

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

ОТЧЕТ №1
по лабораторной работе
по дисциплине «Объектно-ориентированное программирование»
ТЕМА: СОЗДАНИЕ КЛАССОВ, КОНСТРУКТОРОВ И МЕТОДОВ

Студент гр. 1381

Дудко М.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2022

Цель работы.

Ознакомиться с работой основных принципов ООП. Написать примитивную игру на языке C++.

Задание.

Реализовать прямоугольное игровое поле, состоящее из клеток. Клетка - элемент поля, которая может быть проходима или нет (определяет, куда может стать игрок), а также содержит какое-либо событие, которое срабатывает, когда игрок становится на клетку. Для игрового поля при создании должна быть возможность установить размер (количество клеток по вертикали и горизонтали), игровое поле должно быть зациклено по вертикали и горизонтали, то есть если игрок находится на правой границе и идет вправо, то он оказывается на левой границе (аналогично для всех краев поля).

Реализовать класс игрока. Игрок - сущность контролируемая пользователем. Игрок должен иметь свой набор характеристик и различный набор действий (например, разные способы перемещения, попытка избежать событие, и так далее).

Требования:

- Реализован класс игрового поля
- Для игрового поля реализован конструктор с возможностью задать размер и конструктор по умолчанию (то есть конструктор, который можно вызвать без аргументов)
- Реализован класс интерфейс события (в данной лабораторной это может быть пустой абстрактный класс)
- Реализован класс клетки с конструктором, позволяющим задать ей начальные параметры.
- Для клетки реализованы методы реагирования на то, что игрок перешел на клетку.
- Для клетки реализованы методы, позволяющие заменять событие. (То есть клетка в ходе игры может динамически меняться)
- Реализованы конструкторы копирования и перемещения, и соответствующие им операторы присваивания для игрового поля и при необходимости клетки
- Реализован класс игрока минимум с 3 характеристиками. И соответствующие ему конструкторы.
- Реализовано перемещение игрока по полю с проверкой допустимости на переход по клеткам.

Примечания:

- При написании конструкторов учитывайте, что события должны храниться по указателю для соблюдения полиморфизма
- Для управления игроком можно использовать медиатор, команду, цепочку обязанностей

Выполнение работы.

В классе `Cell` я обозначаю клетку поля, которая содержит в себе значение для заполнения массива, а так же 4 состояния, для инициализации этой клетки как пустую, событие, или игрока.

В классе `Field` я создаю векторный массив (поле) из клеток (`Cell`) в методе `create_map`, конструктора копирования и перемещения, геттеры для различных приватных полей класса, методы `add_wall` и `add_apple`, которые меняют состояние клетки с пустого на одно из событий. (проходятся по всему полю что бы найти пустое, затем с помощью алгоритма выбираются случайное из этого числа). В методе `move` мы считываем с класса `controller` направление движение игрока и меняем его положение на единицу в соответствующем направлении, инициализируем состояние этой клетки как игрок, и стираем состояние клетки прошлого местонахождение игрока на пустое. Если встречаем на пути стену то движение останавливается. Если встречаем конец поля то телепортируемся в противоположный край.

В классе `Controller` в методе `player_handler` мы считываем какую клавишу нажал игрок, и определяем в каком направлении игрок будет продолжать движение, так же добавляем некоторые геттеры.

В классе `Drawer` в методе `draw` мы загружаем текстуры и спрайты из файла, а далее исходя из состояние клетки отображаем его на игровом поле.

В `main` мы создаем объекты классов, создаем игровое поле с помощью метода, затем динамически с небольшой задержкой мы сначала рисуем поле, затем считываем нажал ли пользователь клавишу для управления, затем двигаем игрока, а затем добавляем события.

