

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: Шаблонные классы

Студент гр. 3341

Самокрутов А.Р.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2024

Цель работы.

Изучить шаблонные классы и их реализацию в языке программирования C++. Разработать структуру управления игрой на основе ранее разработанной системы классов, использующую шаблонные классы для обработки ввода пользователя, взаимодействия с игровой логикой и визуализации текущего состояния игры.

Задание.

а) 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.

б) 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.

в) 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.

г) 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.

Класс `GameController` осуществляет управление игрой. Класс является шаблонным, и в качестве параметров принимает некоторые классы `Input` и `Renderer` — соответственно класс, считывающий пользовательский ввод, и класс, реализующий отображение игры. Данный класс содержит в себе ссылку на игру `Game& curr_session` и единственные экземпляры классов ввода и вывода (подразумевается паттерн проектирования «одиночка»). Помимо этого класс содержит объект `unordered_map` хранящий в себе пары типа «команда — функция».

Конструктор класса `GameController<Input, Renderer>::GameController (Game& game)` инициализирует поля игры и обработчика вывода, а так же ставит каждой команде в соответствие нужную функцию (= метод игры класса `Game`).

Метод `run()` запускает игру, и далее, пока игра не закончилась, считывает команду — ввод игрока. Если нашлась введённая команда, т.е. ей в соответствие была поставлена какая-то функция, то эта функция вызывается, иначе выводится сообщение о том, что введённая команда является неизвестной. После хода игрока производится ход бота, а также проводится проверка условий конца игры или начала нового раунда. Последним шагом в итерации цикла вызывается метод обработчика вывода, визуализирующий текущее состояние игры.

Был реализован шаблонный класс `GameRenderer`, принимающий в качестве параметра некоторый класс `Output` — синглтон, имеющий методы отрисовки поля, менеджера кораблей и сообщений. Данный класс имеет методы, отвечающие за визуализацию отдельных этапов игры на основе методов класса-отрисовщика поля.

Класс `CLGameOutput` осуществляет отображение поля и др. игровых сущностей через консоль. Данный класс реализует шаблон проектирования «одиночка» на основе синглтона Майерса.

Класс `CLGameInput` осуществляет считывание команд и данных из консоли. Данный класс содержит в себе свой единственный экземпляр (опять же, паттерн проектирования «одиночка»), пути к файлам содержащим соответствия типа «ввод — команда», вспомогательные методы для считывания чисел, координат и строк, а так же основные — для считывания размера поля, размеров кораблей, координат корабля при его постановке и атаке.

Диаграмма классов, разработанных в ходе выполнения лабораторной работы, представлена на рис.1

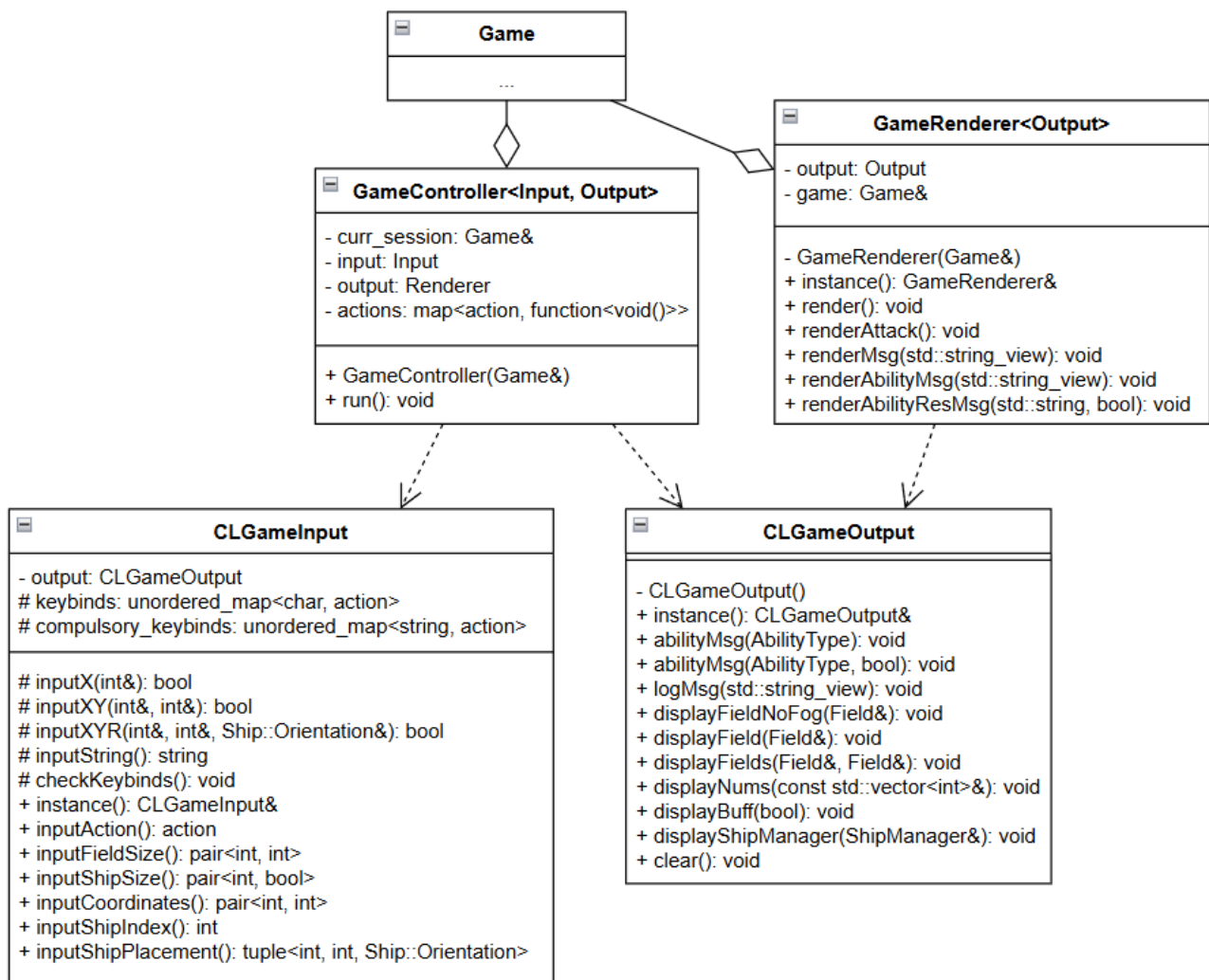


Рисунок 1 – диаграмма классов

Выводы.

В результате выполнения лабораторной работы были изучены шаблоны и шаблонные классы в языке программирования C++.

Был реализован шаблонный класс управления игрой, шаблонный класс отображения игры, класс, обрабатывающий ввод пользователя из терминала, а также класс-отрисовщик поля.

Реализованные классы были успешно интегрированы в существующую систему.