|  |  |
| --- | --- |
|  | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова» (БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
| БГТУ.СМК-Ф-4.2-К5-02 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Факультет |  | О |  | Естественнонаучный |
|  |  | шифр |  | наименование |
| Кафедра |  | О7 |  | Информационные системы и программная инженерия |
|  |  | шифр |  | наименование |
| Дисциплина | Информационные технологии и программирование | | | |

КУРСОВАЯ РАБОТА

на тему

|  |
| --- |
| Объектно-ориентированная разработка программ с графическим |
| пользовательским интерфейсом «снизу-вверх» |
| Вариант: Игра «Сбор урожая» |

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил студент группы | | И924Б | |
| Фокин В. К. | |  | |
| Фамилия И.О. | |  | |
| **РУКОВОДИТЕЛЬ** | | | |
| Васюков В. М. |  | |  |
| Фамилия И.О. |  | | Подпись |
| Оценка | | | |
| « » | | 2023 г. | |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc138844158)

[1 Постановка задачи 4](#_Toc138844159)

[2 Описание иерархии классов 6](#_Toc138844160)

[2.1 Класс menu 6](#_Toc138844161)

[2.2 Класс pause 7](#_Toc138844162)

[2.3 Класс entity 9](#_Toc138844163)

[2.4 Класс basket 10](#_Toc138844164)

[2.5 Класс fruit 11](#_Toc138844165)

[2.6 Демонстрация иерархии классов 11](#_Toc138844166)

[3 Использованные ресурсы и сторонние библиотеки 14](#_Toc138844167)

[4 Демонстрация работы 16](#_Toc138844168)

[ЗАКЛЮЧЕНИЕ 22](#_Toc138844169)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc138844170)

[ПРИЛОЖЕНИЕ А 25](#_Toc138844171)

# **ВВЕДЕНИЕ**

Целью курсовой работы является разработка интерактивного графического приложения с использованием мультимедийной библиотеки SFML. SFML представляет собой кроссплатформенную мультимедийную библиотеку, предоставляющую программистам доступ к функциям ввода, вывода видео и звука, необходимым для создания игр и приложений на различных операционных системах.

В ходе исследования будет изучена архитектура и функциональность библиотеки SFML, а также ее основные компоненты, такие как управление окном, рисование графики, обработка пользовательского ввода. Для достижения поставленной цели необходимо решить следующие задачи:

* описать основные требования к разрабатываемому приложению;
* составить иерархию классов;
* разработать приложение;
* продемонстрировать работоспособность приложения.

# **Постановка задачи**

Игра «Сбор урожая».

Правила игры. Сверху произвольным образом (в разных местах экрана и с разной скоростью) падают фрукты: яблоки и груши, красные и зеленые, крупные и мелкие, целые, червивые и совершенно гнилые. Управляя корзинкой, игрок должен собрать как можно больше фруктов. За крупные спелые фрукты начисляется большее количество очков, за мелкие зеленые – меньшее. За червивые фрукты очки убавляются. Изначально игроку дается 5 жизней. Каждая пойманная гнилушка убавляет 1 жизнь. Игра заканчивается, когда не осталось жизней. Результатом является количество набранных очков. 10 лучших результатов хранятся в файле и выводятся на экран по запросу пользователя.

Программа при выполнении должна иметь графический интерфейс, включающий заставку, меню, справку и средства для управления программой.

Заставка должна выводится на экран при запуске программы и давать представление о ней.

Главное меню должно в обязательном порядке включать в себя пункты «Запустить», «Справка», «Выход», «Смена игрока» и «Рекорды».

При выборе пункта «Запустить» запускается игровой процесс, создаётся файл и т. д. в зависимости от варианта.

При выборе пункта «Справка» выводится информация о возможностях программы/правилах игры, а также описываются действия, которые нужно выполнить для достижения определённых целей.

При выборе пункта «Выход» приложение закрывается.

При выборе пункта «Смена игрока» предлагается ввести новое имя игрока.

При выборе пункта меню «Просмотр результатов» на экран должно в табличном виде выводиться содержимое двоичного файла результатов и подменю, позволяющее выполнить следующие действия: «очистить список» (при выборе данного пункта очищается файл результатов), «подготовить файл для печати» (при выборе данного пункта создается текстовый файл с данными для печати), «вернуться в главное меню».

В процессе игры на экране должна присутствовать справочная строка с актуальной для текущего момента информацией.

Взаимодействия пользователя с приложением осуществляется с помощью мыши и/или клавиатуры.

Во время игры должна быть возможность поставить её на паузу или завершить. Если реализована пауза, то из неё или идет продолжение, или выход в главное меню с завершением.

Десять лучших результатов (очки, минуты и т. п.) сохраняются в файле. Для каждого результата должно быть сохранено имя пользователя и очки.

Выполнение курсовой работы должно быть основано на принципах объектно-ориентированного программирования. В работе должна быть построена и обоснована иерархия классов, описывающих предметную область конкретной задачи.

Программа должна являться графическим приложением, написанным на языке C++ с использованием библиотеки SDL2 или SFML.

# **Описание иерархии классов**

В процессе разработки приложения были созданы следующие классы:

* menu – класс, описывающий логику работы и вывода главного меню;
* pause – класс, описывающий логику работы и вывода меню паузы;
* entity – класс, описывающий основные поля и методы классов сущностей;
* basket – класс, описывающий поведение управляемого игроком персонажа;
* fruit – класс, описывающий поведение собираемых игроком объектов.

Далее будет приведено описание членов каждого из этих классов.

## **2.1 Класс menu**

Данный класс представлен на рисунке 1.

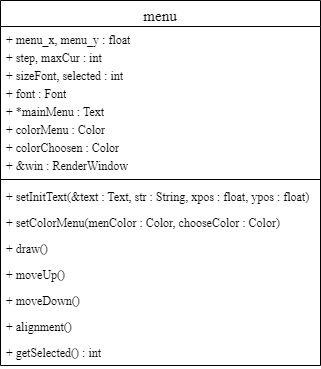


Рисунок 1 – Класс menu

Данный класс имеет следующие публичные поля:

* float menu\_x, menu\_y – поля, содержащие координаты пунктов меню;
* int step – поле, содержащее значение расстояние между пунктами меню;
* int maxCur – поле, содержащее количество пунктов меню;
* int sizeFont – поле, содержащее размер шрифта;
* int selected – поле, содержащее номер выбранного пункта меню;
* Font font – поле, содержащее шрифт, использованный для реализации пунктов меню;
* Text\* mainMenu – поле, содержащее указатель на название пунктов меню;
* Color colorMenu – поле, содержащее цвет пунктов меню;
* Color colorChoosen – поле, содержащее цвет выбранных пунктов меню;
* RenderWindow& win – поле, содержащее ссылку на окно для вывода меню.

Также класс содержит следующие публичные методы:

* void setInitText(Text& text, String str, float xpos, float ypos) – метод, инициализирующий текст меню;
* void setColorMenu(Color menColor, Color chooseColor) – метод, устанавливающий цвет для всех пунктов меню, в том числе выбранного;
* void draw() – метод, выводящий пункты меню;
* void moveUp()– метод, позволяющий перемещаться между пунктами меню вверх;
* void moveDown() – метод, позволяющий перемещаться между пунктами меню вниз;
* void alingment() – метод, выравнивающий кнопки меню;
* int getSelected() – метод, возвращающий выбранный пункт меню.

## **2.2 Класс pause**

Данный класс представлен на рисунке 2.

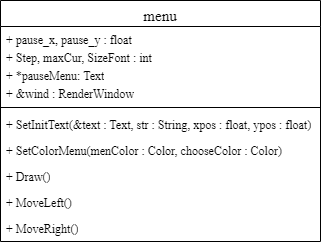


Рисунок 2 – Класс pause

Данный класс содержит следующие публичные поля:

* float pause\_x, pause\_y – поля, содержащие координаты пунктов меню;
* int Step – поле, содержащее значение расстояние между пунктами меню;
* int MaxCur – поле, содержащее количество пунктов меню;
* int SizeFont – поле, содержащее размер шрифта;
* Text\* pauseMenu – поле, содержащее указатель на название пунктов меню;
* RenderWindow& wind – поле, содержащее ссылку на окно для вывода.

Также класс содержит следующие публичные методы:

* void SetInitText(Text& text, String str, float xpos, float ypos) – метод, инициализирующий текст меню;
* void SetColorMenu(Color menColor, Color chosencolor) – метод, устанавливающий цвет для всех пунктов меню, в том числе выбранного;
* void Draw() – метод, выводящий пункты меню;
* void MoveUp() – метод, позволяющий перемещаться между пунктами меню вверх;
* void MoveDown() – метод, позволяющий перемещаться между пунктами меню вниз.

## **2.3 Класс entity**

Данный класс представлен на рисунке 3.

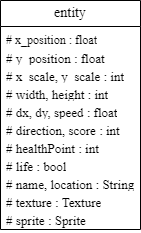


Рисунок 3 – Класс entity

Данный класс содержит следующие защищенные поля:

* float x\_position, y\_position – поля, содержащие координаты объекта;
* int width, height – поля, содержащие значения ширины и высоты;
* int x\_scale, y\_scale – поля, содержащие значения для увеличения объекта в размере;
* float dx, dy – поля, содержащие значения ускорения объекта по оси X и оси Y соответственно;
* float speed – поле, содержащее значение скорости объекта;
* int direction – поле, содержащее значения направления объекта;
* int score – поле, содержащее количество очков, свойственное объекту;
* int healthPoint – поле, содержащее количество очков здоровья, свойственное объекту;
* bool life – поле, обозначающее, достаточно ли очков здоровья у объекта;
* String name – поле, содержащее никнейм объекта;
* String location – поле, содержащее название файла с текстурой объекта;
* Texture texture – поле, содержащее текстуру объекта;
* Sprite sprite – поле, содержащее спрайт объекта;

## **2.4 Класс basket**

Данный класс представлен на рисунке 4.

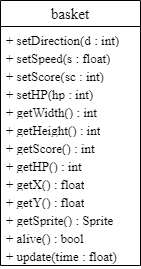


Рисунок 4 – Класс basket

Данный класс содержит следующие публичные методы:

* void setDirection(int d) – метод, устанавливающий направление объекту;
* void setSpeed(float s) – метод, устанавливающий скорость объекту;
* void setScore(int sc) – метод, устанавливающий количество очков объекту;
* void setHP(int hp) – метод, устанавливающий количество очков здоровья объекту;
* int getWidth() – метод, возвращающий ширину объекта;
* int getHeight() – метод, возвращающий высоту объекта;
* int getScore() – метод, возвращающий количество очков объекта;
* int getHP() – метод, возвращающий количество очков здоровья объекта;
* float getX() – метод, возвращающий положение объекта на оси X;
* float getY() – метод, возвращающий положение объекта на оси Y;
* Sprite getSprite – метод, возвращающий спрайт объекта;
* bool alive() – метод, возвращающий состояние объекта относительно очков здоровья;
* void update(float time) – метод, обновляющий состояние объекта, то есть обрабатывает логику и выводит на экран.

## **2.5 Класс fruit**

Данный класс представлен на рисунке 5.

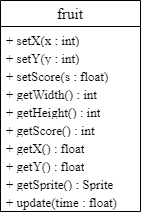


Рисунок 5 – Класс fruit

Данный класс содержит следующие публичные методы:

* void setX(int x) – метод, устанавливающий положение объекта на оси X;
* void setY(int y) – метод, устанавливающий положение объекта на оси Y;
* void setSpeed(float s) – метод, устанавливающий скорость объекта;
* int getWidth() – метод, возвращающий ширину объекта;
* int getHeight() – метод, возвращающий высоту объекта;
* int getScore() – метод, возвращающий количество очков объекта;
* float getX() – метод, возвращающий положение объекта на оси X;
* float getY() – метод, возвращающий положение объекта на оси Y;
* Sprite getSprite – метод, возвращающий спрайт объекта;
* void update(float time) – метод, обновляющий состояние объекта, то есть обрабатывает логику и выводит на экран.

## **2.6 Демонстрация иерархии классов**

Разработанная иерархия сущностей наглядно продемонстрирована на диаграмме, приведённой на рисунке 6.

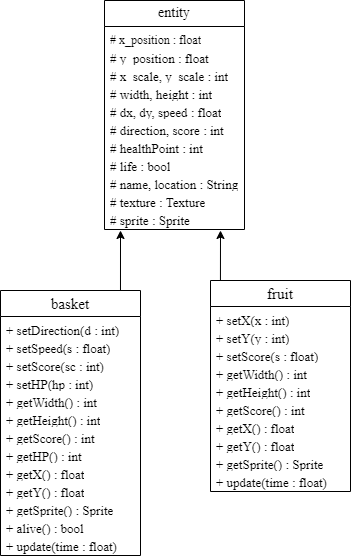


Рисунок 6 – Иерархия сущностей

Разработанная иерархия текстовых окон наглядно продемонстрирована на диаграмме, приведённой на рисунке 7.

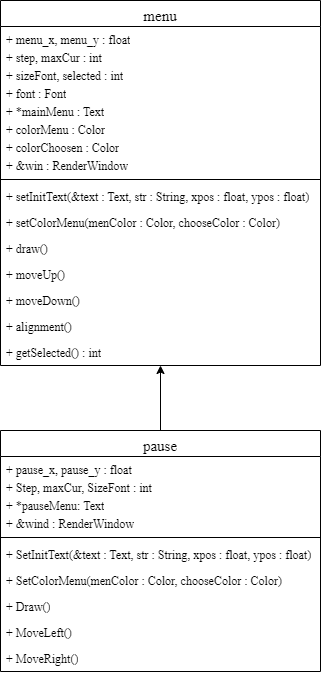


Рисунок 7 – Иерархия текстовых окон

# **3 Использованные ресурсы и сторонние библиотеки**

В процессе работы использованы следующие мультимедийные ресурсы и сторонние библиотеки:

* splash.png – изображение заставки, нарисовано с помощью бесплатного онлайн-редактора pixilart.com [1];
* menu.png – изображение фона меню, нарисовано с помощью бесплатного онлайн-редактора pixilart.com;
* game.png – изображение фона игры, нарисовано с помощью бесплатного онлайн-редактора pixilart.com;
* info.png – изображение фона справки, нарисовано с помощью бесплатного онлайн-редактора pixilart.com;
* records.png – изображение фона таблицы рекордов, нарисовано с помощью бесплатного онлайн-редактора pixilart.com;
* change.png – изображение фона окна изменения имени персонажа, нарисовано с помощью бесплатного онлайн-редактора pixilart.com;
* basket.png – изображение корзинки, нарисовано с помощью бесплатного онлайн-редактора pixilart.com;
* bugApple.png – изображение яблока с червячком, нарисовано с помощью бесплатного онлайн-редактора pixilart.com;
* rottenTang.png – изображение гнилого мандарина, нарисовано с помощью бесплатного онлайн-редактора pixilart.com;
* cherry.png – изображение вишенки, загружено с сайта itch.io [2];
* dragon.png – изображение драконьего фрукта, загружено с сайта itch.io;
* grape.png – изображение винограда, загружено с сайта itch.io;
* melon.png – изображение арбуза, загружено с сайта itch.io;
* pepper.png – изображение перца, загружено с сайта itch.io;
* pumpkin.png – изображение тыковки, загружено с сайта itch.io;
* avatar.png – фотография друга, загружена с мессенджера;
* capybara.png – картинка капибары, загружена с сайта artfile.ru [3];
* pixelFont.ttf – шрифт Pixeloid Sans, загружен с сайта ofont.re [4];
* SFML – загружена с официального сайта SFML [5].

# **4 Демонстрация работы**

При запуске игры показывается заставка, что показано на рисунке 8.



Рисунок 8 – Заставка

Затем появляется главное меню, что показано на рисунке 9.



Рисунок 9 – Главное меню

При нажатии на пункт меню «Старт» запускается игра. Если не ввести имя в «Игроки», то присваивается имя «Гость». Игрок, управляя корзиной, должен набрать как можно больше очков, собирая свежие фрукты, овощи и ягоды. При сборе червивых или гнилых фруктов очки и жизни убавляются, также как и при пропуске свежих плодов. Со временем скорость фруктов увеличивается, что усложняет сбор. Игровой процесс показан на рисунке 10.

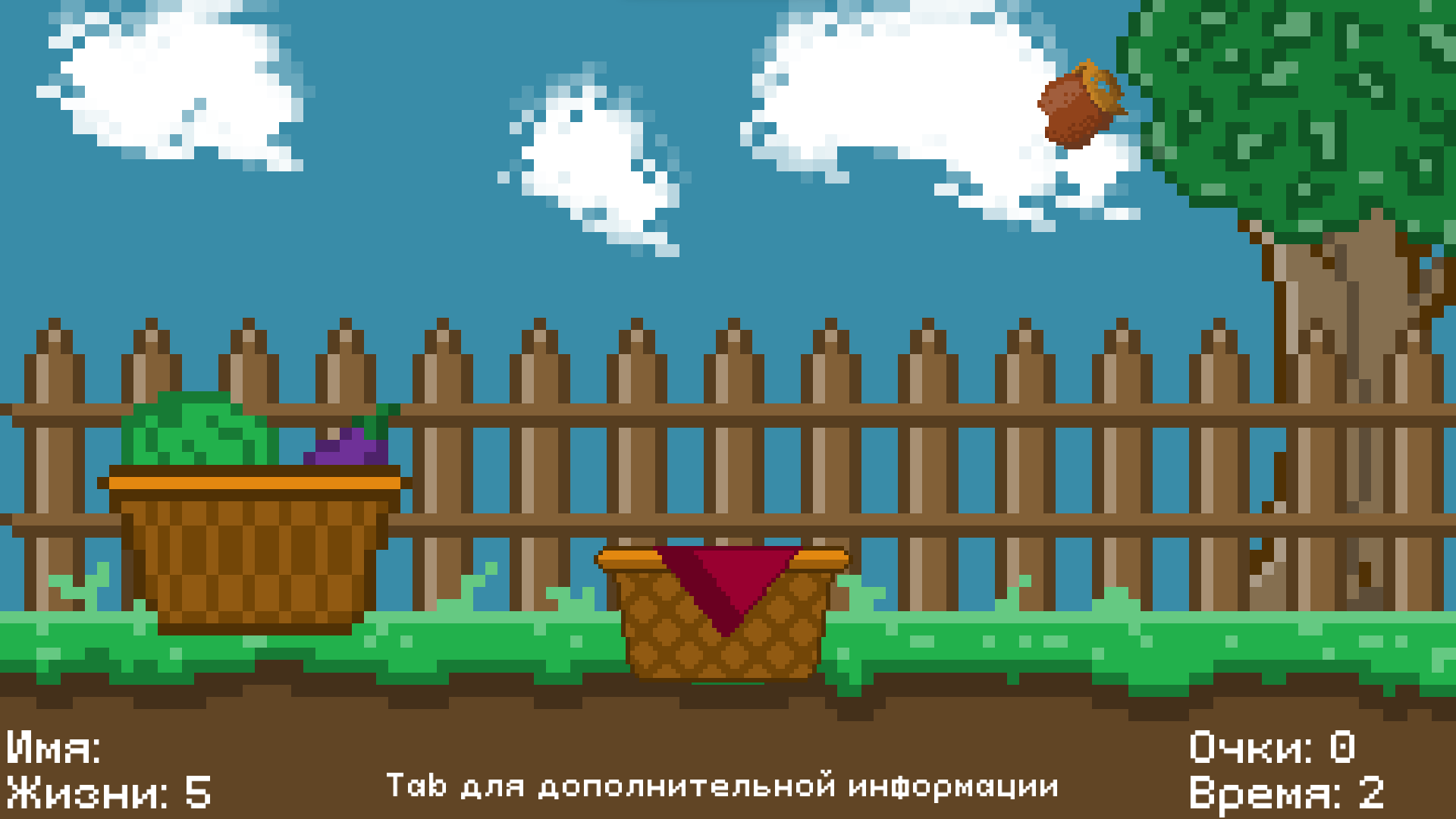


Рисунок 10 – Игровой процесс

При нажатии на Tab во время игрового процесса будут выведены управление и правила игры, что показано на рисунке 11.

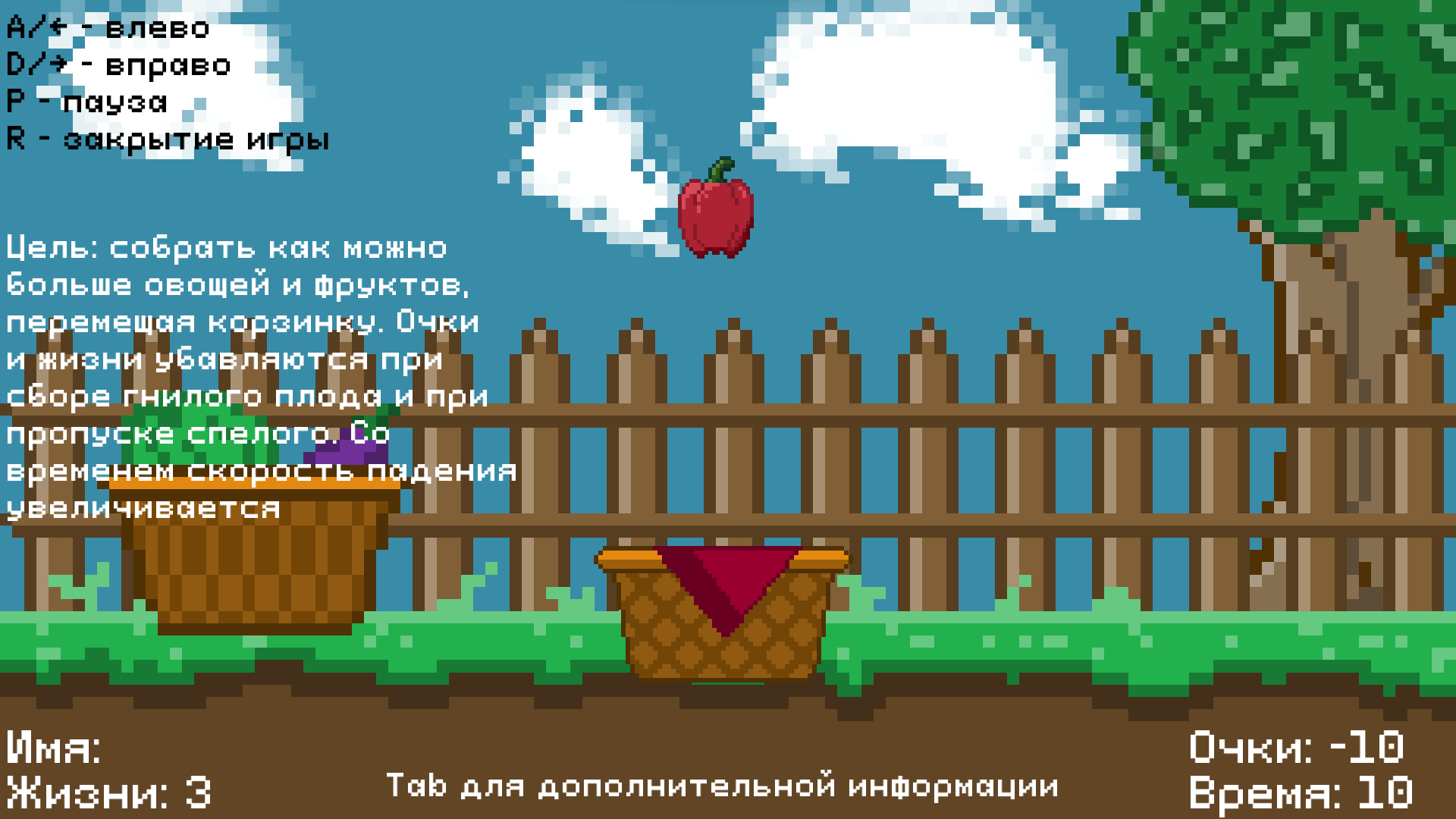


Рисунок 11 – Справочная информация

Если кончаются жизни, появляется заставка конца игры и происходит возврат в главное меню. Заставка показана на рисунке 12.



Рисунок 12 – Заставка конца игры

Также в игре реализована пауза при нажатии на P. При выборе пункта «Рестарт» игра начнется заново, при выборе пункта «Выход в меню» откроется главное меню. При повторном нажатии на P игра продолжится. Меню паузы показано на рисунке 13.



Рисунок 13 – Пауза

При нажатии в главном меню на пункт «Справка» появится окно, где описаны управление и правила игры, что показано на рисунках 14 и 15.



Рисунок 14 – Справка, первая страница



Рисунок 15 – Справка, последняя страница

При выборе пункта меню «Рекорды» появится список рекордов, содержащий 10 лучших результатов. При нажатии на клавишу «S» список будет подготовлен к печати, то есть создается .txt файл, содержащий лучшие результаты. При нажатии на «С» список будет очищен. Окно пункта меню показано на рисунках 16 и 17, файл для печати показан на рисунке 18.



Рисунок 16 – Список рекордов



Рисунок 17 – Подготовка файла к печати

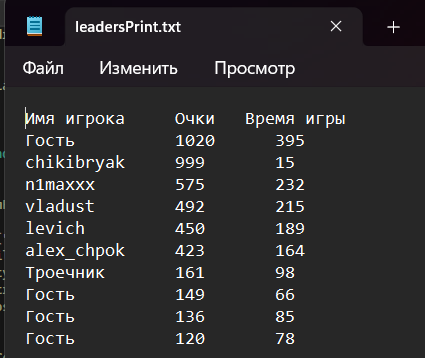


Рисунок 18 – Файл для печати

При выборе пункта меню «Игроки» появится окно, где можно задать имя персонажу. Имя должно содержать не более 10 символов. Окно «Игроки» показано на рисунке 19.

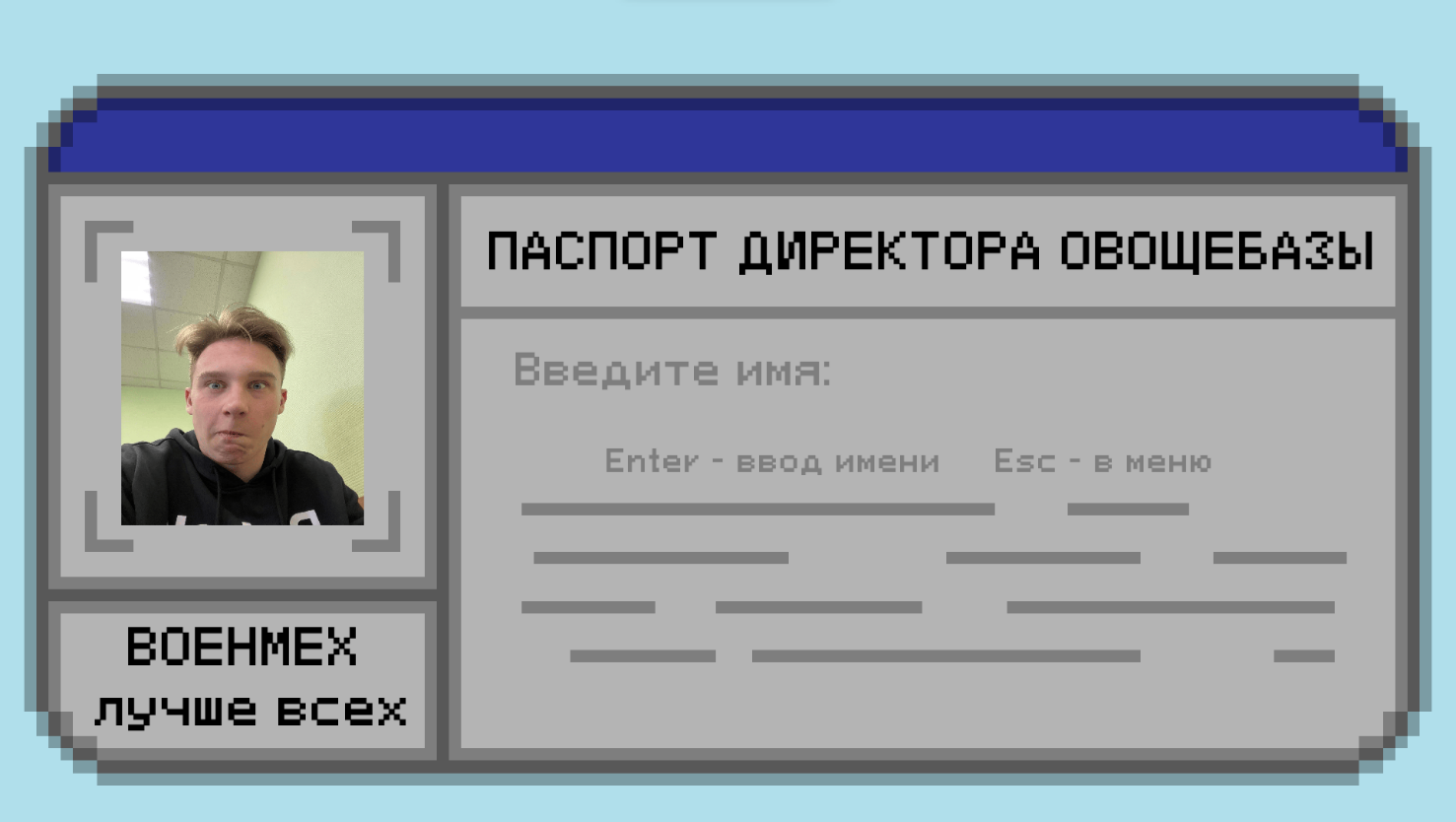


Рисунок 19 – Окно смены имени игрока

При выборе пункта «Выход» работа программы завершится.

# **ЗАКЛЮЧЕНИЕ**

В результате работы была реализована игра «Сбор урожая».

Были решены следующие задачи:

* описаны основные требования к разрабатываемому приложению;
* составлена иерархия классов, реализованы все требуемые поля и методы;
* разработано приложение, реализованы все указанные окна и функции;
* продемонстрирована работоспособность приложения.

При запуске приложения открывается главное меню, содержащее три кнопки. Из главного меню пользователь может попасть в другие окна путем нажатия соответствующих кнопок.

Окно «Об игре» содержит информацию об авторе и краткое описание игры. Присутствует возможность выйти в главное меню с помощью нажатия соответствующей кнопки.

Окно «Рекорды» содержит таблицы 10 лучших игр. Реализованы возможности возврата в меню, подготовки файлов к печати, очистки таблиц лидеров.

Окно игры реализует возможности управления корзинкой, подбора плодов с целью получения очков, просмотра дополнительной информации в виде управления и правил игры, есть возможность паузы, перезапуска игры и выхода в главное меню.

Выполнение курсовой работы основано на принципах объектно-ориентированного программирования. В работе была построена и обоснована иерархия классов, описывающих предметную область конкретной задачи.

Программа является графическим приложением, написанным на языке C++ с использованием библиотеки SFML, исходные тексты которой приведены в Приложении А. Отображаемые на экране элементы пользовательского интерфейса принадлежат к одной из двух иерархий классов: «сущности» или «меню». Предполагается возможность использования объектов одной иерархии как полиморфных.

Все задачи курсовой работы выполнены, цель – достигнута.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Бесплатный онлайн-редактор. – URL: <https://www.pixilart.com/> (дата обращения 22.06.2023).
2. Набор спрайтов овощей, фруктов и ягод. – URL: <https://helm3t.itch.io/free32x32-fruits-and-veggies-asset-packs> (дата обращения 23.06.2023).
3. Изображение капибары. – URL: <https://www.artfile.ru/i.php?i=654925> (дата обращения 22.06.2023).
4. Шрифт Pixeloid Sans. – URL: <https://ofont.ru/view/5693> (дата обращения 22.06.2023).
5. Simple and Fast Multimedia Library. – URL: https://www.sfml-dev.org/download.php (дата обращения 18.06.2023).

# **ПРИЛОЖЕНИЕ А Текст программы**

Исходные тексты программы располагаются на прилагаемом электронном носителе.