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

ОТЧЕТ

по лабораторной работе № 1
по дисциплине «Объектно-Ориентированное Программирование»
Тема: Создание классов, конструкторов и методов

Студент гр. 1384 _____ Шаганов В.А.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург
2022

Цель работы.

Необходимо реализовать классы, отвечающие за поле и его клетки, игрока, события. Код должен быть написан в ООП-стиле.

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.

Центральный класс игры – GameCore. Он хранит информацию о поле, игроке и объектах в целом.

Для взаимодействия с пользователем используется класс-посредник GameMediator, который передаёт данные между классами GameCore и SfmlEventReader. При нажатии на клавишу, SfmlEventReader отлавливает его и переводит в тип UserEventType, и далее передаёт его в посредника, который в свою очередь передаёт его классу GameCore. В зависимости от события, который получил GameCore, выполняются некоторые действия (передвижение игрока, закрытие окна). Такая конструкция позволяет отделить пользовательский ввод от логики, таким образом контроллер получает не код нажатой клавиши, а тип события, на который он должен отреагировать.

Для отрисовки карты используется метод класса LevelPainter – drawWindow. Тот, в свою очередь, пробегается по всем клеткам карты и передаёт тип клетки TileType класса SpriteManager, который возвращает спрайт клетки,

который будет отображён в окне. Таким же образом отображается и игрок – у класса Creature есть поле m_type типа CreatureType. Метод getCreatureSprites класса SpriteManager возвращает таблицу спрайтов, ключом которой выступает тип Direction – сторона, в которую смотрит существо. SpriteManager использует TextureManager, который хранит текстуры и выдаёт их по аргументу типа TextureType. Что SpriteManager, что TextureManager реализованы, используя паттерн проектирования Singleton, т.к. они хранят общие ресурсы и ведут себя одинаково, вне зависимости от контекста их использования, то есть нам не нужно более одного объекта данных классов.

Класс FieldMap отвечает за карту – он хранит двумерный массив указателей на клетки. В классе реализованы методы для получения информации о клетке на заданной позиции. Клетка представляет из себя объект класса Cell или производного от него. В клетке хранится информация о её проходимости и о событии, происходящем при взаимодействии клетки с игроком.

Игрок является объектом класса Player, производным от класса Creature, который, в свою очередь, дочерний по отношению к классу Object. Creature хранит характеристики существа и имеет методы для их изменения.

Тестирование.

Программа запускается и отображает карту с персонажем:

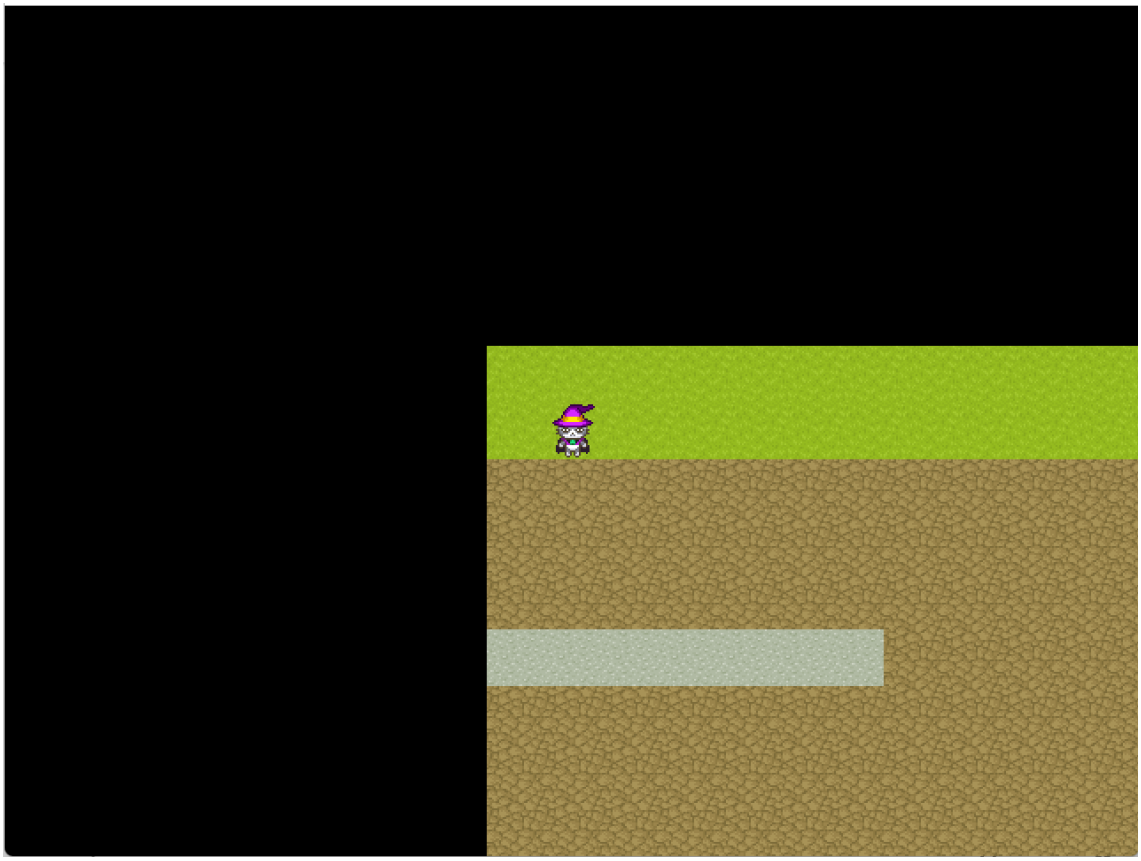

Рисунок 1 – Запуск

При нажатии кнопок управления – W,A,S,D, персонаж передвигается:

Рисунок 2 – Передвижение игрока

При попытке зайти за границу, игрок оказывается с противоположного края карты:

Рисунок 3 - Перед выходом за границу

Рисунок 4 - После выхода за границу

Выводы.

В ходе работы были разработаны классы, являющиеся основой игры. Были реализованы классы, отвечающие за: игровое поле; клетки на этом поле; игрока; отрисовку состояния игры, а также класс-контроллер, хранящий информацию о карте и об объектах на ней. При в классах, отвечающих за логику игры, нет ничего связанного с вводом и выводом информации – таким образом, бизнес-логика отделена от интерфейса.

Были освоены и применены такие паттерны проектирования, как Singleton и Mediator.