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. 0383 _____ Коротков А.В.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург

2021

Цель работы.

Изучить материал по теме «Управление, разделение на уровни абстракции» и, на основе полученных знаний, реализовать управление игрой, обеспечив выполнение необходимых требований.

Задание.

Необходимо организовать управление игрой (номинально через CLI). При управлении игрой с клавиатуры должна считываться нажатая клавиша, после чего происходит перемещение игрока 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.

Потенциальные паттерны проектирования, которые можно использовать:

- Команда (*Command*) - передача команд с информацией через единый интерфейс. помещение команд в очередь.
- Посредник (*Mediator*) - организация взаимодействия различных модулей.

Выполнение работы.

Большая часть управления и логики игры, требуемая была реализована в предыдущих лабораторных работах, например:

- Базовое управление и перемещение персонажа
- Разделение управления от логики и отрисовки
- Возможность запустить и пройти игру с реализованным условиями прохождения игры

В ходе выполнения лабораторной работы был реализован новый класс *KeyAdapter,* а также изменён класс *Game*.

KeyAdapter – класс, который преобразует ивент нажатия на клавишу в элемент перечисления реакций на действия пользователя *EventReaction*. Класс хранит в себе коды клавиш, отвечающих за передвижение героя вверх, вниз, влево и вправо соответственно. Этот класс создан, чтобы предотвратить жесткое определение клавиш управления в коде, и иметь возможность быстро их заменить.

Данный класс имеет метод *setControlKeyBinding(sf::Keyboard::Key, sf::Keyboard::Key, sf::Keyboard::Key, sf::Keyboard::Key),* принимающий в качестве параметров четыре константы, которые выставляются в качестве клавиш управления героем.

Также класс имеет метод *processKeyCode(),* обрабатывающий код нажатой пользователем клавиши и возвращающей направление передвижение персонажа, которое впоследствии через класс *Game* передается в класс *HeroMover*, где непосредственно и происходит передвижение игрока на поле.

Класс *Game* был расширен новым полем для хранения объекта класса *KeyAdapter* и функцией *initControl()* для удобства инициализации и поддержания разделения конструктора на отдельные приватные функции. В функции *initControl()* задаются конкретные значения кодов клавиш управления персонажем.

Ранее были реализованы классы *HeroMover* и *EnemyMover*, позволяющие перемещать персонажа и врагов по полю. Взаимодействие данных классов и передача ввода пользователя в программу также происходит в классе *Game,* хранящую указатели на объекты данных классов.

В классе *Game* был добавлена функция *bool isMissionCompleted()* - функция для проверки выполнения игроком условий открытия возможности выхода с поля.

UML-диаграмму классов смотреть в приложении А.

Тестирование программы и процесса логирования смотреть в приложении В.

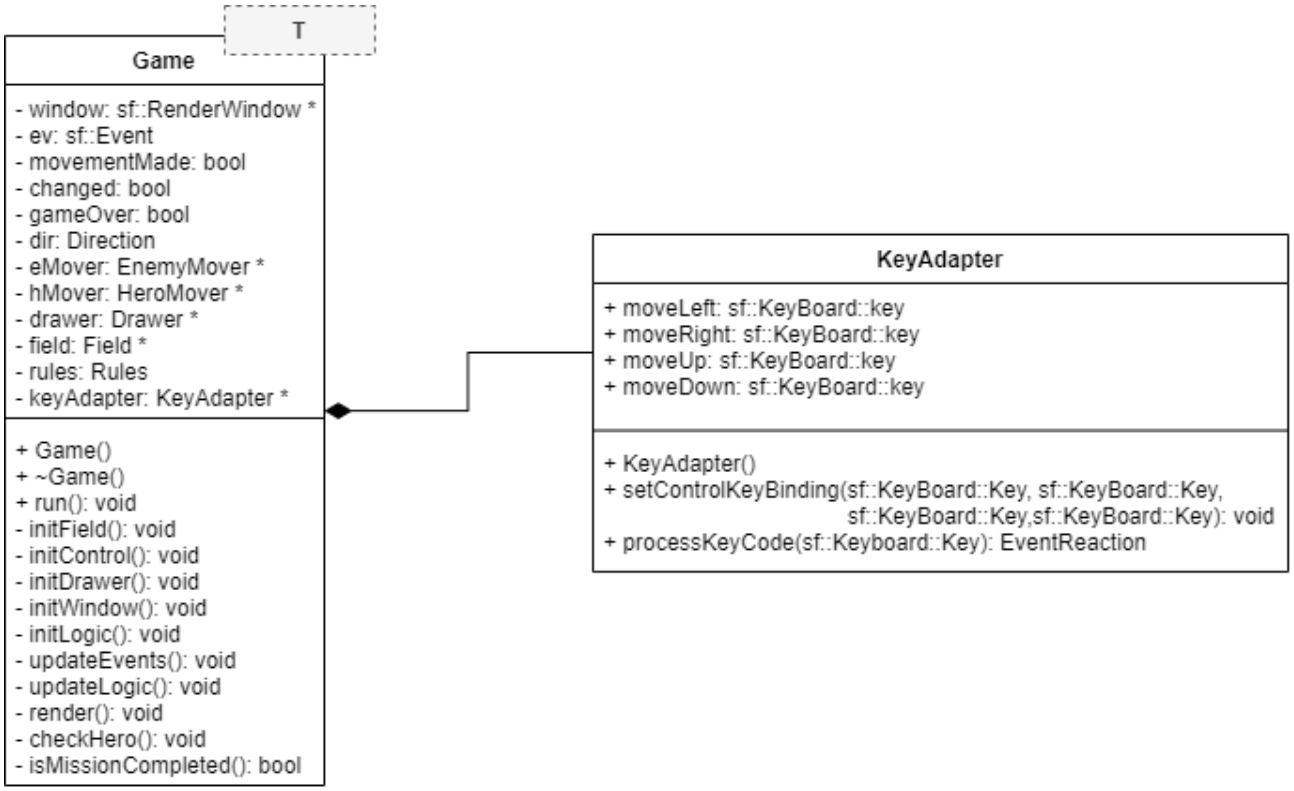
Выводы.

В ходе работы был изучен материал по теме «Управление, разделение на уровни абстракции». Был реализован класс и методы, позволяющие реализовать управление игрой, создана UML-диаграмма классов. Результатом выполнения лабораторной работы стало реализованное управление игрой, удовлетворяющее поставленным требованиям.

Приложение А.

UML-диаграмма.

UML-диаграмма классов



Приложение В.

Тестирование работы программы.

Пример запуска программы: