

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

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

Студент гр. 1381 _____ Исайкин Г. И.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург

2022

Цель работы.

Реализовать настраиваемое управление в игре на C++.

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 выполнения работы сделаны/изменины классы CommandReader и CommandReaderKeyboard.

enum CommandOperand. Набор команд, который может ввести пользователь, а именно:

- движение (всего 8 направлений)

- остаться на месте (ничего не делает)
- запуск/перезапуск игры
- выход

Является дополненным и переименованным enum `Direction`.

Класс `CommandReader`. Абстрактный класс-интерфейс для получения команд. Объявляет виртуальные методы `CommandOperand get_command()`, которая получает команду, и `std::string get_symbol(CommandOperand co)`, который возвращает строку, указывающую название команды или кнопки, которой она соответствует.

Класс `CommandReaderKeyboard`. Класс-наследник `CommandReader`. Считывает при создании файл с настройками, где каждой команде присваивается символ. При вводе символа в терминал вызывается соответствующая ему команда. Делается это через переопределённый метод `CommandOperand get_command()`.

Другие изменения.

- Во всех необходимых классах `Direction` заменён на `CommandOperand`.
- Добавлены новые состояния в `GameStatus` (`interrupt` и `restart`). Метод класса `Observer` - `log_finish_game(GameStatus status)` также был дополнен в соответствии с добавленными состояниями.
- Изменён класс `FacadePlayer`.

UML-диаграмму межклассовых отношений см. в приложении А.

Выводы.

Реализована возможность изменить настройки управления игрой на языке C++.

ПРИЛОЖЕНИЕ А

UML-ДИАГРАММА МЕЖКЛАССОВЫХ ОТНОШЕНИЙ

