

2024 Spring OOP Assignment Report

과제 번호 : Assignment #4

학번 : 20230563

이름 : 김홍근

Povis ID : hongsimi7

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 class를 사용한 객체 지향 프로그래밍을 통해 텍스트 기반의 RPG 게임이다.
- Class를 통해 게임 속의 다양한 객체를 생성하고 관리한다.
- 생성자와 소멸자를 통해 객체가 생성되고 소멸될 때에 객체를 관리한다.
- Character 클래스는 플레이어를 생성하고, 현재 플레이어의 정보를 담는다.
- Monster 클래스는 보스 혹은 몬스터를 생성하고, 현재 몬스터의 정보를 담는다.
- Map 클래스는 게임 맵을 생성하고, 현재 맵의 정보를 담는다.
- Shop 클래스는 게임 상점을 생성하고, 현재 상점의 상태를 담는다.
- Vector 딕셔너리 파일을 가져와, 2차원 리스트로 맵을 구현한다.
- Cstdlib의 rand() 함수를 통해 랜덤으로 프로그램이 작동하게 구현한다.

2. 프로그램의 구조 및 알고리즘

- [Class Character]
 1. Character 클래스는 플레이어가 조종하는 캐릭터에 대한 정보를 관리하는 클래스다.
 2. 멤버 변수로는 attack, defense, gold, level, max_hp, curr_hp, max_mp, curr_mp, max_exp, curr_exp를 가지고 각각은 공격력, 방어력, 골드, 레벨, 최대 HP, 현재 HP, 최대 MP, 현재 MP, 최대 경험치, 현재 경험치를 의미하고 초기의 정보는 생성자를 통해 입력된다.
 3. attack_basic, attack_skill 메서드는 각각 캐릭터가 몬스터에게 기본 공격, 공격 스킬을 가하는 메서드로, 몬스터의 방어력을 매개 변수로 받아 데미지를 계산해 리턴하는 메

서드다.

4. `attack_fire`, `attack_grass`, `attack_water` 메서드는 각각 캐릭터가 몬스터에게 불 속성 공격 스킬, 풀 속성 공격 스킬, 물 속성 공격 스킬을 가하는 메서드로, 몬스터의 방어력과 타입을 매개 변수로 받아 데미지를 계산해 리턴하는 메서드이다.
5. `random_char`은 캐릭터가 랜덤칸에 갔을 때, 랜덤으로 캐릭터의 상태를 변화시키는 메서드이다.
6. `shop_char`은 상점에서 산 아이템을 캐릭터에 적용하는 메서드로 플레이어의 입력을 매개변수로 받아, 캐릭터의 상태를 변화시키는 메서드이다.
7. `reward_gold`는 몬스터를 물리쳤을 때 받는 골드를 더하는 메서드로 몬스터의 `reward`를 매개변수로 받아, 캐릭터의 골드에 추가하는 메서드이다.
8. `lost_hp`는 캐릭터가 몬스터로부터 받은 데미지를 계산하는 메서드로 몬스터가 입힌 데미지를 매개변수로 받아, 캐릭터의 HP에 빼서 계산하는 메서드이다.
9. `show`, `show_gold`, `show_defense`, `show_hp`, `show_mp`는 각각 캐릭터의 상태, 골드, 방어력, HP, MP를 출력하는 메서드이다. 이는 모든 멤버 변수가 `private`에 있어 이를 호출하기 위해 만들어졌다.
10. `level_up`는 경험치 획득을 통해 캐릭터가 레벨업하는 메서드로, 경험치를 매개변수로 받고, 캐릭터의 경험치, 레벨을 계산하고, 레벨업 시에 캐릭터의 상태를 변화시키는 메서드이다.
11. `attack_up`, `defense_up`는 상점에서 각각 산 공격업, 방어업 아이템이 적용된 모습을 출력하는 메서드로 공격력업, 방어업을 매개변수로 받고, 이전 모습과 현재 모습을 출력한다.
12. `exp_up`은 상점에서 산 경험치업 아이템이 적용된 모습을 출력하는 메서드로, 경험치업을 매개변수로 받고, 이전 모습과 현재 모습을 출력하고 `level_up` 메서드에 매개변수를 0을 넣어 사용해 만약 레벨업을 했을 경우, 레벨업한 모습까지 출력한다.
13. `restore_hp`, `restore_mp`는 상점에서 산 HP, MP회복 아이템이 적용된 모습을 출력하는 메서드로 이전 모습과 현재 모습을 출력한다.
14. `Rounds` 함수는 유리수를 매개변수로 받아 반올림을 하여 정수로 바꾸어 리턴하는 함수이다.

□ [class Monster]

1. Monster 클래스는 몬스터의 정보를 관리하는 클래스로 몬스터와 보스를 생성한다.
2. 멤버 변수로는 max_hp, curr_hp, attack, defense, reward가 있고 각각 최대 HP, 현재 HP, 공격력, 방어력, 보상을 의미한다.
3. 생성자는 오버로딩을 통해 두 가지로 나뉘는데, 매개변수가 없는 것은 보스를 만들기 위한 생성자이고, 시작점으로부터의 거리가 매개변수로 있는 생성자는 일반 몬스터를 만들기 위한 생성자이다. 이때, 일반 몬스터는 거리에 따라 몬스터의 상태가 더 강해진다.
4. show, show_defense, show_hp, show_reward, show_type은 각각 몬스터의 상태, 방어력, HP, 보상, 타입을 출력하는 메서드로, 이는 모든 멤버 변수가 private에 있어 이를 호출하기 위해 만들어졌다.
5. monster_attack은 캐릭터 방어력에 대한 몬스터의 공격 데미지를 출력하는 메서드로, 캐릭터의 방어력을 매개변수로 받고, 데미지를 계산하여 리턴한다.
6. lost_hp는 캐릭터의 공격으로 인한 몬스터의 HP를 감소시키는 메서드로, 캐릭터가 입힌 데미지를 매개변수로 받아, 몬스터의 HP에 빼서 계산하는 메서드이다.

□ [Class map]

1. Map 클래스는 격자 형태 맵의 정보를 관리하는 클래스이다.
2. 멤버 변수로는 element, condition, char_row, char_col이 있고 각각은 맵의 요소, 맵의 개척 현황, 캐릭터의 위치(row), 캐릭터의 위치(column)을 의미한다.
3. w_move, a_move, s_move, d_move은 각각 위, 왼쪽, 아래, 오른쪽으로 캐릭터가 이동할 수 있는지 확인하는 메서드로 이동할 수 있으면 0, 불가능하면 1을 출력한다.
4. check_map은 캐릭터가 이동한 칸의 이벤트를 확인하는 메서드로, 빈칸이면 1, 랜덤칸이면 2, 몬스터칸이면 3, 상점칸이면 4, 보스칸이면 5를 출력한다.
5. make_empty는 캐릭터가 이동한 칸의 이벤트가 다시 일어나지 않도록 빈칸으로 만드는 메서드다.
6. show는 현재 맵을 보여주는 메서드이다.
7. manhattan_dist는 출발점으로부터 거리를 출력하는 함수로, 이는 거리에 따른 몬스터의 강함을 적용하기 위해 만들어졌다.

□ [Class Shop]

1. Shop 클래스는 맵에 존재하는 상점의 정보를 관리하는 클래스이다.
 2. 멤버 변수로는 attack_up, defense_up, exp_up가 있고, 각각 공격업의 재고, 방어업의 재고, 경험치의 재고를 의미하고, 이는 생성자로 초기화를 한다.
 3. show_list는 상점 재고 및 현황을 보여주는 메서드이다.
 4. stock_attack, stock_defense, stock_exp는 각각 공격업, 방어업, 경험치업의 재고를 확인하는 메서드로 각각 attack_plus, defense_plus, exp_plus를 리턴한다.
 5. sold_item은 재고 수를 줄이는 메서드로 플레이어의 입력을 매개변수로 받아 플레이어가 선택한 아이템의 재고 수를 줄이는 메서드이다.
- 메인 소스 파일에서 변수 user, exit, direction, game_over, monster_out, shop_out을 생성한다. 이 때 각각 유저의 입력, 나가기 신호, 유저 방향 입력, 유저 사망 신호, 몬스터 사망 신호, 상점 나가기 신호를 의미한다.
 - 초기 화면을 출력하고 while문을 이용하여 잘못된 입력을 받았을 때 다시 입력을 할 수 있도록 한다. 이때 플레이어가 2를 입력했다면 프로그램을 종료한다. 만약 플레이어가 1을 입력했다면 게임이 시작되고, 맵, 캐릭터, 상점을 생성한다.
 - while문을 이용해 게임에서의 플레이어의 동작을 입력 받는다. 이때, 3을 입력하면 게임은 종료되어 프로그램이 종료된다. 만약 2를 입력했다면, 캐릭터의 현재 상태를 출력하고 다시 플레이어의 동작을 입력 받는다.
 - 만약 1을 입력했다면, 어떤 방향으로 이동할 것인지 입력을 받는다. 이때, if else문을 통해 어떤 입력이 들어왔는지 확인하고, Map 클래스의 메서드인 w_move, a_move, s_move, d_move를 통해 이동 가능한지 확인 후, 불가능하면 다시 입력을 받는다. 만약 이동할 수 있다면 이동을 한다.
 - 만약 이동한 칸이 빈칸이라면, 플레이어가 이동한 현재 맵을 출력 후, 다시 플레이어의 동작을 입력 받는다.
 - 만약 이동한 칸이 랜덤칸이라면, Character 클래스의 메서드인 random_char를 통해 캐릭터의 상태를 임의로 조정하고, Map 클래스의 메서드인 make_empty를 이용해 빈칸으로 만들고, 다시 플레이어의 동작을 입력받는다.
 - 만약 이동한 칸이 일반 몬스터 칸이라면, 동적 할당을 통해 몬스터를 생성한다. 그리고 damage라는 변수를 생성해 데미지를 저장하는 용도로 사용한다. 플레이어와 몬스터의 상태를 출력하고 플레이어가 어떤 공격을 할 것인지 입력을 받는다. 만약 스킬 공격을 입력 받았는데 MP가 부족하다면 공격을 하지 않는다. 정상적으로 공격이 가능하다면, Character 클래스의 메서드인 attack_basic, attack_skill, attack_fire, attack_grass, attack_water를 통해 데미지를 계산하고, Monster 클래스의 메서드인 lost_hp를 통해 피해

를 입힌다. 그리고 Monster 클래스의 매서드인 `monster_attack`을 통해 데미지를 계산하고, Character 클래스의 매서드인 `lost_hp`를 통해 피해를 입는다. 만약 캐릭터가 사망했으면 프로그램을 종료하고, 그렇지 않고 몬스터가 죽으면 보상을 주고 몬스터를 동적 할당 해제한 다음, 다시 플레이어의 동작을 입력 받는다. 두 상황 둘 다 아니라면, 다시 몬스터와 배틀을 진행한다.

- 만약 상점칸에 갔다면, 어떤 아이템을 살 것인지 입력을 받는다. 이때, 6을 입력받으면, 상점에 나와 플레이어의 동작을 입력 받는다. 만약 1에서 5 사이의 수를 입력 받았는데, 돈이 부족하면 다시 입력을 받는다. 돈이 충분하다면 산 아이템을 캐릭터에 적용하고 상점을 나와 다시 플레이어의 동작을 입력 받는다.
- 만약 보스칸에 도착했다면, 동작 할당을 통해 보스를 생성한다. 그리고 몬스터의 배틀처럼 똑같이 진행한다. 이때, 다른 점이라고 하면, 보스가 죽었을 때이다. 만약 보스가 죽는다면 승리했다는 문구를 출력한 다음, 프로그램을 종료한다.

3. 토론 및 개선

- 클래스를 정의하는 방법 및 클래스를 통해 객체를 생성하여 유용하게 사용하는 방법에 대해 알 수 있었다. 클래스에는 `private`와 `public`이 있어 데이터를 숨길 수 있다는 것을 알 수 있었다.
- 클래스를 통해 몬스터 객체를 동적 할당하고, 해제함으로 메모리를 효율적으로 사용했다. 그러나 코딩 도중에 생각났기에 다른 `character`, `map`, `shop`은 동적 할당을 하지 않아서 메모리 낭비가 생겼다. 따라서 이에 대해 수정하면 더 좋은 코딩이 될 것이다.
- 생성자를 오버로딩하여 같은 이름의 생성자를 통해 보스와 일반 몬스터가 만들어지도록 했다.
- `Vector` 디렉터리를 통해 2차원 리스트를 사용하여 맵을 더 효율적으로 구성되게 했다.
- `Cstdlib`의 `rand()` 함수를 통해 랜덤으로 프로그램이 작동하게 구현한다.
- 모든 멤버 변수를 `private`에 넣어 `public`으로 메서드를 새로 만들어야만, 메인 소스 파일에서 접근할 수 있도록 하여 상대적으로 많은 양의 코딩을 했다. 따라서 이를 `public`에 옮기는 것도 한 번 해봐야 한다고 생각했다.

4. 참고 문헌

- 해당사항 없음.