**2024 Spring OOP Assignment Report**

과제 번호 : 4-1

학번 : 20230499

이름 : 김재환

Povis ID : carotinoid

**명예서약 (Honor Code)**

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

Mingw(g++): g++ -fdiagnostics-color=always -o main.exe Map.cpp Monster.cpp Character.cpp Shop.cpp main.cpp  
vs(cl): Visual Studio 2022 Community 64bit 17.9.1v(.NET : 4.8.09032), ISO C++14

1. **프로그램 개요** 
   * 다음은 프로젝트 내에 포함된 파일에 대한 설명이다.
     + main.cpp : 프로그램을 실행하는 메인 파일
     + (class).h : 클래스의 프로토타입을 정의하는 헤더파일
     + (class).cpp : 클래스의 멤버함수를 정의하는 소스파일

클래스는 Character, Map, Monster, Shop이 있다.

* 이번 프로그램은 텍스트 기반 RPG게임이다. 위 네 가지 클래스를 이용해 게임을 구현해야한다. 게임을 시작하면 맵의 왼쪽 위 끝에서 시작되며, 맨 오른쪽 아래에 위치한 보스를 잡는 것이 목표이다. 게임을 시작하고 난 이후 플레이어는 게임 종료하거나, 현재 스테이터스를 확인하거나, 지도에서 상하좌우 네 가지 방향 중 하나를 선택해 이동할 수 있다. 플레이어가 이동한 칸이 처음 방문한 칸이라면 지도의 칸에 따라서 상점 / 몬스터 / 랜덤 이벤트가 발생하거나 이벤트가 발생하지 않을 수 있다. 몬스터는 시작지점에서 멀어질수록 강력해지며, 보상 또한 증가한다. 플레이어는 몬스터를 잡아 경험치를 얻어 레벨업을 하면 강해질 수 있고, 이를 반복해 보스를 잡아내는 것이 목표이다.

1. **프로그램의 구조 및 알고리즘**

* 메인 소스 파일에 구현된 함수에 대한 설명은 아래 페이지에 기술하였다. (p.3 ~ p.4)
* 각 클래스의 구현 방식에 대한 설명은 아래 페이지에 기술하였다. (p.5 ~ p.7)
* 속성(타입)은 열거형 타입을 통해 구현된다. 그에 해당하는 코드는 다음과 같다.  
  enum type {FIRE, GRASS, WATER, NONE}  
  열거형을 통해 0에서 3 사이의 정수를 통해 속성을 표현할 수 있다.

1. **토론 및 개선**

* 본격적으로 코드를 작성하기 전에 충분히 클래스를 설계했다고 생각했으나 코드를 작성하면서 계획과는 다른 방식으로 코드를 작성해야 할 경우가 생겼다. 가장 큰 문제는 캐릭터와 몬스터의 전투 부분을 작성하면서 순환 종속성이 발생한 부분으로, 캐릭터 인스턴스가 몬스터에게 데미지를 주기 위해서 몬스터 객체를 매개변수로 받으며, 반대로 몬스터가 캐릭터에게 데미지를 주기 위해 캐릭터 객체를 받게 설계한 부분이 문제였다. 순환 종속성을 해결하는 방법으로 전방선언을 하는 방법이 있다. 각 클래스 헤더 맨 앞에 다른 클래스의 프로토타입을 정의하는 방법이 있지만, 클래스 포인터로만 선언 가능하며 다른 헤더 클래스의 멤버함수를 사용할 수 없다. 다른 방법으로는 중재자를 작성하거나 종속성 관계를 끊는 방법이 있다. 이번 과제에서는 몬스터의 공격방식을 멤버함수에서 메인 파일로 옮겨서 종속성을 제거했다.
* 본 게임을 플레이해보면 현재 캐릭터 스테이터스, 맵, 현재 상황을 한눈에 확인하기 어렵다는 문제가 있다. 이를 해결하기 위해 TUI 인터페이스를 작성할 수 있다. TUI 인터페이스는 CLI와 GUI의 중간 정도의 개념으로, 텍스트를 이용해 그래픽을 사용한 것처럼 인터페이스를 만드는 것이다. TUI를 적용해 전체 화면을 세 구역으로 나누고, 아래에는 캐릭터 스테이터스, 오른쪽에는 맵, 왼쪽에는 기존 텍스트 정보를 출력하면 가독성이 높아질 것이다. 이를 구현하기 위해 TUI 라이브러리를 사용하면 좋을 것이다.
* 클래스를 작성하면서 모든 변수를 프라이빗 구역에 정의하였고, 이 변수를 가져오거나 변경하기 위해 멤버함수를 작성하였다. 다만 단순히 변경 및 접근하기 위해서 이런 방식으로 코딩하는 것은 비효율적이라는 생각이 들었다. 인스턴스 변수를 퍼블릭 구역에 정의하거나, 프라이빗 변수에 접근하는 멤버 함수의 기능을 더욱 구체화하면 좋을 것 같다. 또한 변수를 언제 프라이빗 구역에 정의하는지, 아니면 퍼블릭 구역에 정의하는지 공부할 것이다.
* 각 클래스를 설계할 때, 이번 과제에서 작성한 것 처럼 다른 클래스에 직접 접근해서 값을 변경하는 방식으로 작성한다면, 추후 많은 양의 클래스를 만들었을 때 종속성 문제가 발생 할 것이다. 따라서 이런 방식이 아닌, 메인 파일에서 각 클래스를 연결하거나, 각 클래스를 서로 연결하는 새로운 클래스를 만들면 더욱 깔끔한 코드가 될 것이다.

1. **참고 문헌**.

* Prata, S. (2013). C Primer Plus (6th ed.).
* Lecture note, 조성현.

main.cpp 소스 파일에 구현되어 게임을 진행하는 함수에 대한 설명이다. 다음 5가지 함수는 모두 출력 전용으로 실행된다. 따라서, 출력 전용 함수의 반환형은 모두 보이드 형이다.

* print\_logo()  
  프로그램 실행시 Text RPG 아스키 아트를 출력한다.
* show\_map(Map&, Character&)  
  맵 클래스와 캐릭터 클래스의 인스턴스로부터 현재 맵의 정보와 플레이어블 캐릭터의 위치를 받아와 적절한 형태로 출력한다.
* show\_character\_status(Character&)  
  플레이어블 캐릭터의 정보를 받아와 현재 스테이터스를 출력한다.
* show\_changed\_character\_status(Character&, int, int, int, int, int, bool=false, bool=false)  
  플레이어블 캐릭터의 스테이터스에 상태가 변화됨을 출력하는 함수다. 단, 실제로 상태를 변경하지 않음에 주의해야 한다. 5가지 정수형 매개변수를 받으며, 각각 공격력, 방어력, 체력, 마력, 경험치의 변화량을 뜻한다. 받은 경험치로 인하여 레벨이 증가한다면, 그에 따른 변화량 또한 출력에 포함한다. 이 함수는 변화량이 있을 때에만 화살표 기호를 사용하여 변화를 출력하지만, 상점 아이템 구매 등으로 체력, 마력에는 변화가 없을 경우에도 출력해야할 상황이 있다. 이 때 논리형 매개변수가 참인 경우 강제로 화살표 기호를 사용해 변화를 출력한다.
* show\_changed\_shop\_stock(Shop&, int)  
  정수형 매개변수 번호에 맞는 상품의 재고가 감소한 상태로, 재고의 변화를 출력한다.

다음 4가지 함수는 게임의 진행에 사용되는 함수이다. 게임 진행에 필요한 인스턴스 변수의 수정은 레퍼런스를 통해 이루어지도록 작성하였고, 함수의 반환형은 모두 보이드 형이다.

* game()  
  게임을 진행하는 함수이다. 메인 함수에 게임 전체를 넣는 것 대신에, 메인 함수에서 게임을 시작할 것인지 물어본 후, 1을 선택하면 이 함수가 호출되며 게임이 시작된다. 함수가 호출될 때 캐릭터, 맵, 상점 클래스의 인스턴스를 생성한다. 현재는 게임을 승리 또는 패배 후, 현재는 프로그램이 종료된다. 그러나 이 함수를 다시 호출하도록 하면 손 쉽게 게임을 재시작하도록 수정할 수 있다.
* fight(Character&, Monster&,, bool)  
  몬스터와 전투를 담당한다. 이 함수가 호출되기 전, 맵 클래스의 거리를 구하는 멤버함수를 통해 난이도를 구하고, 이에 맞는 몬스터 인스턴스를 생성한 후 함수에 넣어준다. 3번째 논리형 매개변수는 현재 전투하는 몬스터가 보스인지 아닌지 확인한다.
* shopping(Shop&, Character&)  
  캐릭터 인스턴스가 상점 인스턴스에 방문하여 아이템을 구매할 수 있다. 다음 5가지 상품을 구매할 수 있다. 상품을 하나 구매한 이후 다시 상점에서 나가진다  
  1. 공격력 증가, 2. 방어력 증가, 3. 경험치 증가, 4. 체력 회복, 5. 마력 회복
* random\_event(Character&)  
  매개변수로 전달받은 캐릭터 인스턴스에게 다음 4가지 이벤트 중 하나를 일으킨다.   
  1. 현재 체력의 30%를 회복한다.  
  2. 현재 체력의 30%를 잃는다.  
  3. 100 골드를 획득한다.  
  4. 아무 일도 발생하지 않는다.

다음 함수는 선택지에서 입력한 선택이 올바른지 확인하는 함수이다.

* template <typename T> bool check\_in(T, T\*, int)  
  함수 템플릿으로 구현되며, 첫 번째 매개변수 T가, 두 번째 포인터 매개변수에서 정수형 매개변수 크기로 구성된 배열에 존재하는지 확인하는 함수이다. 문제 사항에서 제시되지 않은 타입에 대한 입력은 고려하지 않는다.

다음은 클래스에 대한 설명이다. 클래스의 변수는 모두 프라이빗 구역에 정의되어 있어 온점 기호의 접근자를 통한 접근을 제한하며, 생성자를 포함한 멤버 함수들은 퍼블릭 구역에 정의된다.

캐릭터 클래스(및 헤더파일)은 지도 클래스와 몬스터 클래스에 종속성을 갖는다. 캐릭터 클래스는 공격력, 방어력, 골드, 레벨, 최대 체력, 현재 체력, 최대 마력, 현재 마력, 최대 경험치, 경험치, 지도에서의 좌표 x, y를 나타내는 정수형 변수를 갖는다. 아래는 캐릭터 클래스가 갖는 멤버함수에 대한 설명이다.

* Character(Map&)  
  생성자. 지도에서 시작 위치를 가져온 후, 플레이어블 캐릭터의 스테이터스를 초기화한다.
* int get\_A()   
  A 대신 사용되는 변수의 목록은 다음과 같다.  
  x, y, exp, gold, atk, def, hp, mp, level, max\_hp, max\_mp  
  해당되는 변수의 값을 가져오기 위해 사용된다.
* void change\_A(int)  
  A 대신 사용되는 변수의 목록은 다음과 같다. atk, def, exp, hp, gold  
  해당되는 변수의 값을 변경하기 위해 사용된다. 호출시 전달되는 정수형 매개변수는 변화량, 델타를 뜻하며 이는 음수의 값을 가질 수 있다.
* bool go\_DIRECTION()  
  DIRECTION 대신 사용되는 문자열은 다음과 같다. up, down, left, right  
  제시된 방향으로 이동한다. 만약 이동이 가능하다면 참을 반환하고, 지도 밖으로 나가는 등의 이유로 이동하지 못했으면 거짓을 반환한다.
* int attack\_TYPE(Monster&)  
  TYPE 대신 사용되는 문자열은 다음과 같다. basic, skill, fire, grass, water  
  주어진 공격 방식에 따라 공격력의 계수를 지정하고, 매개변수로 받은 몬스터의 방어력을 가져와 데미지를 입힌 후 가한 데미지를 반환한다. 방어력이 공격력과 계수의 곱 보다 높다면 데미지는 0이며, 공격에 필요한 마력이 없을 경우 -1을 리턴한다.
* void level\_up()  
  레벨을 증가시키고, 그에 따른 체력회복, 마력회복, 스탯을 증가시키도록 하는 함수이다.
* void recovery\_A()  
  A 대신 사용되는 변수는 다음과 같다. hp, mp  
  주어진 값을 최대로 회복한다.
* bool is\_dead()  
  캐릭터가 죽어있다면(체력이 0 이하라면) 참, 아니라면 거짓을 반환한다.

몬스터 클래스는 다른 헤더파일을 필요로 하지 않는다(종속성이 없다). 몬스터 클래스는 현재 체력, 공격력, 방어력, 타입(속성), 보상 경험치, 보상 골드를 나타내는 정수형 변수를 갖는다. 아래는 몬스터 클래스가 갖는 멤버함수에 대한 설명이다.

* Monster(int)  
  생성자. 정수형 매개변수로 난이도를 받아 그에 해당하는 몬스터 인스턴스를 생성한다. 만약 음수 난이도를 받는다면 보스를 생성한다.
* int get\_A()  
  A 대신 사용되는 변수의 목록은 다음과 같다.   
  atk, def, type, hp, reward\_gold, reward\_exp  
  프라이빗 구역에 정의된 변수의 값을 가져오기 위해 사용된다.
* void change\_hp(int)  
  정수형 매개변수는 변화량, 델타이며 그에 해당하는 값 만큼 체력을 변화시킨다.
* bool is\_dead()  
  몬스터가 죽어있다면 참, 아니라면 거짓을 반환한다.

지도 클래스 또한 종속성이 없다. 지도 클래스는 시작지점의 좌표를 나타내는 정수형 변수 두개를 가지며, 지도를 저장하는 문자형 이차원 배열, 방문 여부를 나타내는 논리형 이차원 배열을 갖는다. 아래는 지도 클래스가 갖는 멤버함수에 대한 설명이다.

* Map()  
  생성자. 지도를 초기화한다.
* int distance(int, int)  
  매개변수로 주어진 좌표가 지도의 시작지점과 얼마나 떨어져 있는지, 맨허튼 거리를 구해 반환한다.
* int get\_start\_A():  
  A 대신 사용되는 변수의 목록은 다음과 같다. x, y  
  프라이빗 구역에 정의된 변수에 접근하기 위해 사용된다.
* void veil(int, int)  
  매개변수로 주어진 좌표 구역의 방문여부를 참으로 변경한다.
* bool isnot\_veil(int, int)  
  매개변수로 주어진 좌표에 방문한 적이 없다면 참, 있다면 거짓을 반환한다.
* char get\_map(int, int)  
  매개변수로 주어진 좌표에 해당하는 구역의 종류를 표현하는 문자를 반환한다.  
  E : empty, M : monster, R : random, S : shop, B : boss

상점 클래스는 캐릭터 헤더를 필요로 한다. 상점 클래스는 각 상품의 가격과 재고를 프라이빗 구역에 변수로 갖는다. 다음은 상점 클래스의 멤버 함수에 대한 설명이다.

* Shop()  
  생성자. 상품의 가격과 재고를 초기화하여 상점 인스턴스를 생성한다.
* int get\_A(int)  
  A대신 사용되는 문자열은 다음과 같다. price, stock  
  프라이빗 구역에 정의된 인스턴스의 변수를 얻어오기 위해 사용되는 함수이다. 상점에는 5가지 상품이 있고, 원하는 값을 얻어오기 위해 상품 번호를 정수형 매개변수로 받는다.
* void atk\_up(Character&), void def\_up(Character&), void\_exp\_up(Character&)  
  매개변수로 전달받은 캐릭터 인스턴스에 상품 효과를 적용한다. 각각 공격력 10, 방어력 5, 경험치 300이 증가한다. 가격은 모두 700원이다.
* void recovery\_hp(Character&), void recovery\_mp(Character&)  
  매개변수로 전달받은 캐릭터 인스턴스에 상품 효과를 적용한다. 각각 체력과 마력을 최대로 회복하며, 가격은 모두 500원이다. 상품 효과를 적용할 때 체력이나 마력의 변화가 없더라도 화살표 기호를 이용해 변화를 출력한다.