

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: Шаблонные классы

Студент гр. 3384

Поляков Г.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2024

Цель работы

Изучит работу с шаблонными классами в языке программирования C++. В соответствии с заданными условиями лабораторной работы создать собственные шаблонные классы, реализующие функционал игры «Морской бой», а именно шаблонный класс управления игрой, для которого в параметре шаблона указывается класс для обработки ввода, который также является шаблонным и принимает класс для обработки команды, шаблонный класс отображения игры, который занимается отрисовкой игрового поля.

Задание

Создать шаблонный класс управления игрой. Данный класс должен содержать ссылку на игру. В качестве параметра шаблона должен указываться класс, который определяет способ ввода команда, и переводящий введенную информацию в команду. Класс управления игрой, должен получать команду для выполнения, и вызывать соответствующий метод класса игры.

Создать шаблонный класс отображения игры. Данный класс реагирует на изменения в игре, и производит отрисовку игры. То, как происходит отрисовка игры определяется классом переданном в качестве параметра шаблона.

Реализовать класс считывающий ввод пользователя из терминала и преобразующий ввод в команду. Соответствие команды введенному символу должно задаваться из файла. Если невозможно считать из файла, то управление задается по умолчанию.

Реализовать класс, отвечающий за отрисовку поля.

Примечание:

Класс отслеживания и класс отрисовки рекомендуется делать отдельными сущностями. Таким образом, класс отслеживания инициализирует отрисовку, и при необходимости можно заменить отрисовку (например, на GUI) без изменения самого отслеживания

После считывания клавиши, считанный символ должен сразу обрабатываться, и далее работа должна проводить с сущностью, которая представляет команду.

Для представления команды можно разработать системы классов или использовать перечисление enum.

Хорошей практикой является создание “прослойки” между считыванием/обработкой команды и классом игры, которая сопоставляет

команду и вызываемым методом игры. Существуют альтернативные решения без явной “прослойки”

При считывания управления необходимо делать проверку, что на все команды назначена клавиша, что на одну клавишу не назначено две команды, что на одну команду не назначено две клавиши.

Выполнение работы

В процессе выполнения работы был создан шаблонный класс управления игрой (class GameHandler). Данный класс в качестве шаблона получает класс, определяющий способ обработки ввода(в данном случае будет использоваться class InputHandler). Перечислим методы класса GameHandler:

void startNewGame() — начинает новую игру, вызывает метод для расстановки кораблей на поле пользователя, учитывая координаты, которые вводит пользователь, а также метод для расстановки вражеских кораблей, после вызывается метод класса play().

void play() — реализует игровой цикл, в нём вызывается метод класса makeMove(), а после проверяется статус игры, если пользователь победил, то начинается новый раунд, его корабли остаются, а корабли врага расставляются снова, если же пользователь проиграл, тогда вызывается метод класса startNewGame() для начала новой игры.

void makeMove() — вызывает метод класса, переданного в шаблоне, для получения вводимой команды, если команда — выстрелить, тогда вызывается метод из класса игры с введёнными координатами, если же команда — применить способность, тогда вызывается соответствующий метод класса игры, аналогично при вводе команд для сохранения и загрузки.

Класс для определения способа ввода команды (class InputHandler) является также шаблонным, в качестве шаблона передаётся класс для считывания команды (в данном случае будет использоваться class CommandHandler). Перечислим методы класса InputHandler:

Command enterCommand() - сначала вызывает метод для вывода всех доступных команд, которые были считаны из файла, по умолчанию файл — settings.txt, далее получает команды от пользователя.

std::pair<int,int> getAttackCoordinates() - считывает координаты для атаки.

`std::pair<int,int> getAbilityCoordinates()` - считывает координаты для применения способностей.

`Layout readShipLayout()` - считывает ориентацию корабля для расположения на поле пользователя.

`std::pair<Layout,std::pair<size_t,size_t>> readShipFromUser(int shipSize)` — считывает от пользователя координаты корабля, также вызывает предыдущий метод для считывания ориентации.

Класс для считывания ввода пользователя из терминала и преобразования ввода в команду (`class CommandHandler`). Перечислим методы класса `CommandHandler`:

`bool loadFile(const std::string &filePath)` — загружает из файла информацию о командах, если файл не соответствует формату (необходимо задавать команды в виде: буква — наименование команды), либо в файле несколько ключей соответствует одной команде, либо один ключ соответствует нескольким командам, то это приводит к ошибке, ключи и команды будут установлены по умолчанию.

`std::string turnCommandToString(Command command)` — переводит команду(`enum` значение) в строковый вид, для вывода пользователю на экран.

`std::string allCommands()` - собирает все команды в строку для вывода в виде списка.

`Command getCommand(char value)` — возвращает команду по ключу (букве), если запрашиваемой команды нет, тогда выбросится ошибка.

Шаблонный класс для отрисовки игры (`class GameDrawer`) принимает в виде шаблона класс (в данном случае будет `class FieldDrawer`), который будет реагировать на изменения игры и выводить на экран игровое поле.

Данный класс имеет один метод `void drawFields(PlayingField userPlayingField, PlayingField enemyPlayingField)` — вызывает методы шаблонного для вывода игрового поля на экран.

Класс FieldDrawer необходим для отслеживания изменений в игре и вывода текущего игрового поля на экран.

Данный класс имеет один метод void printPlayingField(PlayingField field, bool isEnemyField) — принимает текущее поле и флаг (для проверки: является ли поле вражеским), выводит текущее поле на экран.

Также был создан отдельный класс для вывода типовых предложений на экран (class GameMessage).

Связи между классами представлены на рисунке 1:

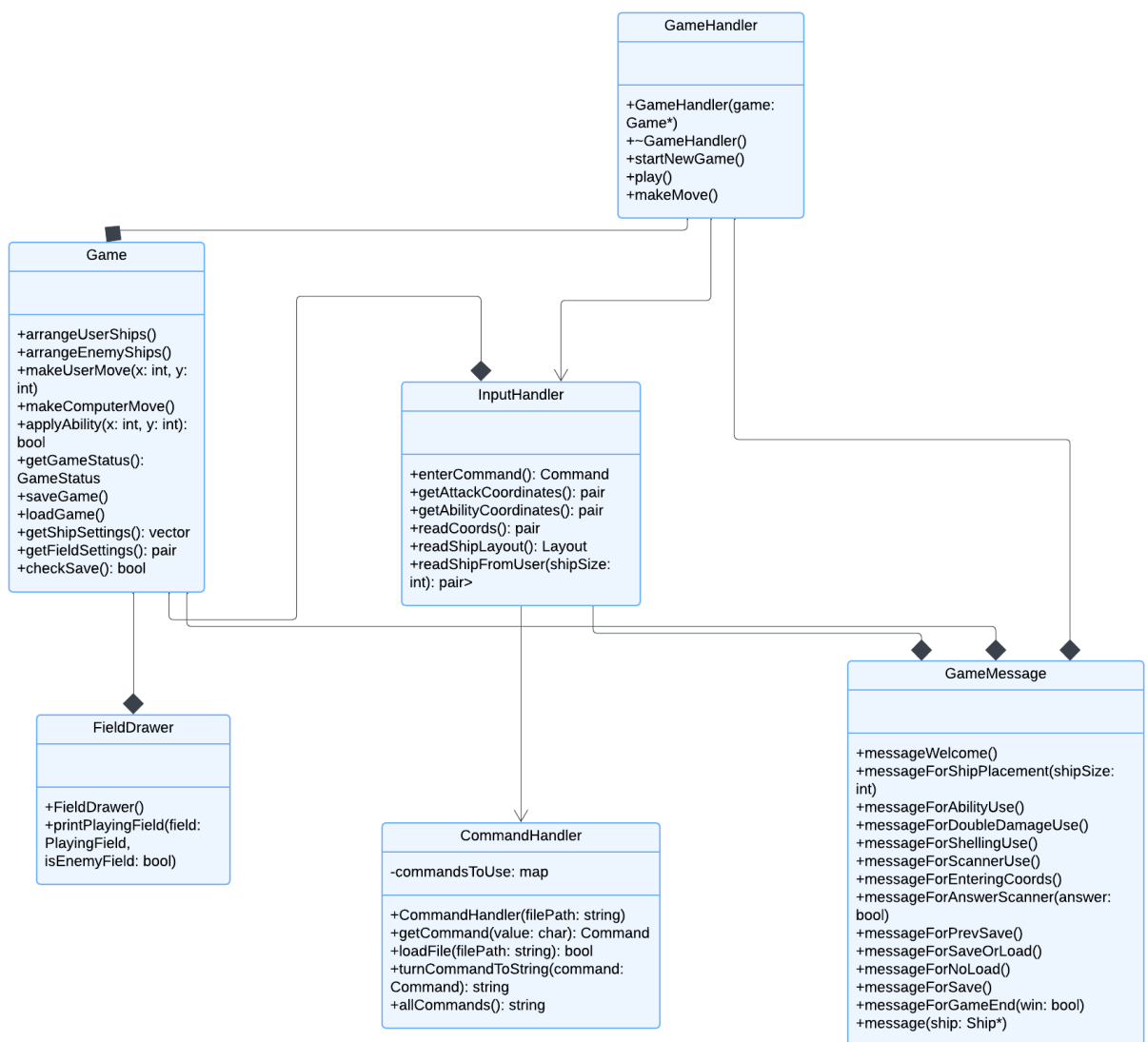


Рисунок 1 — UML диаграмма классов

Выводы

Была изучена работа с шаблонными классами в языке программирования C++. В соответствии с заданными условиями лабораторной работы созданы собственные шаблонные классы, реализующие функционал игры «Морской бой», а именно шаблонный класс управления игрой, в параметре шаблона указывается класс для обработки ввода, который также является шаблонным и принимает класс для обработки команды, шаблонный класс отображения игры, который занимается отрисовкой игрового поля.