

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Объектно-ориентированное программирование»
Тема: «Уровни абстракции, управление игроком»

Студент гр. 1384

Белокобыльский И. В.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2022

Цель работы.

Научиться реализовывать управление игроком при помощи абстрактных команд, не зависящих от конкретных клавиш посредством загрузки данных из файла конфигурации.

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 (определяет какая команда/нажатие клавиши отвечает за управление. Например, w - вверх, s - вниз, и.т.д)

Реализованные классы позволяют добавить новый способ ввода команд без изменения существующего кода (например, получать команды из файла или по сети). По умолчанию, управление из терминала или через GUI, другие способы реализовывать не надо, но должна быть такая возможность.

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.

Для реализации загрузки информации из файла был создан класс HandlerFileReader, реализующий интерфейс IHandlerReader. Он считывает файл config.txt и выдает на выходе словарь, у которого ключи – клавиши, а значения – команды из файла.

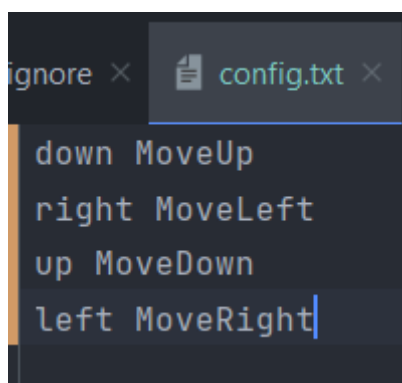
От этого класса зависит `CommandCreator`, который в свою очередь на основе выданного `IHandlerReader` словаря загружает необходимые команды в словарь с ключом клавишей и значением объектом команды.

Он вызывается классом `KeyboardHandler` на каждой итерации SFML-цикла, причем при помощи словаря клавиша из `enum` переводится к строке. Данный класс выданной для выданной команды вызывает метод `execute`.

Команда – объект класса, реализующего интерфейс `ICommand`. Всего реализовано 4 типа команд: `MoveUp`, `MoveLeft`, `MoveDown`, `MoveRight`. Все они обращаются к `GameMediator` для передвижения игрока.

Тестирование программы.

В качестве примера в файле `config.txt` были инвертированы команды перемещения.



Выводы.

В рамках данной лабораторной работы была доработана программа на языке C++ из предыдущей работы. Реализован механизм чтения команд передвижения игрока из файла.

ПРИЛОЖЕНИЕ 1

UML-диаграмма классов:

