Министерство науки и высшего образования Российской Федерации
Пензенский государственный университет
Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

по курсу «Программирование»

на тему «Разработка программы сложной структуры методом нисходящего программирования»

Выполнил
студент группы 22ВОА2
Маслов Ю.А.

Принял
Митрохин М. А.

хор. 28.12.23

Пенза 2023

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет Вычислительной техники
Кафедра «Вычислительная техника»

«УТВЕРЖДАЮ»
Зав. кафедрой ВТ
Митрохин М. А.
«____» _____ 2023 г

ЗАДАНИЕ
на курсовое проектирование по курсу

Программирование
Студенту Маслову Юрию Андреевичу Группа 22ВОА2
Тема проекта: Разработка программы сложной структуры методом нисходящего программирования

Исходные данные (технические требования) на проектирование
Наименование программы: Компьютерная игра «Охота на птиц».

База данных включает следующие поля:
Название, дата выхода, разработчик, издатель, количество очков.

Задание. Программа должна выполнять следующие функции:
- Набор очков посредством попадания по целям.
- вывод количества очков.
- Меню.
- Магазин.

Обязательные требования к программе:
1. Многомодульность.
2. Использование сложных типов данных — списков, структур, файлов.
3. Режим работы видеосистемы — текстовый/графический.
4. Устройство ввода информации — клавиатура, мышь.
5. Пользовательский интерфейс должен быть построен на основе меню и панели инструментов.
6. Наличие заставки.
7. Операционная система — Windows.
8. Система программирования — C/C++ на базе среды Microsoft Visual Studio 2019.
9. Язык программирования — C++.

Объем работы по курсу

1. Расчетная часть

Разработка программы.

2. Графическая часть

Структура модулей программы, схема алгоритма программы.

3. Экспериментальная часть

Отладка программы.

Срок выполнения проекта по разделам

1. В соответствии с графиком выполнения курсового проекта.
2.
3.

Дата выдачи задания «9» сентября 2023 г

Дата защиты проекта «____» _____ 2023 г

Руководитель _____

Задание получил Маслов Ю.А. « 9 » сентября 2023 г

Студент Маслов Юрий Андреевич

Содержание

Введение 5
1. Постановка задачи 6
2. Выбор решения 7
  2.1 Определение необходимых модулей программы 7
  2.2 Определение алгоритма программы 9
3. Описание разработки программы 12
4. Отладка и тестирование 13
5. Описание программы 15
6. Руководство пользователя 15
Заключение 188
Список использованных источников 19
Приложение А Листинг программы 20
  Приложение А.1 Исходный файл «Main.cpp» 20
  Приложение А.2 Исходный файл «Game.cpp» 22
  Приложение А.3 Исходный файл «Settings.cpp» 26
  Приложение А.4 Заголовочный файл «Game1.h» 30
  Приложение А.4 Заголовочный файл «Settings.h» 30
Приложение Б – Результаты работы программы (снимки экрана) 31

Введение

С развитием технологий компьютеры все больше и больше вливаются в повседневную жизнь человека. Персональные компьютеры - это уже не просто машины для математических вычислений, их общедоступность значительно расширила область применения. Это позволило развиться такому виду развлечений как видеоигры. Они имеют большое значение для людей, на ряду с кинематографом и другими развлечениями. Они помогают людям узнать для себя что-то новое, расслабится и весело провести время и просто дарят массу положительных эмоций. Как культура, игры не так давно обосновались, но плотно «врезались» в общество, уже каждый, по крайней мере, 3 игрок ассоциирует видеоигру, как маленькое произведение искусства. Целю этой курсовой работы является создание игры жанра аркада, а также получение практических навыков написания и разработки приложений с использование программной среды разработки Microsoft Visual Studio 2022.

1 Постановка задачи

Необходимо разработать игру «Bird hunting». В игре будут реализованы следующие правила:

1. При запуске игры в окне с определенным интервалом появляются птицы летящие горизонтально.
2. При нажатии на них левой кнопкой мыши птица исчезает и воспроизводится характерный звук выстрела.
3. За каждую сбитую птицу пользователю будут начисляться очки.
4. Основной целью игры является набор большого количества очков.

Необходимо также реализовать меню, с помощью которого можно будет запустить игру и выйти из приложения, а также магазина, в котором можно приобретать за очки разные модельки птиц.

Необходимо реализовать минималистичный и удобный интерфейс, чтобы у пользователя не возникало затруднений при использовании программы. В программе также будут использоваться сложные типы данных. Устройством управления игрой является компьютерная мышь. Все действия в игре должны осуществляться с помощью нажатия левой клавиши мыши.

2 Выбор решения

2.1 Определение необходимых модулей программы

Разработанная программа состоит из трёх модулей:

1. Main.cpp
2. Settings.cpp
3. Game.cpp
4. Game1.h
5. Settings.h

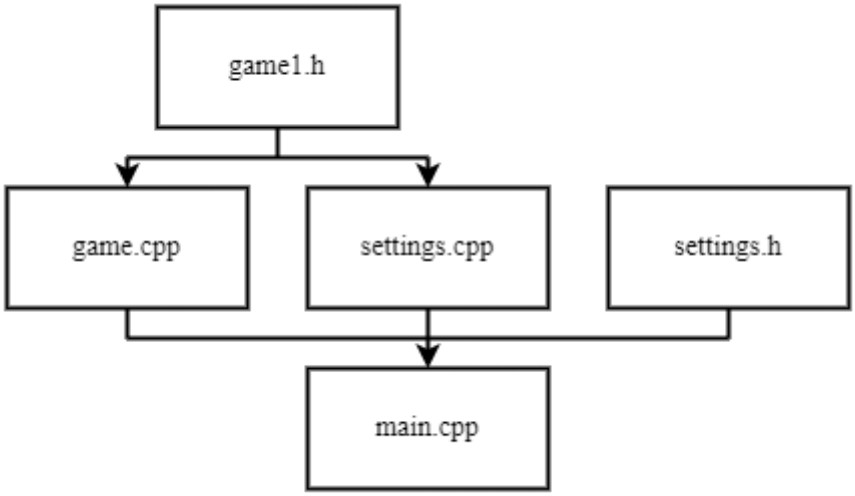
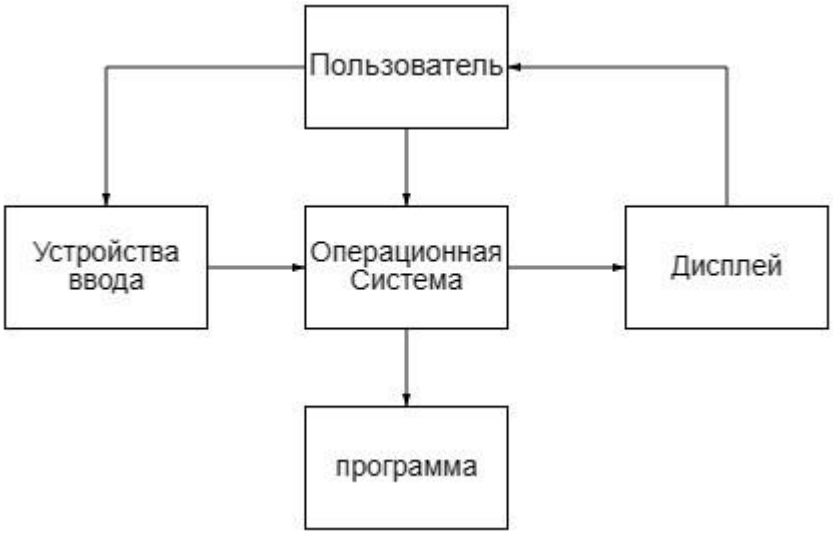


Рисунок 1 — Схема взаимодействия модулей

1. Game1.h – заголовочный файл блока Game.cpp.
2. Settings.h – заголовочный файл блока Settings.cpp.
3. Game.cpp – модуль в котором реализована игра.
4. Settings.cpp – модуль в котором реализован магазин.
5. Main.cpp – главный модуль программы. Именно здесь происходит генерация меню игры. В этом модуле осуществляется запуск игрового процесса и магазина.

Ниже на рисунке 2 представлена схема данных программы

Рисунок 2 — Схема данных



2.2 Определение алгоритма программы

Алгоритм игрового процесса основан на простейших операциях и функциях:

1) Загрузить и включить меню при запуске игры;

2) Загрузка картинок и звука попадания.

3) Обработка нажатия пользователя на птицу;

Для представления принципа работы уровня игры ниже приведена его функциональная схема (рис.3).

Рисунок 3 — алгоритм, реализующий принцип работы игры

Алгоритм проверки на нажатие клавиш движения основан на простом условии, если нажата кнопка мыши и курсор находится на птице, то птица пропадает и счет увеличивается. Алгоритм движения представлен на (рис.4).

Рисунок 4 — алгоритм проверки на нажатия клавиш движения

Алгоритм работы звуков попадания и анимации движения птиц очень простой и состоит из проверки условий работы игры, если игра запущена, то генерируются птицы, если игрок нажал на птицу проигрывается звук(рис.5).

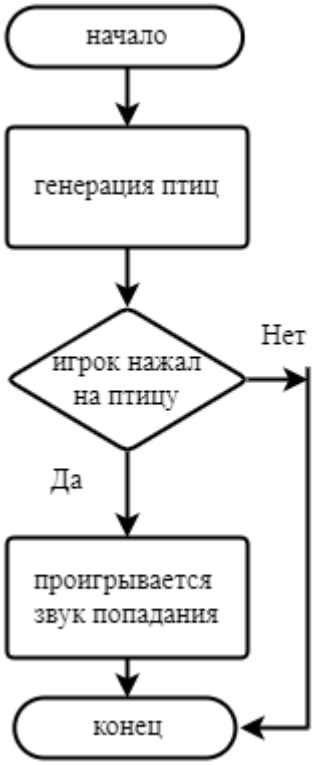


Рисунок 5 — алгоритм музыки и анимации

3 Описание разработки программы

Для написания программы был использован язык программирования C++ а также библиотека SFML, с помощью которого очень удобно спроектировать исполнение игры охота на птиц и обеспечить быстродействие игры. В качестве среды разработки была использована программа Microsoft Visual Studio 2022.

Для начала были разработаны меню, магазин, поле, движок игры «Охота на птиц». Для работы используется модуль Main.cpp. Здесь пользователь может запустить игру и закрыть её.

При выборе элемента меню «Начать игру» начинаться сам процесс игры. Запускается игровая сессия. Проектирование игрового поля и птиц происходит с помощью методов из модулей Game.cpp и Game1.h. С помощью библиотеки SFMLGraphics мы выбираем оформление игры.

При выборе элемента меню «Настройки» открывается окно магазина, в котором пользователь может за накопленные очки приобрести различные вариации моделек птиц. Проектирование поля настроек происходит с помощью методов и модулей game1.h и settings.cpp.

В процессе, когда игрок будет попадать по птицам, он зарабатывает очки, которые равны 10 за птицу. Информация о счёте выводиться на экран.

Для анализа программы необходимо выявить ее поведение (табл.1):

Таблица 1 — Анализ программы

Описание	Результат
Запуск программы	Включается меню с выбором пунктов.
Появление птиц	Появление птиц в случайных местах левой границы поля после запуска игры.
Проверка на попадание	Вывод количества очков в случае попадания.

4 Отладка и тестирование

Средой разработки была выбрана программа VisualStudio 2022. Программа содержит в себе все необходимые средства для успешной отладки и тестирования модулей программы, а также всей программы в целом. Данная среда разработки выпущена относительно недавно, что позволяет проводить необходимые операции над разработанной программой наиболее производительно и эффективно. Для выполнения отладки применялись средства Visual Studio, такие как: точки останова, анализ содержимого переменных, выполнение кода по шагам (по строкам), анализ памяти. В данном случае точка остановки была установлена после нажатия на птицу, что позволяет проверить получение очков счёта, представлен на (рис.6).

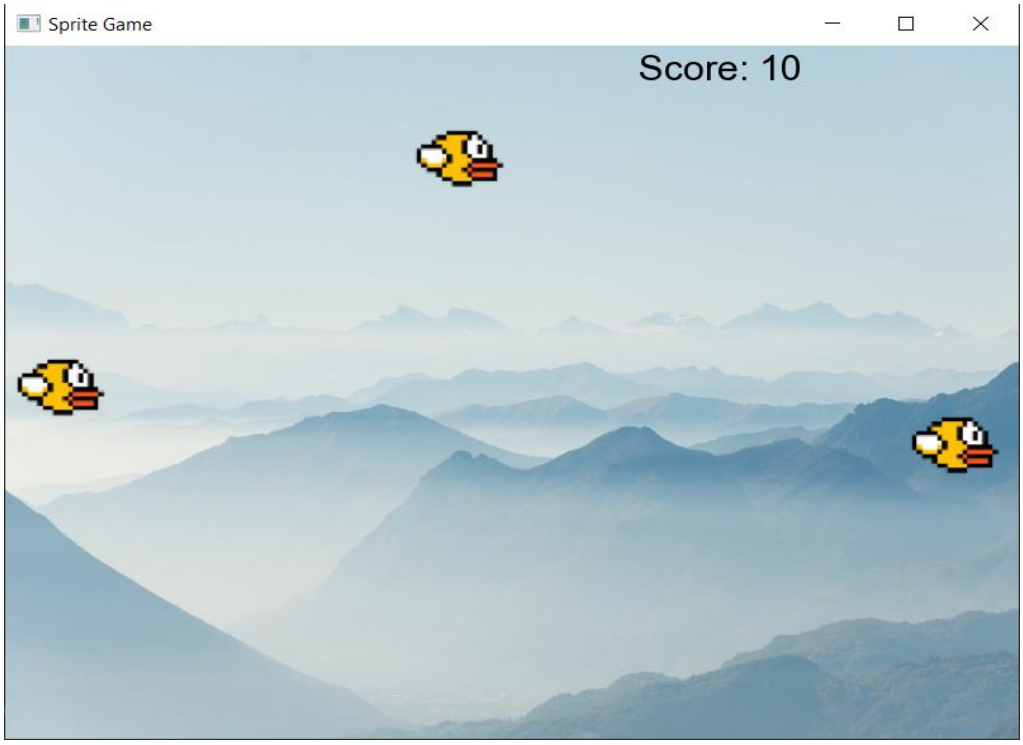


Рисунок 6 — Отладка программы

Тестирование проводилось в рабочем порядке, на всех этапах разработки программы, в том числе после завершения написания кода. Тестирование позволило выявить и исправить множество проблем и недочетов.

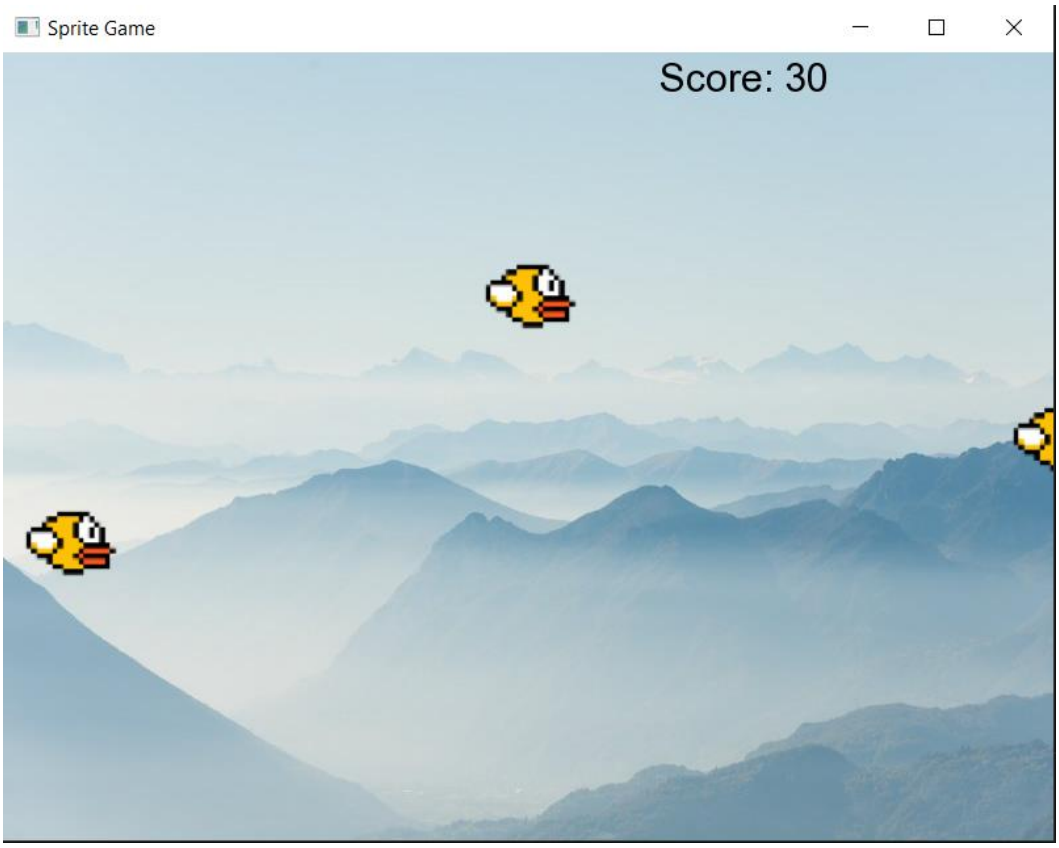


Рисунок 7 — Тестирование программы (появление птиц)

5 Описание программы

Данная программа представляет собой приложение на языке C++, которое позволяет запустить игру «Охота на птиц».

Пользователь может нажать кнопку «Начать игру», после чего ему предоставляется управление игрой. Затем пользователь может начать играть, нажимая клавишу мыши.

После запуска игры происходит появление птиц.

Также пользователь может нажать кнопку «Настройки», после чего ему предоставляется выбор моделей птиц. Пользователь может выбирать их, нажимая клавишу мыши

6 Руководство пользователя

Вид окна меню при запуске приведен на рис 8.

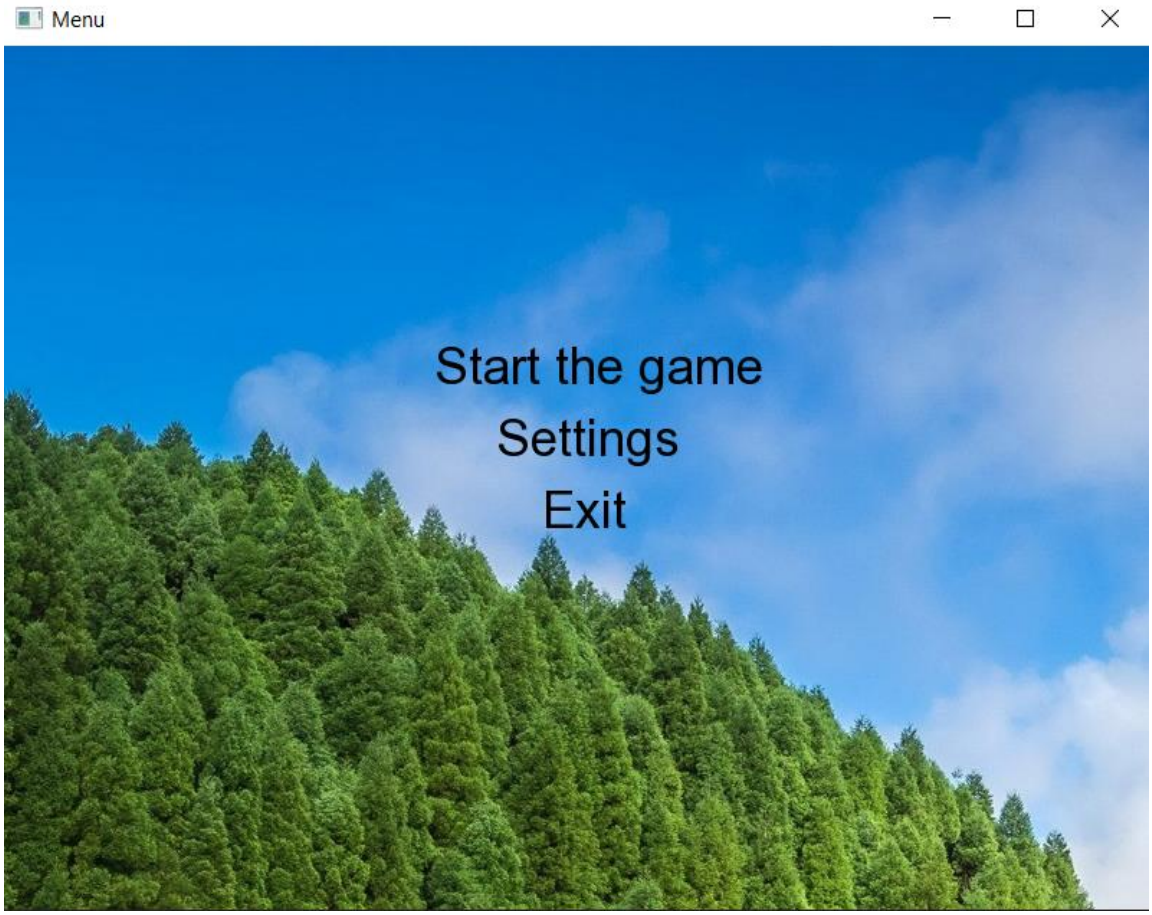


Рисунок 8 — вид окна меню

При выборе действия «Start» – запуститься игра на рис.9.

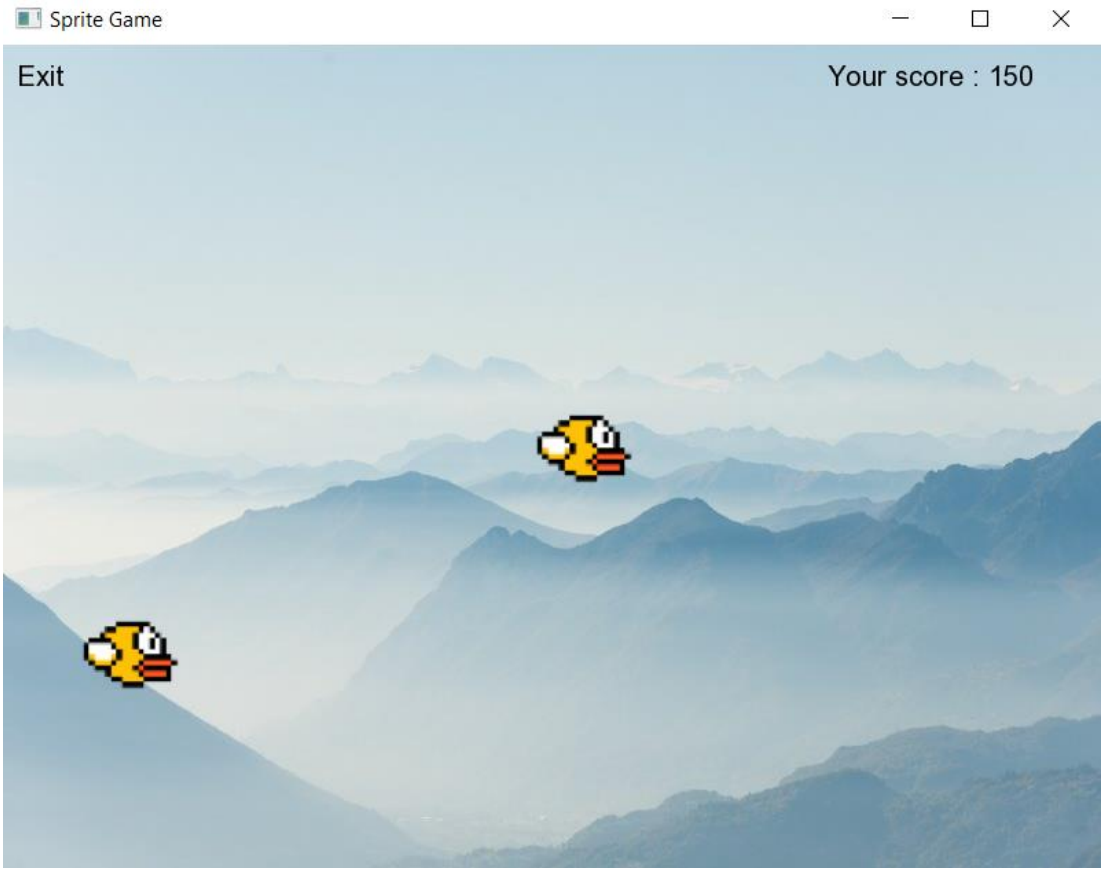


Рисунок 9 – выбор кнопки «Начать игру»

Далее представлено появление очков после попадания на рис.10.

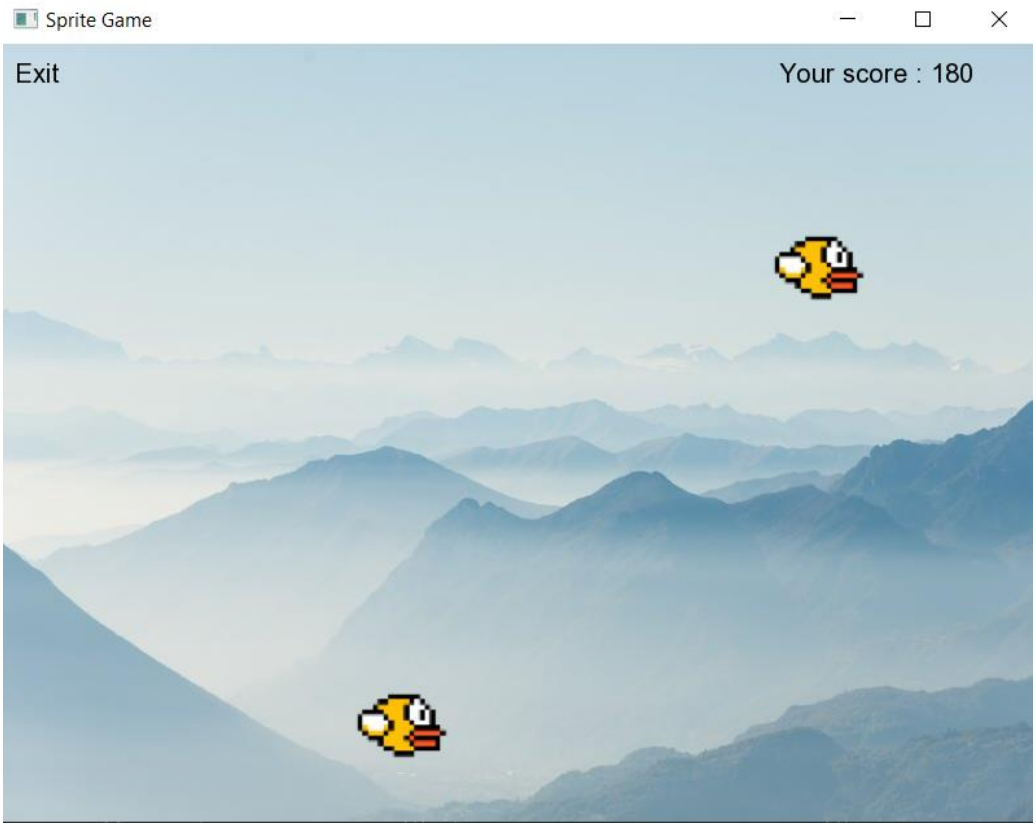


Рисунок 10 — появление очков после попадания

При выборе действия «Settings» – запуститься магазин на рис.11.

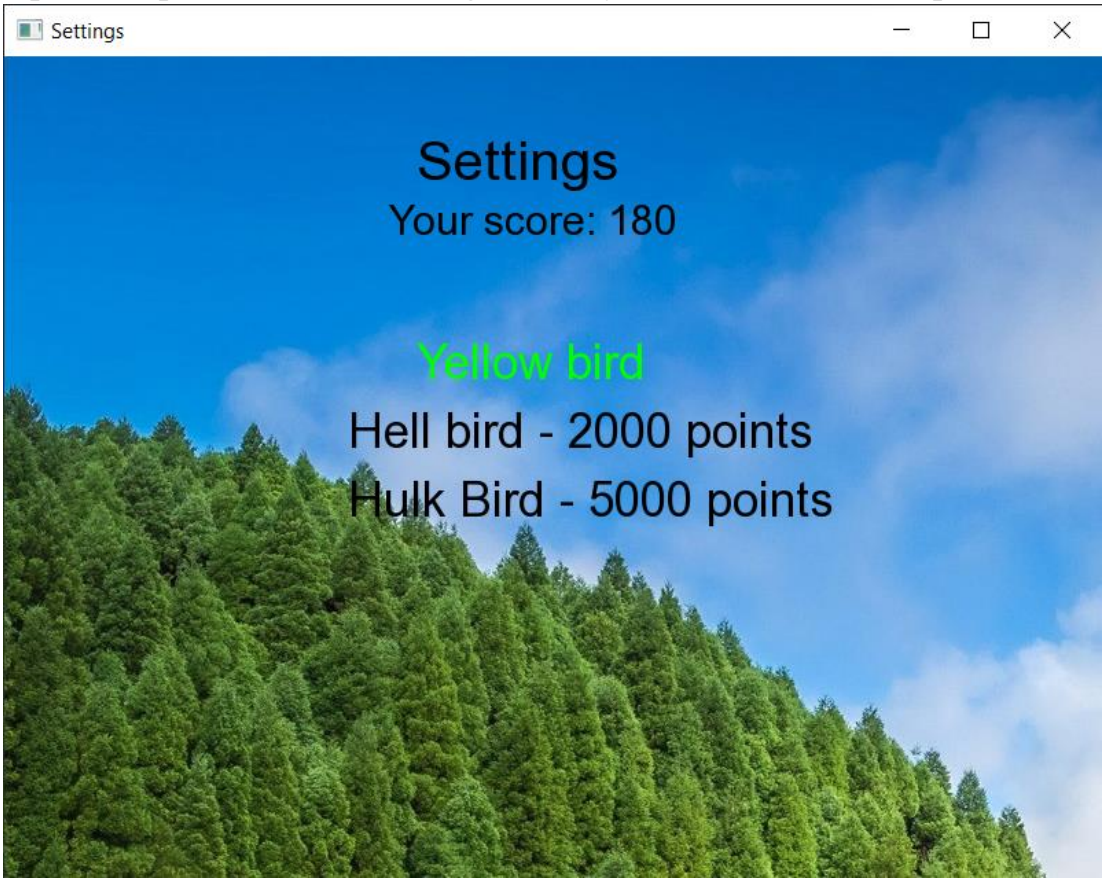


Рисунок 11 — меню магазина.

Заключение

В ходе выполнения курсовой работы были получены и расширены навыки разработки многомодульных программ на языке C++, освоены функции работы с библиотекой SFML, способы интерпретации событий от мыши. Освоены на практике в большом объеме функции для работы с изображениями. Закреплены приемы работы в компиляторе Visual Studio 2022, отточены навыки программирования на языке C++.

В данной курсовой работе была написана программа, используемая для запуска игры «Bird Hunt». Соответственно, программа не требует большого количества системных ресурсов, благодаря, в том числе использованию языка C++, обеспечивает быструю обработку информации.

Программу в будущем можно улучшать, добавлять новые функции. В целом, поставленная в начале курсового проекта цель была достигнута. В программе выполняются все необходимые функциональные требования.

Список использованных источников

1. Электронный ресурс. habr.com Официальный сайт. – URL: https://habr.com/ru/articles/710100/ (дата обращения 19.10.23)
2. Электронный ресурс. sfml-dev.org Официальный сайт. – URL: https://www.sfml-dev.org/tutorials/2.5/ (дата обращения 20.10.23)
3. Электронный ресурс. Learn.microsoft.com Официальный сайт. – URL: https://learn.microsoft.com/ru-ru/cpp/cpp/cpp-language-reference?view=msvc-170 (дата обращения 25.10.23)
4. Электронный ресурс. habr.com Официальный сайт. – URL: https://habr.com/ru/companies/skillfactory/articles/588356/ (дата обращения 28.10.23)
5. Электронный ресурс. habr.com Официальный сайт. – URL: https://habr.com/ru/articles/537682/ (дата обращения 29.10.23)

Приложение А Листинг программы

Приложение А.1 Исходный файл «Main.cpp»

```
#include <SFML/Graphics.hpp>
#include <iostream>
#include "game1.h"
#include "settings.h"

using namespace std;
using namespace sf;

int main()
{
    RenderWindow window(VideoMode(800, 600), "Menu");
    window.setFramerateLimit(60);

    Font font;
    font.loadFromFile("C:/Windows/Fonts/Arial.ttf");

    Text startText("Start the game", font, 35);
    startText.setPosition(300, 200);

    Text settingsText("Settings", font, 35);
    settingsText.setPosition(343, 250);



    Text exitText("Exit", font, 35);
    exitText.setPosition(375, 300);

    bool isStartHighlighted = false;
    bool isSettingsHighlighted = false;
    bool isExitHighlighted = false;

    Texture background;
    background.loadFromFile("images/111.png");
    Sprite backgroundsprite(background);

    while (window.isOpen())
    {
        Event event;

        while (window.pollEvent(event))
```

```
{
    if (event.type == Event::Closed)
        window.close();

    if (event.type == Event::MouseMoved)
    {
        Vector2f mousePosition(event.mouseMove.x, event.mouseMove.y);

        if (startText.getGlobalBounds().contains(mousePosition))
            isStartHighlighted = true;
        else
            isStartHighlighted = false;

        if (settingsText.getGlobalBounds().contains(mousePosition))
            isSettingsHighlighted = true;
        else
            isSettingsHighlighted = false;

        if (exitText.getGlobalBounds().contains(mousePosition))
            isExitHighlighted = true;
        else
            isExitHighlighted = false;
    }
    if (event.type == Event::MouseButtonPressed)
    {
        int mouseX = event.mouseButton.x;
        int mouseY = event.mouseButton.y;

        if (startText.getGlobalBounds().contains(mouseX, mouseY)) {
            game();
        }

        if (settingsText.getGlobalBounds().contains(mouseX, mouseY)) {
            settings();
        }

        if (exitText.getGlobalBounds().contains(mouseX, mouseY)) {
            window.close();
        }
    }
}
```

```
        window.clear();

        if (isStartHighlighted)
            startText.setFillColor(Color::Blue);
        else
            startText.setFillColor(Color::Black);

        if (isSettingsHighlighted)
            settingsText.setFillColor(Color::Blue);
        else
            settingsText.setFillColor(Color::Black);

        if (isExitHighlighted)
            exitText.setFillColor(Color::Blue);
        else
            exitText.setFillColor(Color::Black);

        window.draw(backgroundsprite);

        window.draw(startText);
        window.draw(settingsText);
        window.draw(exitText);

        window.display();
    }

    return 0;
}
```

Приложение А.2 Исходный файл «game.cpp»

```
#include <SFML/Graphics.hpp>
#include <SFML/Audio.hpp>
#include <vector>
#include <random>
#include <string>
#include <fstream>
#include <iostream>
#include "game1.h"

int saveScoreToFile(int score)
{
```

```
    std::ofstream file("score.txt");
    if (file.is_open())
    {
        file << score;
        file.close();
    }
    return 0;
}

int loadScoreFromFile()
{
    int score;
    std::ifstream file("score.txt");
    if (file.is_open())
    {
        file >> score;
        file.close();
    }
    return score;
}

int game()
{
    sf::RenderWindow window(sf::VideoMode(800, 600), "Sprite Game");

    sf::Texture spriteTexture;
    switch (selectedBird)
    {
    case 1:
        spriteTexture.loadFromFile("images/333.png");
        break;
    case 2:
        spriteTexture.loadFromFile("images/666.png");
        break;
    case 3:
        spriteTexture.loadFromFile("images/555.png");
        break;
    }


    int score = loadScoreFromFile();
```

```
sf::Texture background;
background.loadFromFile("images/222.png");
sf::Sprite backgroundsprite(background);

std::vector<sf::Sprite> sprites;

std::random_device rd;
std::mt19937 gen(rd());
std::uniform_int_distribution<> xPosDist(50, 500);
std::uniform_int_distribution<> timeDist(800, 801);

sf::Clock clock;
sf::Time elapsedTime;

sf::Font font;
font.loadFromFile("C:/Windows/Fonts/Arial.ttf");
sf::Text counter;
counter.setFont(font);
counter.setCharacterSize(30);
counter.setPosition(window.getSize().x - 300, 0);
counter.setFillColor(sf::Color::Black);

sf::Text exitText;
exitText.setString("Exit");
exitText.setFont(font);
exitText.setCharacterSize(20);
exitText.setFillColor(sf::Color::Black);
exitText.setPosition(10, 10);

sf::Text scoreText;
scoreText.setString("Your score : " + std::to_string(score));
scoreText.setFont(font);
scoreText.setCharacterSize(20);
scoreText.setFillColor(sf::Color::Black);
scoreText.setPosition(600, 10);

sf::SoundBuffer soundBuffer;
if (!soundBuffer.loadFromFile("images/444.wav"))
{
    return 1;
}
```

```
    sf::Sound sound;
    sound.setBuffer(soundBuffer);

    while (window.isOpen())
    {
        sf::Event event;
        while (window.pollEvent(event))
        {
            if (event.type == sf::Event::Closed)
            {
                window.close();
            }

            if (event.type == sf::Event::MouseButtonPressed)
            {
                if (event.mouseButton.button == sf::Mouse::Left)
                {
                    sf::Vector2i mousePosition = sf::Mouse::getPosition(window);
                    if
(exitText.getGlobalBounds().contains(static_cast<sf::Vector2f>(mousePosition)))
                    {
                        window.close();
                    }
                    for (auto it = sprites.begin(); it != sprites.end(); ++it)
                    {
                        if
((*it).getGlobalBounds().contains(static_cast<sf::Vector2f>(mousePosition)))
                        {
                            sound.play();
                            sprites.erase(it);
                            score += 10;
                            scoreText.setString("Your score : " +
std::to_string(score)); // Исправление
                            break;
                        }
                    }
                }
            }
        }
```

```
        window.clear();

        window.draw(backgroundsprite);

        elapsedTime = clock.getElapsedTime();

        if (elapsedTime.asMilliseconds() > timeDist(gen))
        {
            sf::Sprite sprite;
            sprite.setTexture(spriteTexture);
            sprite.setPosition(sf::Vector2f(0, xPosDist(gen)));

            sprites.push_back(sprite);

            clock.restart();
        }

        for (auto& sprite : sprites)
        {
            sprite.move(sf::Vector2f(0.30, 0));
            window.draw(sprite);
        }

        saveScoreToFile(score);

        window.draw(counter);
        window.draw(exitText);
        window.draw(scoreText);
        window.display();
    }

    return 0;
}
```

Приложение А.3 Исходный файл «settings.cpp»

```
#include <SFML/Graphics.hpp>
#include <iostream>
#include <fstream>
#include "game1.h"

int selectedBird = 1;

int settings()
```

```
{
    sf::RenderWindow window(sf::VideoMode(800, 600), "Settings"); // Создаем окно с названием "Settings"
    window.setFramerateLimit(60); // устанавливаем ограничение кадров, чтобы предотвратить перегрев системы


    int pool = loadScoreFromFile(); // пример количества очков у игрока

    sf::Font font;
    font.loadFromFile("C:/Windows/Fonts/Arial.ttf"); // загружаем шрифт

    sf::Texture background;
    background.loadFromFile("images/111.png");
    sf::Sprite backgroundsprite(background);


    sf::Text title;
    title.setFont(font);
    title.setString("Settings");
    title.setFillColor(sf::Color::Black);
    title.setCharacterSize(40); // устанавливаем размер текста
    title.setPosition(300, 50); // устанавливаем позицию текста

    sf::Text scoreText;
    scoreText.setFont(font);
    scoreText.setString("Your score: " + std::to_string(pool)); // выводим текущее количество очков игрока
    scoreText.setFillColor(sf::Color::Black);
    scoreText.setCharacterSize(30);
    scoreText.setPosition(280, 100);

    sf::Text bird1("Yellow bird", font, 35);
    bird1.setPosition(300, 200);

    sf::Text bird2("Hell bird - 2000 points", font, 35);
    bird2.setPosition(250, 250);


    sf::Text bird3("Hulk Bird - 5000 points", font, 35);
    bird3.setPosition(250, 300);
```

```
    while (window.isOpen())
    {
        sf::Event event;
        while (window.pollEvent(event))
        {
            if (event.type == sf::Event::Closed)
            {
                window.close();
            }
            if (event.type == sf::Event::KeyPressed)
            {
                if (event.key.code == sf::Keyboard::Escape)
                {
                    window.close(); // закрываем окно при нажатии клавиши ESC
                }
            }
            bird1.setFillColor((selectedBird == 1) ? sf::Color::Green :
sf::Color::Black);
            bird2.setFillColor((selectedBird == 2) ? sf::Color::Green :
sf::Color::Black);
            bird3.setFillColor((selectedBird == 3) ? sf::Color::Green :
sf::Color::Black);
            if (sf::Mouse::isButtonPressed(sf::Mouse::Left))
            {
                sf::Vector2i mousePos = sf::Mouse::getPosition(window);
                if (bird1.getGlobalBounds().contains(mousePos.x, mousePos.y))
                {
                    selectedBird = 1;
                }
                else if (bird2.getGlobalBounds().contains(mousePos.x, mousePos.y))
                {
                    if (pool >= 2000) // убедимся, что у игрока достаточно очков
                    {
                        selectedBird = 2;
                        pool -= 2000; // списываем очки

                    }
                }
                else if (bird3.getGlobalBounds().contains(mousePos.x, mousePos.y))
                {
```

```
                    if (pool >= 5000) // убедимся, что у игрока достаточно очков
                    {
                        selectedBird = 3;
                        pool -= 5000; // списываем очки

                    }
                }
            }
        }

        std::ofstream file("score.txt");
        if (file.is_open())
        {
            file << pool;
            file.close();
        }

        scoreText.setString("Your score: " + std::to_string(pool));



        window.clear(sf::Color::White);

        window.draw(backgroundsprite);
        window.draw(title);
        window.draw(scoreText);
        window.draw(bird1);
        window.draw(bird2);
        window.draw(bird3);

        window.display(); // отображаем содержимое окна

    }


    return 0;
}
```

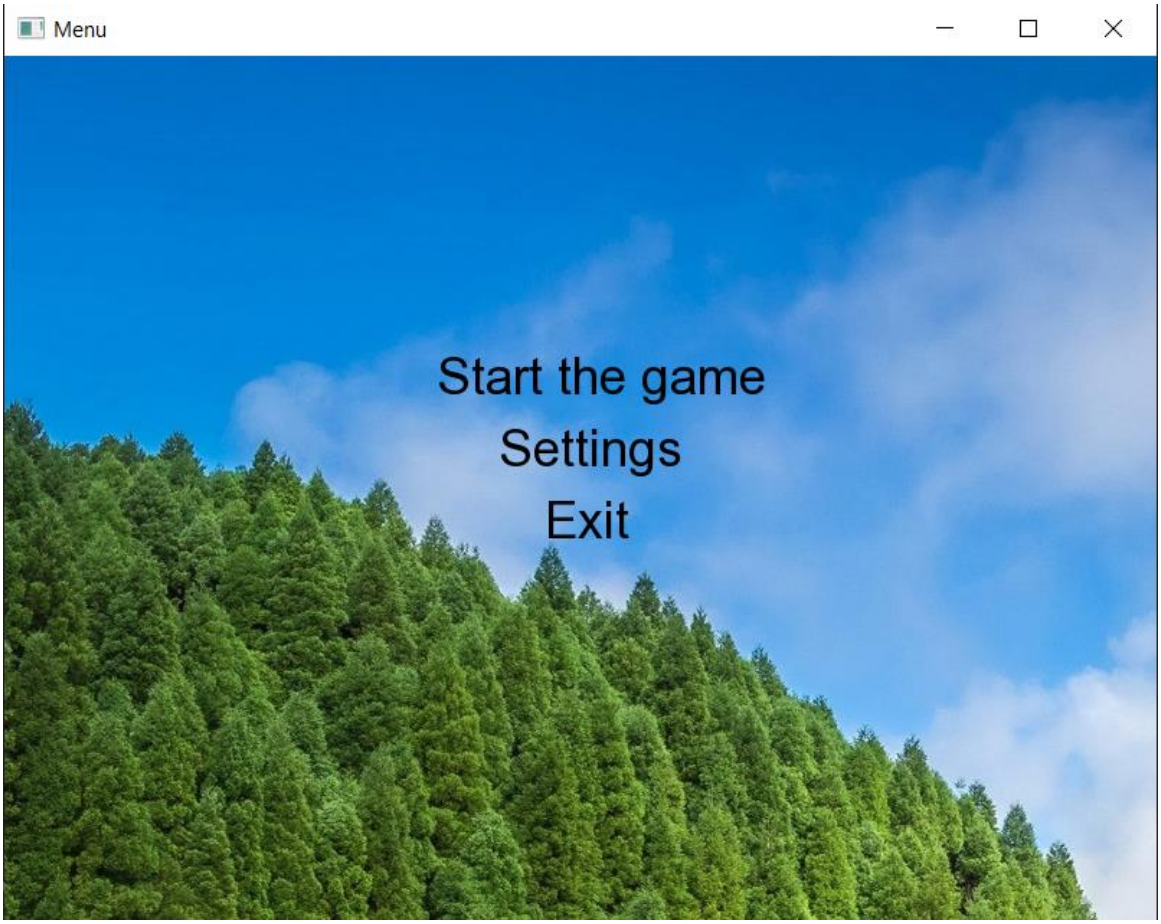
Приложение А.4 Заголовочный файл «Game1.h»

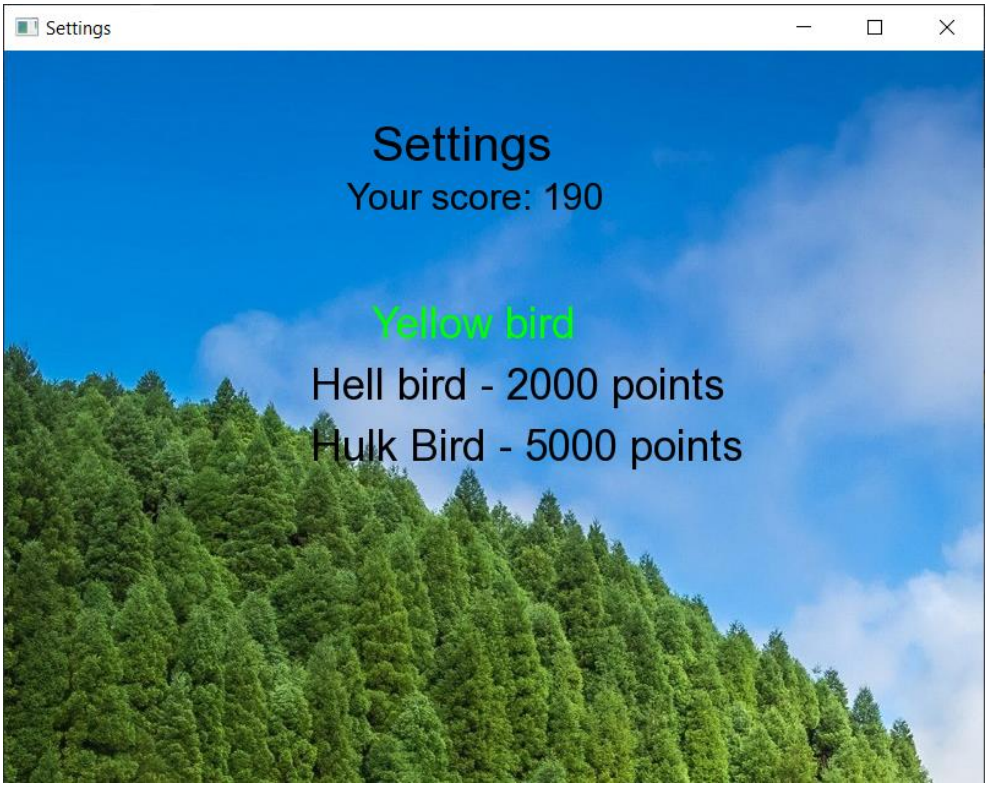
```
#pragma
int game();
int saveScoreToFile();
int loadScoreFromFile();
extern int selectedBird;
```

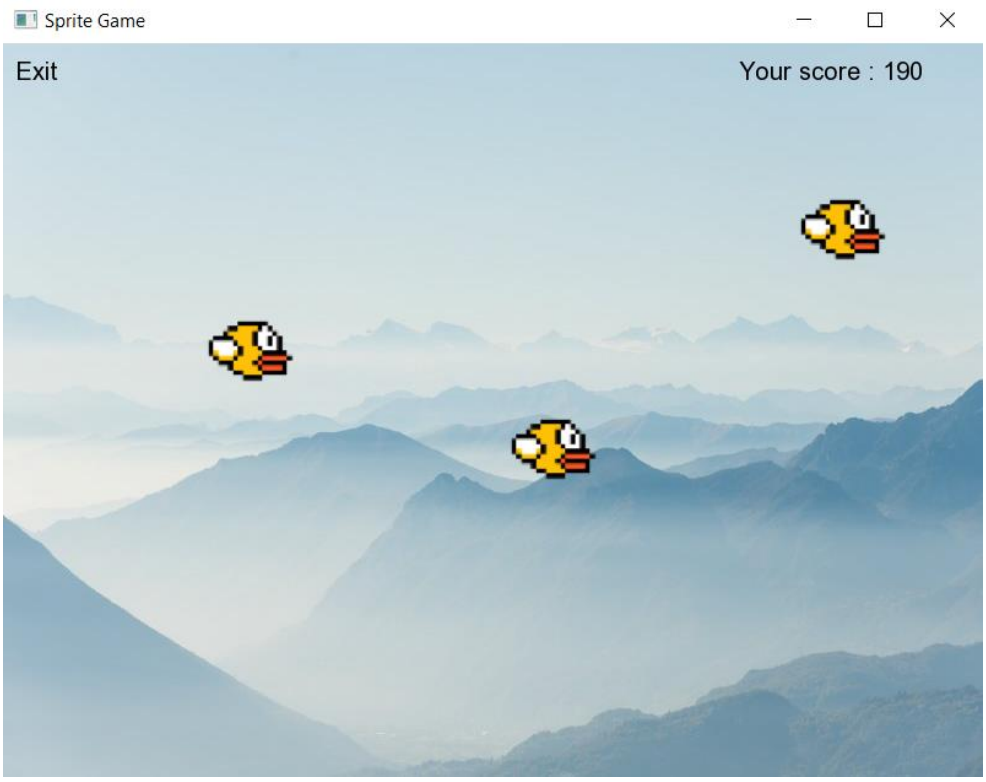
Приложение А.5 Заголовочный файл «Settings.h»

```
#pragma
int settings();
```

Приложение Б Результаты работы программы (снимки экрана)




Settings
Settings
Your score: 190
Yellow bird
Hell bird - 2000 points
Hulk Bird - 5000 points


Sprite Game
Exit
Your score : 190