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

ОТЧЕТ №6

по лабораторной работе

по дисциплине «Объектно-ориентированное программирование»

ТЕМА: СЕРИАЛИЗАЦИЯ, ИСКЛЮЧЕНИЯ

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 систему классов позволяющих проводить сохранение и загрузку состояния игры. При загрузке должна соблюдаться транзакционность, то есть при неудачной загрузки, состояние игры не должно меняться. Покрыть программу обработкой исключительных состояний.

Требования.

- Реализована загрузка и сохранение состояния игры
- Сохранение и загрузка могут воспроизведены в любой момент работы программы.
- Загрузка может произведена после закрытия и открытия программы.
- Программа покрыта пользовательскими исключениями.
- Пользовательские исключения должны хранить полезную информацию, например значения переменных при которых произошло исключение, а не просто сообщение об ошибке. Соответственно, сообщение об ошибке должно учитывать это поля, и выводить информацию с учетом значений полей.
- Исключения при загрузке обеспечивают транзакционность.
- Присутствует проверка на корректность файла сохранения. (Файл отсутствует; в файле некорректные данные, которые нарушают логику; файл был изменен, но данные корректны с точки зрения логики).

Выполнение работы.

В классе Hash в методе hash вычисляет остаток от деления числа. В методе. В методе make_hash все числа в файле сначала пропускаются через метод hash, а затем записываются в сумму. В методе check_hash считается хэш и сравнивается с хэшем в конце файла. Если они не совпадают, то возвращает -1.

Создан класс Save_Field, который наследуется от класса Hash. Имеет поля 2 файловых потока (ofstream, ifstream), путь к файлу, методы сохранения и загрузки поля. В методе сохранения save_field метод принимает ссылку на поле и записывает в файл сначала размеры поля, с помощью методов получения размеров поля из объекта класса поля. А затем проходится по массиву, проверяя какое событие находится в клетке (с помощью методов класса клетки is_*some_event* использующий dynamic_cast) и записывает в файл цифру от нуля до семи. (0 – пустая, 1 – стрела, 2 – ключ и т.д.). В итоге получается массив размера игрового поля, заполненный порядковыми номерами событий. Затем в конец записывается хэш с помощью метода make_hash. В методе загрузки load_field сначала происходит проверка методом класса Hash check_hash. Если проверка не выполняется то с помощью throw в консоль выдает ошибку и программа заканчивает свою работу. Затем метод записывает из файла размеры поля, создает объект класса поля с заданными размерами, а дальше заполняет его событиями, идя по массиву, который получился в файле. С помощью метода поля который, устанавливает в клетку с заданными в методе координатами, заданное событие. После сохранения игры и возвращает поле. При записи события игрока, так же вызываются методы смены позиции игрока на поле.

Класс Save_Player работает аналогичным образом, но там всего 2 характеристики игрока, которые при вызове метода загрузки присваиваются текущему игроку.

В классе Command_Reader добавлены 2 кнопки (Save/Load). Для сохранения и загрузки соответственно.

В классе Controller_P в методе controller добавлено реагирование на нажатие этих кнопок. Так же в этом классе созданы объекты класса Save_Field / Save_Player. При нажатии кнопки Load в текущее поле перемещается поле возвращаемое при вызове метода load_field. А так же вызывается метод load_player, который меняет характеристики игрока на сохраненные. При нажатии кнопки Save, вызываются методы save_field и save_player соответственно.

UML диаграмма (на рис.1)

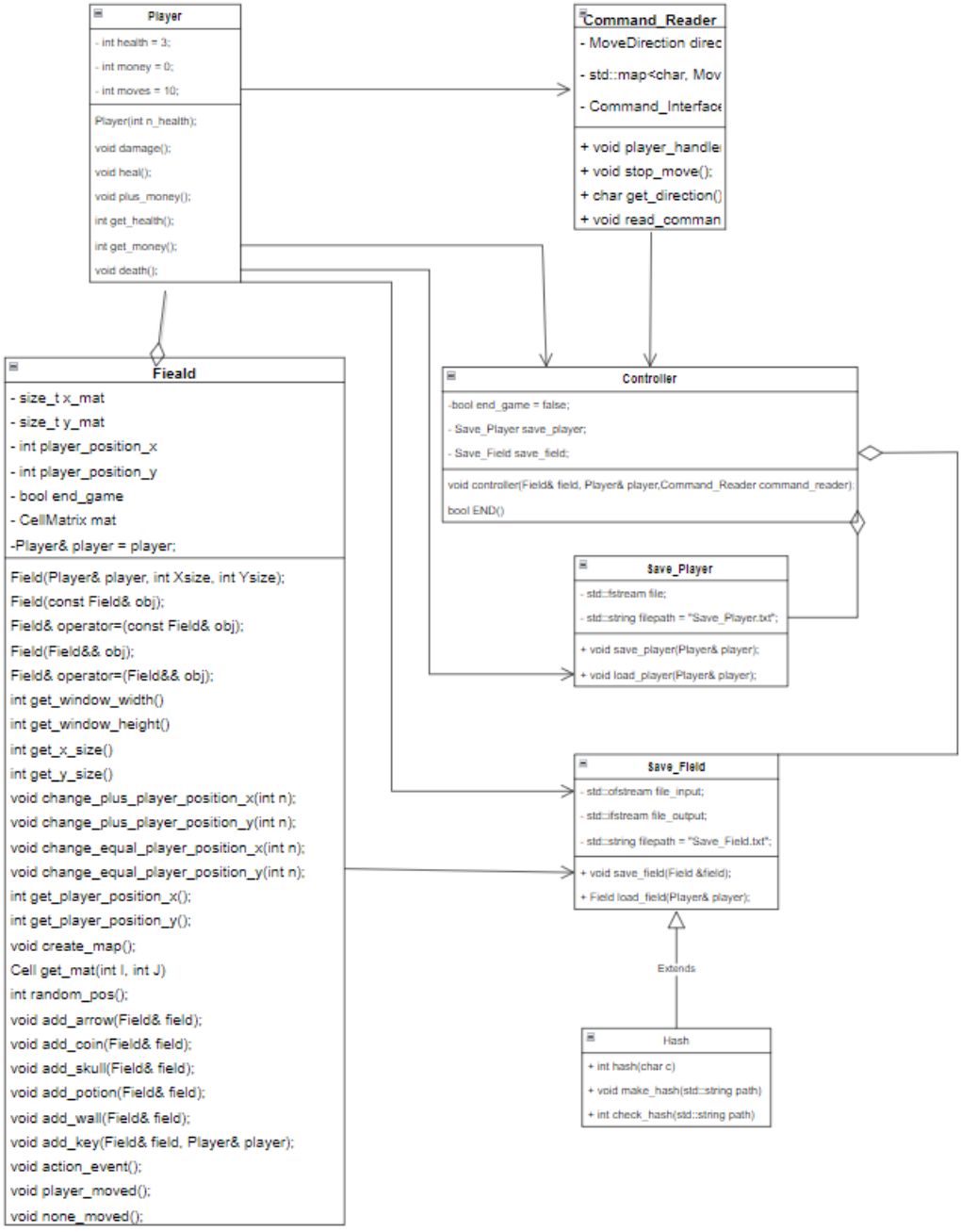

Рисунок 1- UML Диаграмма

Вывод

Изучены принципы вводы и вывода данных, а так же механизмы работы с файлами, изучены уровни абстракции, постигнут дзен, достигнута нирвана, познана вселенная.