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

ОТЧЕТ

по лабораторной работе № 5

по дисциплине «Объектно-ориентированное программирование»

Тема: Управление, разделение на уровни абстракции.

Студент гр. 0303 _____ Болкунов В.О.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург 2021

Цель работы.

Организовать управление игрой.

Задание.

Необходимо организовать управление игрой (номинально через CLI). При управлении игрой с клавиатуры должна считываться нажатая клавиша, после чего происходит перемещение игрок или его взаимодействия с другими элементами поля.

Требования:

- Реализовать управление игрой. Считывание нажатий клавиш не должно происходить в классе игры, а должно происходить в отдельном наборе классов.
- Клавиши управления не должны жестко определяться в коде. Например, это можно определить в отдельном классе.
- Классы управления игрой не должны напрямую взаимодействовать с элементами игры (поле, клетки, элементы на клетках)
- Игру можно запустить и пройти.

Основные теоретические положения.

GUI (Graphical User Interface) или ГИП (графический интерфейс пользователя) — это одна из разновидностей пользовательских интерфейсов, элементы которого выполнены в виде графических изображений. То есть все основные объекты, присутствующие в этом интерфейсе — иконки, функциональные кнопки, объекты меню и т.д. — выполнены в виде изображений.

Qt — фреймворк для разработки кроссплатформенного программного обеспечения на языке программирования C++.

Выполнение работы.

Класс Assets — синглтон, представляет собой структуру для хранения игровых констант:

- названия файлов игровых ассетов, является объектом внутренней закрытой структуры NAMES.
- спрайты (загружаются при создании синглтона), является объектом внутренней закрытой структуры IMAGES
- клавиши управления, является объектом внутренней закрытой структуры KEYS
- размеры спрайтов и частота обновления игры

Класс ObjectView — базовый класс для отображения объекта в графической сцене QT, наследуется от QGraphicsItem.

Класс CellView — отображение клетки поля, получает нужный спрайт из словаря с помощью переданной ссылки на AbstractCell.

Класс EntityView — отображение объектов на поле, получает нужный спрайт из словаря с помощью переданного укзателя на Entity.

Класс Window — класс графического интерфейса игры, наследуется от QmainWindow.

- Содержит указатель на объект игры (передаётся в конструкторе).
- Содержит объекты и QGraphicsItem QraphicsView, с помощью которых происходит отрисовка поля.

Нажатие клавиш обрабатывается переопределённым методом окна keyPressedEvent. Таймер QTimer раз в заданное в Assets время вызывает метод для хода ботов. Для синхронизации команд управления и автоматической работы ботов используется QMutex. При смерти персонажа всплывает окно с информацией о проигрыше. Также при выполнении условий игры и прохождения к выходу всплывает окно с информацией о прохождении игры. В статус-баре окна выводится информация о параметрах игрока. Примеры поведения игры представлены на рисунках 1-3.

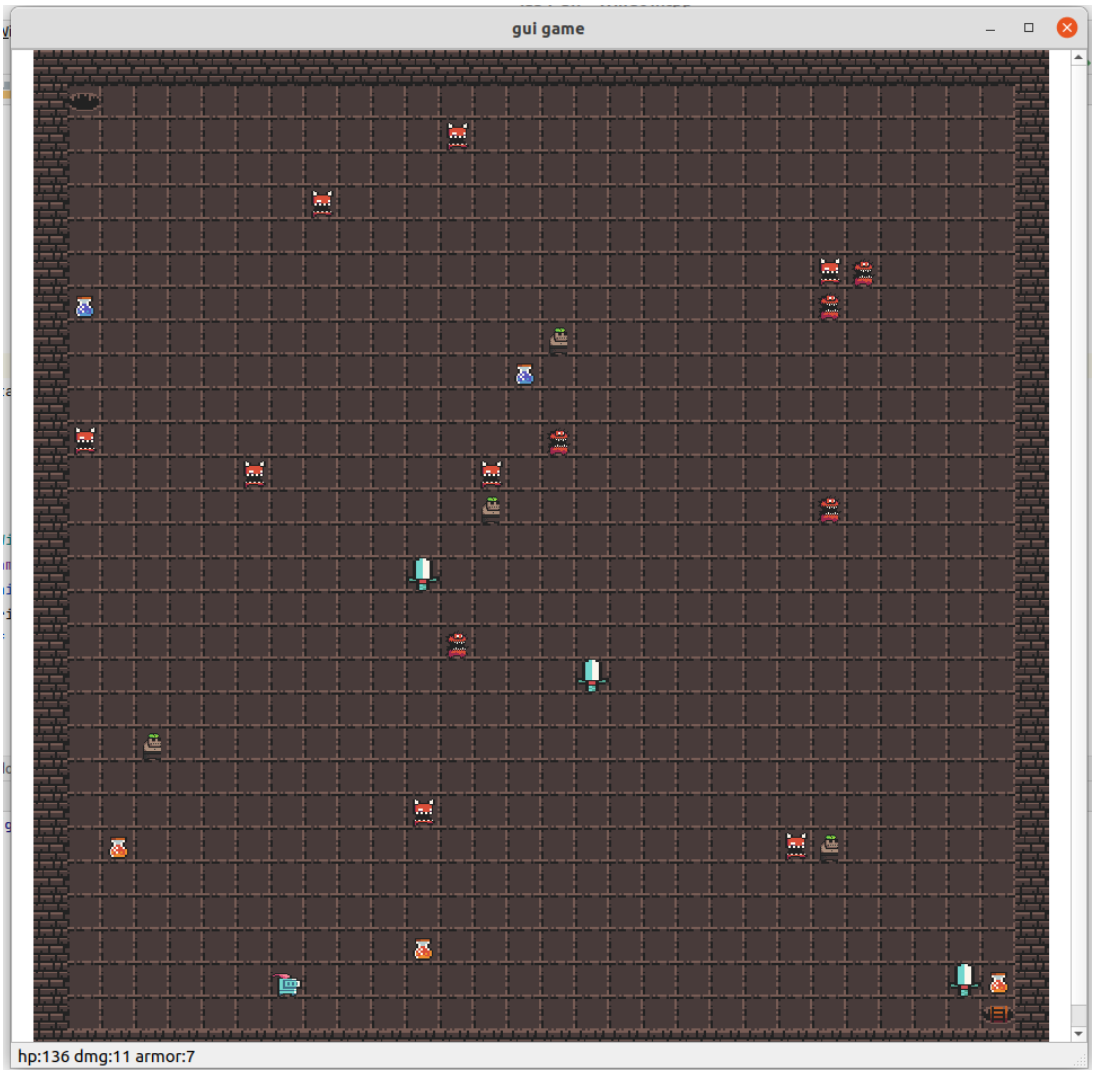

Рисунок 1: Пример игрового окна

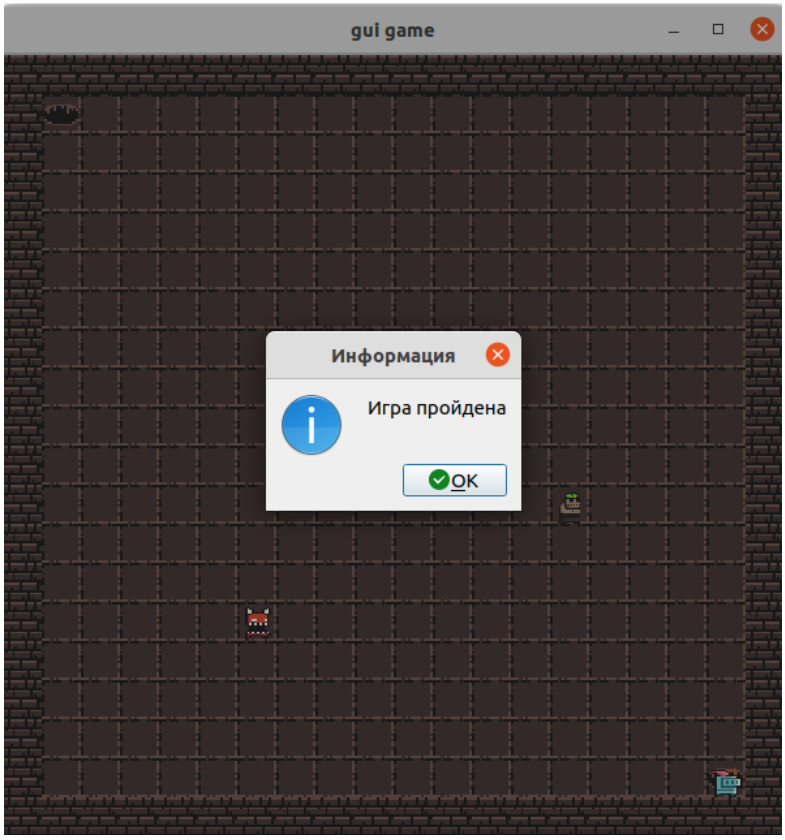

Рисунок 2: Игра пройдена

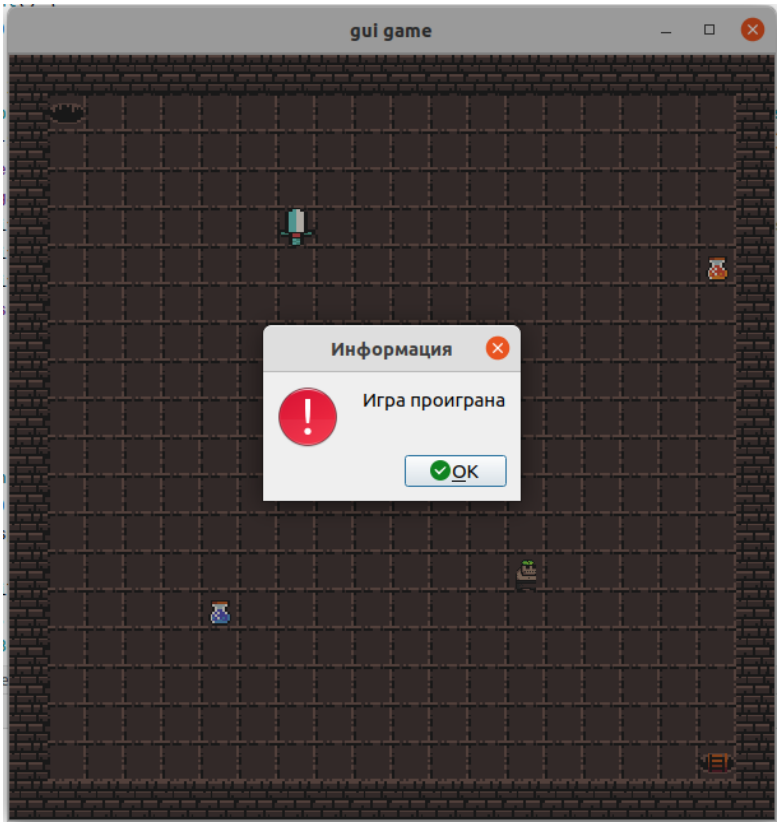

Рисунок 3: Игра проиграна

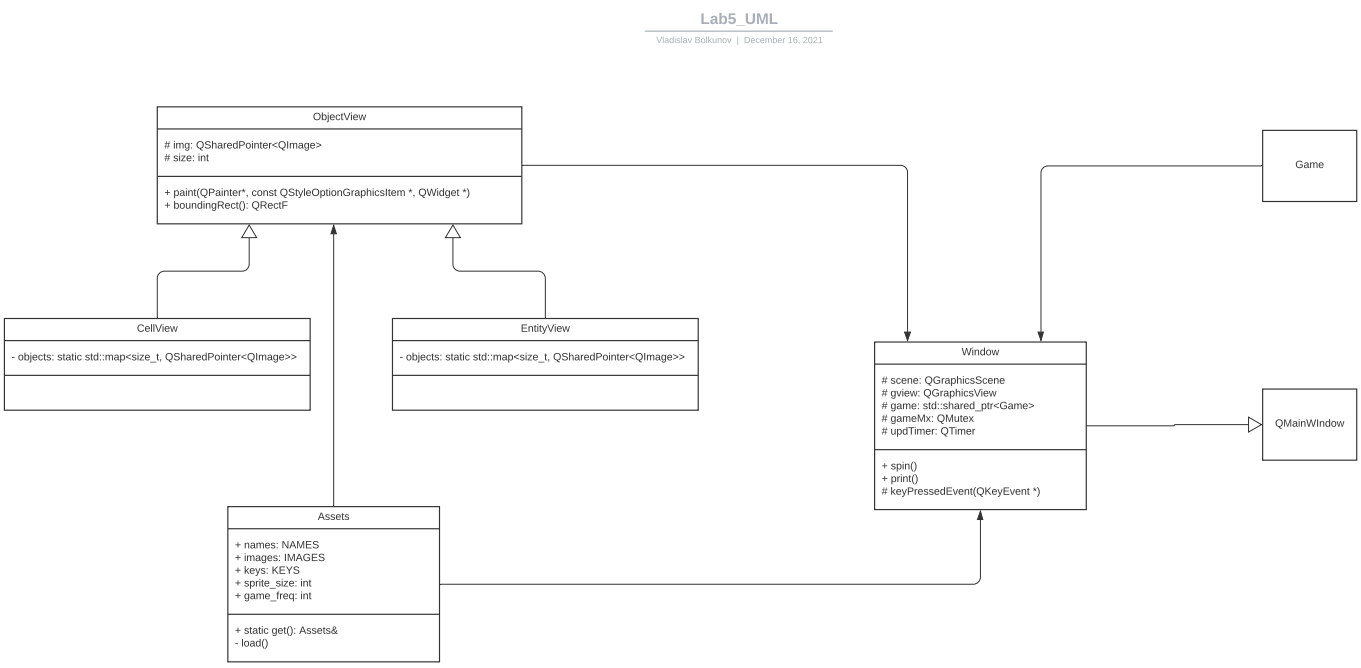

Рисунок 4: UML диаграмма

Разработанный программный код см. в директории */game_lib, /game и /gui*.

Разработанную диаграмму классов UML см. в Lab4_UML.pdf .

Тестирование.

Разработанные тесты см. в директории /tests.

Выводы.

Был реализован набор классов для непосредственного управления игроком. Создан графический интерфейс игры, которая обновляется в реальном времени.