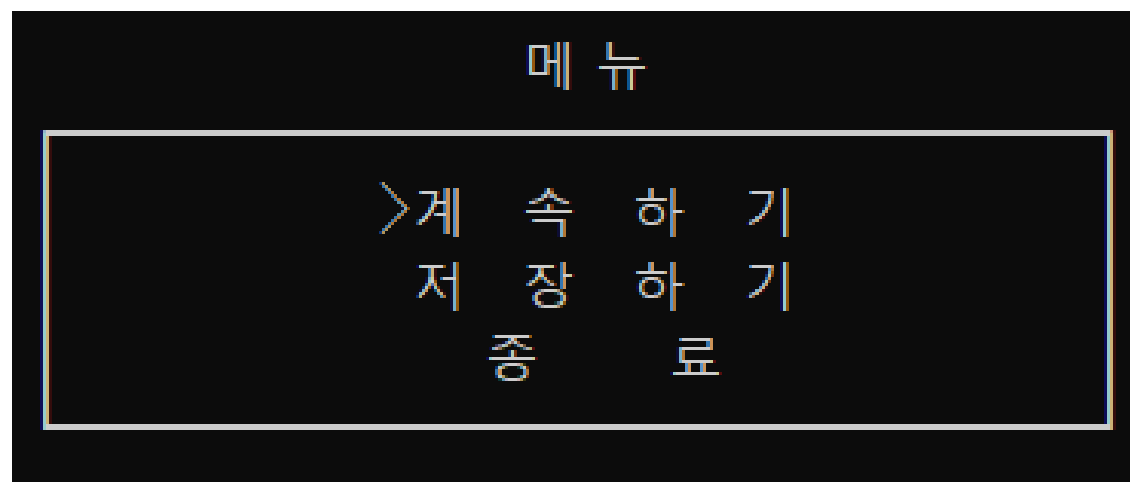


# 02 상점

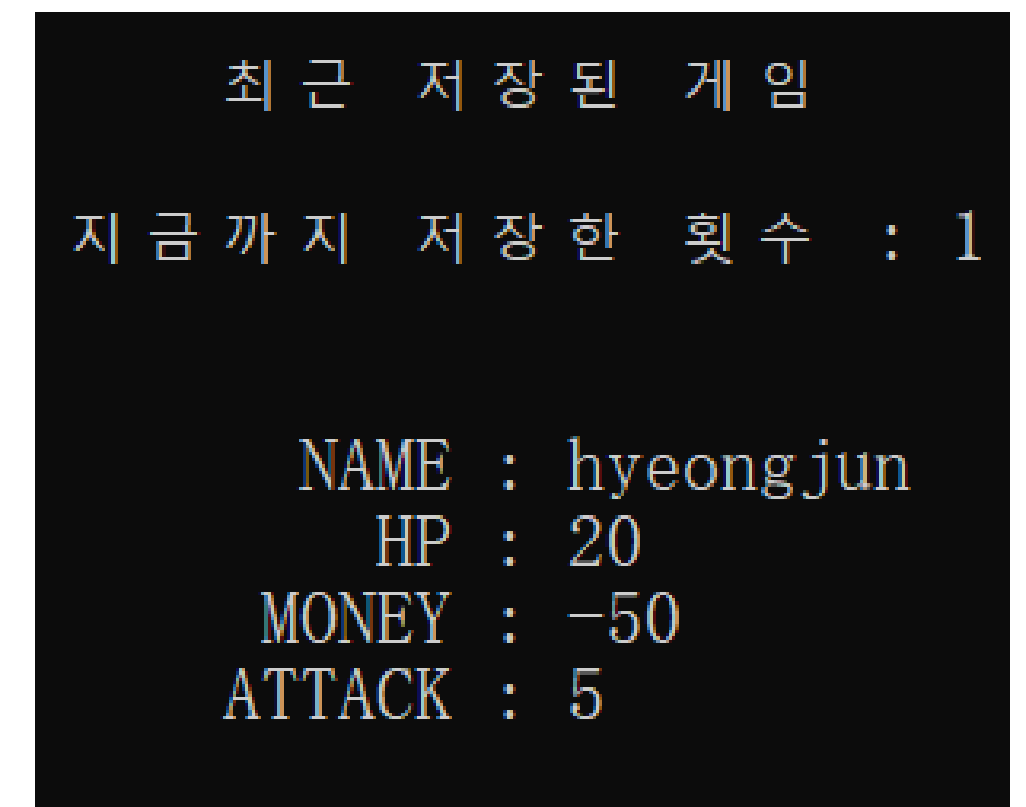


# 기능

# 03 메뉴

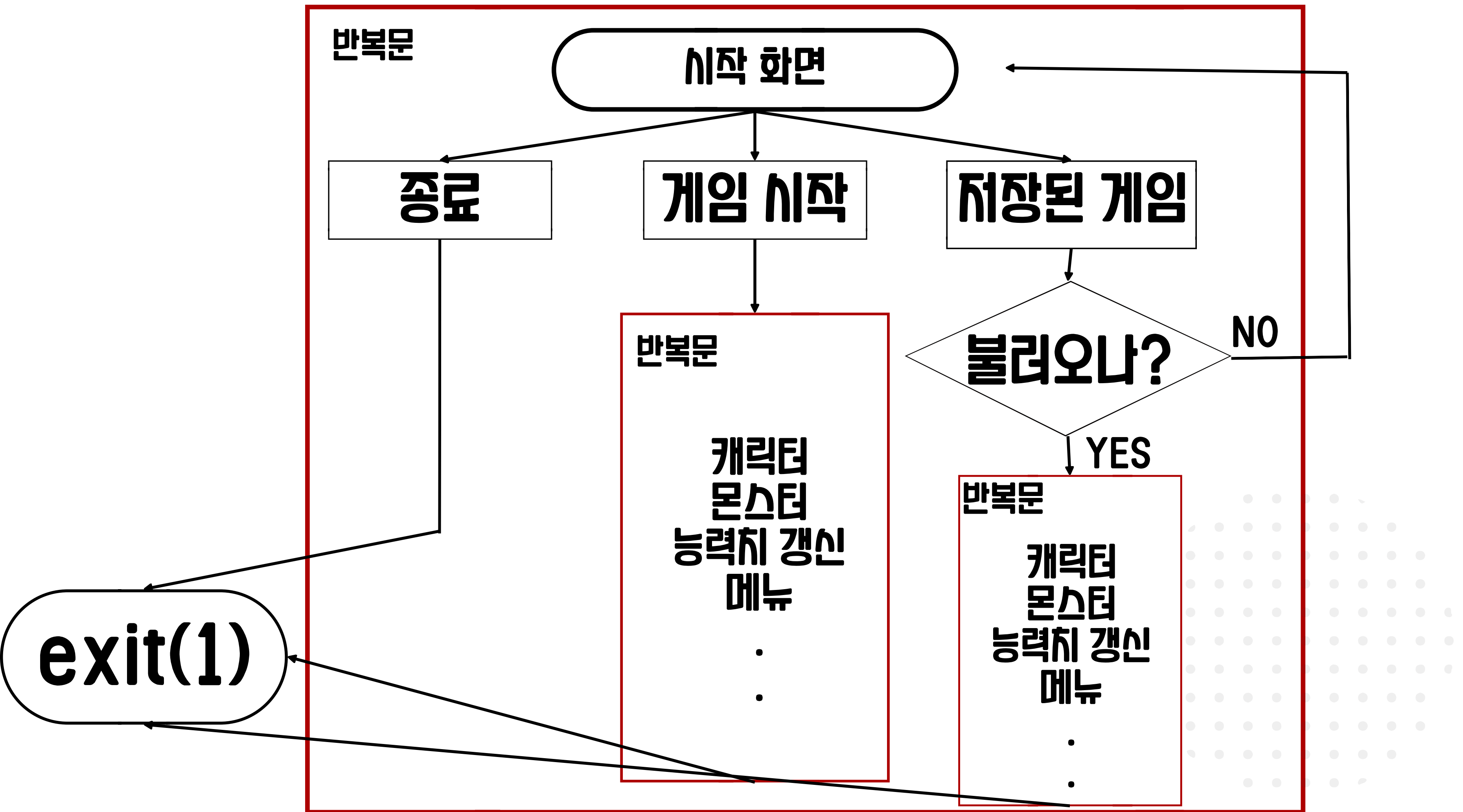


# 04 최근 게임 수치 저장

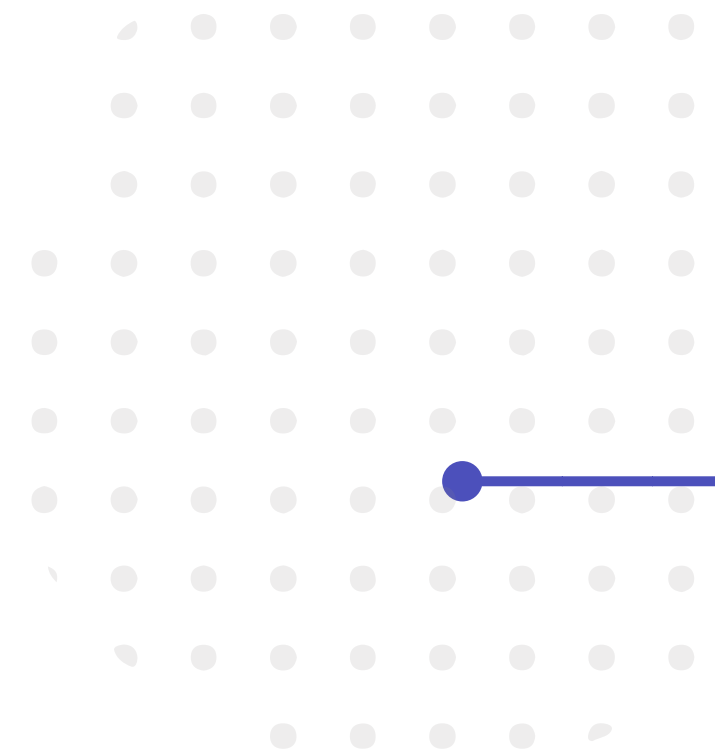




# 전체적인 알고리즘



움직임



현재 x, y

현재 위치에서  
그림 출력

방향키 눌림

바뀐 x, y 값을  
포인터로

현재 위치  
그림 삭제

다음 x, y

다음 위치  
그림 출력

```
void DrawCharacter(int* x, int* y)
{
    gotoxy(*x, *y);
    printf(" @ ");
    if (characterdirection == 1)
    {
        gotoxy(*x, *y + 1);
        printf("/|o");
        gotoxy(*x, *y + 2);
        printf("/ ");
    }
    else if (characterdirection ==
```

```
case RIGHT:
    direction = 1;
    characterdirection = 1;
    if (*x == 106); //오른쪽 벽 확인
    else {
        for (int i = 0; i < 3; i++) {
            gotoxy(*x, *y + i);
            printf(" ");
        }
        (*x)++;
    }
    break;
```



# 캐릭터/몬스터 상호작용

캐릭터 x, y

몬스터 x, y

## bool형 함수

비교반복문

좌표가 일치하는가?

yes

no

true

false

```
bool Damaged(int* x, int* y, int* a)
{
    for (int i = 0; i < 3; i++) {
        if (*a <= *x + i && *x + i <=
            return true; //공격 받음.
        else
            return false;
    }
}
```

```
//몬스터랑 닮았을 경우
if (Damaged(x, y, a)) {
    (*(user + s)).HP = (*(user + s)).HP - (*bug).attack;
    Sleep(100);
}
```





# 저장



# 저장횟수 변수 & stockcount 변수 선언

저장횟수가 0이면 저장횟수  
파일 생성

파일에서 저장횟수 불러오기  
(0이면 변수에 0 대입)

반복문

시작 화면

게임 시작

반복문

게임 진행

저장

종료

캐릭터 정보와 저장횟수  
두 파일 선언

구조체 동적할당 선언

(저장횟수 변수 + 1 만큼)  
동적할당(realloc)

stockcount변수=1

(저장횟수+1)과 현재 게임 정보  
각각 다른 파일에 저장

stockcount에 따라  
메모리 해제 후 종료



**THANK  
YOU**