**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Объектно-Ориентированное программирование»**

Тема: Интерфейсы, динамический полиморфизм

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1303 |  | Королева П.А |
| Преподаватель |  | Жангиров Т.Р |

Санкт-Петербург

2022

## Цель работы.

Реализовать интерфейс события.

## Задание.

Реализовать систему событий. Событие - сущность, которая срабатывает при взаимодействии с игроком. Должен быть разработан класс интерфейс общий для всех событий, поддерживающий взаимодействие с игроком. Необходимо создать несколько групп разных событий реализуя унаследованные от интерфейса события (например, враг, который проверяет условие, будет ли воздействовать на игрока или нет; ловушка, которая безусловно воздействует на игрока; событие , которое меняет карту; и.т.д.). Для каждой группы реализовать конкретные события, которые по разному воздействуют на игрока (например, какое-то событие заставляет передвинуться игрока в определенную сторону, а другое меняет характеристики игрока). Также, необходимо предусмотреть событие “Победа/Выход”, которое срабатывает при соблюдении определенного набора условий.

Реализовать ситуацию проигрыша (например, потери всего здоровья игрока) и выигрыша игрока (добрался и активировал событие “Победа/Выход”)

## Выполнение работы.

Были созданы две группы событий, наследующиеся от базового Event.

Классы групп:

― EventPlayer – события которые влияют только на игрока

― EventField – события которые также влияют на поле

От каждой группы унаследовано еще по два события:

EventPlayer: Damage, Level\_up

EventField: Bone, BoneGround

Описание каждого события:

*Class Damage*

Безусловно убавляет здоровье игрока на единицу, когда он наступает на колючку.

*Class Level\_up*

Увеличивает уровень игрока, когда он наступает на поле с цветами, но при условии достаточного опыта:

1 уровень (опыт 0-5)

2 уровень (опыт 6-10)

3 уровень (опыт 11-15)

*Class Bone*

Безусловно увеличивает опыт игрока на единицу, когда он становится на клетку с косточкой. При этом тип клетки меняется на обычную и указатель на событие приравнивается к nullptr, чтобы это событие не вызывалось при повторном наступании на клетку.

*Class BoneGround*

Увеличивает опыт игрока на две единицы, когда он наступает на клетку с закопанной в землю костью. Это возможно при условии, что уровень игрока больше 1.

Для создания событий была реализована абстрактная фабрика.

Для этого реализованы классы:

Class CommonEventsFactory является абстрактным классом, интерфейсом для всех конкретных фабрик. Содержит один чисто виртуальный метод createEvent().

От него наследуются конкретные фабрики по созданию событий:

― BoneFactory

― Bone\_from\_groundFactory

― DamageFactory

― Level\_upFactory

Каждая конкретная фабрика имеет конструктор, в котором инициализируются поля, необходимые для создания соответствующего события.

Создание событий в процессе инициализации карты

В контроллере, в методе InitMap() создаются экземпляры четырех конкретных фабрик и, в зависимости от типа клетки, фабрика передается в метод init\_event(), который работает с CommonEventsFactory, т.е интерфейсом общим для всех фабрик. Метод возвращает указатель на Event, который устанавливается в нужное поле клетки.

Завершение игры

Для этого в классе Сontroller реализован метод fail\_or\_win(), который отслеживает характеристики игрока и возвращает число -1, 1, 0 в случае проигрыша, победы или ничего из перечисленного соответственно.

В классе CommandReader, поддерживающем взаимодействие с пользователем, реализован метод NotEndGame(), который на основе значений Controller::fail\_or\_win() либо завершает главный цикл программы и выводит сообщение о победе/проигрыше на экран, либо позволяет циклу продолжаться.

**UML-диаграмма иерархии событий.**

Иерархия классов событий представлена на рисунке 1.

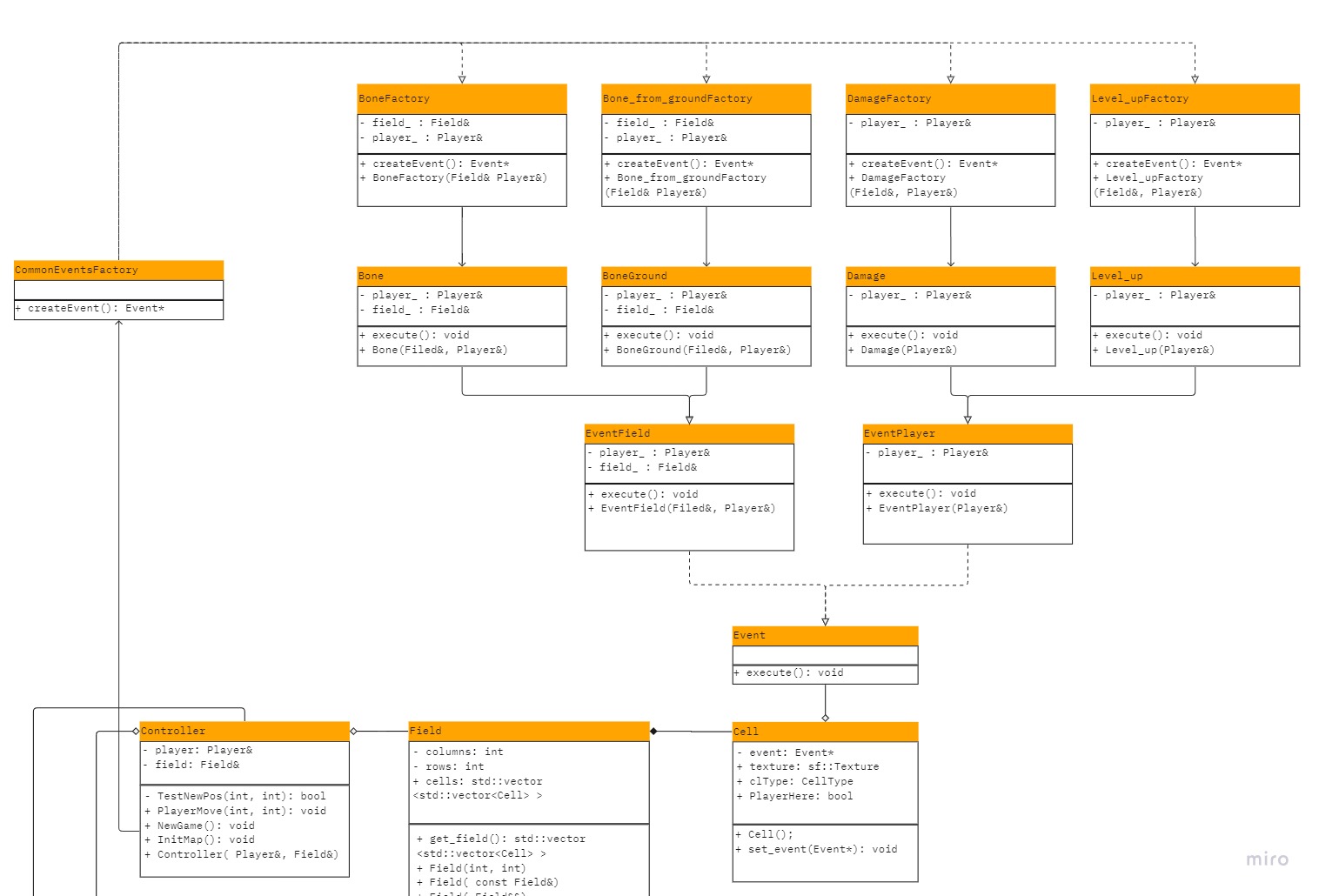


Рис.1 – Диаграмма иерархии событий

## Тестирование.



## Рис.2 – Запуск программы

Проверка корректности работы событий:



Рис.3 – Событие увеличивающее опыт и меняющее тип клетки при поедании косточек



Рис.4 – Событие уменьшающее здоровье при прохождении через колючку



Рис5. – Событие увеличивающее уровень при *условии* достаточного опыта, при попадании на поле с цветами.



Рис.6 – Событие выкапывания косточки из земли при *условии* нужного уровня

## Выводы.

Был реализован интерфейс события.