**Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное**

**учреждение высшего образования**

**ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Физико-технический институт**

Дисциплина «Технология программирования»

Отчет по командному проекту:

«Написание 2D-игры на языке С++ с использованием библиотеки SFML»

Выполнили:

Студенты 3 курса ФТИ группы 21317

**Тужикова Елена Эдуардовна**

**Власов Андрей Андреевич**

**Стафеева Надежда Владимировна**

**Комарова Валерия Андреевна**

Проверил:

кандидат физико-математических наук

**Бульба Артем Владимирович**

Петрозаводск 2018

**ЦЕЛЬ РАБОТЫ**

Командный проект по созданию простой 2D игры с использованием SFML (Simple and Fast Multimedia Library) на языке С++ в Visual Studio.

**ЗАДАЧИ РАБОТЫ**

1. Командная работа .
2. Научиться взаимодействовать с командой в системе контроля версий Git, создавая репозиторий, ветки, фиксируя изменения. Объединять отдельные части программы, созданные членами команды, воедино.
3. Научиться строить UML - диаграммы с использованием CASE - системы WhiteStarUML (диаграмма классов (class diagram) и диаграмма вариантов использования (use case diagram) разработанной программы).
4. Написать простую 2D - игру на языке C++ с использованием SFML.
5. Составить отчет по проделанной работе.

**ТРЕБОВАНИЯ К ПРОГРАММЕ**

1. Программа не должна вызывать системных сбоев или зависать.

2. Программа должна освобождать всю выделенную память.

3. Должны отсутствовать конструкции, приводящие к неопределенному поведению.

4. Не должно быть неиспользуемых переменных.

5. Алгоритмы не должны быть избыточны (нет циклов, массивов, ... без которых можно обойтись).

6. Свои классы необходимо реализовывать в отдельных файлах.

7. Необходимо отделять интерфейс от реализации.

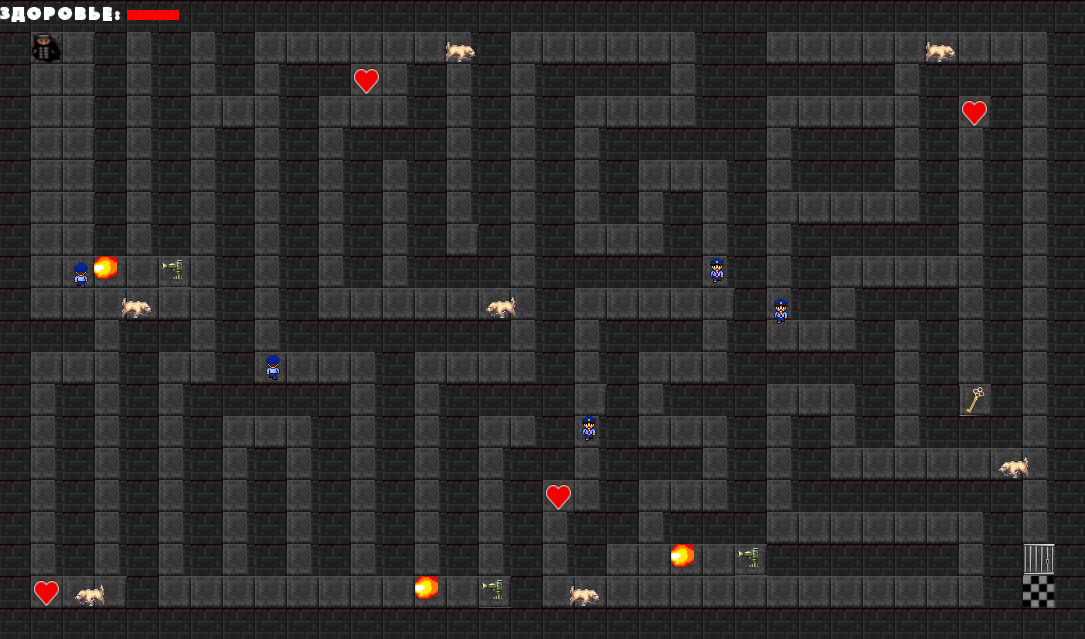
8. Необходимо использовать STL-контейнеры и итераторы.

**СЮЖЕТ ИГРЫ**

Игра “Побег из тюрьмы” - это лабиринт, представляющий собой головоломку, где необходимо проявить свою внимательность.

Главный герой - заключенный, которому необходимо выбраться из заточения. Для успешного финала игроку необходимо найти выход из лабиринта, не попасться на глаза охранником и сторожевым собакам, а а также не умереть от пуль из турелей. Для того, чтобы выход из тюрьмы активировался, герою необходимо найти специальный ключ, которой расположен на карте. В случае, если игрок встретился “лицом к лицу” с охранником, собакой, либо же пулей, то происходит потеря некоторой части жизней. Пополнить их можно с помощью специальных сердечек, которые расположены на карте.

При потере всех жизней игра заканчивается.



**ИСПОЛЬЗОВАННЫЕ ПРОГРАММНЫЕ СРЕДСТВА**

Каждый из членов команды использовал среду разработки приложений Visual Studio 2017 с подключенной библиотекой SFML последней версии.

**СТРУКТУРА ПРОГРАММЫ**

Код программы написан на языке С++ и удовлетворяет принципам объектно-ориентированного подхода (инкапсуляция, наследование, полиморфизм), присутствует раздельная компиляция.

**Заголовочные файлы программы:**

1. **<SFML/Graphics.hpp>** - работа с графикой. SFML.
2. **<SFML/Audio.hpp>** - работа с аудио. SFML
3. **"map.h" -** интерфейс класса карты.(map.cpp)
4. **"globals.h" -** обеспечение взаимодействия объектов с картой, вызов из других классов. Содержит extern и static const.
5. **"player.h" -** интерфейсная часть класса игрока.(player.cpp)
6. **"Enemy.h" -** интерфейсная часть классов врага (охраны и сторожевых собак). (Enemy.cpp)
7. **"Pushka.h"** - интерфейсная часть классов стреляющей пушки и пуль.(Pushka.cpp)

**ГЛАВНЫЙ ФАЙЛ ИСХОДНОГО КОДА**

Главный файл исходного кода(main.cpp) включает в себя:

1. Подключение необходимых заголовочных файлов:

#include <iostream>

#include <SFML/Graphics.hpp>

#include <SFML/Audio.hpp>

#include "map.h"

#include "globals.h"

#include "player.h"

#include "Enemy.h"

#include <list>

#include "Pushka.h"

1. Меню, звуковое сопровождение (строки 14-84).
2. Объекты соответствующих классов:

Map drawing;

Player z("hero\_d.png", 48, 48, 23.0, 23.0);

Map m;

1. Списки и соответствующие итераторы:

А) охраны

std::list<Entity\*> enemy;

std::list<Entity\*>::iterator ee;

Б) пушек

std::list<Push\*> pus;

std::list<Push\*>::iterator pp;

В) пуль

std::list<Push::Bullet\*> bullets;

std::list<Push::Bullet\*>::iterator bul;

1. Вывод врагов на карту:

for (int i = 0; i < HEIGHT\_MAP; i++) {

for (int j = 0; j < WIDTH\_MAP; j++) {

if (TileMap[i][j] == 'L') {

enemy.push\_back(new Police(j, i));

cout << i << " " << j << endl;

}

if (TileMap[i][j] == 'S') {

enemy.push\_back(new Dogs(j, i));

}

if (TileMap[i][j] == 'P') {

pus.push\_back(new Pushka(j, i));

}

}

}

1. Функции прорисовки и взаимодействия объектов:

m.drawing();

z.draw\_p();

for (ee = enemy.begin(); ee != enemy.end(); ee++) {

(\*ee)->drawing(z.GetPlayerCoordinateX(), z.GetPlayerCoordinateY(), z);

}

for (pp = pus.begin(); pp != pus.end(); pp++) {

(\*pp)->drawing();

if ((\*pp)->shoot > 300) {

(\*pp)->shoot = 0;

FloatRect polozh = (\*pp)->polozenie();

float uskor = (\*pp)->uskorenie();

bullets.push\_back(new Push::Bullet(polozh, uskor));

}

(\*pp)->shoot++;

}

for (bul = bullets.begin(); bul != bullets.end(); bul++) {

if ((\*bul)->life) {

(\*bul)->WithPlayer(z);

(\*bul)->drawing();

}

1. Удаление пули при взаимодействии со стеной/игроком:

else {

delete((\*bul));

bullets.remove(\*bul);

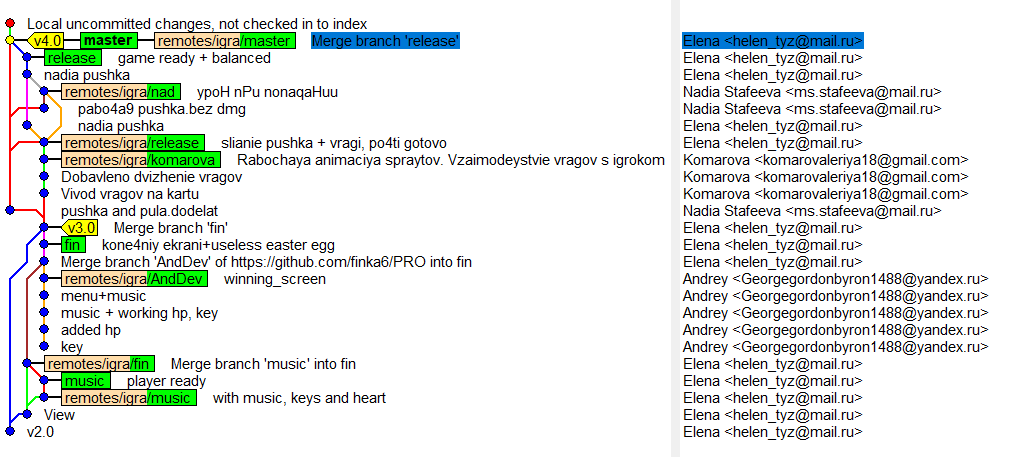
break;

}

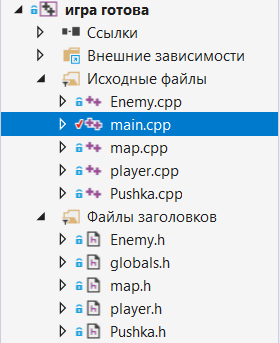
}

**ЗАКЛЮЧЕНИЕ**

При командной работе была создана простая 2D-игра в жанре головоломка на языке С++ в среде разработке приложений Visual Studio с использованием SFML библиотек.

В процессе создания проекта каждый из создателей использовал систему контроля версий Git. 

При создании программы все участники команды руководствовались принципами ООП и раздельной компиляцией.



В коде программы реализована очистка памяти, отсутствуют неиспользуемые переменные.

Были использованы STL-контейнеры и итераторы.

Программа удовлетворяет поставленным требованиям к проекту.

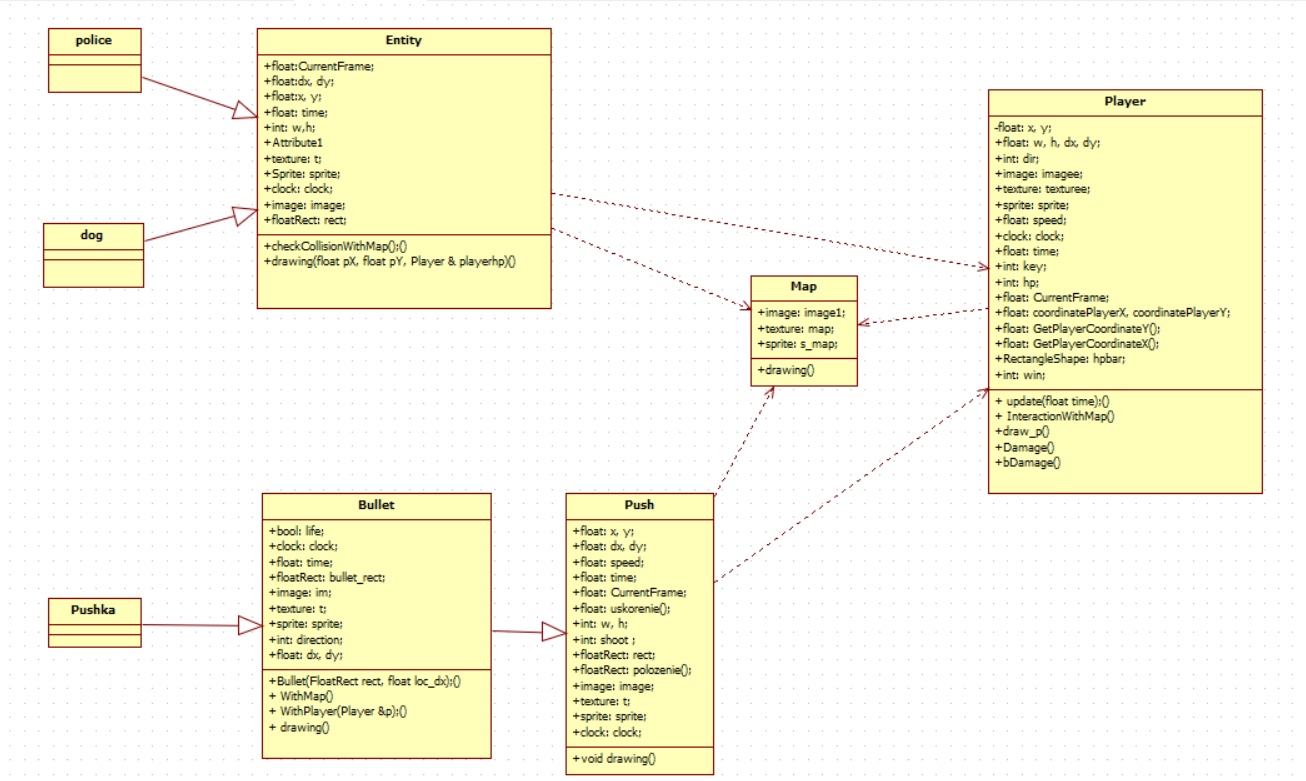
Сбоев или зависаний при запуске программы не выявлено.

Цель работы достигнута – игра исправно работает.

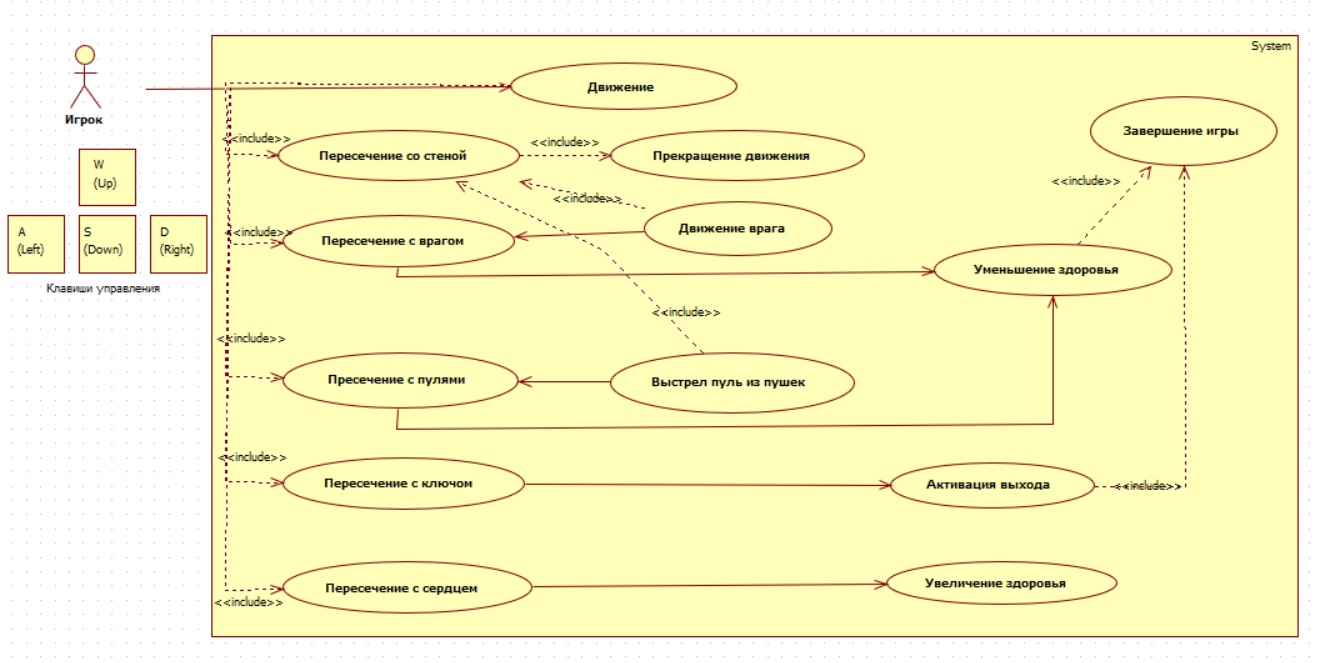
К отчету прилагаются UML-диаграммы (UML-диаграмма классов (class diagram) **(Приложение1)** и диаграмма вариантов использования (use case diagram) **(Приложение 2),** листинг главного кода программы main.cpp **(Приложение 3).**

ПРИЛОЖЕНИЯ

**Приложение 1.** UML-диаграмма классов



**Приложение 2.** Диаграмма вариантов использования



**Приложение 3.** Листинг главного кода программы main.cpp

#include <iostream>

#include <SFML/Graphics.hpp>

#include <SFML/Audio.hpp>

#include "map.h"

#include "globals.h"

#include "player.h"

#include "Enemy.h"

#include <list>

#include "Pushka.h"

using namespace std;

using namespace sf;

void menu(RenderWindow & window) {

Texture menuTexture1, menuTexture2, menuTexture3, aboutTexture, menuBackground;

menuTexture1.loadFromFile("images/play.png");

menuTexture2.loadFromFile("images/authors.png");

menuTexture3.loadFromFile("images/exit.png");

aboutTexture.loadFromFile("images/about.png");

menuBackground.loadFromFile("images/back.jpg");

Sprite menu1(menuTexture1), menu2(menuTexture2), menu3(menuTexture3), about(aboutTexture), menuBg(menuBackground);

bool isMenu = 1;

int menuNum = 0;

menu1.setPosition(50, 30);

menu2.setPosition(50, 130);

menu3.setPosition(50, 230);

menuBg.setPosition(0, 0);

Music prison;

prison.openFromFile("audio/prison.ogg");

prison.setVolume(70);

prison.play();

prison.setLoop(true);

//////////////////////////////МЕНЮ///////////////////

while (isMenu)

{

menu1.setColor(Color::White);

menu2.setColor(Color::White);

menu3.setColor(Color::White);

menuNum = 0;

window.clear(Color(129, 181, 221));

if (IntRect(50, 30, 350, 100).contains(Mouse::getPosition(window))) { menu1.setColor(Color::Yellow); menuNum = 1; }

if (IntRect(50, 130, 350, 100).contains(Mouse::getPosition(window))) { menu2.setColor(Color::Yellow); menuNum = 2; }

if (IntRect(50, 230, 350, 100).contains(Mouse::getPosition(window))) { menu3.setColor(Color::Yellow); menuNum = 3; }

if (Mouse::isButtonPressed(Mouse::Left))

{

if (menuNum == 1) isMenu = false;//если нажали первую кнопку, то выходим из меню

if (menuNum == 2) { window.draw(about); window.display(); while (!Mouse::isButtonPressed(Mouse::Right)); }

if (menuNum == 3) { window.close(); isMenu = false; }

}

window.draw(menuBg);

window.draw(menu1);

window.draw(menu2);

window.draw(menu3);

window.display();

}

}

void winner(RenderWindow & window)

{

Texture winning\_screen;

winning\_screen.loadFromFile("images/pobeg.jpg");

Sprite winscreen(winning\_screen);

winscreen.setPosition(0, 0);

window.draw(winscreen);

window.display();

}

void lose(RenderWindow & window)

{

Texture losee;

losee.loadFromFile("images/porazh.jpg");

Sprite lose(losee);

lose.setPosition(0, 0);

window.draw(lose);

window.display();

}

int main()

{

menu(window);//вызов меню

Map drawing;

Player z("hero\_d.png", 48, 48, 23.0, 23.0);

Map m;

Clock clock;

std::list<Entity\*> enemy;

std::list<Entity\*>::iterator ee;

std::list<Push\*> pus;

std::list<Push\*>::iterator pp;

std::list<Push::Bullet\*> bullets;

std::list<Push::Bullet\*>::iterator bul;

for (int i = 0; i < HEIGHT\_MAP; i++) {

for (int j = 0; j < WIDTH\_MAP; j++) {

if (TileMap[i][j] == 'L') {

enemy.push\_back(new Police(j, i));

cout << i << " " << j << endl;

}

if (TileMap[i][j] == 'S') {

enemy.push\_back(new Dogs(j, i));

}

if (TileMap[i][j] == 'P') {

pus.push\_back(new Pushka(j, i));

}

}

}

Music music;

music.openFromFile("audio/ost.ogg");

music.setVolume(40);

music.play();

music.setLoop(true);

Font font;

font.loadFromFile("Ben\_Krush.ttf");

Text text("", font, 20);

text.setStyle(sf::Text::Bold);

while (window.isOpen())

{

if (z.win >= 1)

{

music.stop();

Music pobeda;

pobeda.openFromFile("audio/pobeda.ogg");

pobeda.setVolume(100);

pobeda.play();

window.close();

RenderWindow window(sf::VideoMode(1400, 700), "GAME");

winner(window); while (!Mouse::isButtonPressed(Mouse::Right));

}

if (z.hp <= 0)

{

music.stop();

Music lost;

lost.openFromFile("audio/lose.ogg");

lost.setVolume(100);

lost.play();

window.close();

RenderWindow window(sf::VideoMode(1400, 700), "GAME");

lose(window); while (!Mouse::isButtonPressed(Mouse::Right));

}

sf::Event event;

while (window.pollEvent(event))

{

if (event.type == sf::Event::Closed)

window.close();

}

window.clear(Color(33, 30, 30));

m.drawing();

z.draw\_p();

for (ee = enemy.begin(); ee != enemy.end(); ee++) {

(\*ee)->drawing(z.GetPlayerCoordinateX(), z.GetPlayerCoordinateY(), z);

}

for (pp = pus.begin(); pp != pus.end(); pp++) {

(\*pp)->drawing();

if ((\*pp)->shoot > 300) {

(\*pp)->shoot = 0;

FloatRect polozh = (\*pp)->polozenie();

float uskor = (\*pp)->uskorenie();

bullets.push\_back(new Push::Bullet(polozh, uskor));

}

(\*pp)->shoot++;

}

for (bul = bullets.begin(); bul != bullets.end(); bul++) {

if ((\*bul)->life) {

(\*bul)->WithPlayer(z);

(\*bul)->drawing();

}

else {

delete((\*bul));

bullets.remove(\*bul);

break;

}

}

text.setString("Здоровье:");

text.setPosition(0, 0);

window.draw(text);

window.draw(z.hpbar);

window.display();

}

return 0;

}