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

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Объектно-ориентированное программирование»

Тема: Добавления логирования

Студент гр. 9381 _____ Камакин Д.В.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2020

Цель работы.

Изучить синтаксис языка программирования C++, ознакомиться с основными принципами объектно-ориентированного программирования. Реализовать классы для логирования.

Задание.

Создан набор классов, которые отслеживают игрока и элементы на поле, и выводят/сохраняют информацию об их изменениях.

Обязательные требования:

Реализована возможность записи логов в терминал и/или файл

Взаимодействие с файлом реализовано по идиоме RAII

Перегружен оператор вывода в поток для всех классов, которые должны быть логированы

Дополнительные требования:

Классы, которые отслеживают элементы, реализованы через паттерн Наблюдатель

Разделение интерфейса и реализации класса логирования через паттерн Мост

Выполнение работы.

Написание работы производилось на базе операционной системы Ubuntu 20.04 в среде разработки QtCreator с использованием фреймворка Qt.

Для игрового поля был реализован класс GameField. Написаны конструкторы копирования и перемещения, а также операторы присваивания и перемещения. Класс содержит двумерный массив клеток собственного класса GameCell, а также переменную и соответствующие функции для реализации паттерна «Одиночка».

Класс GameCell служит для реализации элемента на игровом поле. Каждая клетка содержит поля для хранения её типа, информацию о положении и занятости объектом. Реализованы функции для получения текущего состояния клетки.

Для создания GUI был использован фреймворк Qt, написан класс MainWindow, объект которого отображается в функции main(). В menuBar был добавлен пункт информации об авторе программы. Поле отображается при помощи QGraphicsView, сцена QGraphicsScene которого содержит клетки QGraphicsRectItem, внешний вид которых зависит от типа элемента на игровом поле GameField. Размер игрового поля 20X20.

Кроме того, для удобства работы с координатами был написан класс Point2D, в котором определены функции для получения и установки координат.

Обработка нажатий реализована в классе MainWindow. Поддерживается только **латинская** раскладка.

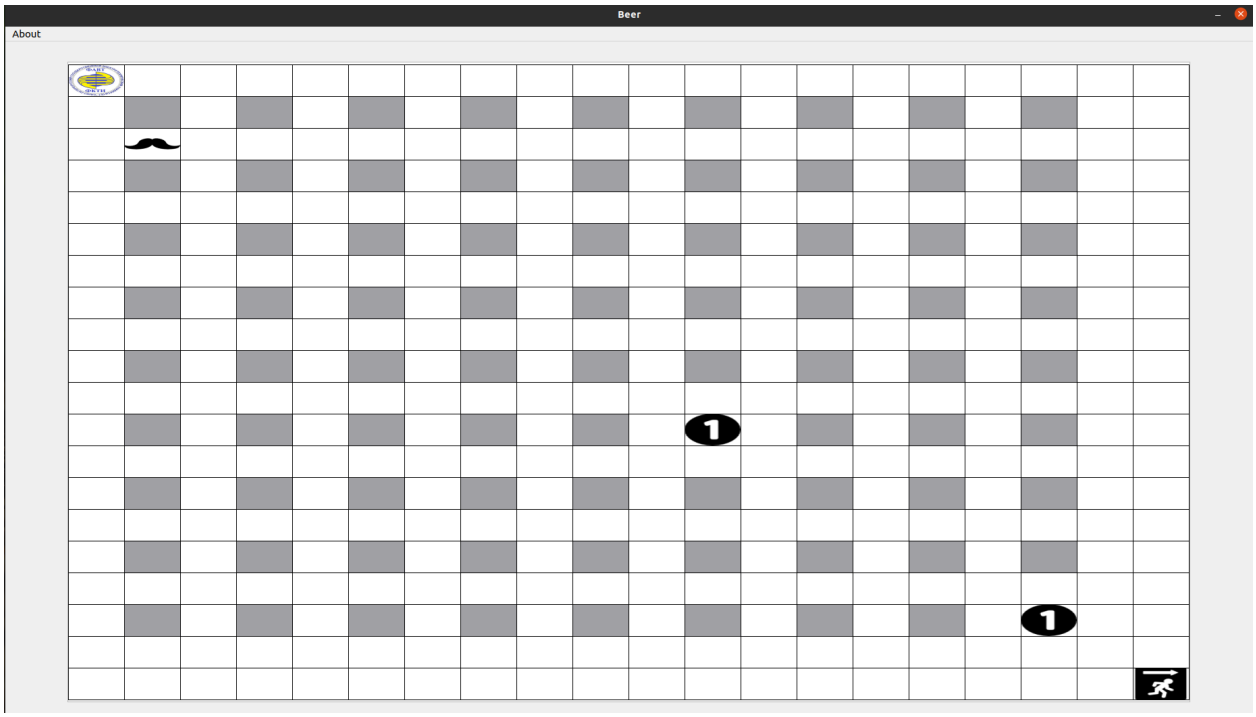
После получения команды от пользователя, направление движение отправляется в класс GameController, который перемещает игрока по полю, возвращая новую координату расположения персонажа.

Объекты на поле наследуются от интерфейса GameObject, в котором объявлена виртуальная функция action. Взаимодействие героя с объектами реализовано при помощи перегруженного оператора «+»

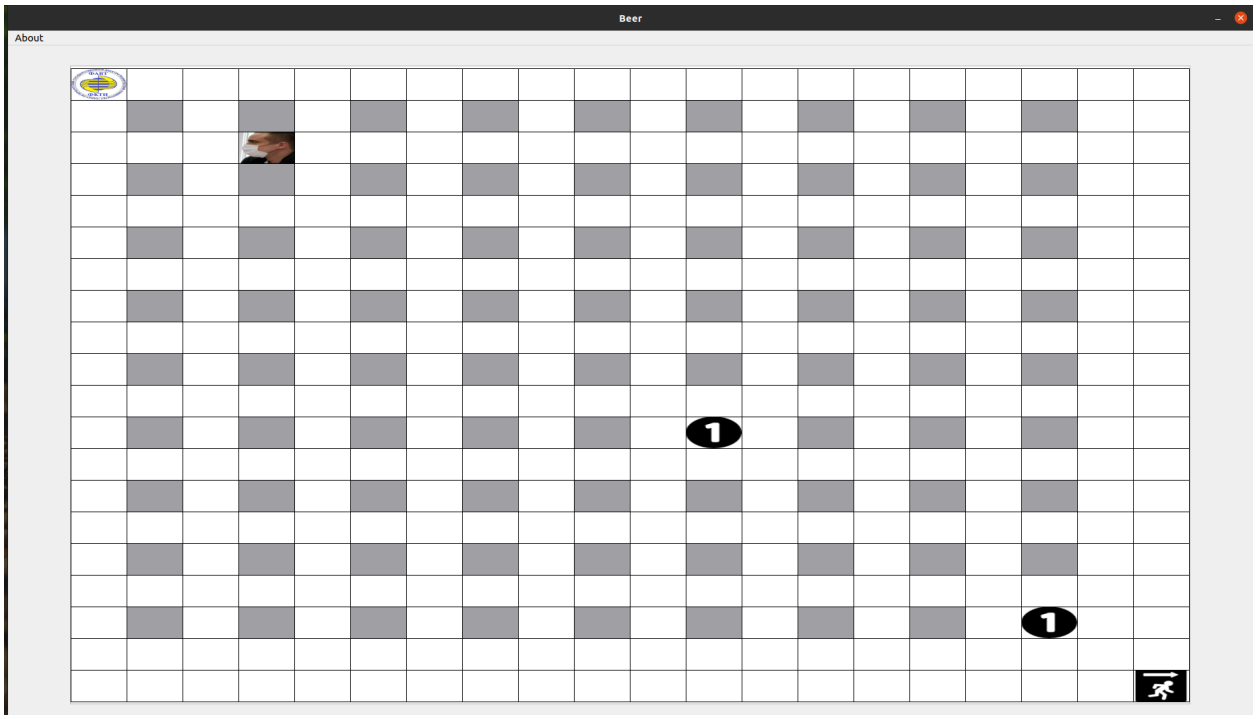
UML диаграмма представлена в файле uml.png.

Тестирование.

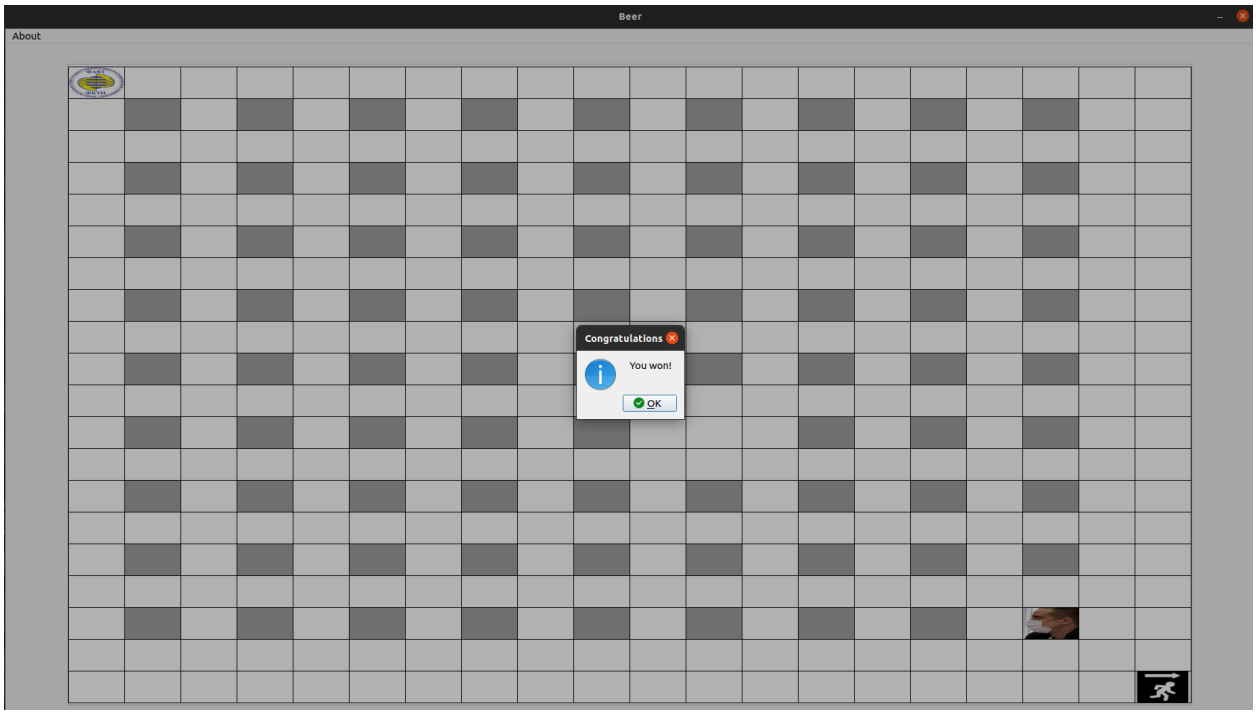
1. Программа успешно запускается, поле корректно отображается.



2. Персонаж перемещается по полю, взаимодействуя с объектами.



3. При сборе всех монет игра завершается



Выводы.

Изучен синтаксис языка программирования C++, реализованы все необходимые классы для логирования (в том числе один абстрактный и две его реализации). Вывод возможен на консоль и/или в файл.