**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Физико-технический институт**

**Командное задание в рамках курсового проекта**

**«Разработка игры «Space invaders»**

Выполнил:

Студент

Физико-технического

Института группы 21318

Шимозёров Илья

Преподаватель:

Бульба Артем Владимирович

Петрозаводск

2023

**Цель работы**

Разработать аналог игры «space invaiders». Вести разработку с помощью системы контроля версий Git через веб-сервис GitHub. Закрепить полученные знания, использования языков С++ с использованием библиотеки SFML и UML.

**Задачи**

1. Познакомится с принципами работы в команде.
2. Научится использовать при работе систему контроля версии Git.
3. Научится строить UML-диаграммы.
4. Научится использовать SFML библиотеки в программной среде С++.
5. Написание отчёта о проделанной работе.

**Кратко о программной реализации**

Программная реализация выполнена на языке c++ с использованием библиотеки SFML, среда разработки – IDE Qt Creator (Qt 5.4.2).

**Единицы компиляции**

* Game.h
* Game.cpp
* Main.cpp

**Варианты использования**

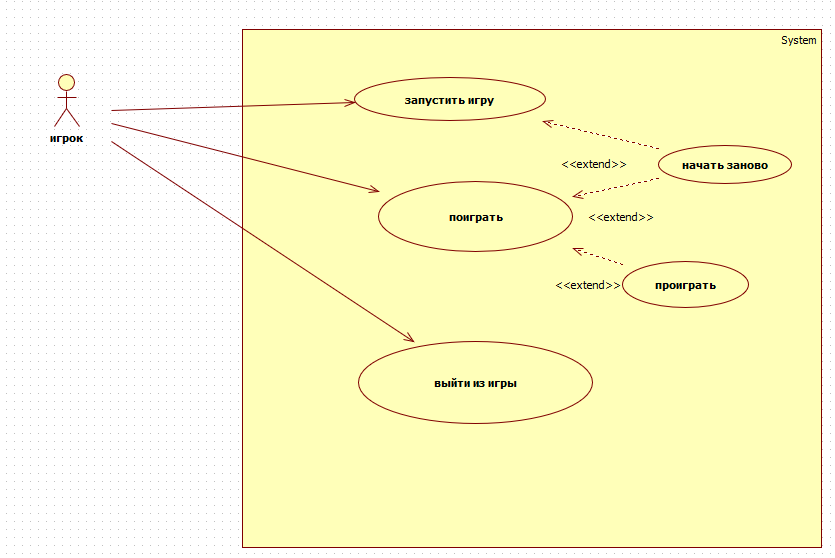


Рис1. Диаграмма вариантов использования

Запустить игру

Игрок запускает игру

Поиграть

С помощью стрелок игрок передвигает самолетик, уворачиваясь от пуль противника, а также стреляет в ответ.

Выйти из игры

Игрок выходит из игры

Проиграть

Если игрок не сможет увернутся от вражеской пули, игроку покажут его счет

Начать заново

При желании, игрок может начать игру заново сбросив свой счет и восстановив убитых врагов

**Заголовочный файл**

Game.h

#ifndef GAME

#define GAME

#include <iostream>

#include <SFML/Graphics.hpp>

#include <random>

using namespace sf;

#include "globals.h"

#include "player.h"

#include "enemies.h"

class Game{

private:

Font font;

Text\* text;

std::mt19937 engine;

Texture background\_texture;

Sprite background\_sprite;

Clock clock;

RenderWindow\* window;

Enemies\* enemies;

Player\* player;

public:

Game(RenderWindow\* wind);

void start();

};

#endif // GAME

**Исполнительные файлы**

Game.cpp

#include "game.h"

Game::Game(RenderWindow\* wind):window(wind){

font.loadFromFile(TIMES\_NEW\_ROMAN);

text=new Text("", font, 16);

text->setColor(Color::Blue);

engine.seed(std::time(nullptr));

background\_texture.loadFromFile(BACKGROUND);

background\_sprite.setTexture(background\_texture);

enemies=new Enemies(window);

Image player\_img;

player\_img.loadFromFile(PLAYER\_IMG);

player=new Player(player\_img, 0,SCREEN\_HEIGHT-BASE\_SIZE-1, BASE\_SIZE, BASE\_SIZE, window, enemies->get\_bullet\_list());

}

void Game::start(){

while (window->isOpen()){

window->clear(); //РћС‡РёС‰Р°РµРј СЌРєСЂР°РЅ

sf::Event event; //РџРµСЂРµРјРµРЅРЅР°СЏ РґР»СЏ СЃРѕР±С‹С‚РёСЏ

while (window->pollEvent(event)) //РћРїСЂРѕСЃ СЃРѕР±С‹С‚РёР№

{

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

window->close();//Р—Р°РєСЂС‹РІР°РµРј РѕРєРЅРѕ, РµСЃР»Рё СЃРѕР±С‹С‚РёРµ вЂњClosedвЂќ

}

if(Keyboard::isKeyPressed(Keyboard::Escape)){

window->close();

}

if(Keyboard::isKeyPressed(Keyboard::Space)){

enemies->reset();

player->reset(false);

}else{

float time = clock.getElapsedTime().asMicroseconds();

clock.restart();

time = time / 800;

if (player->get\_life()&& !(enemies->get\_below())){

if(enemies->get\_enemies\_size()==0){

player->reset(true);

enemies->reset();

}

window->draw(background\_sprite);

enemies->update(time, engine,player->get\_player\_bullet\_list());

player->*update*(time);

text->setPosition(0, -5);

text->setString("score:"+player->get\_score());

window->draw(\*text);

}else{

text->setPosition(0, -5);

text->setString("score:"+player->get\_score());

text->setPosition(60, 60);

if (!player->get\_life()){

text->setString("Game Over(you were killed)");

}else{

text->setString("Game Over(enemies reached you)");

}

window->draw(\*text);

text->setPosition(110, 77);

text->setString("Your score:"+player->get\_score());

window->draw(\*text);

text->setPosition(90, 93);

text->setString("Press 'space' to start");

window->draw(\*text);

}

}

window->display(); //РћС‚РѕР±СЂР°Р¶Р°РµРј РєСЂСѓРі РЅР° СЌРєСЂР°РЅ

}

}

Main.cpp

#include "game.h"

int main()

{

RenderWindow window(VideoMode(SCREEN\_RESIZE \* SCREEN\_WIDTH, SCREEN\_RESIZE \* SCREEN\_HEIGHT), "Space Invaders", Style::Close);

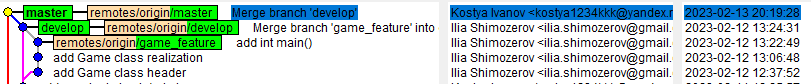
window.setView(View(FloatRect(0, 0, SCREEN\_WIDTH, SCREEN\_HEIGHT)));

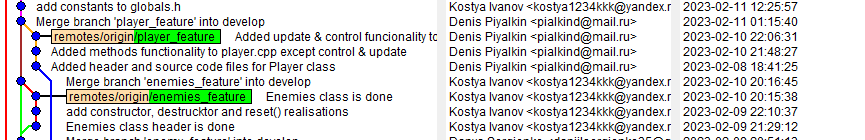
Game game(&window);

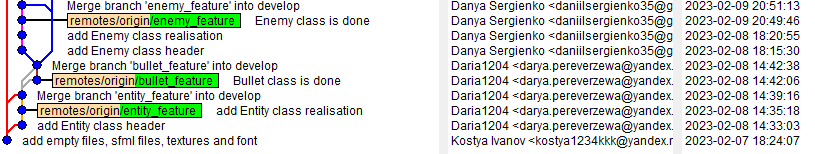
game.start();

}

**История проекта Github**







**История моих коммитов**

* **add Game class header**

Добавлен готовый заголовочный файл класса Game.

* **add Game class realization**

Добавлена реализация методов.

* **add int main()**

Добавлена функция main().

* **Merge branch 'game\_feature' into develop**

Слияние ветки game\_feature в ветку develop

**Вывод**

В ходе выполнения командного задания в рамках курсового проекта использовалась среда разработки Qt язык C++ с использованием библиотеки SFML. Применялась система контроля версий git. Реализованы все прецеденты. В ходе тестирования игры зависания или сбои не были обнаружены. Для классов Player и Enemies, хранящих в себе списки пуль, была реализована очистка памяти в их деструкторах. Неиспользуемые переменные в ходе разработки старались удалять. В алгоритмы методов старались не добавлять не нужных циклов, массивов и др. Мы считаем, что наша команда достигла поставленных целей.