

C++ 프로그래밍 실습

첫 번째 중간 보고서

제출자명: 강예준

제출자학번: 234091

프로그램 실행 방법

(main.cpp), (main.exe), (save_load.cpp), (user.cpp), (user.h) 파일 다운로드

한 폴더에 위 파일들 넣기 (save파일 생성)

첫 번째 방법: vsCode에서 main.cpp 실행

두 번째 방법:

1. cmd실행
2. main.exe파일 붙여넣기
3. 엔터키 누르고 실행

명령어 목록

up 플레이어를 위로 이동.
down 플레이어를 아래로 이동.
left 플레이어를 왼쪽으로 이동.
right 플레이어를 오른쪽으로 이동.
map 현재 맵 상태 출력.
save 게임 상태 저장.
exit 게임 종료 및 저장.
inventory 인벤토리 확인. // 오류 발생 미완성
use <아이템> 특정 아이템 사용 (예: use 포션).

기능 별 구현 상황

```
void saveGame(const string& user_id, const vector<vector<int>>& map, int user_x, int user_y, const User& user, bool weapon, int armor) {
    ofstream save_file(user_id + "_game_save.txt");

    if (save_file.is_open()) {
        // 맵 저장
        for (const auto& row : map) {
            for (int cell : row) {
                save_file << cell << " ";
            }
            save_file << endl;
        }
        // 플레이어 상태 저장
        save_file << user_x << " " << user_y << endl;
        save_file << user.GetHP() << endl;
        save_file << weapon << " " << armor << endl;

        save_file.close();
    }
}
```

```
// 맵과 사용자 상태를 저장
void saveGame(const string& user_id, const vector<vector<int>>& map, int user_x, int user_y, int user_hp, bool weapon, int armor) {
    string filename = user_id + "_game_save.txt";

    ofstream save_file(filename);

    if (save_file.is_open()) {
        // 맵 데이터 저장
        for (const auto& row : map) {
            for (int cell : row) {
                save_file << cell << " ";
            }
            save_file << endl;
        }

        // 사용자 위치
        save_file << user_x << " " << user_y << endl;
        // 사용자 체력
        save_file << user_hp << endl;
        // 무기 갑옷 상태
        save_file << weapon << " " << armor << endl;

        save_file.close();
        cout << "게임 상태가 '" << filename << "' 파일에 저장되었습니다." << endl;
    }
    else {
        cout << "파일 저장에 실패했습니다!" << endl;
    }
}
```

게임 저장 (11/17)

입력:

- user_id (string): 사용자 ID, 저장 파일 이름에 사용
- user (User): 플레이어 객체, 체력 등 상태 정보
- weapon (bool): 플레이어의 무기 보유 여부
- armor (int): 플레이어가 보유한 갑옷의 수량

설명:

user_id를 기반으로 파일명을 생성하고, 현재 게임 상태를 해당 파일에 저장.

파일에 저장되는 정보:

1. 맵 상태(각 타일의 값).

2. 플레이어의 X, Y 좌표.
3. 플레이어의 체력(HP).
4. 무기 상태 및 갑옷 수량.

파일 저장이 완료되면 사용자에게 저장 완료 메시지 출력.

```
bool loadGame(const string& user_id, vector<vector<int>>& map, int& user_x, int& user_y, User& user, bool& weapon, int& armor) {
    ifstream load_file(user_id + "_game_save.txt");

    if (!load_file.is_open()) return false;

    // 맵 로드
    for (auto& row : map) {
        for (int& cell : row) {
            load_file >> cell;
        }
    }
    // 플레이어 상태 로드
    load_file >> user_x >> user_y;
    int hp;
    load_file >> hp;
    user.IncreaseHP(hp - user.GetHP()); // HP 초기화
    load_file >> weapon >> armor;

    load_file.close();
    return true;
}
```

게임 로드 (11/17)

입력:

user_id (string): 사용자 ID, 로드할 파일 이름에 사용됨.

user (User&): 플레이어 객체, 불러온 체력 상태를 저장.

weapon (bool&): 불러온 무기 상태를 저장할 참조 변수.

armor (int&): 불러온 갑옷 수량을 저장할 참조 변수.

반환값:

bool: 파일 로드 성공 여부.

true: 저장된 파일이 정상적으로 로드됨.

false: 파일을 찾을 수 없거나 읽기 실패.

설명:

user_id를 기반으로 파일명을 생성하고, 해당 파일이 존재하면 게임 상태를 읽어 변수에 저장.

로드되는 정보:

맵 상태.

플레이어의 X, Y 좌표.

플레이어의 체력(HP).

무기 상태 및 갑옷 수량.

파일이 없거나 읽기에 실패하면 false를 반환하며, 사용자에게 메시지 출력.

인벤토리 (11/17) 미완성

```
else if (user_input == "inventory") {
    user.DisplayInventory(); // 인벤토리 출력
} else if (user_input.rfind("use", 0) == 0) { // "use 포션"과 같은 명령
    if (user_input.length() > 4) {
        string item = user_input.substr(4); // 명령어 뒤의 아이템 이름 추출
        user.UseItem(item);
    } else {
        cout << "사용할 아이템의 이름을 입력해주세요! (예: use 포션)" << endl;
    }
}
```

```
//아이템 사용
void User::UseItem(const string& item) {
    if (inventory[item] > 0) {
        inventory[item]--;
        cout << item << "을(를) 사용했습니다. 남은 수량: " << inventory[item] << endl;

        if (item == "포션") {
            IncreaseHP(5);
            cout << "포션 사용으로 HP +5. 현재 HP: " << GetHP() << endl;
        }
    } else {
        cout << item << "이(가) 인벤토리에 없습니다!" << endl;
    }
}

// 인벤토리 표시
void User::DisplayInventory() const {
    cout << "=== 인벤토리 ===" << endl;
    for (const auto& pair : inventory) {
        cout << pair.first << ": " << pair.second << endl;
    }
    cout << "======" << endl;
}
```

플레이어가 획득한 아이템을 인벤토리에 저장하고, 필요할 때 확인할 수 있도록 하여 관리 요소를 추가

// 인벤토리 추가에는 성공했으나 포션이나 아이템 사용시 오류 발생

```
명령어를 입력하세요 (up,down,left,right,map,save,exit): use 포션
사용할 아이템의 이름을 입력해주세요! (예: use 포션)
명령어를 입력하세요 (up,down,left,right,map,save,exit): 잘못된 입력입니다.
```

실행 결과

```
사용자 ID를 입력하세요 : 1111
새로운 게임을 시작합니다.
명령어를 입력하세요 (up,down,left,right,map,save,exit):
```

새로운 유저 접속

```
명령어를 입력하세요 (up,down,left,right,map,inventory,save,exit): inventory
=== 인벤토리 ===
갑옷 : 1
=====
```

인벤토리 시스템

```
1111_game_save.txt (up,down,left,right,map,inventory,save,exit): save
게임 상태가 저장되었습니다.
```

SAVE 시스템

```
사용자 ID를 입력하세요 : 1111
저장된 게임을 불러옵니다.
명령어를 입력하세요 (up,down,left,right,map,inventory,save,exit): map
|아이템|  적  |      |목적지|
-----
USER |      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
```

LOAD 시스템

```
명령어를 입력하세요 (up,down,left,right,map,inventory,save,exit): use 포션
사용할 아이템의 이름을 입력해주세요! (예: use 포션)
명령어를 입력하세요 (up,down,left,right,map,inventory,save,exit): 잘못된 입력입니다.
명령어를 입력하세요 (up,down,left,right,map,inventory,save,exit): use포션
>이(가) 인벤토리에 없습니다!
명령어를 입력하세요 (up,down,left,right,map,inventory,save,exit):
```

USE 시스템 오류

++SAVE시 아이템 사라짐 이슈