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

ОТЧЕТ

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

Студент гр. 8381 _____ Почаев Н.А.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург
2020

Цель работы.

Разработать и реализовать набора классов для взаимодействия пользователя с юнитами и базой. Основные требования:

- Должен быть реализован функционал управления юнитами
- Должен быть реализован функционал управления базой

Задание.

- Выполнены все основные требования к взаимодействию
- Добавлен функционал просмотра состояния базы
- Имеется 3+ демонстрационных примера
- Реализован паттерн “Фасад” через который пользователь управляет программой
- Объекты между собой взаимодействуют через паттерн “Посредника”
- Для передачи команд используется паттерн “Команда”
- Для приема команд от пользователя используется паттерн “Цепочка обязанностей”

Выполнение работы.

Написание работы производилось на базе операционной системы Windows 10 в среде разработки Qt Creator, для компиляции и отладки использовалась UNIX-подобная среда Cygwi и набор адаптированных инструментов MiniGW. Были задействованы пакеты GCC, CMake, а также GDB.

Реализованные классы

Классы, добавленные в программу в данной лабораторной работе и их функционал представлены в табл. 1. В ней приведено общее описание классов, отдельные моменты пояснены в комментариях к коду.

Таблица 1 – Основные добавленные классы

Класс	**Назначение**
Game	Класс реализующий паттерн “Фасад”. Скрывает в себе сложное взаимодействие классов игры, оставляя функции main лишь возможность запуска игры (и соответствующего потока выполнения GUI), а пользователю – взаимодействие с интерфейсом.
MainWindow	Класс главного окна программы – стартовое меню. Даёт возможность выбора количества игроков, а также запустить окно с основной игрой.
GameWindow	Класс окна игрового процесса.
UnitAttackMediator	Класс реализует паттерн “Посредник” между юнитами при их взаимодействии – атаке. Когда один из юнитов хочет атаковать другой, он отправляет соответствующий сигнал, содержащий возможный наносимый урон. Пройдя через проверки прокси поля на возможность проведения данного действия, другой юнит получает и сигнал и урон.

Выводы.

В ходе выполнения лабораторной работы были написаны требуемые класс, а также реализовано требуемое взаимодействие между функциональными классами игры и пользователем.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ. MAIN.CPP

```
#include <iostream>

#include <QApplication>
#include <QGridLayout>
#include <QWidget>
#include <QLabel>
#include <QScreen>

#include "Tests/examples.h"
#include "Game/game.h"

int main(int argc, char *argv[])
{
    // fieldBasedTest();
    // ObserverDeathTest();
    // landscapeTest();
    unitInteractionTest();

    Game *game = new Game(argc, argv);
    game->start();

    return 0;
}
```