

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

ОТЧЕТ
по лабораторной работе № 5
по дисциплине «Объектно-Оrientированное Программирование»
Тема: Шаблонные классы, генерация карты

Студент гр. 1384

Шаганов В.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2022

Цель работы.

Разработать классы, позволяющие задать клавиши для управления. Разработать классы, обрабатывающие команды пользователя таким образом, чтобы без изменения существующего кода можно было внедрить иной способ ввода команд.

Задание.

Реализовать шаблонный класс генерирующий игровое поле. Данный класс должен параметризоваться правилами генерации (расстановка непроходимых клеток, как и в каком количестве размещаются события, расположение стартовой позиции игрока и выхода, условия победы, и.т.д.). Также реализовать набор шаблонных правил (например, событие встречи с врагом размещается случайно в заданном в шаблоне параметре, отвечающим за количество событий)

Требования:

- Реализован шаблонный класс генератор поля. Данный класс должен поддерживать любое количество правил, то есть должен быть variadic template.
- Реализовано не менее 6 шаблонных классов правил
- Классы правила должны быть независимыми и не иметь общего класса-интерфейса
- При запуске программы есть возможность выбрать уровень (не менее 2) из заранее заготовленных шаблонов

Примечания:

- Для задания способа генерации можно использовать стратегию, компоновщик, прототип

Выполнение работы.

Был создан шаблонный класс-генератор карты MapGenerator, принимающий любое количество шаблонных параметров. В нём был создан публичный метод generate(), возвращающий сгенерированную карту. Создание карты сводится к следующим пунктам:

- Найти среди правил то, что может создать карту (имеет метод createMap). Использовать такое правило в первую очередь, иначе используется стандартный конструктор карты.
- Пройтись по всем правилам, умеющим модифицировать карту, и модифицировать ранее созданную карту с их помощью.
- Вернуть сгенерированную карту.

Проверка на то, может ли правило создать карту реализовано при помощи механизма SFINAE. И реализуется данный механизм при помощи шаблонного класса CanCreateMap.

Были созданы следующие правила генерации карты:

- FieldSizeRule<int, int> определяет размеры поля. Создаёт поле заданного размера.
- WinEventPositionRule<int, int> определяет координаты события, приводящего к победе.
- PlayerPositionRule<int, int> определяет изначальное положение игрока.
- ObstaclesRule<ObstaclesVariant> определяет способ расставления препятствий.
- LoseEventPositionRule<int, int> определяет координаты события, приводящего к поражению.
- DamageEventFrequencyRule<int, int> определяет процент заполнения карты событиями урона + определяет урон этих событий.

На выбор пользователю предоставляется два уровня, каждый создан с разными правилами генерации.

Диаграмма разработанных в ходе лабораторной работы классов:

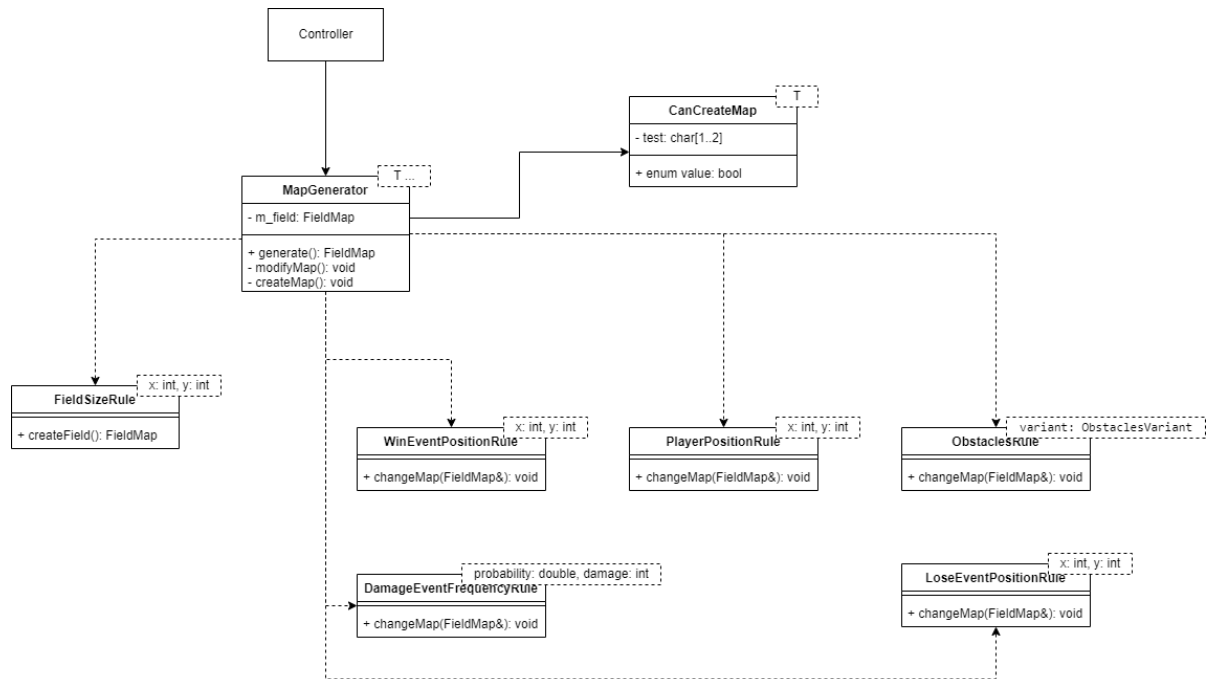


Рисунок 1 – Диаграмма классов

Тестирование.

При запуске пользователю предлагается на выбор 2 карты, каждая из которых генерируется с разными шаблонными правилами.

```

try {
    if (mapType == Map::MapType::Dungeon) {
        Map::MapGenerator<
            Map::ObstaclesRule<Map::ObstaclesVariant::ROOMS>,
            Map::FieldSizeRule<10,10>,
            Map::DamageEventFrequencyRule<50, 5>,
            Map::WinEventPositionRule<-2,-2>,
            Map::PlayerPositionRule<2,2>
        > mg;
        m_map = mg.generate();
    }
    else if (mapType == Map::MapType::Overworld) {
        Map::MapGenerator<
            Map::ObstaclesRule<Map::ObstaclesVariant::SPIRAL>,
            Map::FieldSizeRule<8,12>,
            Map::WinEventPositionRule<(8-1)/2,12/2>,
            Map::PlayerPositionRule<1,1>
        > mg;
        m_map = mg.generate();
    }
}

```

Рисунок 2 – Создание карты

Выводы.

В ходе выполнения лабораторной работы был разработан шаблонный класс – генератор карты, использующий классы-правила, передаваемые

генератору в качестве шаблонных параметров. При этом правила не имеют общего базового класса и полностью независимы.