

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

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Объектно-ориентированное программирование»
Тема: шаблонные классы, генерация карты

Студент гр. 1384

Прошичев А.В.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2022

Цель работы.

Освоение и использование шаблонов. Создать генератор карты, используя шаблонные классы.

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.

Класс генератор создает поле, а не принимает его.

Класс генератор не должен принимать объекты классов правил в каком-либо методе, а должен сам создавать (в зависимости от реализации) объекты правил из шаблона.

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

Классы правила не должны быть только “хранилищем” для данных.

Так как используются шаблонные классы, то в генераторе не должны быть `dynamic_cast`

Примечания:

Для задания способа генерации можно использовать стратегию, компоновщик, прототип

Не рекомендуется делать static методы в классах правилах

Выполнение работы.

Класс, определяющий уровень генерации карты, `LevelManager` при создании в конструктор одним из параметров принимает число – уровень, который будет выбран для генерации карты. В зависимости от уровня(числа) `LevelManager` создаёт класс данной уровня `Level`. Применяет метод `GenerateLevel()`, который создаёт поле и возвращает его. Далее этот класс будет хранить поле и уничтожит его при своём разрушении.

Класс `Level` является интерфейсом класс уровней. Он содержит чистый виртуальный метод `GenerateLevel()` и виртуальный деструктор этого класса. Его реализациями являются `FirstLevel` и `SecondLevel`. Наследники используют для создания поля с определёнными параметрами класс `Generator` с определённым набором правил. Далее вызывают у объекта генератора метод создания поля и возвращают полученное поле.

Класс `Generator` отвечает за обработку правил. Имеет такой класс всего один метод `Fill()` и является `variadic template`, то есть принимает любое число правил. Правила распаковываются и обрабатываются с помощью функторов, определённых у каждого правила по-своему.

Правила:

- правило `BushRule`: принимает позицию на карте в виде двух чисел. Случайным порядком выбирает два числа. Если первое число чётное, то на позицию устанавливается куст с медведем, иначе – куст с зайцем. Аналогично вокруг этого куста устанавливаются ещё 8 однотипных кустов. Если число чётное, то куст с ягодами, иначе – куст с волками.

- правило EarthquakeRule: принимает позицию на карте в виде двух чисел.

На эту позицию устанавливается событие землетрясения. По горизонтали и вертикали, содержащих эту точку, устанавливаются события водных пещер.

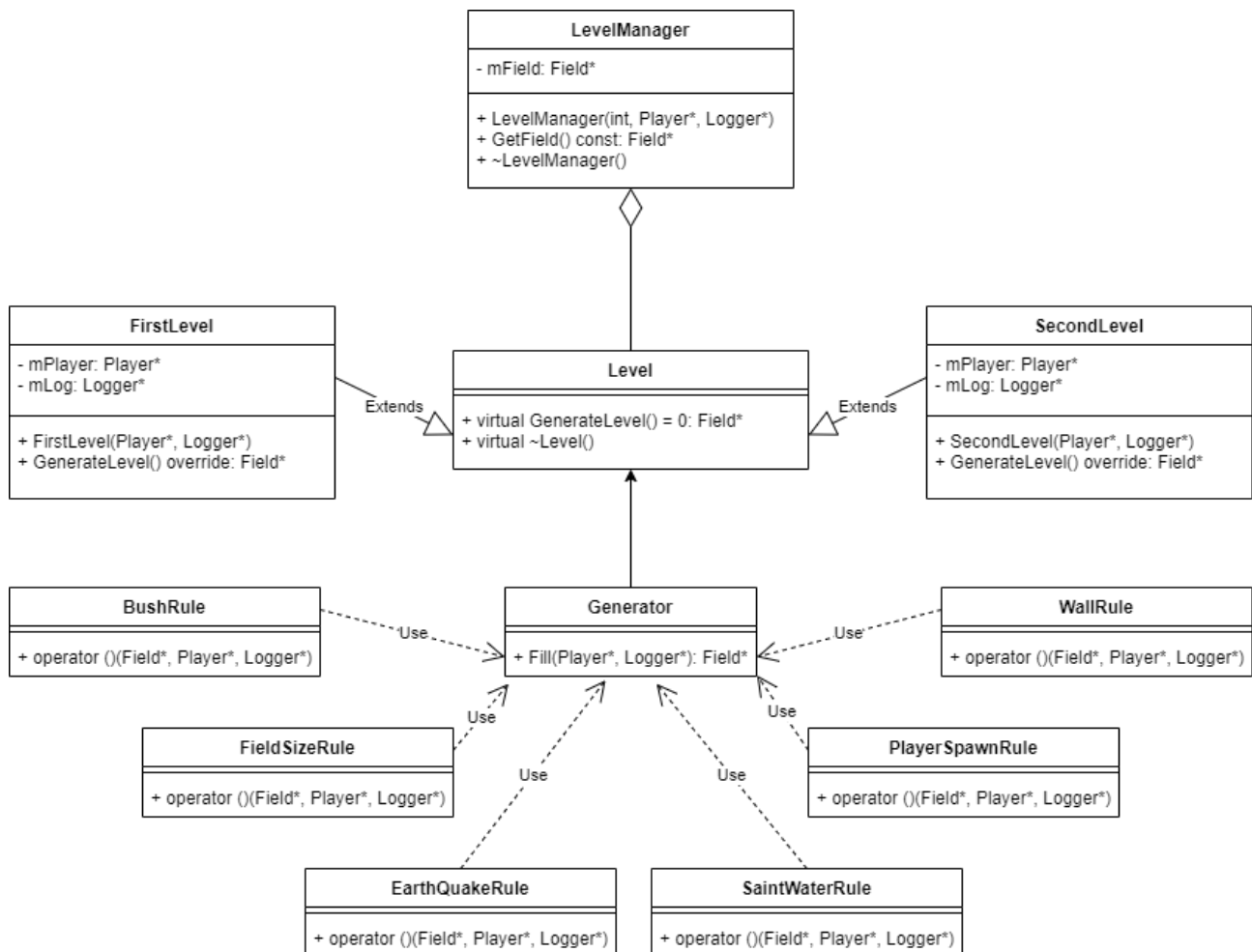
- правило FieldSizeRule: принимает пары чисел: длину и ширину поля. По данным числам создаёт поле требуемого размера.

- правило PlayerSpawnRule: принимает позицию на карте в виде двух чисел. На эту позицию устанавливает игрока.

- правило SaintWaterRule: принимает позицию на карте в виде двух чисел.

На эту позицию устанавливает событие святой воды.

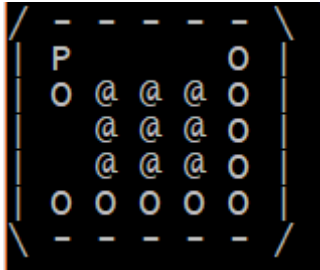
- правило WallRule: принимает параметром число, данное число задаёт вероятность, с которой на карте появятся стены. Чем больше число, тем меньше стен.



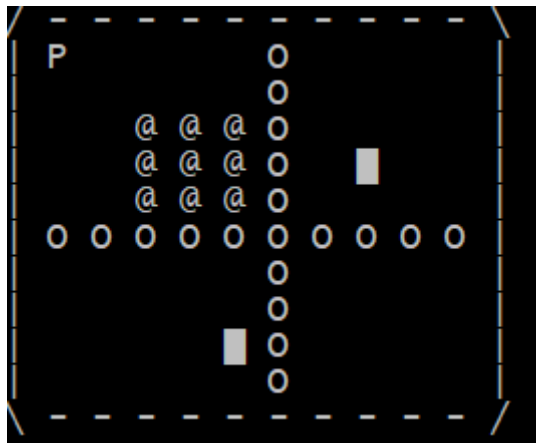
Тестирование.

```
which level do you want to choose? Enter 1 or 2
1_
```

Изображение №1. Выбор уровня



Изображение №2. Первый уровень



Изображение №3. Второй уровень

Выводы.

Освоил и шаблоны и работу с ними. Создал генератор карты, используя шаблонные классы.