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: _____ Самара Р.Д.

Преподаватель: _____ Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы.

Создать классы игрока, предметов, врагов на игровом поле.

Задание.

Могут быть три типа элементов располагающихся на клетках:

Игрок - объект, которым непосредственно происходит управление. На поле может быть только один игрок. Игрок может взаимодействовать с врагом (сражение) и вещами (подобрать).

Враг - объект, который самостоятельно перемещается по полю. На поле врагов может быть больше одного. Враг может взаимодействовать с игроком (сражение).

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

Выполнение работы.

Порядок выполнения поставленной задачи программой:

1. Был создан класс GridEntity, от которого наследуются классы Creature — игрок и враги, а также класс Item — предметы. Все наследуемые классы будут обладать общими методами (уничтожить, переместиться, получить информацию и т. д.)
2. Вещи наследуются от класса Item с виртуальным методом effect. Среди них есть Medkit (частично восстанавливает здоровье), Power (увеличивает силу атаки персонажа), Gaze (увеличивает дальность атаки персонажа) и Thorn (шипы, отнимающие здоровье если на них наступить).
3. Был создан класс Creature, который имеет общие для всех существ (игрока, врагов) методы, такие как получить урон, восстановить здоровье. При уровне здоровья меньше либо равным 0 существо уничтожается методом destroy().
4. Был создан класс Player — игрок, который определяет метод move из интерфейса IMove, отвечающего за передвижение всех существ на поле. Игрок может дойти до клетки-выхода, тогда переменная levelComplete станет равной true. При попытке сдвинуть игрока на

пределенную клетку происходит проверка, что находится на этой клетке, и не выходит ли она за границы. Игрок не двигается, если на клетке что-то находится, или эта клетка край поля.

5 Интерфейс IEnemy содержит виртуальные методы isPlayerReachable(находится ли игрок в зоне видимости врага), attack (атака персонажа) и nextAction (расчет дальнейших действий врага). Все враги наследуются от IEnemy, к ним относятся Archer (лучник, не двигающийся на поле, стреляющий на дальнюю дистанцию), Knight (рыцарь, ходящий на 3 клетки туда и обратно) и Wolf (волк, ходящий в виде знака бесконечности и передвигающийся сразу на 2 клетки за 1 ход).

Диаграмму с зависимостями классов см. в приложении А

Выводы.

В ходе работы было изучены способы создания собственных классов, их конструкторов и деструкторов, полей и методов.

Разработана программа по типу мини игры, генерирующая поле с непроходимыми клетками и стартом/финишем.

ПРИЛОЖЕНИЕ А

UML-ДИАГРАММА КЛАССОВ