

2024 Spring OOP Assignment Report

과제 번호 : 4
학번 : 20230642
이름 : 이채영
Povis ID : ch yng

명예서약 (Honor Code)

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.
I completed this programming task without the improper help of others.

1. 프로그램 개요

- 본 프로그램은 플레이어 캐릭터를 성장시키며 보스 몬스터를 잡는 것을 목표로 하는 텍스트 기반의 RPG이다.
- 플레이어 캐릭터는 8*8 크기의 맵 내에서 이동한다. 맵을 이동하며 empty 칸, random 칸, shop 칸, monster 칸, boss monster 칸에 진입하게 된다. monster칸에서 몬스터를 마주쳐 이길 경우 경험치, 골드를 얻어 레벨을 올리고 상점에서 아이템을 사기 위한 자금을 모을 수 있다. random칸에 진입하게 되면 골드를 얻거나, hp를 얻거나, hp를 잃거나, 아무 일도 일어나지 않게 된다. shop칸에 진입하게 되면 몬스터와 싸워서 이겼을 때 얻은 골드로 아이템을 살 수 있다.(공격력 증가, 방어력 증가, 경험치 증가, 체력 회복, 마나 회복 아이템) boss monster칸에 진입하게 되면 일반 몬스터보다 강력한 보스 몬스터와 싸우게 되며, 이길 경우 게임을 클리어하게 된다.

2. 프로그램의 구조 및 알고리즘

- Class 설명:
 - Character : 플레이어 캐릭터를 다루는 클래스. 게임을 실행했을 때 생성됨.
 - ◆ 멤버 변수: (Private) 최대 체력, 최대 마나, 최대 경험치, 레벨 / (Public) 골드, 공격력, 방어력, 현재 체력, 현재 마나, 현재 경험치, 플레이어의 x, y 좌표
 - ◆ Constructor: 플레이어 캐릭터의 멤버 변수의 초기값을 설정한다.
 - ◆ void show_status(): 현재 플레이어 캐릭터의 정보(공격력, 방어력, 경험치 등)를 출력한다.
 - ◆ void random_room(): 플레이어 캐릭터가 random 칸에 진입했을 때 기능을

수행한다. 각각 1/4의 확률로 현재 체력의 30%를 회복하거나(현재 체력 *1.3 > 최대 체력 시 최대 체력까지 회복), 현재 체력의 30%를 잃거나, 100 골드를 획득하거나, 아무 일도 일어나지 않는다.

- ◆ void check_level(int exp): 몬스터와 싸워 이겨 경험치를 얻은 후, 경험치가 레벨업을 하기에 충분한지 확인한다. 레벨업 조건을 만족하면 print_level_up(int exp) 함수를 호출해 레벨업 시의 메시지를 출력하고, 공격력, 방어력, 레벨 등의 정보를 업데이트한다. 레벨업 조건을 만족하지 않으면 print_state(int exp) 함수를 호출해 레벨업을 하지 않았을 때의 메시지를 출력한다.
- ◆ void print_level_up(int exp): 레벨업 시의 메시지를 출력한다.
- ◆ void print_state(int exp): 레벨업을 하지 않았을 때의 메시지를 출력한다.
- ◆ void two_level_up(int exp): 한번에 두 레벨이 올랐을 때의 메시지를 출력하고, 플레이어의 정보를 업데이트한다.
- ◆ void enter_shop(): 상점에 진입했을 시 메시지를 출력한다. 플레이어가 보유하고 있는 골드를 출력하기 위해 Character 클래스의 멤버 함수로 설정하였다.
- ◆ void buy_attack(): 상점에서 attack 아이템을 살 때 메시지를 출력한다. 아이템을 사기 전 공격력과 산 후의 공격력, 다른 정보들을 모두 출력한다.
- ◆ void buy_defense(): 상점에서 defense 아이템을 살 때 메시지를 출력한다. 아이템을 사기 전 방어력, 산 후의 방어력, 다른 정보들을 모두 출력한다.
- ◆ void restore_hp(): 상점에서 restore_hp 아이템을 살 때 메시지를 출력한다. 아이템 구매 전 체력, 구매 후 체력, 다른 정보들을 모두 출력한다.
- ◆ void restore_mp(): 상점에서 restore_mp 아이템을 살 때 메시지를 출력한다. 아이템 구매 전 마나, 구매 후 마나, 다른 정보들을 모두 출력한다.
- Map: 플레이어가 움직이는 map을 관리하는 클래스. 게임을 실행할 때 생성된다.
 - ◆ 멤버 변수: mapping[8][8]는 맵의 정보를 관리한다. (Random, empty, monster 등의 칸이 어느 위치에 있는지) visit[8][8] 는 플레이어가 방문한 칸을 표시한다.
 - ◆ Constructor: init_mapping[8][8]이라는 char 형태의 2차원 배열에 맵의 정보를 입력하고, mapping[8][8]에 이 정보를 복사하여 저장한다. 또한 visit[8][8]의 모든 값을 0으로 저장하고, visit[0][0] 칸만 1을 할당한다. (캐릭터가 방문

한 칸은 visit을 1로 표시, 방문하지 않은 칸은 0으로 표시)

- ◆ void show_map(Character &ch): terminal에 맵을 표시하는 함수이다. 캐릭터가 있는 칸은 You로, 방문하지 않은 칸은 ?로, 방문한 칸은 빈칸으로 표시한다.
- ◆ int move(Character &ch, Shop &shop): 캐릭터를 실제로 이동시키며, 캐릭터가 Random, Monster, Shop, Boss monster 칸에 들어갔을 때 알맞은 함수들을 호출하여 기능을 수행한다. (실질적으로 main과 같은 기능)
- ◆ int go_up(Character &ch) / int go_down(Character &ch) / int go_left(Character &ch) / int go_right(Character &ch): 각각 캐릭터를 위, 아래, 왼쪽, 오른쪽으로 이동시킬 때 호출되는 함수이다. 캐릭터의 x, y 좌표값을 바꾸어 캐릭터를 이동시킨다.
- ◆ int ready(Character &ch): 맵을 표시한 뒤 사용자에게 1. 캐릭터가 이동할 방향 선택 2. 캐릭터의 상태 표시 3. 게임 종료 중 하나를 선택하도록 한다.
- Monster: 캐릭터가 Monster 칸에 진입했을 때 생성되며, 몬스터의 정보를 관리한다.
 - ◆ 멤버 변수: 몬스터의 difficulty, 초기 체력, 현재 체력, 공격력, 방어력, 보상 골드, 보상 경험치, 속성
 - ◆ Constructor: 매개 변수 (int)difficulty로 플레이어의 x좌표+y좌표를 받아 몬스터의 초기 체력, 공격력, 방어력, 보상 골드, 보상 경험치를 초기화하고 랜덤으로 속성을 결정한다.
 - ◆ int fight(Character &ch): 몬스터 칸에 진입될 시 호출되는 함수이다. Character 클래스의 함수를 불러와 플레이어 캐릭터의 정보를 출력하고, 몬스터의 정보를 출력한다. 그리고 공격 방법을 선택하여 각각에 맞는 함수를 호출하고, 몬스터와 플레이어의 체력을 검사하여 캐릭터가 죽었을 때 / 승리했을 때 각각 알맞은 함수를 호출한다. 몬스터의 체력이 0 미만이 되면 캐릭터가 승리하고, 플레이어의 체력이 0 미만이 되면 몬스터의 체력이 0 미만이 되더라도 플레이어가 죽게 되어 게임이 종료된다.
 - ◆ void show_monster_state(): 몬스터의 공격력, 방어력, 현재 체력/초기 체력을 출력한다.
 - ◆ void attack_basic(Character &ch): 플레이어가 몬스터에게 기본 공격을 수행한다. 몬스터의 체력은 (캐릭터의 공격력)-(몬스터의 방어력)만큼 감소한다.
 - ◆ void attack_skill(Character &ch): 플레이어가 몬스터에게 일반 마법 공격을

수행한다. 플레이어의 마나가 15만큼 소모되며, 몬스터의 체력은 (캐릭터의 공격력*1.5)-(몬스터의 방어력)만큼 감소한다.

- ◆ void attack_fire(Character &ch): 플레이어가 몬스터에게 불 속성 공격을 수행한다. 플레이어의 마나가 15만큼 소모되며, 몬스터가 불 속성인 경우 몬스터의 체력은 (캐릭터의 공격력*2)-(몬스터의 방어력)만큼 감소하고, 이외의 몬스터는 (캐릭터의 공격력)-(몬스터의 방어력)만큼 체력이 감소한다.
- ◆ void attack_grass(Character &ch): 플레이어가 몬스터에게 풀 속성 공격을 수행한다. 플레이어의 마나가 15만큼 소모되며, 몬스터가 물 속성인 경우 몬스터의 체력은 (캐릭터의 공격력*2)-(몬스터의 방어력)만큼 감소하고, 이외의 몬스터는 (캐릭터의 공격력)-(몬스터의 방어력)만큼 체력이 감소한다.
- ◆ void attack_water(Character &ch): 플레이어가 몬스터에게 물 속성 공격을 수행한다. 플레이어의 마나가 15만큼 소모되며, 몬스터가 불 속성인 경우 몬스터의 체력은 (캐릭터의 공격력*2)-(몬스터의 방어력)만큼 감소하고, 이외의 몬스터는 (캐릭터의 공격력)-(몬스터의 방어력)만큼 체력이 감소한다.
- ◆ void attack_by_monster(Character &ch): 몬스터가 플레이어에게 공격을 가한다. 공격을 받은 플레이어의 체력은 (몬스터의 공격력)-(캐릭터의 방어력)만큼 감소한다.
- ◆ void boss(): 보스 몬스터를 마주쳤을 때, 보스 몬스터의 초기 체력, 공격력, 방어력을 재설정하는 함수이다.
- ◆ int fight_boss(Character &ch): 보스 몬스터를 마주쳤을 때, void boss() 함수 다음으로 호출되는 함수이다. 보스 몬스터와 플레이어가 싸우게 되고, 플레이어가 죽었을 때는 0을, 보스 몬스터를 물리쳤을 때는 1을 리턴한다.
- Shop: 게임을 실행했을 때 생성되며, 맵 내의 존재하는 상점의 재고, 판매 아이템 등의 정보를 관리한다.
 - ◆ 멤버 변수: 상점에 남은 attack_item, defense_item, exp_item의 재고를 관리하는 변수
 - ◆ Constructor: attack_left(공격 아이템 재고), defense_left(방어 아이템 재고), exp_left(경험치 아이템 재고)를 1로 초기화한다.
 - ◆ Void select(Character &ch): 상점에 진입했을 때 호출되는 함수이다. 어떤 아이템을 살지 사용자로부터 번호를 selection이라는 변수에 입력받고, 1~5의 수를 입력하면 번호에 맞는 아이템을 관리하는 함수가 호출된다. Selection 값이 6이 되면 상점을 나가게 된다.

- ◆ `Int item_attack(Character &ch)`: 공격력 아이템을 산다고 선택했을 때 호출되는 함수이다. 아이템의 재고가 남아있지 않으면 메시지를 출력하고 1이 반환되고, 골드가 부족할 경우에도 1이 반환된다. 사용자의 골드가 충분하고, 재고가 남아있는 경우에는 아이템을 구매했다는 메시지를 출력하고, 골드를 700 차감시키고 재고에서 1을 빼고, 공격력을 10 증가시킨다. 그리고 (sold out) 되었다는 메시지를 출력하고, selection 값으로 6을 리턴해 상점에서 나가도록 만든다.
- ◆ `Int item_defense(Character &ch)`: 방어력 아이템을 산다고 선택했을 때 호출되는 함수이다. 아이템의 재고가 남아있지 않으면 메시지를 출력하고 1이 반환되고, 골드가 부족할 경우에도 1이 반환된다. 사용자의 골드가 충분하고, 재고가 남아있는 경우에는 아이템을 구매했다는 메시지를 출력하고, 골드를 700 차감시키고 재고에서 1을 빼고, 방어력을 5 증가시킨다. 그리고 (sold out) 되었다는 메시지를 출력하고, selection 값으로 6을 리턴해 상점에서 나가도록 만든다.
- ◆ `Int item_exp(Character &ch)`: 공격력 아이템을 산다고 선택했을 때 호출되는 함수이다. 아이템의 재고가 남아있지 않으면 메시지를 출력하고 1이 반환되고, 골드가 부족할 경우에도 1이 반환된다. 사용자의 골드가 충분하고, 재고가 남아있는 경우에는 아이템을 구매했다는 메시지를 출력하고, 골드를 700 차감시키고 재고에서 1을 뺀다. 그리고 경험치를 충전하고 Character 클래스의 멤버 함수인 `check_level` 함수를 호출하여 레벨업 기준을 충족할 경우 레벨 업하고, 그렇지 않을 경우에는 경험치 값 변화만 출력한다. (sold out) 되었다는 메시지를 출력하고, selection 값으로 6을 리턴해 상점에서 나가도록 만든다.
- ◆ `Int item_hp(Character &ch)`: 체력 회복 아이템을 산다고 선택했을 때 호출되는 함수이다. 사용자의 골드가 충분하지 않을 경우 1이 반환된다. 사용자의 골드가 충분할 경우 골드를 500 차감시키고, 플레이어의 현재 체력을 최대 체력까지 회복시킨다. 그리고 selection 값으로 6을 리턴해 상점에서 나가도록 한다.
- ◆ `Int item_mp(Character &ch)`: 마나 회복 아이템을 산다고 선택했을 때 호출되는 함수이다. 사용자의 골드가 충분하지 않을 경우 1이 반환된다. 사용자의 골드가 충분할 경우 골드를 500 차감시키고, 플레이어의 현재 마나를 최대 마나까지 회복시킨다. 그리고 selection 값으로 6을 리턴해 상점에서 나가도록 한다.

□ 프로그램 실행 흐름

■ 게임 시작

- ◆ main() 함수에서 게임이 시작되며, 초기 화면을 출력하고 사용자 입력(게임 시작/종료)을 받는다.

■ 게임 루프

- ◆ 게임이 시작되면 'Character', 'Map', 'Shop' 객체가 생성된다.
- ◆ 게임의 주요 루프가 시작되며, Map 클래스의 ready() 함수를 통해 플레이어의 다음 행동을 받아 이동, 상태 확인, 게임 종료 중 하나를 선택하게 한다.
- ◆ 플레이어가 이동을 선택하면 Map의 move() 함수가 실행되어 새로운 위치의 이벤트(몬스터 전투, 상점 방문, 랜덤 이벤트)를 처리한다.

■ 이벤트 처리

- ◆ 몬스터와의 전투: Monster 클래스가 전투 로직을 수행하며, 플레이어의 Character 객체와 상호작용한다.
- ◆ 상점 방문: Shop 클래스가 상점 메뉴를 표시하고 플레이어의 선택에 따라 아이템을 구매할 수 있게 .
- ◆ 랜덤 이벤트: Character 클래스의 random_room() 메소드가 실행되어 랜덤 결과(HP 손실, 회복, 금화 획득 등)가 적용된다.

■ 게임 종료 조건

- ◆ 플레이어가 몬스터에 의해 사망하거나 게임에서 승리(보스 몬스터 처치) 또는 사용자가 게임 종료를 선택할 경우 게임 루프가 종료된다.

3. 토론 및 개선

- 본 프로그래밍 과제에서는 클래스를 활용해 다양한 객체들을 관리하는 프로그램을 설계하였다. 캐릭터, 몬스터, 맵, 상점 등 각 요소를 독립적인 객체로 처리함으로써 각 클래스 내 특정 기능을 캡슐화하여 관리하고, 각각의 상태와 행동을 명확히 정의하여 유지 보수와 기능 확장에 유리하게 구현할 수 있었다. 또한 각 클래스끼리의 상호작용을 통해 게임 로직을 구현함으로써 코드의 모듈성과 확장성을 높이는 법을 배울 수 있었다.

- 본 프로그램에 추가할 수 있는 기능으로는 다양한 캐릭터 유형이나 클래스를 선택할 수 있게 하여, 각기 다른 능력치와 특수 능력을 가진 캐릭터를 제공해 플레이어 선택 옵션을 확장하는 방법이 있다. 또한 마법 아이템, 특수 능력 아이템 등의 아이템 시스템 능력을 확장할 수 있고, 플레이어의 진행 상황을 저장하고 불러오는 기능을 추가할 수 있다. 그리고 그래픽 사용자 인터페이스(GUI)를 도입하면 더욱 몰입감 있는 게임을 구현할 수 있다.
- 본 프로그램에서 개선할 수 있는 부분은, 객체 지향 원칙을 좀 더 준수하여 클래스를 분리하는 것이다. 현재 프로그램의 Character 클래스에서 플레이어의 스펙 관리, 전투 관리를 모두 처리하고 있는데 이를 전투 관리 클래스로 분리할 수 있다.

4. 참고 문헌

<https://en.cppreference.com/w/cpp/language/classes>

- 클래스를 활용하기 위해 참고하였다.