

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

ОТЧЕТ
по лабораторной работе № 4
по дисциплине «Объектно-Оrientированное Программирование»
Тема: Уровни абстракции, управление игроком

Студент гр. 1384

Алиев Д.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2022

Цель работы.

Разработать классы, позволяющие задать клавиши для управления. Разработать классы, обрабатывающие команды пользователя таким образом, чтобы без изменения существующего кода можно было внедрить иной способ ввода команд.

Задание.

Реализовать набор классов отвечающих за считывание команд пользователя, обрабатывающих их и изменяющих состояния программы (начать новую игру, завершить игру, сохраниться, управление игроком, и т.д.). Команды/клавиши, определяющие управление, должны считываться из файла.

Требования:

- Реализован класс/набор классов обрабатывающие команды
- Управление задается из файла
- Реализованные классы позволяют добавить новый способ ввода команд без изменения существующего кода (например, получать команды из файла или по сети)
- Из метода, считывающего команду, не должно быть “прямого” управления игроком

Примечания:

- Для реализации управления можно использовать цепочку обязанностей, команду, посредника, декоратор, мост, фасад

Выполнение работы.

Был создан класс конфигурации, хранящий символы, соответствующие той или иной пользовательской команде. Класс имеет два конструктора: по умолчанию и принимающий строку – название файла, из которого будет считана конфигурация управления. При вызове конструктора по умолчанию назначается стандартное управление (на кнопки WASD).

Был разработан класс интерфейс считывателя команд `EventReader`, имеющий чистую виртуальную функцию `readCommands`, которую нужно будет переопределить в классах наследниках. В методе `readCommands` в классах, наследующих интерфейс, должен вызываться метод `notify` хранящегося по указателю медиатора, передавая распознанную команду.

От абстрактного класса был унаследован класс `CommandReader`, который определяет команду, в зависимости от привязанной к ней клавише, и уведомляет `Mediator` о полученной команде, которая затем передаётся в `Controller`.

Диаграмма разработанных в ходе лабораторной работы классов:

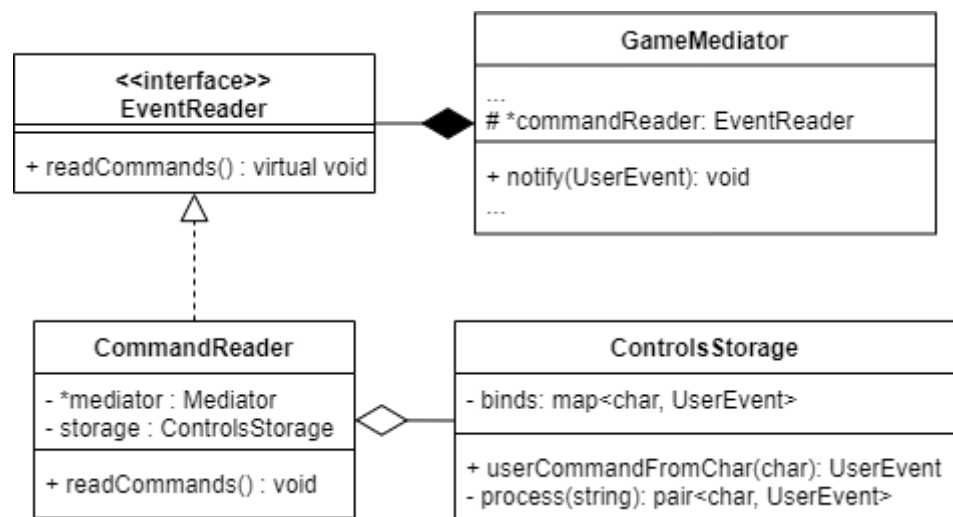


Рисунок 1 – Диаграмма классов

Тестирование.

В файле controls.txt были описана конфигурация управления:

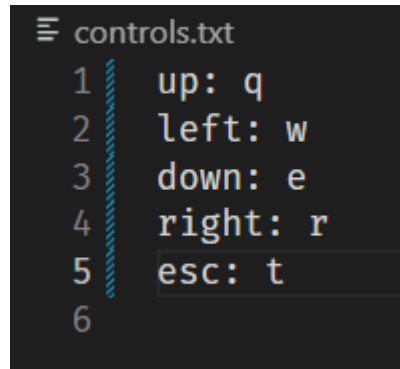


Рисунок 3 – Файл конфигурации управления

Управление в игре сменилось с WASD на QWER, выход осуществляется на кнопку T.

Выводы.

В ходе выполнения лабораторной работы был разработан шаблонный абстрактный класс считывателя пользовательских команд, от которого наследуются классы, выполняющие считывание команд. Классы, задействованные в обработке пользовательских команд разработаны таким образом, что можно внедрить новый способ ввода команд, не меняя существующий код – для этого достаточно унаследовать абстрактный класс считывателя команд.