

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

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Объектно ориентированное программирование »
Тема: Логическое разделение классов.

Студент гр.8382

Синельников М.Р

Преподаватель

Жангиров Т.Р

Санкт-Петербург

2020

Цель работы.

Разработать и реализовать набор классов для взаимодействия пользователя с юнитами и базой.

Ход выполнения.

Для управления игрой был реализован класс *Game*. Определение содержится в файле *Game.h*, описание в файле *Game.cpp*. При запуске игры класс *Game* вызывает метод класса *BaseAction* для размещения баз. Определение класса *BaseAction* содержится в файле *BaseAction.h*, описание в файле *BaseAction.cpp*. После установления баз запускается цикл *while(1)*, каждая итерация которого обозначает очередной ход в игре. При очередном ходе пользователь имеет право

- завершить игру
- пропустить ход
- выбрать юнита
- выбрать базу

При выборе юнита запускается метод класса *UnitAction*, который предоставляет пользователю всевозможные действия юнита. При выборе конкретного действия запускается соответствующий метод класса *Interaction*, который и реализует основной функционал. Определение классов в файлах *UnitAction.h* и *Interaction.h*, описание в файлах *UnitAction.cpp* и *Interaction.cpp*.

При выборе базы запускается метод класса *BaseAction*, который предоставляет пользователю всевозможные действия с базой. При выборе конкретного действия запускается соответствующий метод класса *Interaction*, который и реализует основной функционал.

Класс *UnitAction* позволяет просмотреть характеристики юнита, передвинуть юнита, атаковать другого юнита, если он находится в одной из соседних клеток, атаковать чужую базу, воспользоваться услугами нейтрального объекта, если он находится с юнитом на одной клетке или же починить свою базу, если юнит находится с базой на одной клетке.

Взаимодействие всех объектов реализовано с помощью класса *Interaction*.

Класс *BaseAction* позволяет просмотреть характеристики базы или создать юнита.

Из дополнительных требований реализован паттерн Фасад в виде класса *Game* и паттерн Медиатор в виде класса *Interaction*.

Выводы.

В ходе выполнения работы были разработаны классы для взаимодействия пользователя с юнитами и базой.