

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

Преподаватель

Сосновский Д.Н.

Жангиров Т. Р.

Санкт-Петербург

2020

Задание.

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

Выполнение работы

Для выполнения лабораторной работы были созданы следующие классы:

Таблица 1 – Основные добавленные классы

Имя класса	Назначение
Handler, AbstractHandler, StartGameHandler, AttackHandler, AddBaseHandler, AddUnitHandler, MoveHandler	Классы, реализующие паттерн цепочка обязанностей . Получают на вход функции handle() данные и отправляют специальные команды в фасад .
Command, MoveCommand, AttackCommand, AddBaseCommand, AddUnitCommand, StartNewGameCommand	Классы, реализующие паттерн команда . Имеют функцию execute(), запускающую их исполнение. Вызывают необходимые функции фасада .
ClientInputData, AttackClientData, MoveClientData, StartNewGameData, AddUnitData, AddBaseData	Вспомогательные классы, которые используются в паттерне цепочка обязанностей . Хранят информацию о пользовательском вводе.
Game	Основной класс игры. Управляет игровыми составляющими.

GameFacade	Класс, реализующий паттерн Фасад . Сочетает в себе управление игровыми составляющими, отслеживанием игрового процесса и так же посылает команды в интерфейс IView для отображения.
IView	Абстрактный класс, описывающий вид для отображения информации от фасада.
FightMediator	Класс, реализующий паттерн Посредник . Используется для выполнения битвы между юнитами.
FieldWidget, FieldCellWidgetItem	Классы для отображения игры в виде GUI.
GameWindow	Окно, в котором есть кнопки управления, окна отображения поля и пр. Реализует интерфейс IView.

Создание UML-диграммы.

UML-диаграмма была создана при помощи сервиса Lucidchart. Диаграмма идёт в приложении к лабораторной работе.

Тестирование программы.

Для демонстрации работы программы были разработаны 3 примера.

Пример 1.

Создаётся поле размера 4x4, устанавливаются две базы. Далее каждый из игроков ставит по юниту на поле. Итог приведён на рисунке.

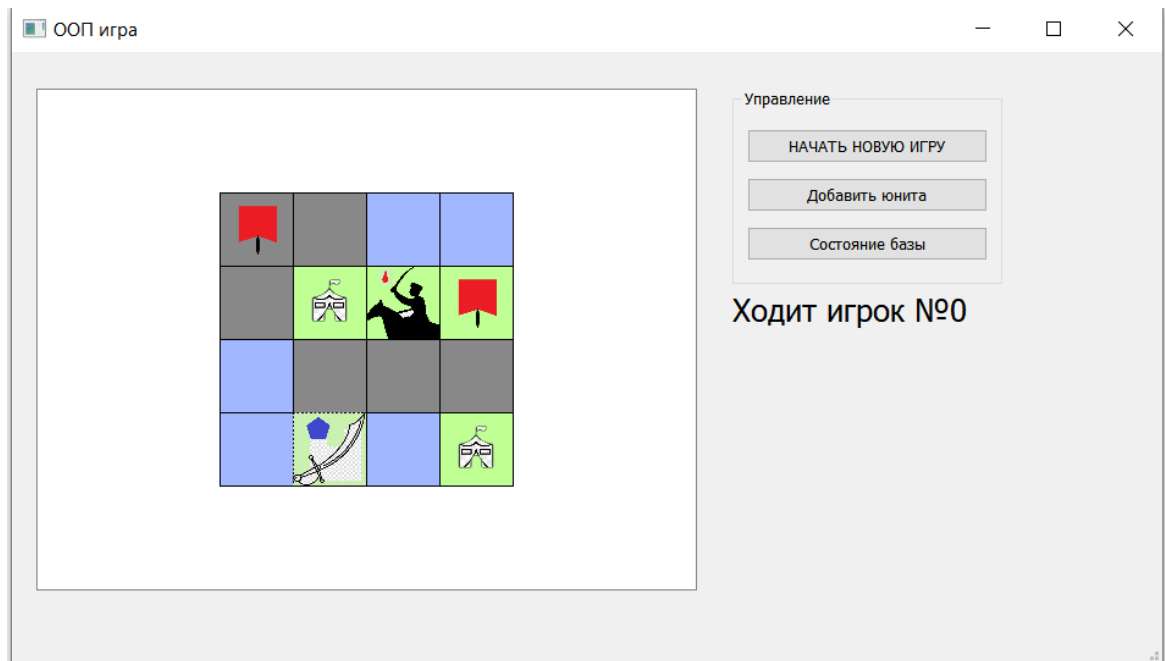


Рисунок 1 - пример 1

Пример 2.

Теперь говорим кавалерии перейти на одну клетку вправо.

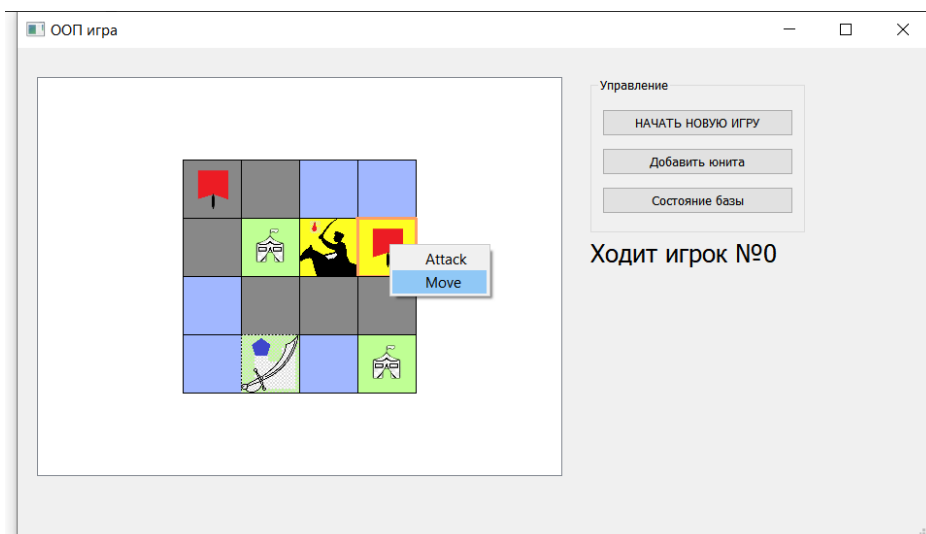


Рисунок 2 – движение

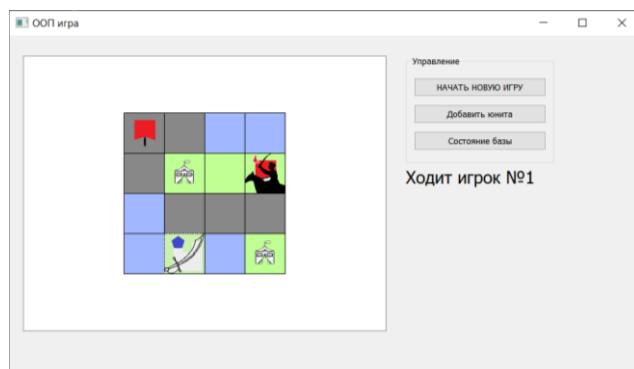


Рисунок 3 - после движения

Пример 3.

Теперь за несколько ходов добиваемся уничтожения базы игрока №0 путём атаки с помощью пехоты.

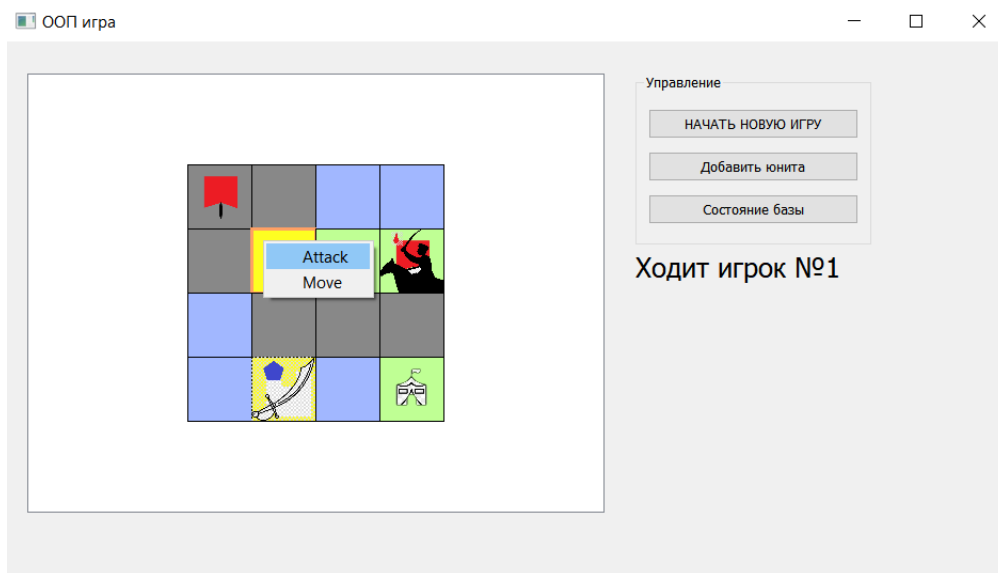


Рисунок 4 – атака



Рисунок 5 - игрок №0 уничтожен. Его юниты тоже удалены.

Вывод.

В ходе выполнения лабораторной работы была написана программа, в которой реализованы классы для функционала программы и взаимодействия пользователя с программой. Был использован объектно-ориентированный стиль программирования, были изучены и применены его основные положения, а также реализованы некоторые паттерны проектирования.