

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

**ОТЧЕТ**  
**по лабораторной работе № 2**  
**по дисциплине «Объектно-Оrientированное Программирование»**  
**Тема: Интерфейсы, динамический полиморфизм**

Студент гр. 1384

\_\_\_\_\_

Шаганов В.А.

Преподаватель

\_\_\_\_\_

Жангиров Т.Р.

Санкт-Петербург

2022

### **Цель работы.**

Разработать абстрактные классы, реализующие ранее написанный интерфейс события. Разработать конкретные события – наследники абстрактных классов, оказывающие влияние на игровой мир.

### **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 реализовав 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”)

#### **Требования:**

- Разработан интерфейс события с необходимым описанием методов
- Реализовано минимум 2 группы событий (2 абстрактных класса наследников события)
- Для каждой группы реализовано минимум 2 конкретных события (наследники от группы события)
- Реализовано минимум одно условное и безусловное событие (условное - проверяет выполнение условий, безусловное - не проверяет).

- Реализовано минимум одно событие, которое меняет карту (меняет события на клетках или открывает расположение выхода или делает какие-то клетки проходимыми (на них необходимо добавить события) или не непроходимыми
- Игрок гарантированно имеет возможность дойти до выхода

### **Выполнение работы.**

Был создан интерфейс события – класс Event. Его реализуют абстрактные классы – события определённого типа, от которых наследуются классы определённых событий. Например, цепочка наследований может выглядеть следующим образом:

ChangeCellsEvent → MapEvent → Event

Где MapEvent – абстрактный класс, символизирующий событие, влияющее на игровое поле, а ChangeCellsEvent – конкретное событие, меняющее клетки на поле.

Класс клетки Cell хранит указатель на событие и имеет метод trigger, которое просто вызывает метод trigger хранящегося события.

Нужно отметить, что аргументы событию передаются в конструкторе. Таким образом, сначала создаётся объект события, и уже этот объект передаётся клетке через метод setEvent.

Вызов метода trigger клетки происходит из контроллера – каждый раз, когда игрок меняет свою координату, вызывается метод реагирования для клетки на текущей координате персонажа.

Описание разработанных событий:

- ChangeCellEvent: Меняет клетки на заданных позициях на заданные клетки;
- ChangeMapEvent: Меняет текущую карту (поле) на заданное. Также перемещает игрока на заданную позицию;
- WinStateEvent: Изменяет состояние игры на “WIN”;
- LoseStateEvent: Изменяет состояние игры на “LOSS”;

- DamagePlayerEvent: Отнимает у игрока заданное количество очков здоровья;
- PlayerSpeedMultiplier: Умножает показатель скорости игрока на заданную величину, при этом существуют минимальная и максимальные скорости, при достижении которых событие не меняет скорость;
- PlayerHealEvent: Если здоровье игрока не максимально, поднимает уровень здоровья до максимума.

Диаграмма классов представлена ниже:

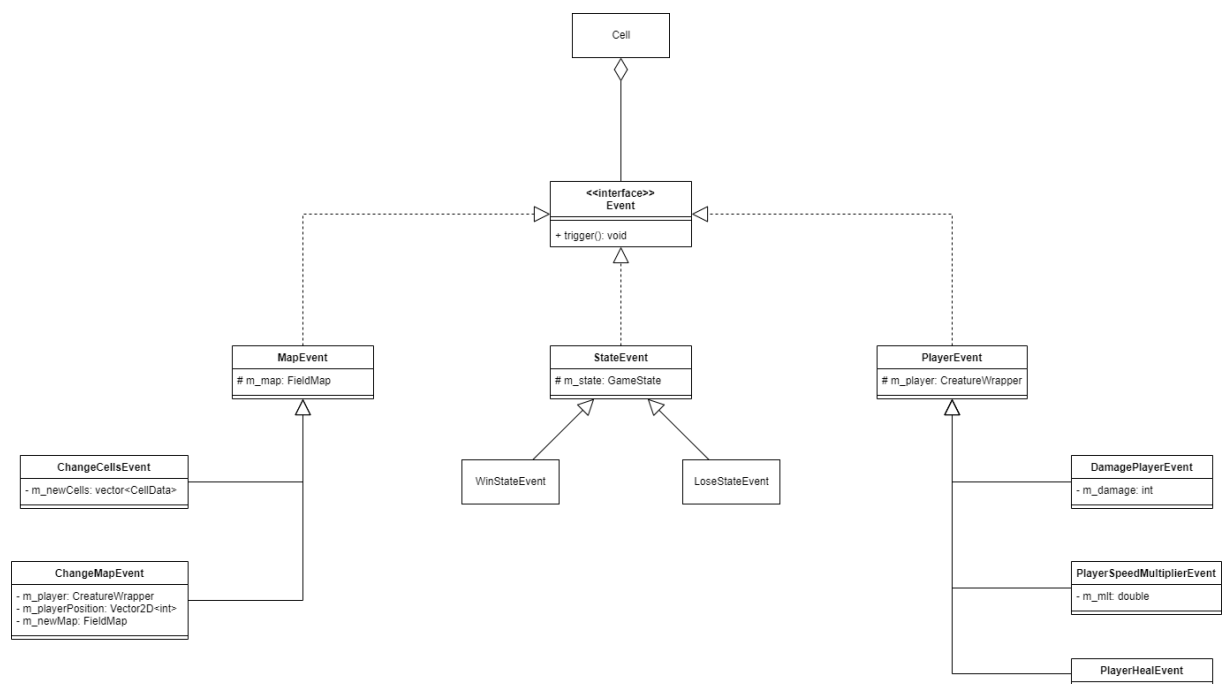


Рисунок 1 – Диаграмма классов событий

## Тестирование.

Для тестирования события были установлены на определённые клетки.

Результат выполнения события **ChangeCellsEvent**:



Рисунок 2 – ChangeCellsEvent: до наступления



Рисунок 3 – ChangeCellsEvent: после наступления

На игровом поле появляются (сменяют существующие) заданные клетки.

Результат выполнения события DamagePlayerEvent:



Рисунок 4 – DamagePlayerEvent: до наступления



Рисунок 5 – DamagePlayerEvent: после наступления

Можно видеть, что здоровье игрока понижается – с 100 до 80.

Результат выполнения события PlayerHealEvent:



Рисунок 6 – PlayerHealEvent: до наступления



Рисунок 7 – PlayerHealEvent: после наступления

Здоровье игрока поднимается с 40 до 100.

Результат выполнения события ChangeMapEvent:



Рисунок 8 – ChangeMapEvent: до наступления



Рисунок 9 – ChangeMapEvent: после наступления

Игровое поле заменяется, позиция игрока также меняется.

## Выводы.

В ходе работы было разработано 3 абстрактных класса, реализующих интерфейс Event. Наследники этих классов, конкретные события, оказывают непосредственное воздействие на игровой мир. Было проведено тестирование программы – проверена работоспособность созданных событий.