МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе № 2

по дисциплине «Объектно ориентированное программирование»

Тема: Интерфейсы, полиморфизм

Студент гр. 0383 _____ Коротков А.В.

Преподаватели _____ Жангиров Т.Р.

Санкт-Петербург
2021

Цель работы:

Могут быть три типа элементов располагающихся на клетках:

1. Игрок - объект, которым непосредственно происходит управление. На поле может быть только один игрок. Игрок может взаимодействовать с врагом (сражение) и вещами (подобрать).
2. Враг - объект, который самостоятельно перемещается по полю. На поле врагов может быть больше одного. Враг может взаимодействовать с игроком (сражение).
3. Вещь - объект, который просто располагается на поле и не перемещается. Вещей на поле может быть больше одной.

Требования:

- Реализовать класс игрока. Игрок должен обладать собственными характеристиками, которые могут изменяться в ходе игры. У игрока должна быть прописана логика сражения и подбора вещей. Должно быть реализовано взаимодействие с клеткой выхода.
- Реализовать три разных типа врагов. Враги должны обладать собственными характеристиками (например, количество жизней, значение атаки и защиты, и.т.д. Желательно, чтобы у врагов были разные наборы характеристик). Реализовать логику перемещения для каждого типа врага. В случае смерти врага он должен исчезнуть с поля. Все враги должны быть объединены своим собственным интерфейсом.
- Реализовать три разных типа вещей. Каждая вещь должна обладать собственным взаимодействием на ход игры при подборе. *(например, лечение игрока).* При подборе, вещь должна исчезнуть с поля. Все вещи должны быть объединены своим собственным интерфейсом.
- Должен соблюдаться принцип полиморфизма

Потенциальные паттерны проектирования, которые можно использовать:

- *Шаблонный метод (Template Method) - определение шаблона поведения врагов*
- *Стратегия (Strategy) - динамическое изменение поведения врагов*
- *Легковес (Flyweight) - вынесение общих характеристик врагов и/или для оптимизации*
- *Абстрактная Фабрика/Фабричный Метод (Abstract Factory/Factory Method) - создание врагов/вещей разного типа в runtime*
- *Прототип (Prototype) - создание врагов/вещей на основе "заготовок"*

Выполнение работы.

С помощью интерфейсов и абстрактных классов была сформирована архитектура, позволяющая взаимодействовать следующим классам между собой.

Был реализован класс игрока (*Hero*) с собственными характеристиками: броней, силой атаки и количеством жизней. В это классе реализована логика сражения и подбора вещей для увеличения характеристик.

Кроме игрока были реализованы классы врагов (*Slime*, *Goblin* и *Troll*) с разными характеристиками: силой атаки и количеством жизней. Для врагов была прописана логика перемещения по полю и сражения с героем.

Также был реализован класс предметов (*Item*), который в нескольких экземплярах неподвижно располагаются случайным образом на клетках и увеличивают характеристики героя после взаимодействия с последним, например: здоровье (не превышая максимального значения), урон и броню, позволяющую игнорировать часть получаемого героем урона.

Также для тестирования взаимодействия был реализован простейший внутриигровой GUI для отображения характеристик персонажа.

Реализованные классы обеспечивают полиморфизм.

UML диаграмму см. в приложении А.

Выводы.

В ходе выполнения лабораторной работы была написана программа, реализующая классы игрока, врагов и вещей, их взаимодействие а также управление персонажем и перемещение врагов в простейшей форме.

Приложение А

UML-диаграмма