

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

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
по курсу «Программирование»
на тему «Разработка программы сложной структуры
методом нисходящего программирования»

Выполнил:
студентка группы 22ВОА1
Гольяпина Д.А.

Принял:
Митрохин М.А.

Пенза 2023

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

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
по курсу «Программирование»
на тему «Разработка программы сложной структуры
методом нисходящего программирования»

Выполнил:
студентка группы 22BOA1
Гольяпина Д.А.

Принял:
Митрохин М.А.

25.12.23

Пенза 2023

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет Вычислительной техники
Кафедра «Вычислительная техника»

«УТВЕРЖДАЮ»
Зав. кафедрой ВТ
Митрохин М. А.
«___» _____ 2022

Г

ЗАДАНИЕ
на курсовое проектирование по курсу

Программирование

Студенту Гольяпиной Джастине Александровне Группа 22ВОА1

Тема проекта: Разработка программы сложной структуры методом нисходящего программирования

Исходные данные (технические требования) на проектирование

Наименование программы: Игра «Doodle Jump».

Задание. Программа должна выполнять следующие функции:

- Запускать игру.
- Выводить информацию об авторе.
- Содержать меню игры.
- Содержать выбор скина.

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. Операционная система — Windows.
7. Система программирования — C++ на базе среды Microsoft Visual Studio 2022.
8. Язык программирования — Си++.

Объем работы по курсу

1. Расчетная часть
Разработка программы.

2. Графическая часть
Схема алгоритма программы, иерархическая структура программы, схема данных.

3. Экспериментальная часть
Отладка программы.

Срок выполнения проекта по разделам

1. В соответствии с графиком выполнения курсового проекта.

Дата выдачи задания « 9 » сентября 2023
 Г _____

Дата защиты проекта « ____ » _____ 2023
 Г _____

Руководитель _____ « ____ » _____ 2023

Задание получил _____

Г _____

Студент Голоткина Д. А. _____

Оглавление

1 Постановка задачи	5
2 Выбор решения	6
2.1 Определение необходимых модулей программы	6
2.2 Определение алгоритма программы	7
3 Описание разработки программы	9
4 Отладка и тестирование	11
5 Описание программы	12
6 Руководство пользователя	19
Заключение	27
Список использованных источников	28
Приложение А Исходный код программы.....	29
Приложение А.1 Файл «Menu.cpp».....	29
Приложение А.2 Файл «Game.cpp».....	36
Приложение А.3 Файл «main.cpp»	40
Приложение А.4 Файл «bibl.h».....	41
Приложение А.5 Файл «game.hpp»	42
Приложение Б Снимки экрана.....	43

Прообразом привычного нам компьютера стала разностная машина английского математика Чарльза Бэббиджа. Работу над ней он начал ещё в 1822 году и надеялся, что однажды она заменит людей, которые регулярно допускали ошибки в навигационных, астрономических и математических таблицах.

Бэббидж ничего не знал о микросхемах и транзисторах, но ему удалось придумать аппарат, который мог функционировать и производить вычисления. Всех поставленных целей Бэббиджу достичь не удалось, но он сделал главное – указал направление развития компьютерного мира./1/

Задачей данной курсовой работы является создание игры «Doodle Jump». Для платформы iOS игра вышла 6 апреля 2009 года, а для платформ Android и BlackBerry — 2 марта 2010 года. В *Doodle Jump* необходимо подняться как можно выше по области, похожей на лист тетради в клетку, постоянно перепрыгивая с одной платформы на другую. Разработанная программа, может использоваться людьми, не имеющими квалификации программиста, Программа разработана с использованием современных концепций программирования — объектно-ориентированного программирования и языка C++. /2/

Пояснительная записка содержит техническое задание, описание программы, программу и методику испытаний, описание применения, текст программы.

Необходимо разработать игру «Doodle Jump». В игре будет реализовано следующее:

1. Вывод на экран персонажа.
2. Вывод информации об авторе.
3. Сохранение результата и того, кто играл в файл.
4. Вывод информации об игроках из файла.

Необходимо реализовать меню, с помощью которого можно будет запустить игру или узнать необходимую информацию.

Необходимо реализовать минималистичный и удобный интерфейс, чтобы у пользователя не возникало затруднений при использовании программы.

Программа должна быть многомодульной. Это облегчит обнаружение и исправление ошибок при ее отладке и тестировании.

В программе также будут использоваться сложные типы данных.

Режим работы видеосистемы графический.

Устройствами управления игрой является клавиатура или мышка. Все действия в игре должны осуществляться с помощью нажатия клавиш клавиатуры или клика мыши.

При разработке программы необходимо использовать среду Visual Studio 2022. Язык программирования - C++. Язык программирования C++ имеет возможность применения классов при использовании обычного стиля программирования и поддерживает визуальное программирование с немедленным отображением графических объектов.

2.1 Определение необходимых модулей программы

Программа состоит из 3 модулей: Main.cpp, Game.cpp, Menu.cpp.

Описание модулей приведено в таблице 2.

Таблица 2 – Описание модуля

Имя модуля	Форма	Примечание
Main	Main. cpp	Модуль программы отвечающий за инициализацию объекта Menu и вызов его метода run
Game	Game.cpp	Модуль программы отвечающий за отрисовку и логику игры
Menu	Menu.cpp	Модуль программы отвечает за отображение меню

Схема взаимодействия модулей программы представлена на рисунке 1.

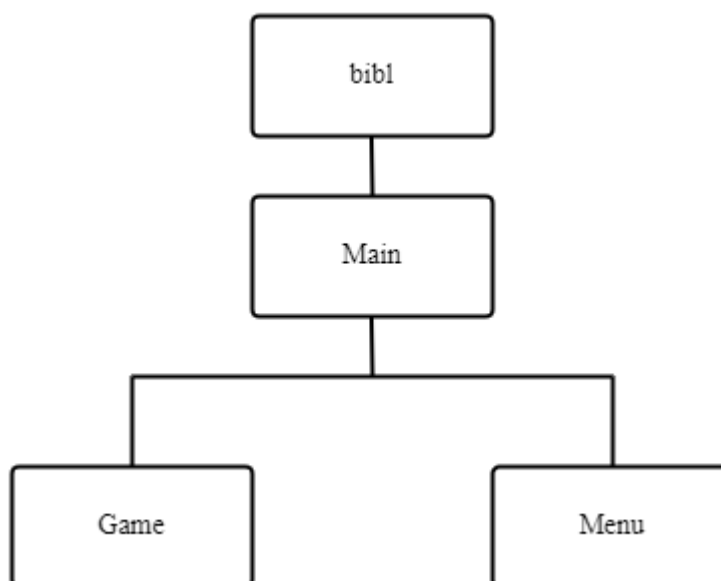


Рисунок 1 - Схема взаимодействия программ

2.2 Определение алгоритма программы

Для представления принципа работы программы ниже приведена ее функциональная схема на рисунке 2.

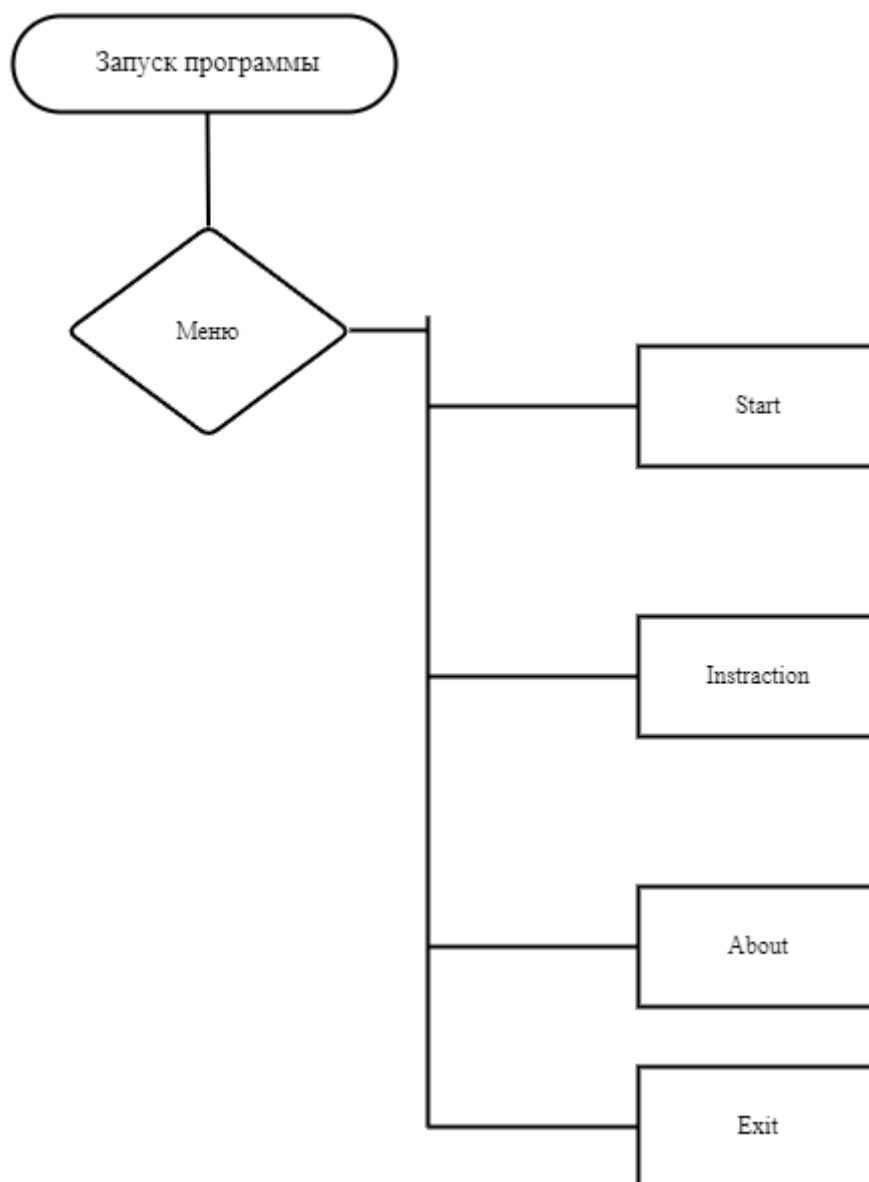


Рисунок 2 — Функциональная схема программы

После запуска программы появляется меню. Оно содержит четыре пункта. Первый пункт «Start», после выбора которого запускается игровой

процесс. Второй пункт «Instruction», после выбора которого на экран выводится инструкция. Третий пункт «About», после выбора которого на экран выводятся данные об авторе. Четвертый пункт «Exit», он позволяет выйти из игры.

Ниже на рисунке 3 представлена схема данных программы.

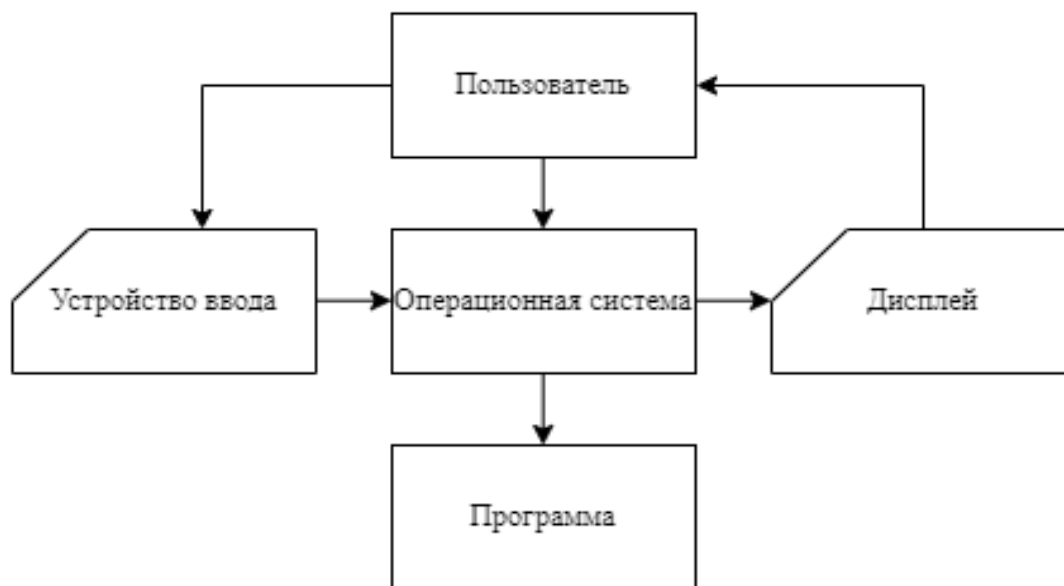


Рисунок 3 — Схема данных

Для написания данной программы будет использован язык программирования C++. Этот язык является распространенным языком программирования. При разработке языка C++ был принят компромисс между низким уровнем языка ассемблера и высоким уровнем других языков. В языке C++ предусмотрено много операций, непосредственно исполняемых большинством микропроцессоров; в то же время он дает программисту возможность выражать свои алгоритмы и данные наиболее подходящими средствами. C++ обеспечивает возможности структурирования данных. Он позволяет разрабатывать большие, но структурированные программы, предоставляя возможность отдельной разработки программы.

Целью создателей языка C++ была разработка мобильного языка, который можно было бы использовать для разработки системного программного обеспечения. Программы, написанные на языке высокого уровня, мобильны, поскольку язык должен быть одним и тем же независимо от того, на каком компьютере и в какой операционной системе он используется. Дополнительное свойство языка C++, называемое условной компиляцией, позволяет программисту изолировать машинно-зависимые операторы и контролировать их компиляцию в другой среде. Это дополнительно повышает мобильность программного обеспечения, написанного на языке C++.

В качестве среды программирования был выбран программный продукт Visual Studio 2022.

Компонентный подход к созданию приложений (или визуальное программирование), реализованный в различных средах разработки приложений, произвел настоящую революцию в технологиях программирования. Суть компонентного подхода проста: приложение собирается из готовых элементов – компонентов, библиотека которых поставляется вместе со средой программирования. В качестве строительной площадки служит форма – окно Windows, которое во время разработки

обладает рядом вспомогательных свойств, облегчающих размещение компонентов.

Интерфейс информационно-вычислительной системы будет представлен по большей части в виде панели инструментов, т.к. это наиболее удобный доступ к данным. Выбор пунктов меню будет осуществляться с помощью клика мыши на графическом интерфейсе.

В качестве среды разработки была выбрана программа Microsoft Visual Studio 2022, которая содержит в себе все необходимые средства для разработки и отладки модулей и программ. Для отладки использовались такие инструменты, как точки останова, пошаговое выполнение кода программы, анализ содержимого глобальных и локальных переменных.

Тестирование проводилось в рабочем порядке, в процессе разработки, после завершения написания программы. В процессе отладки было выявлено и исправлено множество ошибок. Например, некорректное открытие игры при нажатии «Start», проблемы связанные с прыжком персонажа, и генерацией платформ.

Тестирование смены скина:

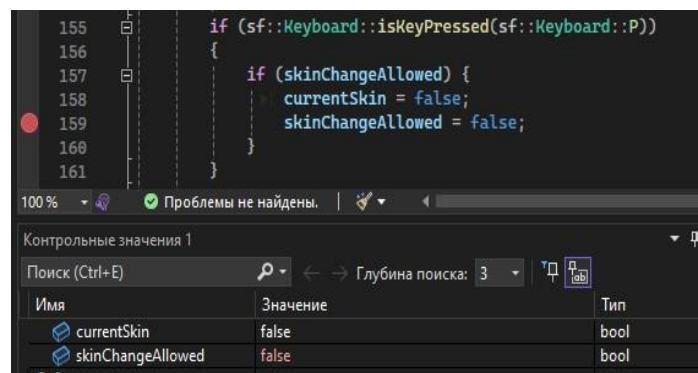


Рисунок 4 – Нажата кнопка P

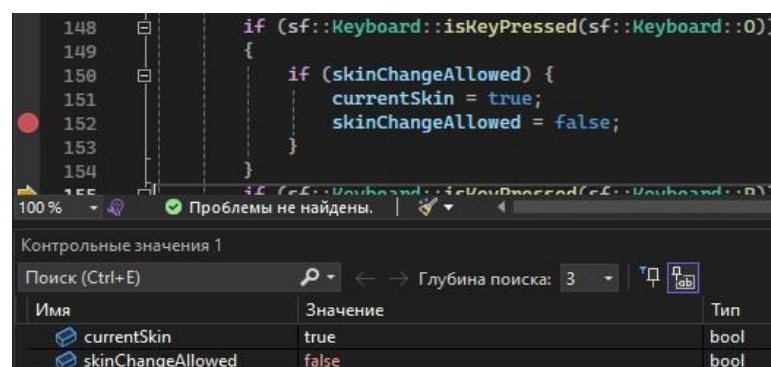


Рисунок 5 – Нажата кнопка O

Проект программы включает в себя главную программу и следующие подпрограммы:

Описание процедур и функций приведено в таблице 3.

Таблица 3 – Описание процедур и функций.

Имя модуля	Наименование процедуры или функции	Примечание
Main	Main run()	Процедура устанавливает и запускает игру
Game	void Game:: run()	Процедура запускает все основные игровые процессы
Menu	Метод run()	Метод выполняет обработку событий, включая клики мыши, отображает различные экраны

Обработчик `gui()` предназначен для отображения меню игры, на экране появляется изображение меню с пунктами.

Алгоритм процедуры `gui()` представлен на рисунках 4-5.

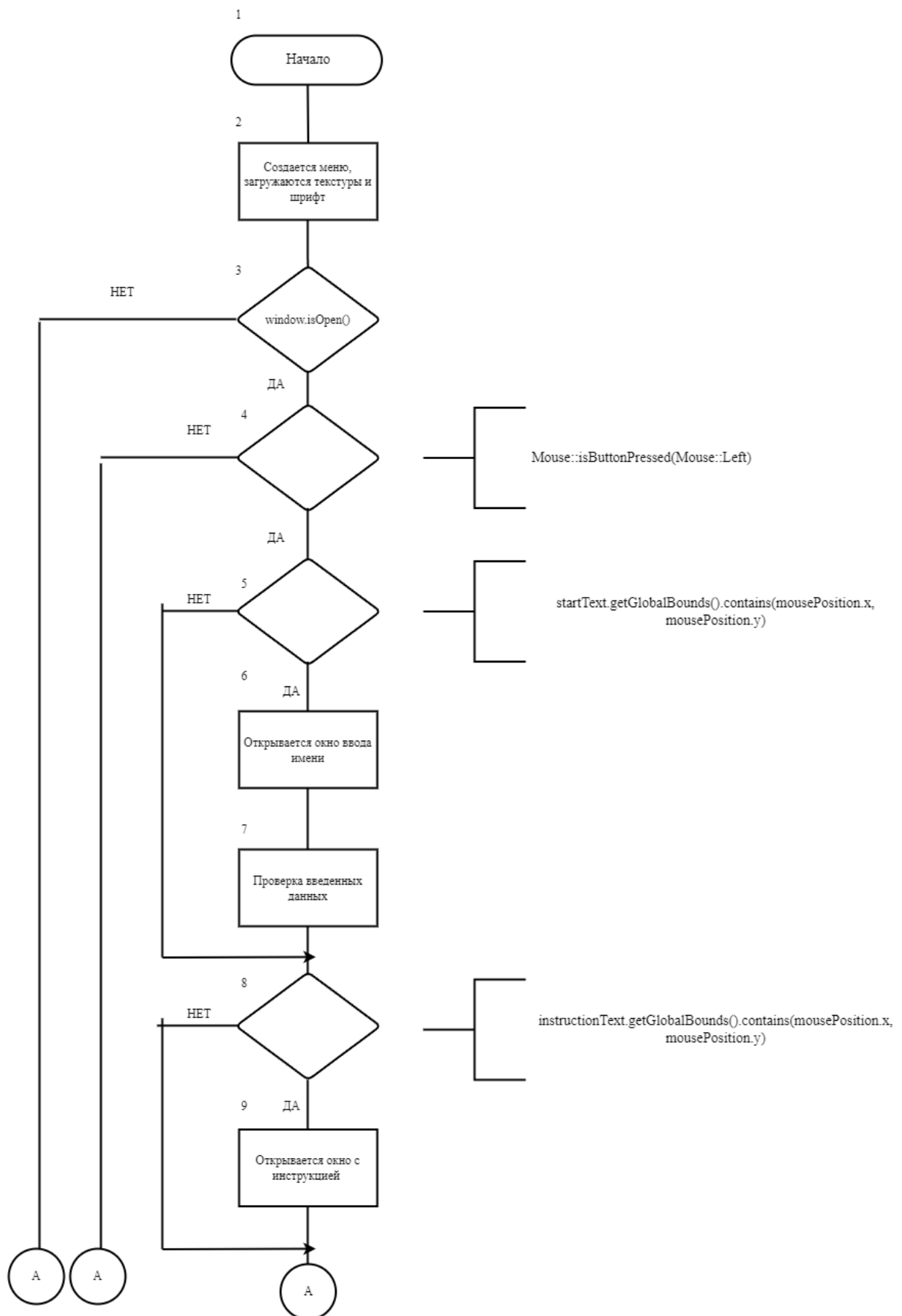


Рисунок 5 – Схема процедуры run()

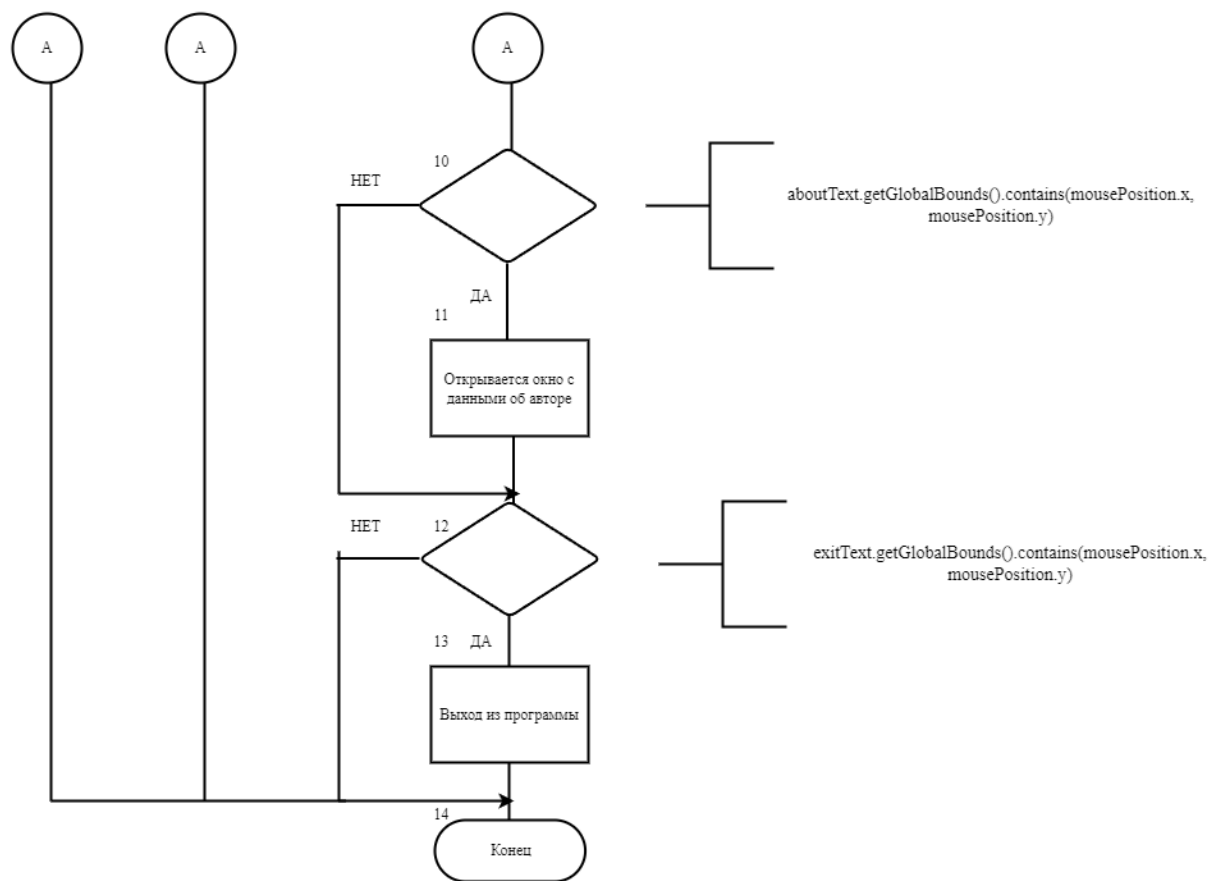


Рисунок 6 - Схема процедуры run()

Алгоритм начинается с создания меню игры, загрузки для него текстур и шрифта (блок 2). Далее идет цикл, который выполняется пока окно меню открыто (блок 3). Если истина, то далее идет условие, которое проверяет была ли нажата левая кнопка мыши (блок 4). Если истина, то проверяется условие нажат ли пункт меню старт (блок 5). Если истина, то появляется окно ввода данных игрока (блок 6) и затем проверяется на правильность введенных символов (блок 7). Если нажат пункт «Инструкция» (блок 8), то на экран выводится инструкция игры (блок 9). Если выбран пункт «Об авторе», (блок 10), то на экран выводятся данные об авторе игры (блок 11). Если выбран пункт «Выход» (блок 12), то происходит выход из программы (блок 13).

Алгоритм процедуры void Game:: run() представлен на рисунке 4.

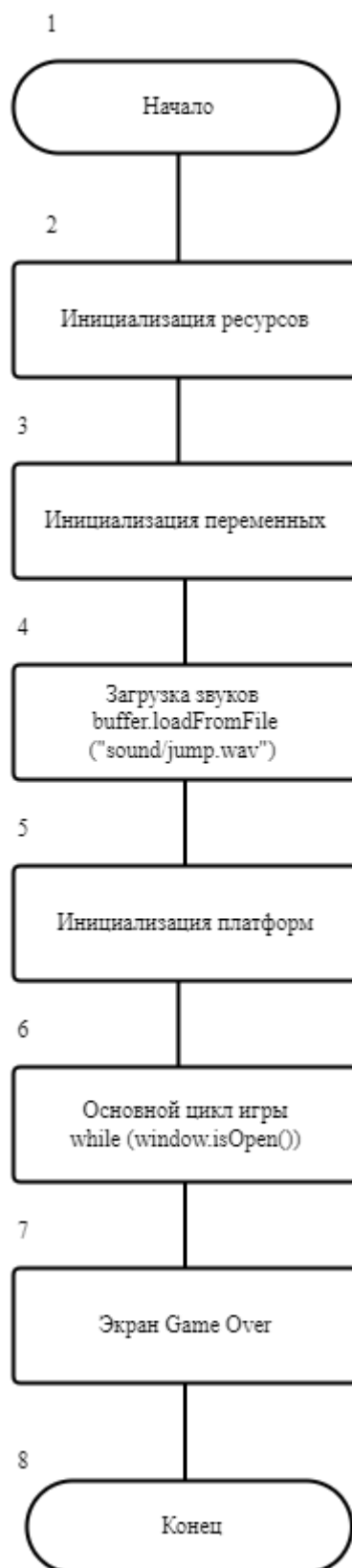


Рисунок 7 - Схема процедуры void Game:: run()

Алгоритм начинается с того, что начинается инициализация ресурсов (блок 2), далее происходит инициализация переменных (блок 3). После

начинается загрузка звуков (блок 4), и происходит инициализация игровых платформ (блок 5). После всего начинается основной цикл игры (блок 6), и заканчивается экраном Game Over (блок 7).

Обработчик Метода `gui()` отвечает за выполнение обработки событий, включая клики мыши, отображает различные экраны.

Алгоритм Метода `gui()` представлен на рисунке 4



Рисунок 8 - Схема main

Алгоритм начинается с создания объекта Menu object (menu) (блок 2), далее выполняется метод `gui` (блок 3). Следом создается объект игры (блок 4).

6 Руководство пользователя

При запуске программы появится меню игры (рисунок 9).

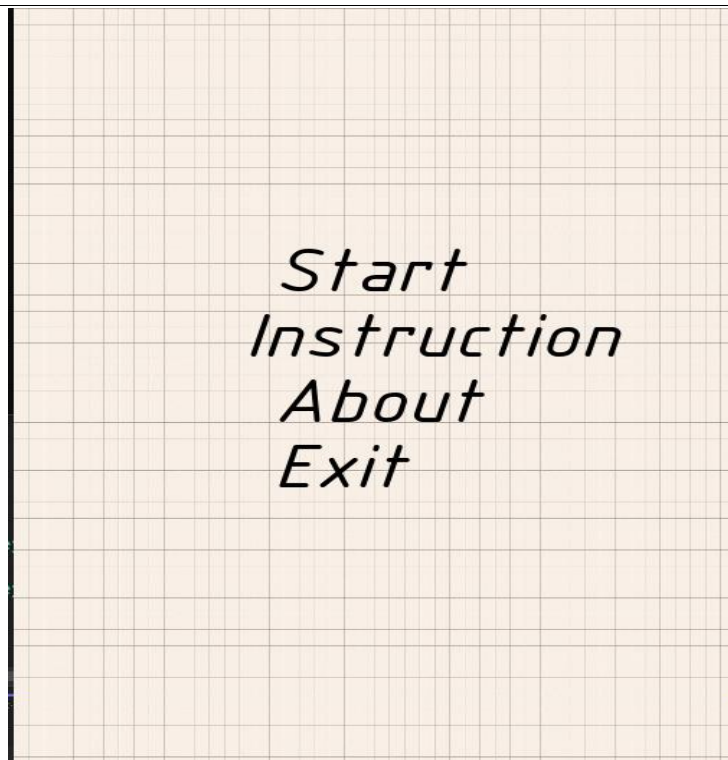


Рисунок 9 – Меню игры

Если выбрать пункт меню «START» (рисунок 10), то будет предложено ввести свое имя (рисунки 11-12). Затем нажать клавишу «Enter».



Рисунок 10 - Пункт «START»

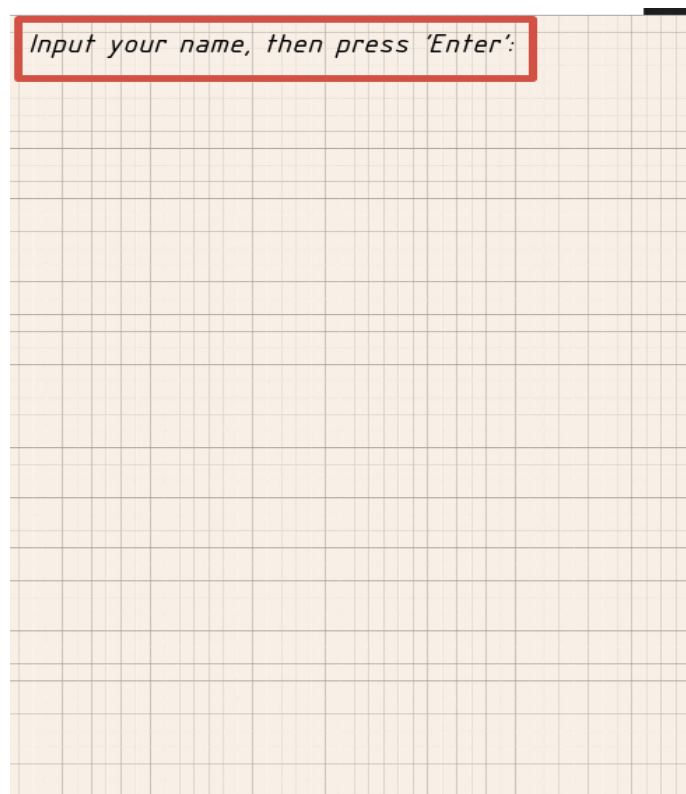


Рисунок 11 – Ввод имени

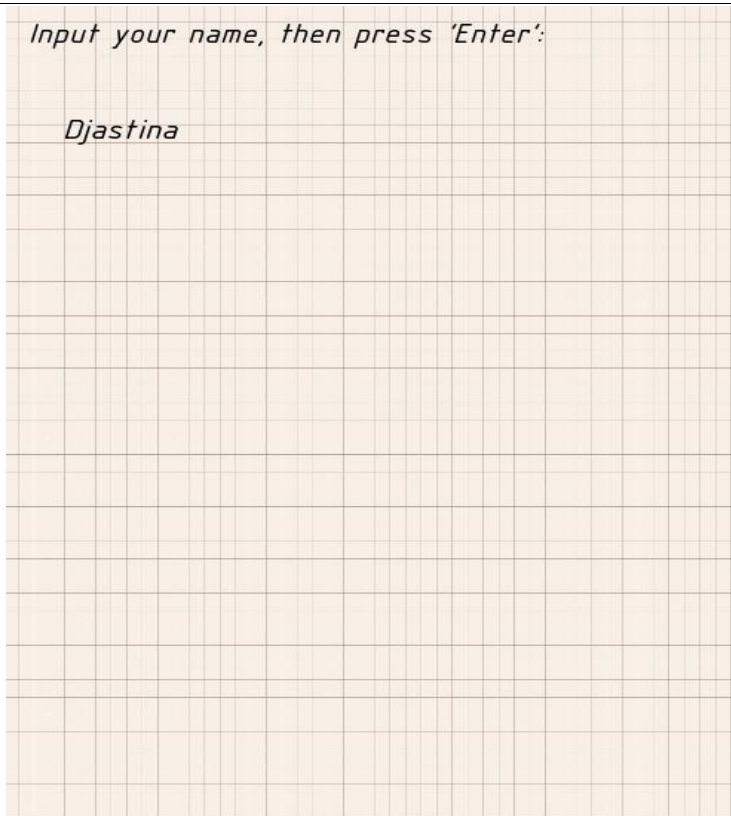


Рисунок 1 2 – Введено имя

После нажатия на «Enter» на экране появится игра (рисунок 13).

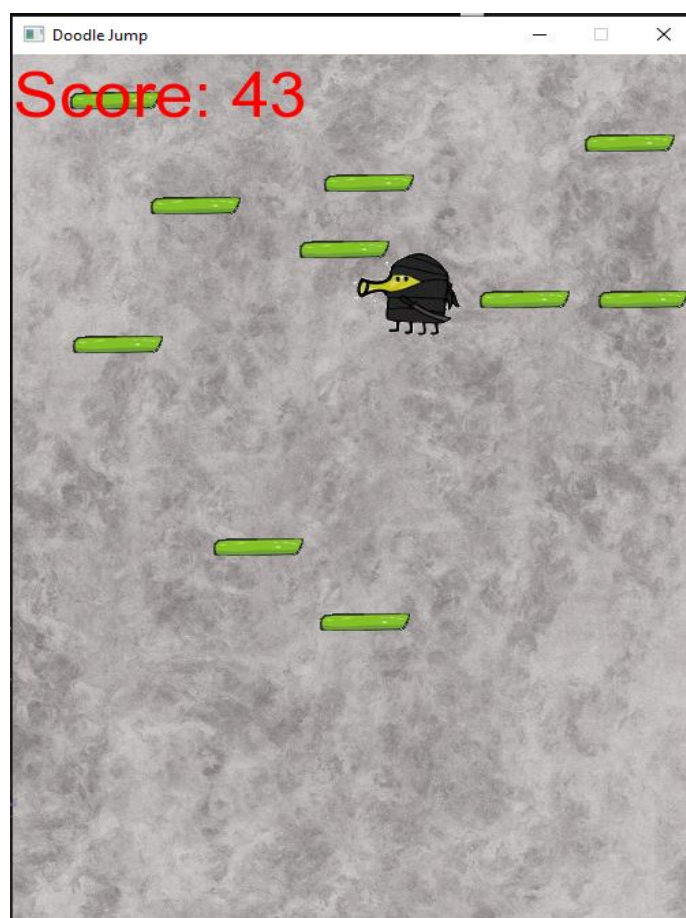


Рисунок 13– Игровой процесс

Если упасть с платформы, то игра будет закончена (рисунок 14).



Рисунок 14 – Проигрыш

После падения с платформы также изображается ваш результат (рисунок 15).



Рисунок 15 – Результат

Если нажать на пункт меню «Instruction» (рисунок 16). Будет открыто окно с дополнительной инструкцией для игры. Чтобы вернуться в меню надо нажать на «BACK».



Рисунок 16 – Пункт меню «Instruction»



Рисунок 17 – Инструкция

Если нажать пункт меню «ABOUT» (рисунок 18), то на экран будет выведены данные о разработчике (рисунок 19). Чтобы вернуться в меню надо нажать на «BACK».



Рисунок 18 – Пункт меню «ABOUT»

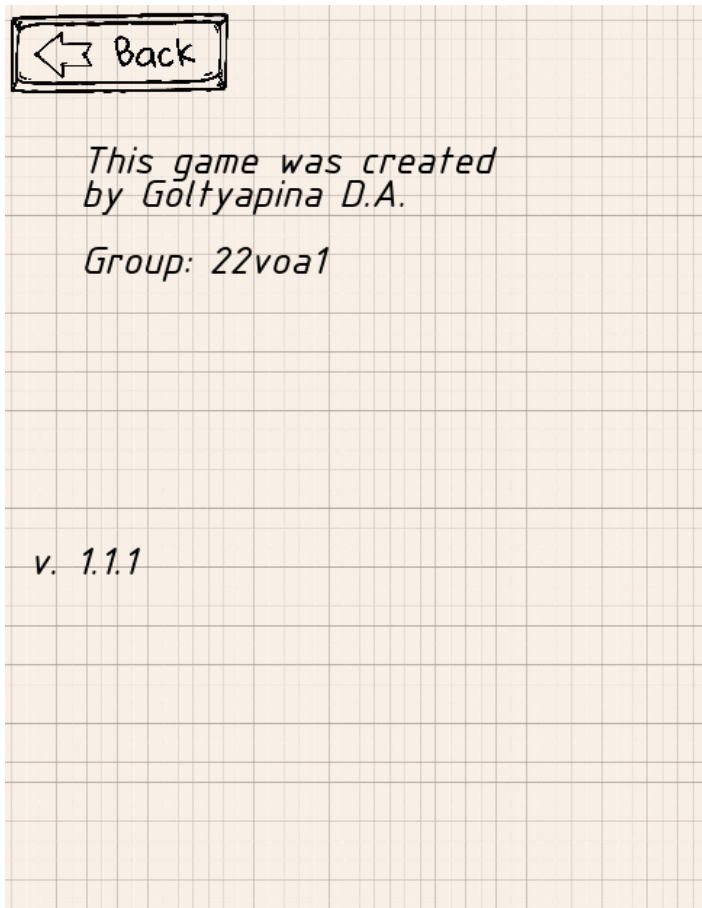


Рисунок 19 – Информация о разработчике

Для выхода из игры надо нажать на пункт меню «EXIT» (рисунок 20).



Рисунок 20 – Пункт меню «EXIT»

Заключение

В ходе выполнения курсовой работы было сформулировано и утверждено техническое задание на разработку игры «Doodle Jump», разработан алгоритм решения задачи, составлена и отлажена программа, оформлена документация на программу. Проведенные испытания показали, что программа все требования технического задания выполнены в полном объёме. Результаты испытаний свидетельствуют о правильности работы программы.

Изучены основные возможности среды программирования Microsoft Visual Studio 2022. Получены навыки отладки и тестирования программ. Были освоены приемы создания и подключения модулей программы. Получены навыки работы со сложными типами данных, такие как файлы.

Достоинства игры:

- Простой и понятный интерфейс;
- Программный продукт малотребователен к системным ресурсам компьютера;
- Игра понравится детям из-за обилия красок и простого игрового процесса;

Недостатки:

- Мало разнообразия.

В дальнейшем игру можно улучшить, добавив больше скинов на персонажа, оформления заднего фона игры.

Список использованных источников

1. Информатика. Базовый курс [Текст]: учебное пособие для студентов высших технических учебных заведений / под ред. С. В. Симановича. - СПб.: Питер, 2019. – 637 с.
2. Основы компьютерной обработки информации [Текст] : учебно-методическое пособие / А. Е. Хохлов, К. М. Буданов ; Пенз. гос. ун-т. - Пенза : Изд-во Пенз. гос. ун-та, 2017. - 113 с.
3. Электронный ресурс. Кафедра автоматизированных систем управлений. Официальный сайт. – URL: <https://asu.tusur.ru/graduate/spec051301/spec051301-lect1.pdf> (дата обращения 17.02.2022).
4. Электронный ресурс. Официальный сайт Брильёновой Натальи Валерьевны преподавателя кафедры информатики и электроники Екатеринбургского государственного института. – URL: <https://natalibrilenova.ru/referat-na-temu-informatsiya/> (дата обращения 18.11.2023).
5. Электронный ресурс. Официальный сайт Laurent Gomila (SFML). – URL: <https://www.sfml-dev.org/> (дата обращения 19.11.2023).

Приложение А

Исходный код программы

Приложение А.1

Файл «Menu.cpp»

```
#define _CRT_SECURE_NO_WARNINGS
#include "bibl.h"

class Menu {
private:
    RenderWindow window, window2, window3;
    Texture menuTexture;
    Texture f, f2, Back;
    Sprite menuSprite;
    RectangleShape background, background2, back;
    Font ft;
    Text startText;
    Text instructionText;
    Text aboutText;
    Text exitText;
    Text author;
    string befString, inputString;
    FILE* fileResults;

public:

    void run() {

        window.create(VideoMode(500, 700), "Doodle Jump", Style::None);
        window.setFramerateLimit(60);

        f.loadFromFile("images/bg2.jpg");
        background.setSize(Vector2f(500, 700));
        background.setTexture(&f);
        background.setPosition(0, 0);

        if (!ft.loadFromFile("images/bukvi2.ttf")) {
            cout << "Error loading font" << endl;
```

```

return;
    }

    startText.setString("Start");
    startText.setFont(ft);
    startText.setCharacterSize(60);
    startText.setFillColor(Color::Black);
    startText.setPosition(180, 200);

    instructionText.setString("Instruction");
    instructionText.setFont(ft);
    instructionText.setCharacterSize(60);
    instructionText.setFillColor(Color::Black);
    instructionText.setPosition(160, 260);

    aboutText.setString("About");
    aboutText.setFont(ft);
    aboutText.setCharacterSize(60);
    aboutText.setFillColor(Color::Black);
    aboutText.setPosition(180, 320);

    exitText.setString("Exit");
    exitText.setFont(ft);
    exitText.setCharacterSize(60);
    exitText.setFillColor(Color::Black);
    exitText.setPosition(180, 380);

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

        if (sf::Mouse::isButtonPressed(sf::Mouse::Left)) {
            sf::Vector2i mousePosition = sf::Mouse::getPosition(window);
            if (startText.getGlobalBounds().contains(mousePosition.x,
mousePosition.y)) {

                sf::RenderWindow window3(sf::VideoMode(500, 700), "Input your
name", Style::None);

```

```

// Создаем текстовое поле для отображения введенной строки
sf::Text text, beftext;

beftext.setFont(ft);
beftext.setCharacterSize(23);
beftext.setFillColor(sf::Color::Black);
beftext.setPosition(17, 10);

text.setFont(ft);
text.setCharacterSize(24);
text.setFillColor(sf::Color::Black);
text.setPosition(40, 90);

f2.loadFromFile("images/bg2.jpg");
background2.setSize(Vector2f(500, 700));
background2.setTexture(&f2);
background2.setPosition(0, 0);

while (window3.isOpen())
{
    sf::Event event;
    while (window3.pollEvent(event))
    {
        if (event.type == sf::Event::Closed)
            window3.close();

        befString = "Input your name, then press 'Enter':";

        // Если была нажата клавиша
        if (event.type == sf::Event::TextEntered)
        {
            // Проверяем, что символ можно отобразить (не является
управляющим символом)
            if (event.text.unicode >= 32 && event.text.unicode <=
126)
                inputString +=
static_cast<char>(event.text.unicode); // Добавляем введенный символ в строку

            // Если нажата клавиша Backspace, удаляем последний
символ из строки
            if (event.text.unicode == 8 && inputString.size() >
0)

```

```

inputString.pop_back();

// Если нажата клавиша Enter, выводим строку в консоль
и очищаем ее

if (event.text.unicode == 13)
{
    std::cout << inputString << std::endl;

    wchar_t* szUsername;
    szUsername = new wchar_t[inputString.length() +
1];

    memset(szUsername, L'\0', (inputString.length() +
1) * sizeof(wchar_t));

    for (int i = 0; i <= inputString.length(); i++)
szUsername[i] = inputString[i];

    fileResults = fopen("Results.txt", "a");
    fprintf(fileResults, "Username: %ws\n",
szUsername);

    delete szUsername;
    fclose(fileResults);

    inputString.clear();
    window3.close();
    window.close();
}
}

text.setString(inputString);
befText.setString(befString);

window3.clear(sf::Color::White);
window3.draw(background2);
window3.draw(text);
window3.draw(befText);
window3.display();
}

}
// window.close();
else if (instructionText.getGlobalBounds().contains(mousePosition.x,
mousePosition.y)) {
    window2.create(VideoMode(500, 700), "Doodle Jump", Style::None);

```

```

f2.loadFromFile("images/bg2.jpg");
    background2.setSize(Vector2f(500, 700));
    background2.setTexture(&f2);
    background2.setPosition(0, 0);

    Back.loadFromFile("images/back.png");
    back.setSize(Vector2f(150, 60));
    back.setTexture(&Back);
    back.setPosition(10, 10);

    // Создаем новый текст для инструкций
    Text instructionText;
    instructionText.setFont(ft);
    instructionText.setCharacterSize(15);
    instructionText.setFillColor(Color::Black);
    instructionText.setPosition(10, 100);
    instructionText.setString("Press O - SKIN NINJA, Press P - SKIN NEW
YEAR ");

    while (window2.isOpen()) {
        sf::Event event;
        while (window2.pollEvent(event)) {
            if (event.type == sf::Event::Closed)
                window2.close();
        }

        if (sf::Mouse::isButtonPressed(sf::Mouse::Left)) {
            sf::Vector2i mousePosition =
sf::Mouse::getPosition(window2);

            if (back.getGlobalBounds().contains(mousePosition.x,
mousePosition.y))
                window2.close();
        }

        window2.clear();
        window2.draw(background2);

        // Отображаем текст инструкции
        window2.draw(instructionText);

        window2.draw(back);
        window2.display();
    }
}

```

```

        //cout << "Result selected" << endl;

        else if (aboutText.getGlobalBounds().contains(mousePosition.x,
mousePosition.y)) {
            window2.create(VideoMode(500, 700), "Doodle Jump", Style::None);
            f2.loadFromFile("images/bg2.jpg");
            background2.setSize(Vector2f(500, 700));
            background2.setTexture(&f2);
            background2.setPosition(0, 0);

            Back.loadFromFile("images/back.png");
            back.setSize(Vector2f(150, 60));
            back.setTexture(&Back);
            back.setPosition(10, 10);

            author.setString("\tThis game was created \n\tby Goltyapina D.A.
\n\n\tGroup: 22voa1 \n\n\n\n\n\n\n\n\n v. 1.1.1");
            author.setFont(ft);
            author.setCharacterSize(30);
            author.setFillColor(Color::Black);
            author.setPosition(10, 100);

            while (window2.isOpen()) {
                sf::Event event;
                while (window2.pollEvent(event)) {
                    if (event.type == sf::Event::Closed)
                        window2.close();
                }

                if (sf::Mouse::isButtonPressed(sf::Mouse::Left)) {
                    sf::Vector2i mousePosition =
sf::Mouse::getPosition(window2);

                    if (back.getGlobalBounds().contains(mousePosition.x,
mousePosition.y))

                        window2.close();

                }

                window2.clear();
                window2.draw(background2);
                window2.draw(author);
                window2.draw(back);

```

```

window2.display();
        }
    }

    else if (exitText.getGlobalBounds().contains(mousePosition.x,
mousePosition.y)) {
        exit(0);

    }
}

window.clear();
window.draw(background);

window.draw(menuSprite);

window.draw(startText);
window.draw(instructionText);
window.draw(aboutText);
window.draw(exitText);
window.display();
    }
}
};

```

Приложение A.2

Файл «Game.cpp»

```
#define _CRT_SECURE_NO_WARNINGS

#include "game.hpp"
#include "bibl.h"
#include "menu.cpp"

Font ft;
Text startText;

void Game::Run()
{
    FILE* fileResults;

    Menu menu;
    sf::RenderWindow window(sf::VideoMode(500, 700), "Doodle Jump", sf::Style::Close);
    window.setFramerateLimit(60);
    sf::Texture backgroundTexture;
    sf::Texture playerTexture;
    sf::Texture platformTexture;
    sf::Texture playerTexture2;
    backgroundTexture.loadFromFile("images/background.png");
    playerTexture.loadFromFile("images/doodle.png");
    platformTexture.loadFromFile("images/platform.png");
    playerTexture2.loadFromFile("images/doodle2.png");

    sf::Sprite background(backgroundTexture);
    sf::Sprite player(playerTexture);
    sf::Sprite platform(platformTexture);
    sf::Sprite player2(playerTexture2);

    sf::RectangleShape gameOverBackground(sf::Vector2f(500, 700));
    gameOverBackground.setFillColor(sf::Color::White);

    sf::Font font;
    font.loadFromFile("font/arial.ttf");
    sf::Text scoreText;
    scoreText.setFont(font);
    scoreText.setCharacterSize(50);
    scoreText.setCharacterSize(50);
    scoreText.setFillColor(sf::Color::Red);
    sf::Text gameOverText;
    gameOverText.setFont(font);
    gameOverText.setString("Game Over!");
    gameOverText.setCharacterSize(80);
    gameOverText.setFillColor(sf::Color::Red);
    sf::Text goMenu;
    goMenu.setFont(font);
    goMenu.setString("Press 'M' to return to the menu");
    goMenu.setCharacterSize(30);
    goMenu.setFillColor(sf::Color::Red);
    bool currentSkin = true;
    bool skinChangeAllowed = true;

    int playerX = 250;
    int playerY = 151;

    float dy = 0;

    int height = 150;

    int score = 0;
```

```

// sound
sf::SoundBuffer buffer;
buffer.loadFromFile("sound/jump.wav");
sf::Sound sound;
sound.setBuffer(buffer);

const int PLAYER_LEFT_BOUNDING_BOX = 20;
const int PLAYER_RIGHT_BOUNDING_BOX = 60;
const int PLAYER_BOTTOM_BOUNDING_BOX = 70;

// initialize platforms
sf::Vector2u platformPosition[10];
std::uniform_int_distribution<unsigned> x(0, 500 - platformTexture.getSize().x);
std::uniform_int_distribution<unsigned> y(100, 700);
std::default_random_engine e(time(0));
for (size_t i = 0; i < 10; ++i)
{
    platformPosition[i].x = x(e);
    platformPosition[i].y = y(e);
}

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

    if (sf::Keyboard::isKeyPressed(sf::Keyboard::A) || sf::Keyboard::isKeyPressed(sf::Keyboard::Left))
        playerX -= 4;
    if (sf::Keyboard::isKeyPressed(sf::Keyboard::D) || sf::Keyboard::isKeyPressed(sf::Keyboard::Right))
        playerX += 4;

    if (playerX > 500)
        playerX = 0;
    if (playerX < -40)
        playerX = window.getSize().x - playerTexture.getSize().x;

    if (playerY == height && dy < (-1.62))
    {
        score += 1;
        scoreText.setString("Score: " + std::to_string(score));
    }

    dy += 0.2;
    playerY += dy;

    if (playerY < height)
    {

```

39

```

for (size_t i = 0; i < 10; ++i)
    {
        playerY = height;
        platformPosition[i].y -= dy;
        if (platformPosition[i].y > 700)
        {
            platformPosition[i].y = 0;
            platformPosition[i].x = x(e);
        }
    }

for (size_t i = 0; i < 10; ++i)
{
    if ((playerX + PLAYER_RIGHT_BOUNDING_BOX > platformPosition[i].x) && (playerX +
PLAYER_LEFT_BOUNDING_BOX < platformPosition[i].x + platformTexture.getSize().x)
        && (playerY + PLAYER_BOTTOM_BOUNDING_BOX > platformPosition[i].y) && (playerY +
PLAYER_BOTTOM_BOUNDING_BOX < platformPosition[i].y + platformTexture.getSize().y)
        && (dy > 0))
    {
        sound.play();
    }
}

```

```

        dy = -10;
    }
}

player.setPosition(playerX, playerY);

window.clear();
window.draw(background);
if (currentSkin == true)
{
    player.setTexture(playerTexture);
}
else if (currentSkin == false)
{
    player.setTexture(playerTexture2);
}
window.draw(player);
if (sf::Keyboard::isKeyPressed(sf::Keyboard::O))
{
    if (skinChangeAllowed) {
        currentSkin = true;
        skinChangeAllowed = false;
    }
}
if (sf::Keyboard::isKeyPressed(sf::Keyboard::P))
{
    if (skinChangeAllowed) {
        currentSkin = false;
        skinChangeAllowed = false;
    }
}

for (size_t i = 0; i < 10; ++i)
{
    platform.setPosition(platformPosition[i].x, platformPosition[i].y);
    window.draw(platform);
}

if (playerY > 700)
{

```

40

```

gameoverText.setPosition(30, 200);
goMenu.setPosition(30, 500);
scoreText.setPosition(150, 400);

fileResults = fopen("results.txt", "a");
fprintf(fileResults, "Score %d\n", score);
fclose(fileResults);

goto gameover;
}

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

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

    window.draw(gameoverBackground);
    window.draw(gameoverText);
    window.draw(goMenu);
    window.draw(scoreText);

```

```

        window.display();

        if (sf::Keyboard::isKeyPressed(sf::Keyboard::M)) {
            window.close();
            menu.run();
            Game().Run();
        }
    }
}

```

41

Приложение А.3

Файл «main.cpp»

```

#include "game.hpp"
#include "bibl.h"
#include "menu.cpp"

extern std::string befString;
extern std::string inputString;
int main()
{
    Menu menu;
    menu.run();
    Game().Run();

    return 0;
}

```

Приложение A.4

Файл «bibl.h»

```
#pragma once
#include <SFML/Graphics.hpp>
#include <SFML/Audio.hpp>
#include <random>
#include <ctime>
#include <iostream>
#include <fstream>
using namespace sf;
using namespace std;
```

43

Приложение A.5

Файл «game.hpp»

```
#pragma once  
#include "bibl.h"
```

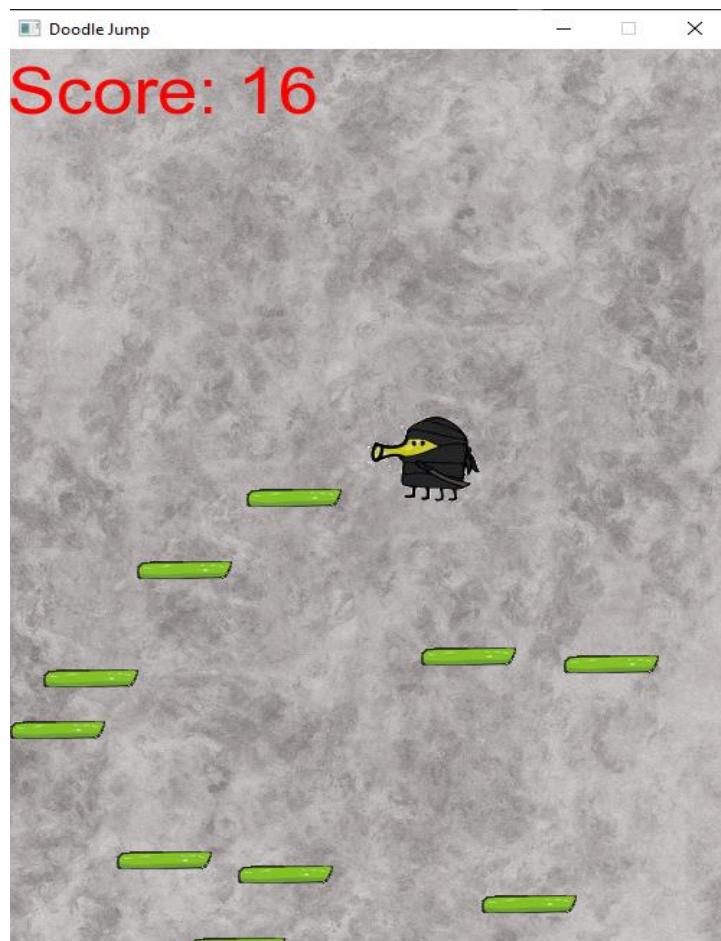
```
class Game  
{  
public:  
  
    void Run();  
  
private:  
  
};
```

Приложение Б

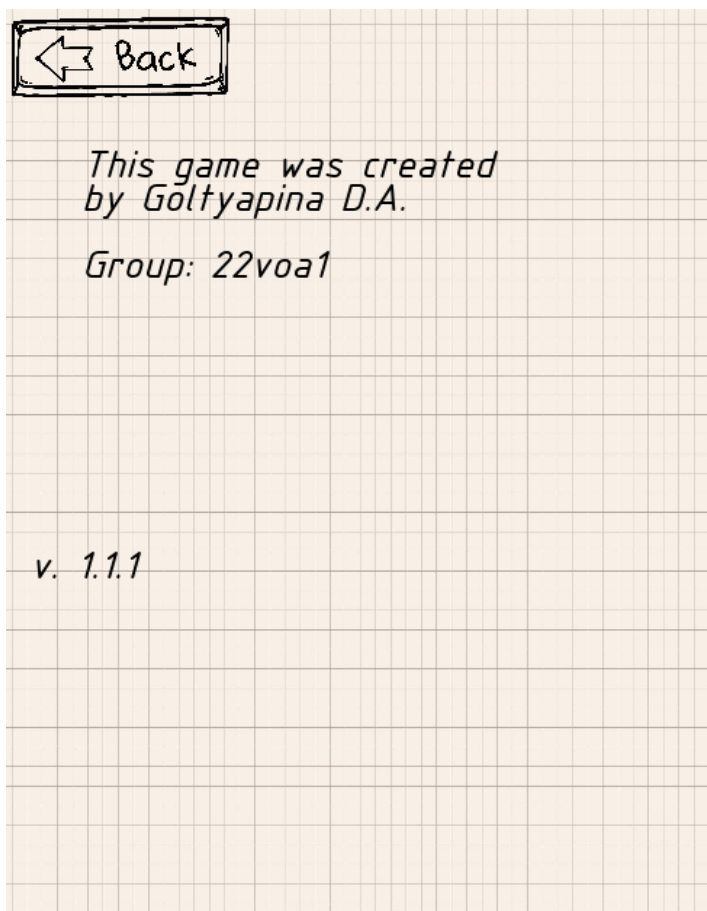
Снимки экрана

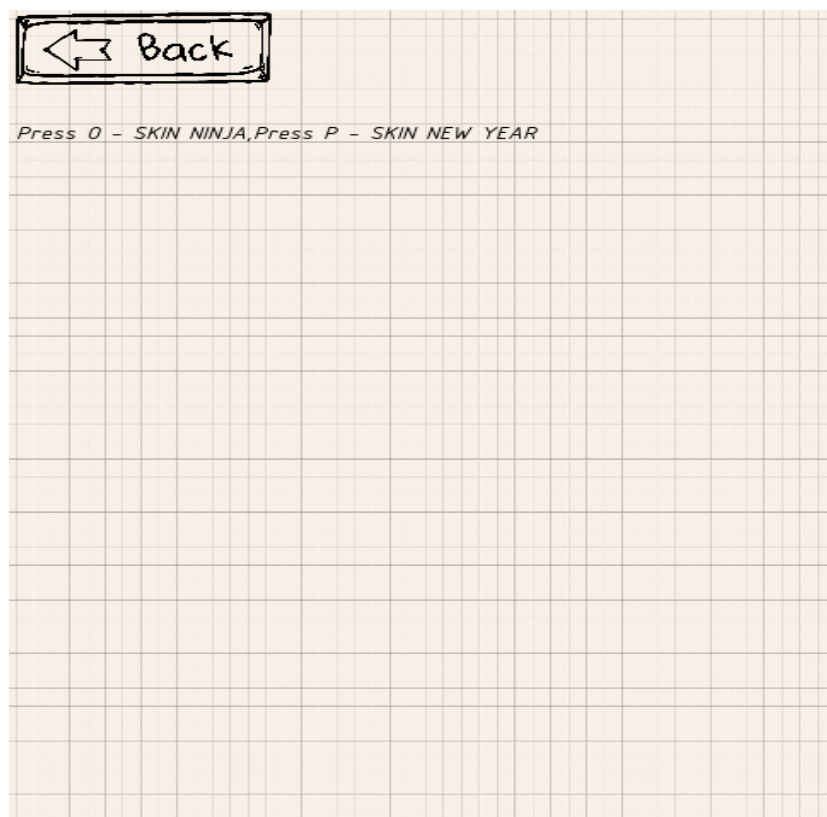


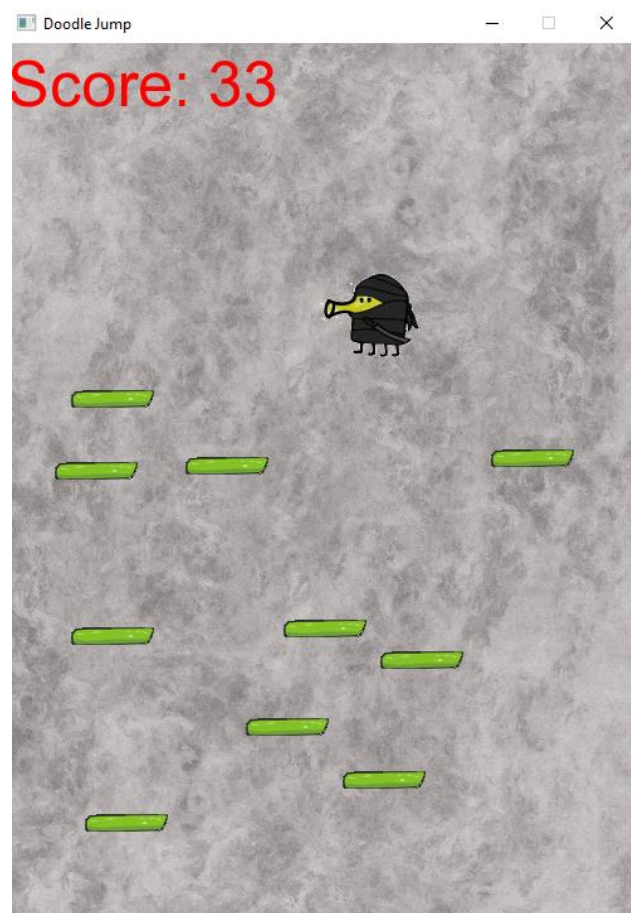
Рисунок 21 – Меню игры



46







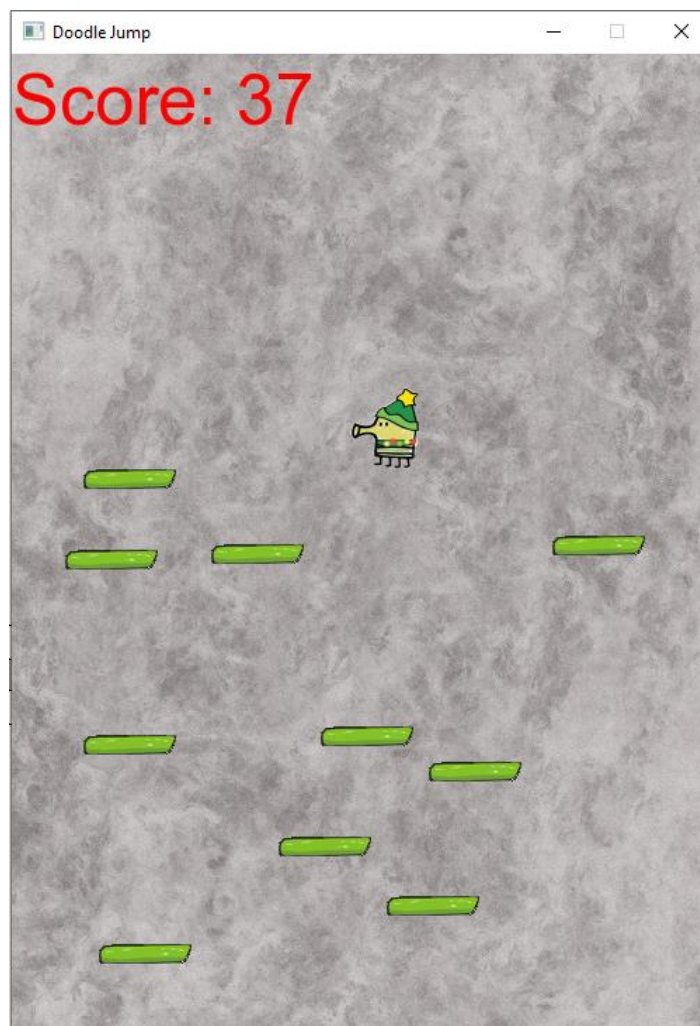


Рисунок 26 – Скин NEW YEAR



Рисунок 27 – Завершение игры

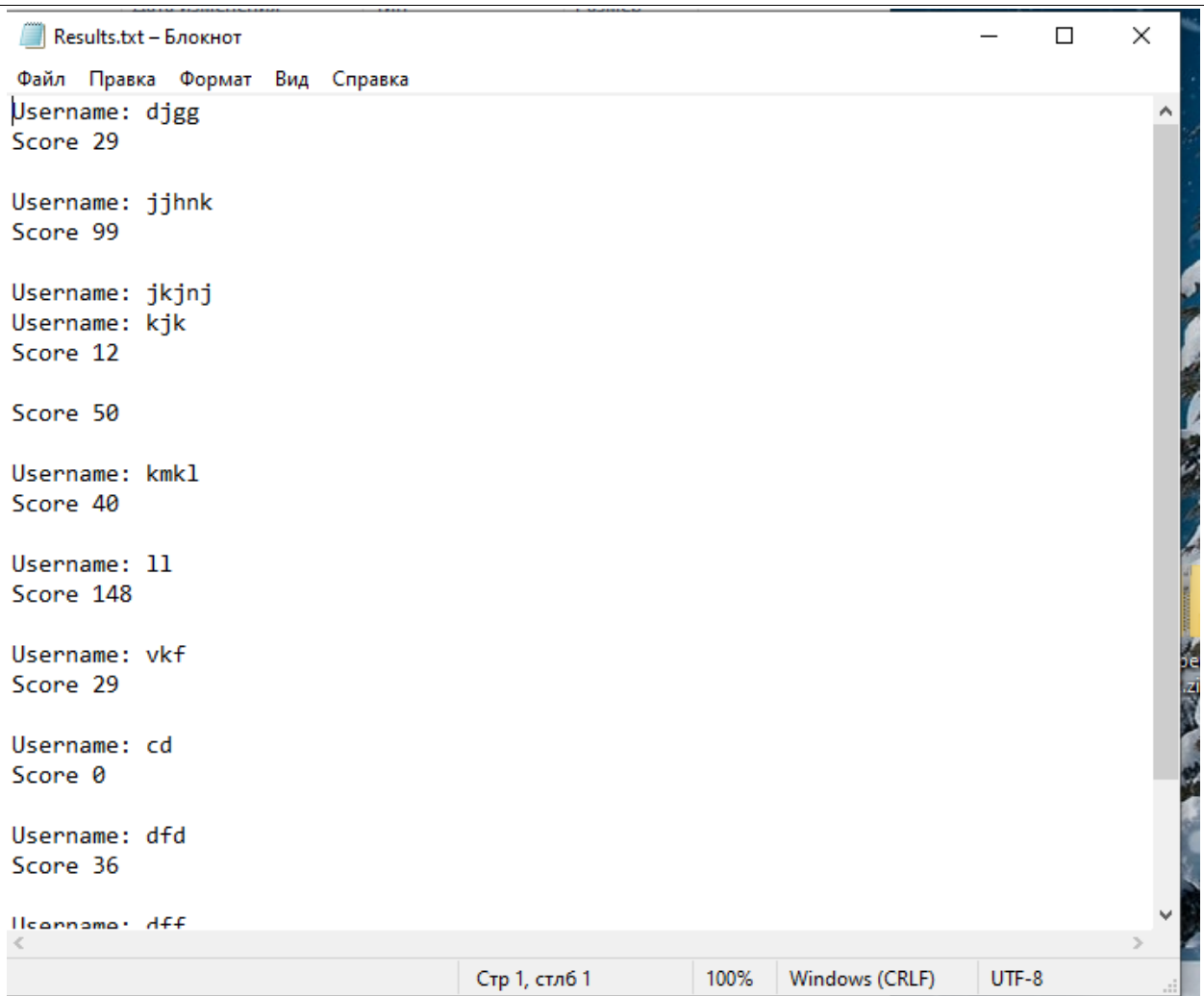


Рисунок 28 – Таблица игроков