

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

ОТЧЕТ
по лабораторной работе №6
по дисциплине «Объектно-ориентированное программирование»
ТЕМА: СЕРИАЛИЗАЦИЯ, ИСКЛЮЧЕНИЯ

Студентка гр. 0382

Андрющенко К.С

Преподаватель

Жангиров Т. Р

Санкт-Петербург

2021

Цель работы.

Создание классов, ход игры, с последующим его восстановлением, загрузкой и сохранением.

Задание.

Сериализация - это сохранение в определенном виде состоянии программы с возможностью последующего его восстановления даже после закрытия программы. В рамках игры, это сохранения и загрузка игры.

Требования.

- Реализовать сохранения всех необходимых состояний игры в файл
- Реализовать загрузку файла сохранения и восстановления состояния игры
- Должны быть возможность сохранить и загрузить игру в любой момент
- При запуске игры должна быть возможность загрузить нужный файл
- Написать набор исключений, который срабатывают если файл с сохранением некорректный
- Исключения должны сохранять транзакционность. Если не удалось сделать загрузку, то программа должна находиться в том состоянии, которое было до загрузки. То есть, состояние игры не должно загружаться частично

Выполнение работы.

UML-диаграмма для данной лабораторной работы см. Рисунок 1.

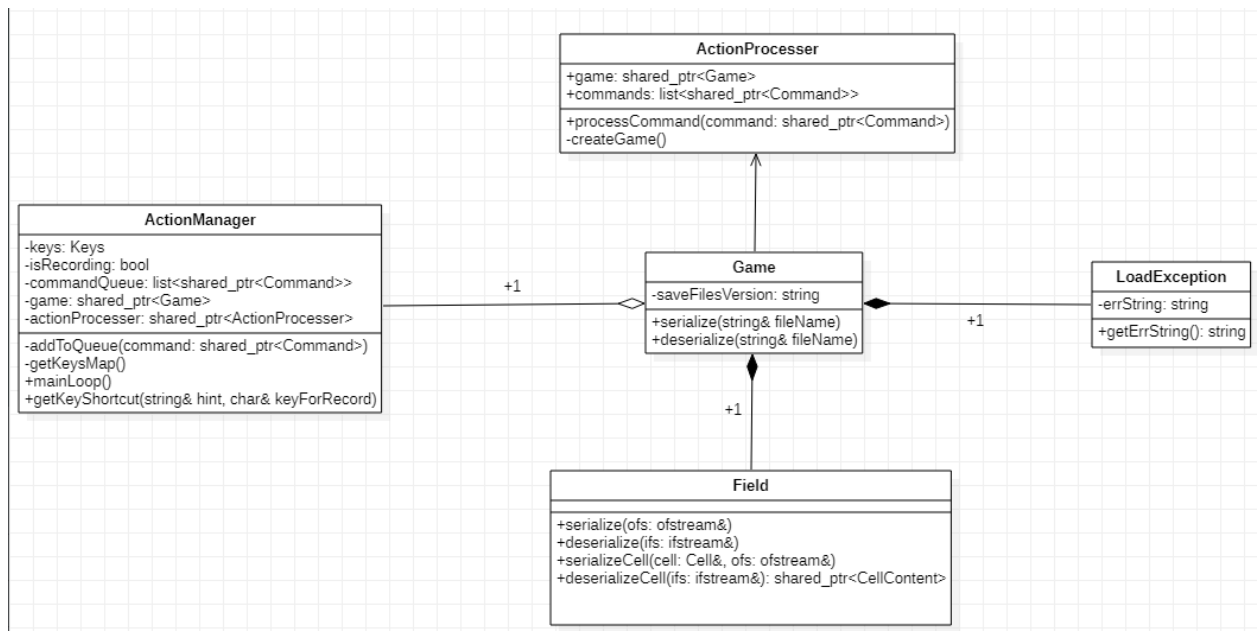


Рисунок 1 – UML-диаграмма лабораторной работы №6

Классы, созданные и модифицированные в данной лабораторной работе, их методы и свойства. Описание диаграммы.

Класс LoadException:

Заголовочный файл см. ПРИЛОЖЕНИЕ А.

Класс обработки исключений: хранит в себе текстовое поле исключения *string errString* и функцию, его возвращающую *string getErrString()*.

Класс ActionProcessor: Заголовочный файл см. Ошибка! Источник ссылки не найден..

В данном классе происходит проверка файла с сохранением игры и вызов исключений в функции *processCommand()*. При выборе команды типа **SAVE** и **LOAD** происходит загрузка файла по имени и сохранение в новый файл.

Класс ActionManager:

Добавлены 3 действия: сохранение игры, загрузка и создание новой игры.

```
else if (selectedKey == keys.SAVE) {
    resultCommand = std::make_shared<Save>();
```

```

    } else if (selectedKey == keys.LOAD) {
        resultCommand = std::make_shared<Load>();
    } else if (selectedKey == keys.NEW_GAME) {
        resultCommand = std::make_shared<NewGame>();
    }

```

Класс Field: Заголовочный файл см. Ошибка! Источник ссылки не найден..

Добавлены функции сериализации и десериализации для параметров поля и элементов клетки.

```

void serialize(std::ofstream& ofs);
void serializeCell(Cell& cell, std::ofstream& ofs);
void deserialize(std::ifstream& ifs);
std::shared_ptr<CellContent> deserializeCell(std::ifstream& ifs);

```

Данные команды сохраняют состояние поля и записывают в файл.

Проверка классов.

Приложим скриншоты тестов.

Запустим игру, пройдем по полю и сохраним состояние в файл incorrect, затем сохраним состояние в файл correct. Правильный файл обладает требованием на заголовок GameSaveFile_ver._1.0.

Удалим данный заголовок из файла incorrect. Проверим загрузку игры после запуска приложения и во время игрового процесса из правильного и неправильного файла. Выведем результат игры.

Main():

```

ActionManager actionManager;
actionManager.mainLoop();

```

Рисунок 2 – Главный цикл игры.

Output(консоль):

Enter ESC command

Enter UP command

Enter DOWN command

Enter LEFT command

Enter RIGHT command

Enter RECORD command

Enter SAVE command

Enter LOAD command

Enter NEW GAME command

Enter your command

Enter your command

0 ent	101	0	102	0
1	0	103	103	0
1	0	102	1	0
0	0	0	0	0 ext

Enter your command

Non traversable for enemy

0 ent	0	101	0	102
1	0	103	103	0
1	0	102	1	0
0	0	0	0	0 ext

Enter your command

Non traversable for enemy

0 ent	0	0	101	102
1	0	103	103	0
1	0	102	1	0
0	0	0	0	0 ext

Enter your command

You have been damaged

BATTLE: Damage: 10

MOVEMENT: Player's current health: 90

BATTLE: Emeny is dead.

Non traversable for enemy

```

Non traversable for enemy
0 ent    0      0      0      101
1        0      103    103    0
1        0      102    1      0
0        0      0      0      0 ext

```

Enter your command

Enter game save file name

i

```

0 ent    0      0      0      101
1        0      103    103    0
1        0      102    1      0
0        0      0      0      0 ext

```

Enter your command

Non traversable for enemy

```

0 ent    0      0      0      101
1        0      103    103    0
1        0      102    1      0
0        0      0      0      0 ext

```

Enter your command

Non traversable for enemy

```

0 ent    0      0      0      101
1        0      103    103    0
1        0      102    1      0
0        0      0      0      0 ext

```

Enter your command

Non traversable for enemy

```

0 ent    0      0      101    0
1        0      103    103    0
1        0      102    1      0
0        0      0      0      0 ext

```

Enter your command

Enter your command

Enter game save file name

c

0 ent	0	0	101	0
1	0	103	103	0
1	0	102	1	0
0	0	0	0	0 ext

Enter your command

C:\Users\Xenia\Desktop\prog\OOP\oop\build>"C:\Users\Xenia\Desktop\prog

```

\OOP\oop\build\oop_lab.exe"
Enter ESC command
Enter UP command
Enter DOWN command
Enter LEFT command
Enter RIGHT command
Enter RECORD command
Enter SAVE command
Enter LOAD command
Enter NEW GAME command
Enter your command
Enter game save file name
incorrect
Save file verison is not correct
Enter your command
Enter your command
Enter game save file name
correct
DESERIALIZED FIELD
0 ent    0      0      0      101
1        0      103    103    0
1        0      102    1      0
0        0      0      0      0 ext

Enter your command
Non traversable for enemy
0 ent    0      0      0      101
1        0      103    103    0
1        0      102    1      0
0        0      0      0      0 ext

Enter your command
Non traversable for enemy
0 ent    0      0      0      0
1        0      103    103    101
1        0      102    1      0
0        0      0      0      0 ext

Enter your command

```



```
Non traversable for enemy
0 ent    0        0        0        0
1        0        103      103      0
1        0        102      1        101
0        0        0        0        0 ext
```

Enter your command

Enter game save file name

incorrect

Save file verison is not correct

Вывод.

В ходе работы были написаны классы обработки команд пользователя для данной игры. Сохранение и загрузка игры. Описаны крайние случаи.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Файл LoadException.hpp

```
#pragma once

class LoadException
{
public:
    LoadException(std::string errString) : errString(errString) { }
    virtual ~LoadException() { }

    std::string getErrString() const
    {
        return errString;
    }

private:
    std::string errString;
};
```

ПРИЛОЖЕНИЕ В

ИСХОДНЫЙ КОД ПРОГРАММЫ

Файл ActionProcessor.hpp

```
#pragma once

#include "../Game.hpp"
#include "Command.hpp"
#include "../LoadException.hpp"

class ActionProcessor
{
public:
    ActionProcessor(std::shared_ptr<Game> game);
    ~ActionProcessor();

    std::shared_ptr<Game> game;

    std::list<std::shared_ptr<Command>> commands;
    void processCommand(std::shared_ptr<Command> command);

private:
    std::shared_ptr<Game> createGame();
};
```

ПРИЛОЖЕНИЕ С

ИСХОДНЫЙ КОД ПРОГРАММЫ

Файл ActionProcessor.hpp

```
#pragma once

#include "Game.hpp"
#include "FieldBuilder.hpp"
#include "Cell.hpp"
#include "VictoryCondition/VictoryCondition.hpp"

class FieldBuilder;
class Game;

class Field
{
public:
    Field(Game* parent, unsigned short width, unsigned short length);
    Field(const Field& other);
    Field(Field&& other);
    Field& operator=(const Field& other);
    Field& operator=(Field&& other);
    ~Field();

    /// @brief Возвращает параметр клетки width.
    unsigned short getWidth();
    /// @brief Возвращает параметр клетки heigth.
    unsigned short getHeigth();

    /// @brief Возвращают клетку с указанными параметрами.
    Cell getCell(unsigned short width, unsigned short heigth);

    /// @brief Устанавливает параметры клетки.
    /// @param[in] width Кордината клетки по оси абсцисс.
    /// @param[in] heigth Кордината клетки по оси ординат.
    /// @param[in] entity Определяет, что находится в клетке.
    /// @param[in] special Определяет, является ли клетка входом, выходом или ни
    тем, ни другим.
    /// @return true если клетка существует, false если нет.

    template<class T>
    bool setCellContent(unsigned short width, unsigned short heigth,
        std::shared_ptr<T> cellContent)
    {
        field_contents[width][heigth].setCellContent(cellContent);
        return true;
    }

    /// @brief Вывод поля на экран.
```

```

void printField();

void serialize(std::ofstream& ofs);
void serializeCell(Cell& cell, std::ofstream& ofs);
void deserialize(std::ifstream& ifs);
std::shared_ptr<CellContent> deserializeCell(std::ifstream& ifs);

private:
    Point entry;
    Point exit;
    /// @brief Ширина поля.
    unsigned short _width = 0;
    /// @brief Длина поля.
    unsigned short _height = 0;

    unsigned short _widthMaxIndex = 0;
    unsigned short _heightMaxIndex = 0;

    /// @brief Указатель на указатель для хранения клеток поля.
    Cell** field_contents = nullptr;

    void buildField();

    /// @brief Проверка параметров клетки.
    bool newCellParametersIsValid(Entity entity, Special special);
    /// @brief Удаление клеток поля.
    void deleteFieldCells();
    /// @brief Копирование параметров клетки.
    void copyStackValues(Field* destination, const Field& source);

    std::unique_ptr<FieldBuilder> fieldBuilder;

    Game* parent;

    friend class FieldBuilder;
    friend class Game;
};

```