

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

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

Студентка гр.1381

Рымарь М.И.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2022

Цель работы.

Реализовать набор классов, отвечающих за считывание команд, введённых пользователем, а также команд, обрабатывающих и изменяющих состояние программы.

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.

В ходе выполнения лабораторной работы созданы классы, отвечающие за считывание команд пользователя, обрабатывающих их и изменяющих состояния программы. Команды, определяющие управление, считываются из файла.

Новые классы

Абстрактный класс *interfaceConfig* хранит в себе два защищённых поля для того, чтобы наследуемые классы имели к ним доступ. Первое поле *config* нужно для того, чтобы пользователь самостоятельно задавал управление. Второе поле *defaultConfig* нужно в том случае, когда пользователь самостоятельно некорректно вводит управление или когда он хочет оставить стандартное управление в игре. В этом классе также есть методы *getDefault()*, *isCorrect()*, *getConfig()*.

Класс *fileConfig* наследуется от предыдущего, он хранит в себе файл. В конструкторе этого файла открывается на считывание файл *config.txt*. В деструкторе файл закрывается, выполняется идиома RAII. В этом классе переопределён метод *getConfig()*, сигнатура которого описана в прошлом классе. Сначала здесь проверяется, открыт ли файл. Если файл открыть не удалось, то возвращается стандартное управление. Далее построчно считывается файл, последний символ является клавишей, которая как-то управляет игрой. Потом до конца считывается файл, проверяется каждая строка. Если во время считывания не было найдено ошибок, то вызывается функция *isCorrect()*.

В классе *interfaceChReader* есть метод *getChar()*. Этот класс нужен для того, чтобы в последующем была возможность ввода команд без изменения существующего кода.

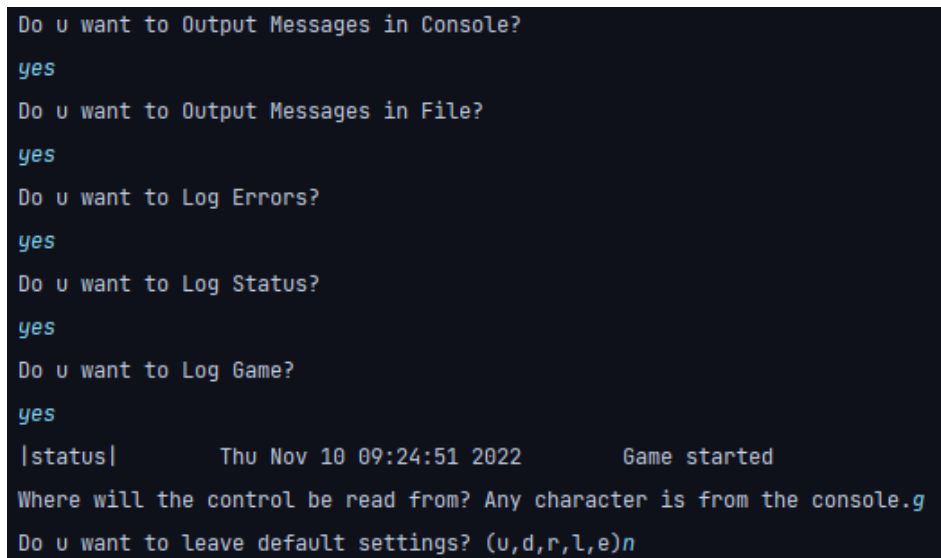
В классе *ChReader* реализован метод *getChar()*. Он считывает объект типа *char* и возвращает его.

Изменены классы *mediator* и *commandReader*. В первом добавлен метод *setConfig()*, который задаёт управление игрой по желанию пользователя. Во втором хранится *interfaceChReader* command*. В методе *getSrc()* под него выделяется память, в деструкторе память очищается. Переписан метод *readStep()*, в прошлой реализации он не принимал объект типа *map*. Сейчас это необходимо, так как заранее неизвестны символы, влияющие на перемещение игрока. Сначала этим методом в *config* считывается *char*. Далее этот символ

ищется в ключах словаря. Если он не найден, то шаг ставится с меткой STOP, которая останавливает движение игрока.

Тестирование.

Интерфейс командной строки при установке управления, заданного через файл, показан на рисунке 1.



```
Do u want to Output Messages in Console?
yes
Do u want to Output Messages in File?
yes
Do u want to Log Errors?
yes
Do u want to Log Status?
yes
Do u want to Log Game?
yes
|status|          Thu Nov 10 09:24:51 2022          Game started
Where will the control be read from? Any character is from the console.g
Do u want to leave default settings? (y,d,r,l,e)n
```

Рисунок 1 – Установка управления через файл

Новое управление, заданное через файл, показано на рисунке 2.

```
- - - - -
|p           w           * |
|x           |             |
|  x         |             |
|    x       |             |
|      x     |             |
|        x   |             |
|          x  |             |
|            x|             |
|              x t |
|k            |             |
- - - - -

Please input player's movement direction (l,u,r,d,e):d
|game|      Thu Nov 10 09:25:55 2022      Player moved to 1 0

- - - - -
|  p           w           * |
|x           |             |
|  x         |             |
|    x       |             |
|      x     |             |
|        x   |             |
|          x  |             |
|            x|             |
|              x t |
|k            |             |
- - - - -
```

Рисунок 2 – Новое управление

UML-диаграмма межклассовых отношений.

На рисунке 3 представлена UML-диаграмма межклассовых отношений.

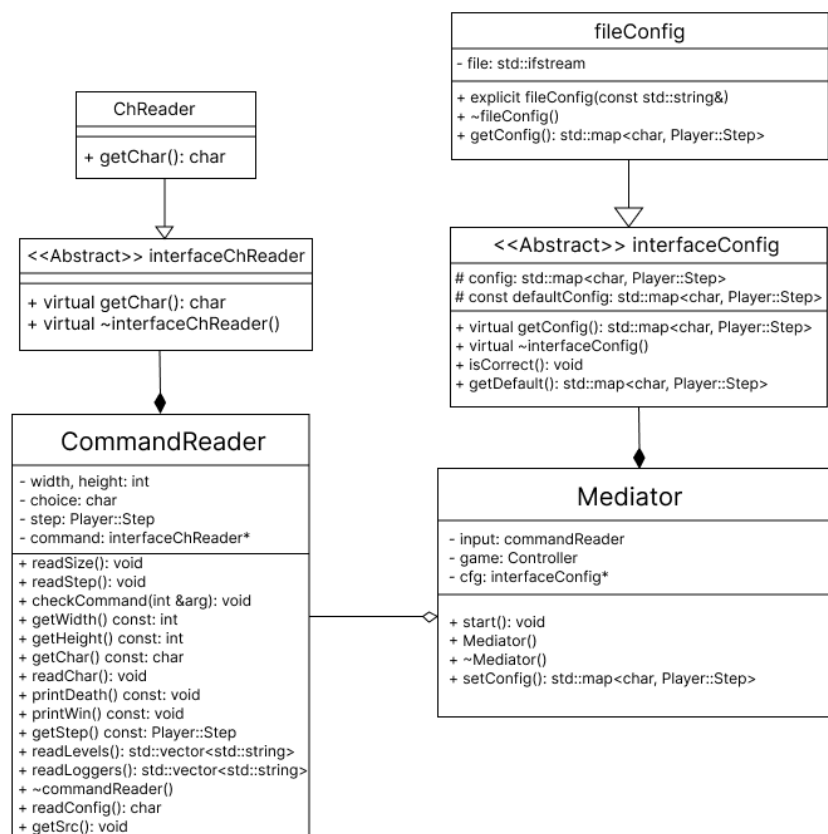


Рисунок 3 – UML-диаграмма

Выводы.

В ходе выполнения лабораторной работы реализован набор классов, отвечающих за считывание команд пользователя, обрабатывающих и изменяющих состояние программы.