**Введение**

В рамках курсового проектирования передо мной стоит задача разработать игровое приложение «Виселица».

Цель проекта - создать интерактивное приложение, которое будет не только развлекать пользователя, но и способствовать развитию его лексического запаса и мышления.

Игра «Виселица» является классической угадайкой слов, где игроку предстоит отгадывать слова по буквам. Это приложение будет полезно широкому кругу пользователей: от школьников, изучающих новую лексику, до взрослых, желающих проверить свои знания или просто интересно провести время.

В первом разделе «Анализ задачи» пояснительной записки будет проведен тщательный анализ задачи. Это включает в себя исследование предметной области, определение функциональных и нефункциональных требований к приложению, а также анализ существующих аналогов. Будет уделено внимание изучению потребностей и предпочтений целевой аудитории, чтобы обеспечить максимальную вовлеченность пользователей.

В разделе «Проектирование» будут рассмотрены архитектура приложения и дизайн пользовательского интерфейса, а также подходы к организации данных и алгоритмы обработки информации.

В разделе «Реализация» будет описан процесс программирования и интеграции компонентов приложения, а также тестирование функциональности и пользовательского интерфейса.

В разделе «Применение» будет рассмотрено, как приложение может быть использовано в различных контекстах, включая образовательные учреждения и домашнее использование. Будут представлены сценарии использования приложения, которые покажут его гибкость и адаптивность к потребностям разных пользователей.

Заключение В заключении будет подведен итог всему проекту. Будут описаны достигнутые результаты, преодоленные трудности и полученные знания в процессе разработки. Также будет дана оценка потенциального влияния приложения на целевую аудиторию и возможности его дальнейшего развития.

Таким образом, данный проект направлен на создание полноценного и функционального приложения «Виселица», которое будет отвечать современным требованиям к мобильным играм и обеспечит пользователю интересное и познавательное времяпрепровождение.

Схема работы системы будет представлена в графической части.

**1 Анализ задачи**

* 1. **Постановка задачи**

Игровое приложение «виселица» — это увлекательное лингвистическое развлечение, которое позволяет игрокам испытать свои знания в области словесности и орфографии. Эта игра, имеющая долгую историю и популярность во многих странах, представляет собой не только способ развлечения, но и эффективный метод обучения и запоминания новых слов.

Принцип игры заключается в следующем: перед игроком стоит задача угадать заранее загаданное слово, выбирая отдельные буквы алфавита. Если предполагаемая буква содержится в слове, она открывается и занимает соответствующие позиции в загаданном слове, что дает игроку дополнительные подсказки для дальнейших попыток. В случае ошибки, когда выбранная буква отсутствует в слове, начинается поэтапное создание изображения виселицы — символического представления неудачных попыток. С каждой ошибкой к рисунку добавляется новый элемент, что увеличивает напряжение и заставляет игрока быть более внимательным и обдуманным в своих действиях.

Игра продолжается до тех пор, пока не будет достигнут один из двух возможных исходов: либо игрок угадает все буквы и тем самым раскроет загаданное слово, либо, постепенно допуская ошибки, он дополнит рисунок до полной «виселицы», что означает проигрыш. Таким образом, «виселица» не только развлекает, но и стимулирует умственную активность, тренирует память и внимание, а также расширяет словарный запас игроков. Это делает ее идеальной для семейных вечеров, образовательных игр в школах и даже для использования в качестве инструмента для изучения иностранных языков.

Автоматизация игрового приложения «виселица» в цифровом формате открывает новые горизонты для любителей этой классической головоломки. В эпоху цифровых технологий, когда доступ к информации и развлечениям стал проще и удобнее, игра “виселица” претерпела значительные изменения, став более интерактивной и доступной для игроков по всему миру.

Разработка интерфейса для цифровой версии игры включает создание удобной и интуитивно понятной платформы для ввода букв. Это может быть как простое текстовое поле для ввода с клавиатуры, так и более сложная система с использованием сенсорного экрана или даже голосовых команд. Важно, чтобы интерфейс был максимально адаптирован под потребности пользователя, обеспечивая легкость в использовании и понимании.

Программная логика игры “виселица” в цифровом формате требует тщательной проработки. Она должна не только отслеживать попытки пользователя и правильно отображать буквы в слове, но и управлять игровым процессом, включая начало и конец игры, подсчет очков и предоставление подсказок.

Периодичность использования данного продукта неограничена. Игровое приложение обладает уникальной гибкостью в плане периодичности использования, что делает её идеальным выбором как для образовательных учреждений, так и для домашнего досуга. В образовательной среде игра может служить в качестве дополнительного инструмента для изучения новых слов и улучшения орфографических навыков. Учителя могут включать “виселицу” в учебный план, используя её для ежедневных или еженедельных занятий, чтобы стимулировать интерес учащихся к языку и расширять их словарный запас.

Для личного использования «виселица» предлагает бесконечные возможности развлечения и самообразования. Игроки могут практиковать игру ежедневно, используя мобильные приложения или онлайн-платформы, что позволяет им улучшать свои лингвистические способности в любое время и в любом месте. Также игровое приложение может быть прекрасным способом для семейного времяпрепровождения в выходные, объединяя членов семьи для совместной игры и обучения.

В цифровую эпоху «виселица» стала ещё более доступной, благодаря чему каждый человек может самостоятельно выбирать, как часто и в каких условиях он хочет играть. Это может быть короткая игра во время перерыва на работе или длительная сессия в кругу друзей во время вечеринки. В любом случае, «виселица» остаётся универсальной игрой, способной адаптироваться к различным потребностям и предпочтениям пользователей.

Новая программа значительно упрощают и обогащают игровой процесс, делая его более интерактивным и привлекательным для пользователей всех возрастов. Автоматическая проверка букв и визуальное отображение прогресса игры являются ключевыми аспектами, которые повышают удобство и увлекательность игры.

Автоматическая проверка букв позволяет игрокам сосредоточиться на выборе правильных букв, не беспокоясь о необходимости вручную проверять каждую букву. Программа мгновенно сообщает игроку, присутствует ли выбранная буква в загаданном слове, что ускоряет игровой процесс и делает его более динамичным. Это также помогает избежать ошибок, связанных с человеческим фактором, таких как неправильное подсчет ошибок или забывчивость относительно уже использованных букв.

Визуальное отображение прогресса игры добавляет визуальный элемент, который делает игру более захватывающей. Игроки могут видеть, как постепенно формируется изображение виселицы с каждой ошибкой, что добавляет элемент напряжения и азарта. Кроме того, визуализация угаданных букв и оставшихся пустых мест в слове помогает игрокам лучше понимать свои шансы на успех и стратегически подходить к выбору следующих букв.

Игровое приложение предлогает различные уровни сложности и темы, что делает игру подходящей для разнообразной аудитории. Они также могут включать функции, такие как подсказки или возможность выбора категорий слов, что делает игру более образовательной и адаптируемой к интересам конкретного пользователя.

Существует множество цифровых версий игры «виселица», включая мобильные приложения и веб-сайты. Новая разработка должна предложить уникальные функции или улучшенный пользовательский опыт, чтобы выделиться на фоне существующих аналогов.

Цель курсового проекта по созданию игрового приложения «Виселица» заключается в разработке интерактивного и образовательного инструмента, который предлагает как развлечение, так и обучение для пользователей разных возрастов. Основные задачи проекта включают:

* Разработка удобного пользовательского интерфейса: Создание интуитивно понятного и привлекательного интерфейса, который позволит пользователям легко взаимодействовать с приложением и погружаться в игровой процесс.
* Интеграция образовательных элементов: Включение функций, которые помогут пользователям расширять свой словарный запас и улучшать орфографические навыки через игровую форму.
* Реализация программной логики: Создание надежной программной логики, которая будет отслеживать ход игры, проверять выбор букв и отображать прогресс игрока.
* Создание динамичного и захватывающего контента: Внедрение разнообразных слов и категорий, а также разработка анимаций, которые сделают игру более интересной и динамичной.

В конечном итоге, целью проекта является создание игрового приложения «Виселица», которое будет не только развлекать, но и способствовать обучению и развитию когнитивных навыков у игроков всех возрастов

**Выходная информация**

**Способы вывода данных:**

* Экранное отображение текущего состояния слова с прочерками на месте неугаданных букв.
* Визуализация «виселицы» с каждой неправильной попыткой.
* Отображение списка уже выбранных букв.

**Характеристики выходных наборов данных:**

Слово или фраза для угадывания, представленное в виде серии прочерков.

Количество оставшихся попыток до завершения игры.

Список использованных букв, отображаемый отдельно от игрового поля.

Описание выходных экранных форм:

Интерфейс должен быть интуитивно понятным, с четким разделением между областью ввода букв и игровым полем.

Визуальные эффекты для правильных и неправильных ответов, такие как подсветка или анимация.

**Входная информация**

**Способы ввода данных:**

* Клавиатурный ввод через физическую или экранную клавиатуру.
* Возможность выбора буквы нажатием на экране в случае мобильных устройств.

**Характеристики входных документов:**

Не требуются физические документы, так как ввод осуществляется напрямую пользователем в интерактивном режиме.

**Постоянная информация**

**Нормативно-справочная информация:**

файл со списком слов, из которого программа будет случайным образом выбирать слово для угадывания.

Требования к составу выполняемых функций

* Генерация случайного слова из предварительно определенного списка слов.
* Прием ввода от пользователя и проверка на соответствие выбранной буквы слову.
* Обновление отображения слова и «виселицы» в соответствии с выбором пользователя.
* Отслеживание количества неправильных попыток и завершение игры при достижении максимального числа.
* Возможность начать новую игру или выбрать другой уровень сложности.
  1. **Инструменты разработки**

Для разработки данного игрового приложения будет использовать среда **Delphi 12**.

Некоторые ключевые особенности и улучшения, которые были внедрены в Delphi 12:

* Поддержка различных платформ: Delphi 12 позволяет разработчикам создавать приложения для Windows, Android, iOS, macOS и Linux, используя единую кодовую базу, что значительно упрощает процесс разработки и поддержки приложений.
* Современный пользовательский интерфейс: Delphi 12 предлагает обновленные элементы управления UI и поддержку API WinRT, а также функции HighDPI, что особенно актуально для приложений Windows 11.
* Работа с базами данных: Delphi 12 предоставляет возможность подключения к более чем 20 базам данных нативно с помощью FireDAC для высокоскоростного прямого доступа.
* Улучшения в написании кода: Умная навигация по коду для рефакторинга, автодополнение кода, выделение неактивного кода в редакторе кода и осведомленность LSP о файлах Include.
* Поддержка облачных технологий: Возможность вызова REST-сервисов или компонентов AWS и Azure с помощью библиотек HTTPS и REST клиента.
* RAD Server: Упрощение разработки и развертывания многозадачного готового к использованию сервера REST API приложений с поддержкой Swagger.

Эти улучшения делают Delphi 12 идеальным инструментом для разработчиков, стремящихся создавать мощные и современные приложения, способные работать на широком спектре устройств и операционных систем.

**Adobe Photoshop CS6 x64** - мощный инструмент для работы с графикой, который позволяет создавать и редактировать изображения и текстуры, которые будут использованы в игре. Photoshop CS6 предлагает широкий спектр инструментов для обработки изображений, включая сложные фильтры и эффекты.

**Microsoft Word 2021** - программа для обработки текстов, которая будет использована для создания документации, необходимых для моего проекта. Word 2021 включает функции совместного редактирования и новые возможности для работы с мультимедийным контентом.

**Microsoft PowerPoint 2021**- программа для создания презентаций, которая будет использована для демонстрации концепции игры, её особенностей и плана разработки. PowerPoint 2021 предлагает улучшенные функции записи слайд-шоу и новые инструменты для рисования.

**Dr.Explain** – это интуитивно понятный инструмент для создания справочных файлов, руководств пользователя и документации. Программа позволяет сохранять проекты в различных форматах, включая HTML Help, Winhelp, MS Help 2.0 / Visual Studio Help, Browser-based Help, PDF и Word RTF.

**Блокнот** - простой текстовый редактор, для создания txt файлов.

**Smart Install Maker** - утилита для создания инсталляционных пакетов, которая позволяет легко создавать профессионально выглядящие установщики для моего приложения. Smart Install Maker предоставляет возможности для редактирования системного реестра, создания ярлыков, регистрации шрифтов и ActiveX компонентов.

Разработка ведется на ноутбуке Legion. У данного ноутбука следующие

параметры:

* процессор AMD Ryzen 7 5800H
* объем ОЗУ 16 гб
* объем места на HDD – 474 гб
* ОС – Windows 11
  1. **Требования к приложению**

Требования к интерфейсу игрового приложения «Виселица» подчеркивает важность создания пользовательского интерфейса, который не только удовлетворяет эстетические предпочтения и функциональные потребности пользователей, но и обеспечивает интуитивно понятное и эффективное взаимодействие с приложением. Интерфейс является ключевым элементом в восприятии приложения, его удобстве.

Пользовательский интерфейс должен быть разработан так, чтобы он был доступен для широкого круга пользователей, включая людей с различными уровнями технической грамотности и физическими возможностями. Он должен сочетать в себе простоту и функциональность, позволяя пользователям легко навигировать по приложению и выполнять необходимые действия без излишних сложностей.

Требования к интерфейсу:

* Интуитивность: Интерфейс должен быть понятным и легким в использовании для всех возрастных групп.
* Адаптивность: Дизайн должен корректно отображаться на различных размерах экранов и ориентациях.
* Доступность: Приложение должно соответствовать стандартам доступности, чтобы быть использованным людьми с ограниченными возможностями.
* Эстетика: Визуальное оформление должно быть привлекательным и современным.

Эти требования обеспечивают создание интерфейса, который будет не только функциональным и удобным, но и приятным в использовании, что способствует повышению удовлетворенности пользователей и их вовлеченности в игровой процесс.

**2 Проектирование задачи**

Проектирование задачи - это критически важный этап в любом проекте разработки программного обеспечения. Это процесс, в котором мы определяем, что именно должна делать наша программа, как она должна это делать и как она будет взаимодействовать с пользователем.

Проектирование задачи помогает нам уточнить наши цели и ожидания от программы. Оно позволяет нам проработать все детали функциональности, интерфейса и структуры данных до того, как мы начнем писать код. Это помогает предотвратить ошибки и недоразумения в дальнейшем.

Более того, хорошо спроектированная задача облегчает процесс разработки. Она позволяет нам легче понять, какие шаги нам нужно предпринять для реализации нашей программы, и помогает нам организовать нашу работу более эффективно.

**2.1 Организация данных**

В нашем приложении «Виселица» мы будем использовать различные типы данных для обеспечения функциональности игры.

**Список слов:** Самый важный элемент данных - это список слов, которые будут использоваться для игры. Этот список будет загружаться из текстового файла при запуске игры. Каждое слово в этом списке будет представлено в виде строки.

**Текущее слово:** Мы будем использовать отдельную переменную для хранения текущего слова, которое игрок пытается угадать. Это слово будет выбираться случайным образом из нашего списка слов при начале каждой новой игры.

**Состояние игры:** Мы будем использовать несколько переменных для отслеживания текущего состояния игры. Это включает в себя количество неправильных ответов (которое определяет, сколько частей виселицы было нарисовано), а также список уже отгаданных букв. Эти данные помогут нам обновлять интерфейс игры и определять, когда игра закончена.

**Другие параметры игры:** В зависимости от того, какие дополнительные функции мы решим включить в игру, нам может потребоваться использовать дополнительные переменные. Например, организация системы подсчета очков, чтобы игроки могли отслеживать свой прогресс.

**2.2 Процессы**

Основной процесс в игровом приложении - это цикл угадывания букв, который можно разделить на следующие этапы:

**Выбор слова:** В начале каждой игры из списка слов случайным образом выбирается слово. Это слово становится «текущим словом», которое игрок должен угадать.

**Инициализация игры:** на экране отображается ряд пустых ячеек, соответствующих количеству букв в текущем слове. Также устанавливается счетчик ошибок, который отслеживает количество неправильно угаданных букв.

**Цикл угадывания:** Игрок выбирает букву. Если выбранная буква присутствует в текущем слове, то она отображается на соответствующих местах в ряду ячеек. Если выбранная буква отсутствует в текущем слове, то увеличивается счетчик ошибок, и на экране отображается еще одна часть виселицы.

**Проверка условий окончания игры:** после каждого выбора буквы игра проверяет, угаданы ли все буквы в слове или исчерпаны ли все попытки (счетчик ошибок достиг максимального значения). Если одно из этих условий выполнено, игра заканчивается.

**Конец игры:** если игрок угадал все буквы, он побеждает. Если же все попытки исчерпаны и слово не угадано, игра объявляется проигранной. В обоих случаях игроку предлагается начать новую игру.

**2.3 Описание внешнего пользовательского интерфейса**

Интерфейс будет разработан с учетом простоты и интуитивности использования. Он будет включать в себя следующие ключевые элементы:

**Главное меню:** Главное меню будет содержать кнопки начать игру , правила игры, настройки где мы сможем настроить звук, и конечно же выход из нашего приложения.

**Меню уровней:** В данном меню мы сможем выбрать тематику, на которую мы захотим угадывать слова, и уровень игры (легкий, средний, сложный) от которого будет зависеть сложность и длина слова.

**Окно игры:** Окно игры подразумевает само начало игры. В данном окне будет присутствовать панель с нашими буквами с помощью которых мы сможем отгадывать слово, кнопки подсказка (добавит одну букву, будет присутствовать на уровнях после легкого), кнопка выход в главное меню и кнопка начать заново игру. Так же мы будем наблюдать в данном окне анимацию нашего бедного героя, у которого будут появляться постепенно части тела на виселице в соответствии с непопаданием на нужную букву. Будет присутствовать панель, где нам покажет, какие буквы игрок уже использовал. И конечно наш рекорд будет отображаться в виде звездочек, которые будут отниматься по мере того, сколько раз игрок ошибся.

**Правила:** Это окно будет отвечать за правила игры, где игрок сможет их прочитать и уже уверенно начать игру зная что к чему.

Таким образом, нашему игровому приложению нужно будет обеспечить приятное сочетание оттенков для интерфейса и понятность пользования.

Структура навигации по проекту представлена на рисунке 1:

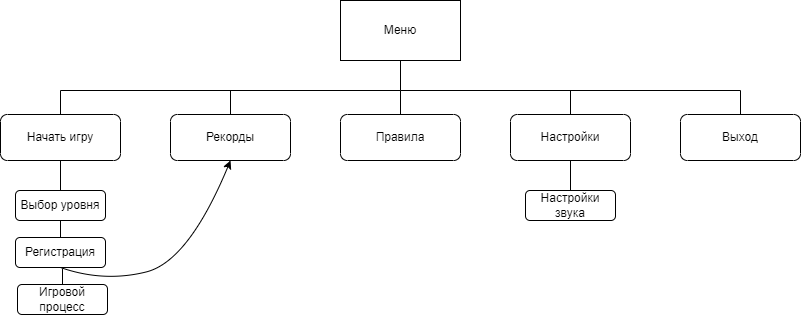


Рисунок 1 – Навигация между окнами программы

**Прототип** — это объект, который является источником тонкостей и особенностей в программировании. Этот термин употребляется для обозначения объекта-примера, который служит основой для создания других подобных объектов. В разных контекстах «прототип» может означать разные вещи: от конкретной личности, послужившей основой для образа в искусстве, до быстрой черновой реализации будущей системы. **Итоговое отображение программы может отличаться от прототипа.** Ниже можно наблюдать прототипы:

1. Логотипа (Рисунок 2)

2. Заставки (Рисунок 3)

3. Главного окна (Рисунок 4)

4. Окна, содержащее меню уровней (Рисунок 5)

5. Окна, содержащее меню уровней развернутое (Рисунок 6)

6. Окна, содержащее настройки звука (Рисунок 7)

7. Окна, содержащее рекорды игры (Рисунок 8)

8. Окна, содержащее игровой процесс (Рисунок 9)

9. Окна, содержащее паузу (Рисунок 10)

10. Окна, содержащее правила игры (Рисунок 11)



Рисунок 2 – Прототип логотипа



Рисунок 3 – Прототип заставки

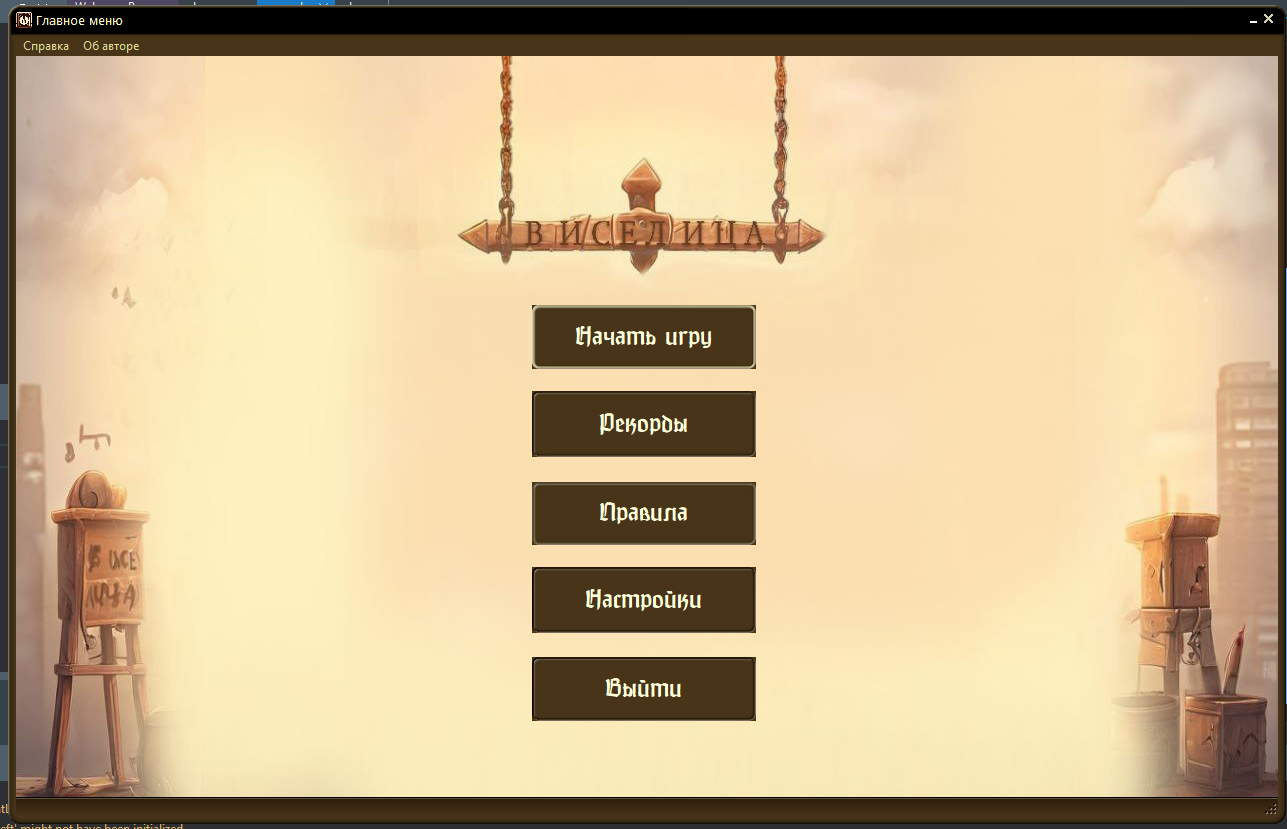


Рисунок 4 – Прототип главного окна

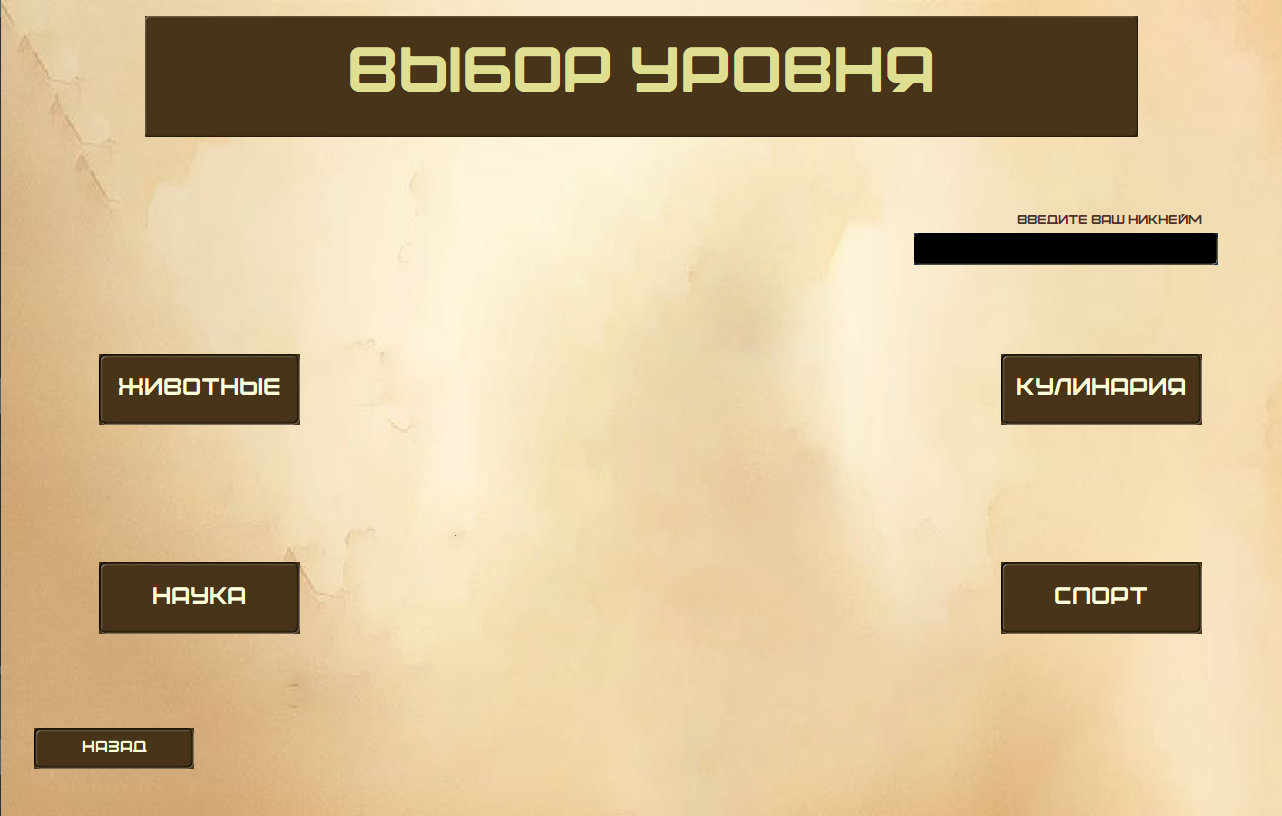
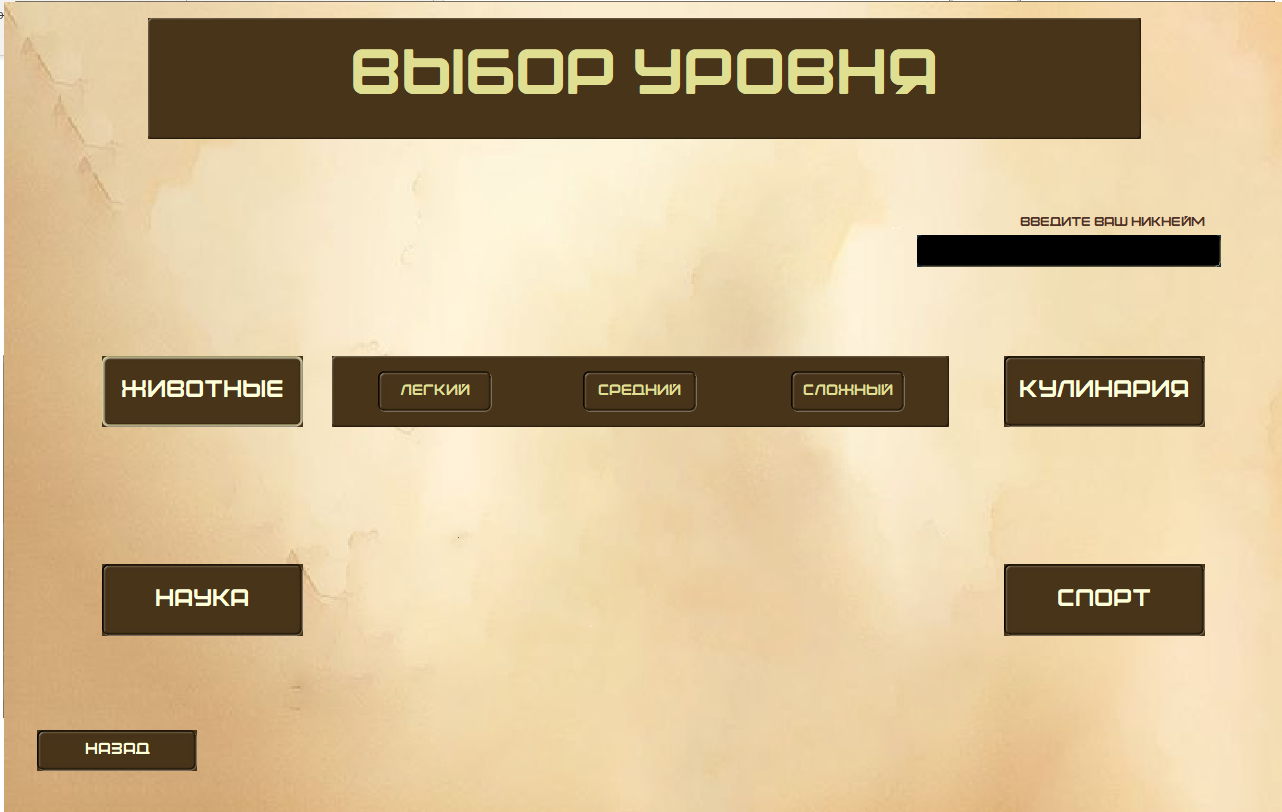


Рисунок 5 – Прототип меню уровней



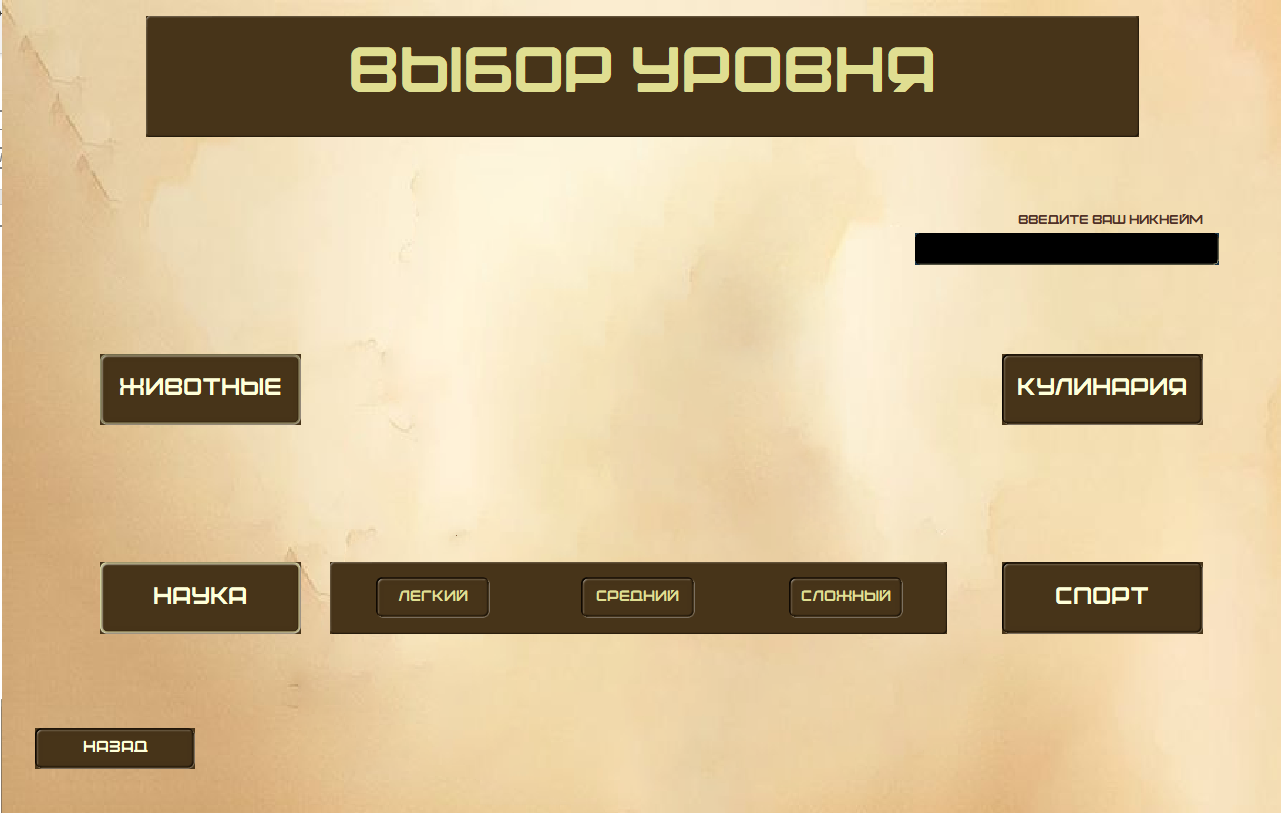


Рисунок 6 – Прототип развернутое меню уровней

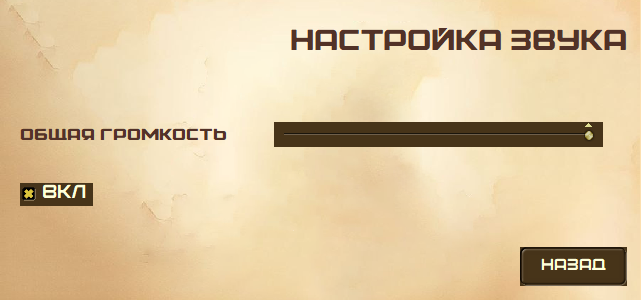


Рисунок 7 – Прототип настроек звука



Рисунок 8 – Прототип рекордов





Рисунок 9 – Прототип игрового окна

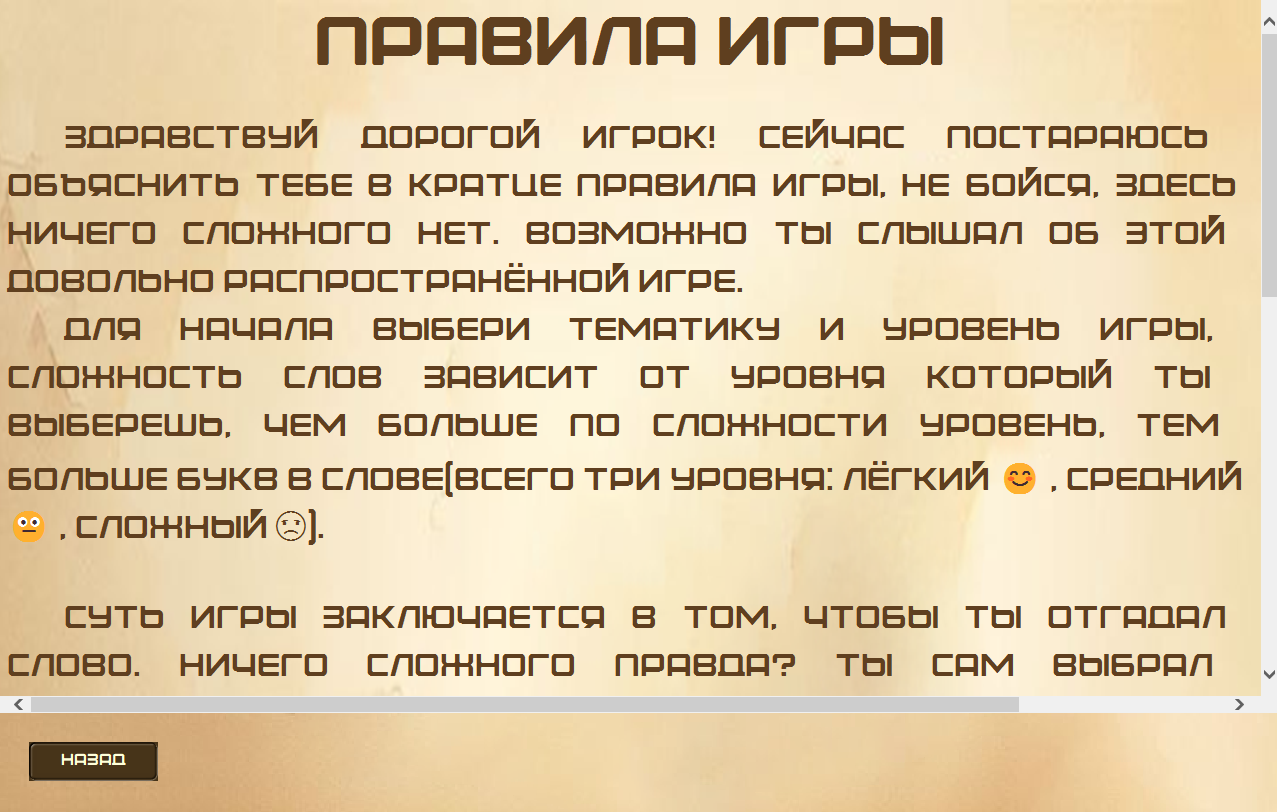


Рисунок 10 – Прототип правил игры

Таким образом, для успешной работы всего проекта в целом следует непременно тесно связать все эти компоненты и заставить их работать.

**3 Реализация**

**3.1 Реализация проекта**

Данный курсовой проект содержит 8 модулей. Далее рассмотрим назначение каждого модуля:

1. Модуль glavnaya – главное меню, модуль за которым закреплена главная форма, можно перейти с неё на любую другую форму.
2. Модуль zastavka – заставка курсовой работы.
3. Модуль igra – меню уровней, можно выбрать любую из предложенных тем и уровень игры, тем самым перейдя в саму игру.
4. Модуль nastroikii – отвечает за настройку звука.
5. Модуль recorrdu – содержит рекорды поставленные игроком(тема, уровень, никнейм, количество набранных звезд)
6. Модуль pravilaa – содержит правила игры
7. Модуль play – содержит саму игру, игровой процесс «виселицы»
8. Модуль oknopobedaproigrsha – меню паузы, где можно перейти в настройки, обратно к уровням или продолжить опять игру, так же данный модуль используется для окна победы при выигрыше игрока, где такие же функции как и в паузе, кроме надписей, и для окна проигрыша.

**3.1.1 Структура и описание процедур и функций пользователя**

Описание разработанных процедур находятся в таблице 1.

Таблица 1 – Процедуры и функции

|  |  |  |  |
| --- | --- | --- | --- |
| Имя процедуры (функции) | В каком модуле находится | За каким компонентом закреплена | Назначение |
| 1 | 2 | 3 | 4 |
| procedure zastavkaa.Timer1Timer(Sender: TObject); | zastavka | Timer1 | Таймер для закрытия загрузочного экрана |
| procedure FormShow(Sender: TObject); | glavnaya | MainForm | Запуск заставки |
| procedure glavnayaa.FormCreate(Sender: TObject); | glavnaya | MainForm | Выполняет инициализацию шрифтов, настройку медиаплеера |
| procedure glavnayaa.SetVolume(Volume: integer); | glavnaya | MainForm | Установка громкости |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| procedure glavnayaa.FormDestroy(Sender: TObject); | glavnaya, igra, nastroikii, recorrdu, pravilaa, play, oknopobedaproigrsha | MainForm | При завершении работы приложения выполняет очистку шрифтов |
| procedure glavnayaa.MediaPlayer1Notify(Sender: TObject); | glavnaya, play | MediaPlayer1 | процедура обрабатывает успешные уведомления от MediaPlayer, чтобы воспроизводить файлы и управлять флагом |
| procedure glavnayaa.НачатьигруClick(Sender: TObject); | glavnaya | Начатьигру | Переход на форму с уровнями |
| procedure glavnayaa.рекордыClick(Sender: TObject); | glavnaya | рекорды | Переход на форму с рекордами |
| procedure glavnayaa.Правила Click(Sender: TObject); | glavnaya | Правила | Переход на форму с правилами игры |
| procedure glavnayaa.настройкиClick(Sender: TObject); | glavnaya | настройки | Переход на форму с настройками |
| procedure glavnayaa.ВыходClick(Sender: TObject); | glavnaya | Выход | Выход из игрового приложения |
| procedure pravila. sButton1Click(Sender: TObject); | pravilaa | sButton1 | Выход с формы обратно в главное меню |
| procedure nastroiki.sCheckBox1Click(Sender: TObject); | nastroikii | .sCheckBox1 | Выключает и включает звук |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| procedure nastroiki.sTrackBar1Change(Sender: TObject); | nastroikii | sTrackBar1Change | Вызывается процедура SetVolume для регулировки громкости звука |
| procedure nastroiki.НазадClick(Sender: TObject); | nastroikii | Назад | Выход с формы обратно в главное меню |
| procedure urovni.Button17Click(Sender: TObject); | igra | Button17 | Выход с формы обратно в главное меню |
| procedure urovni.животныеClick(Sender: TObject); | igra | животные | Открывает панель уровней сложности на тему животные |
| procedure urovni.кулинарияClick(Sender: TObject); | igra | кулинария | Открывает панель уровней сложности на тему кулинария |
| procedure urovni.спортClick(Sender: TObject); | igra | спорт | Открывает панель уровней сложности на тему спорт |
| procedure Turovni.НаукаClick(Sender: TObject); | igra | Наука | Открывает панель уровней сложности на тему наука |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| procedure urovni.Button5Click(Sender: TObject);  procedure urovni.sButton1Click(Sender: TObject);  procedure urovni.sButton6Click(Sender: TObject);  procedure urovni.sButton9Click(Sender: TObject);  procedure urovni.sButton8Click(Sender: TObject);  procedure urovni.sButton7Click(Sender: TObject);  procedure urovni.sButton5Click(Sender: TObject);  procedure urovni.sButton4Click(Sender: TObject);  procedure urovni.sButtonClick(Sender: TObject);  procedure urovni.sButton2Click(Sender: TObject);  procedure urovni.sButton11Click(Sender: TObject);  procedure urovni.sButton10Click(Sender: TObject); | igra | Button5, sButton1, sButton6, sButton9, sButton8, sButton7, sButton5, sButton4, sButton3, sButton2, sButton11, sButton10 | Определяют уровень игры и переходят на форму с игрой |
| procedure urovni.кнопканазадClick(Sender: TObject); | igra | кнопканазад | Выход с формы обратно в главное меню |
| procedure recordu.sButton1Click(Sender: TObject); | recorrdu | sButton1 | Выход с формы обратно в главное меню |
| procedure recordu.LoadRecords; | recorrdu | MainForm | При входе в приложение заново и на форму, сохранённые рекорды записывает в таблицу рекордов из файла |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| procedure Form2.FormClose(Sender: TObject; var Action: TCloseAction); | play | MainForm | Обновляет все компоненты |
| procedure Form2.подсказкаClick(Sender: TObject); | play | подсказка | Использует подсказку, открываю одну букву в слове |
| procedure Form2.proverkabukv(Sender: TObject); | play | MainForm | Проверяет буквы отгаданы они или нет |
| procedure Form2.паузаClick(Sender: TObject); | play | пауза | Переход на форму паузы |
| procedure Form2.зановоClick(Sender: TObject); | play | заново | Игровой процесс обновляется и начинается заново |
| Procedure.Form2.sozdaniepanelee; | play | MainForm | Создает пустые панели в соответствии с длинной слова, если есть старые панели, то они удаляются |
| procedure Form2.SaveRecordToFile; | play | MainForm | Сохранение рекорда за игровой процесс |
| procedure recordu.Button2Click(Sender: TObject); | recorrdu | Button2 | Очищает таблицу рекордов и рекорды в файле |
| procedure recordu.ClearFile; | recorrdu | MainForm | Очистка файла |
| procedure Form2.Timer1Timer(Sender: TObject); | play | Timer1 | Включение таймера при условии если буквы все отгаданы или попытки на ошибку закончились |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| procedure Form2.запуститьClick(Sender: TObject); | play | запустить | Запуск игры |
| procedure Form2.AlphabetButtonClick(Sender: TObject); | play | MainForm | Проверка букв на совпадение с буквами в загаданном слове |
| procedure okno.заново2Click(Sender: TObject); | oknopobedaproigrsha | заново2 | Начать игру заново |
| procedure okno.настройки2Click(Sender: TObject); | oknopobedaproigrsha | настройки2 | Переход на форму с настройками |
| procedure okno.продолжитьClick(Sender: TObject); | oknopobedaproigrsha | продолжить | Переход обратно к игре |
| procedure Tokno.обратнокуровнямClick(Sender: TObject); | oknopobedaproigrsha | обратнокуровням | Выход обратно к меню уровней |

**3.1.2 Описание использованных компонентов**

Описание использованных для разработки приложения компонентов приводится в таблице 2.

Таблица 2 – использованные компоненты

|  |  |  |
| --- | --- | --- |
| Компонент | На какой форме  расположен | Назначение |
| TMainMenu | glavnaya | Используется для вызова справки, закрытия программы. |
| TImage | zastavka, glavnaya, igra, nastroikii, recorrdu, pravilaa, play, oknopobedaproigrsha | Используется как фон, скин на кнопки, элементы интерфейса |
| TTimer | zastavka,play, oknopobedaproigrsha | Используется для закрытие загрузочной формы, открытия формы-оповещения об выигрыше/проигрыше |
| TLabel | glavnaya, igra, nastroikii, recorrdu, pravilaa, play, oknopobedaproigrsha | Отображение надписей на форме |
| TButton | glavnaya, igra, nastroikii, recorrdu, pravilaa, play, oknopobedaproigrsha | Используется для взаимодействия  пользователя с программой: для закрытие программы, перехода на другую формы и т.д. |

Продолжение таблицы 2

|  |  |  |
| --- | --- | --- |
| TListBox | recorrdu | Используется для хранения рекордов |
| TPanel | igra, play | Используется для отображения неугаданных букв, для размещения кнопок |
| TProgressbar | glavnaya, igra, nastroikii, recorrdu, pravilaa, play, oknopobedaproigrsha | Используется для загрузки формы |
| TEdit | igra | Используется для ввода никнейма игрока |
| TMediaPlayer | glavnaya, play, oknopobedaproigrsha | Используется для фонового проигрывания аудиофайлов |
| TWebBrowser | pravilaa, obavtore | Браузер для чтения |
| sSkinManager | glavnaya | Используется для скинов на компоненты |
| TSpeedButton | play, oknopobedaproigrsha | Используется для взаимодействия  пользователя с программой: для закрытие программы, перехода на другую формы, запуска игры и т.д. |

**3.2 Спецификация программы**

Точное название проекта и его состав приводится в таблице.

Таблица 3 – Спецификация программы

|  |  |
| --- | --- |
| Имя файла | Назначение |
| 1 | 2 |
| viselica.exe | Исполняемый файл проекта, используется для запуска программы на выполнение. |
| viselica.dproj | Файл проекта, связывает все файлы из которых состоит приложение. |
| glavnaya.pas | Файл программного модуля главной формы |
| zastavka.pas | Файл программного модуля заставки |
| igra.pas | Файл программного модуля меню уровней |
| nastroikii.pas | Файл программного модуля настроек |
| obavtore.pas | Файл программного модуля информации об авторе |
| oknopobedaproigrsha.pas | Файл программного модуля меню паузы, уведомление об проигрыше/выигрыше |
| play.pas | Файл программного модуля игрового процесса |
| pravilaa.pas | Файл программного модуля правил игры |
| glavnaya.dfm | Главная форма |
| igra.dfm | Форма с уровнями игры |
| nastroikii.dfm | Форма с настройками |
| obavtore.dfm | Форма с информацией об авторе |

Продолжение таблицы 3

|  |  |
| --- | --- |
| oknopobedaproigrsha.dfm | Форма с меню паузы,уведомление об проигрыше/выигрыше |
| play.dfm | Форма с игрой |
| pravilaa | Форма с правилами игры |
| Setup.exe | Установочный пакет программы |
| животныелегкий.txt  животныесложный.txt  животныесредний.txt  кулинариялегкий.txt  кулинариясложный.txt  кулинариясредний.txt  наукалегкий.txt  наукасложный.txt  наукасредний.txt  спортлегкий.txt  спортсложный.txt  спортсредний.txt | Слова разных уровней сложности(лёгкий, средний, сложный) для отгадывания |
| рекорды.txt | Информация о рекордах |

**4. Тестирование**

В процессе разработки данной программы мы активно устраняли возникающие ошибки и недоработки. После завершения этапа написания кода мы провели тщательное функциональное тестирование. Это тестирование гарантировало, что все элементы программы работают корректно в автономном режиме.

Отчет о результатах тестирования представлен в таблице 4

Таблица 4 - Отчет о результатах тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Идентификатор теста | Тест | Ожидаемый результат | Физический результат | Результат тестирование |
| 1 | Проверка входа  пользователя в приложение | Открытие главного окна | Открытие главного  окна | Выполнено |
| 2 | Проверка главного меню кнопки  «Начать игру» | Открытие формы с  меню уровней | Открытие формы с  меню уровней | Выполнено |
| 3 | Проверка главного меню кнопки  «Рекорды» | Переход на форму с таблицей рекордов | Переход на форму с таблицей рекордов | Выполнено |
| 4 | Проверка главного меню кнопки «Правила» | Переход на форму с правилами игры, открытие текстового файла в формате .docx | Переход на форму с правилами игры, открытие текстового файла в формате .docx | Выполнено |
| 5 | Проверка главного меню на кнопку «Закрыть» | Закрытие приложения | Закрытие приложения | Выполнено |
| 6 | Проверка главного меня кнопки «Настройки» | Открытие формы с настройкой звука | Открытие формы с настройкой звука | Выполнено |
| 7 | Проверка настроек звука включение/выключение, регулировка звука | Выключение/Включение, понижение/повышение громкости | Выключение/Включение, понижение/повышение громкости | Выполнено |
| 8 | Проверка на форме рекордов таблицы рекордов | Запись рекордов после победы в таблицу | Запись рекордов после победы в таблицу | Выполнено |

Продолжение таблицы 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 9 | Проверка меню уровней | Открытие кнопок с уровнями сложности, переход на форму с игрой | Открытие кнопок с уровнями сложности, переход на форму с игрой | Выполнено |
| 10 | Проверка кнопки  «Начать» | Генерирование слова в виде пустых панелей, активирование клавиатуры | Генерирование слова в виде пустых панелей, активирование клавиатуры | Выполнено |
| 11 | Проверка кнопки «Подсказка» | Открытие буквы на одной из пустых панелей, отображение уменьшения подсказок | Открытие буквы а одной из пустых панелей, отображение уменьшения подсказок | Выполнено |
| 12 | Проверка кнопки «Заново» | Обновление панелей, количества подсказок, клавиатуры, изображения отображающее ошибки | Обновление панелей, количества подсказок, клавиатуры, изображения отображающее ошибки | Выполнено |
| 13 | Проверка кнопки «Пауза» | Переход на форму меню паузы, остановка фоновой музыки, включение фоновой музыки для паузы | Переход на форму меню паузы, остановка фоновой музыки, включение фоновой музыки для паузы | Выполнено |
| 14 | Проверка кнопки «Выйти» | Переход на форму меню уровней | Переход на форму к меню уровней | Выполнено |
| 15 | Проверка кнопки «Настройки» | Переход на форму настроек | Переход на форму настроек | Выполнено |
| 16 | Проверка кнопки «Обратно» | Переход обратно на форму с игрой, переключение фоновой музыки | Переход обратно на форму с игрой, переключение фоновой музыки | Выполнено |
| 17 | Проверка формы, уведомляющая о победе/проигрыше | Открытие формы с информацией о победе/проигрыше после отгадывания слова или после исчерпанных попыток на ошибку | Открытие формы с информацией о победе/проигрыше после отгадывания слова или после исчерпанных попыток на ошибку | Выполнено |

При разработке программного продукта было решено множество проблем, например, не осуществлялся переход к справке в верхнем меню по нажатию, звук во время паузы останавливался , поэтому в будущем пользователь не столкнется с данными проблемами.

Элементы программы были проверены, и было установлено, что все они работают правильно и выполняют задачи, указанные в процедурах.

**5 Применение**

**5.1 Общие сведения о программном продукте**

Цель игрового приложения «Виселица» заключается в том, чтобы развлекать игроков и проверять их логическое мышление и знание слов. В игре игроку предлагается угадать слово, вводя буквы. Если буква есть в слове, она открывается, иначе на рисунке добавляется новая часть виселицы. Игрок должен угадать слово до того, как виселица будет полностью нарисована.

Создаваемое приложение будет рассчитано для всех, кто хочет развлечься, проверить свои знания слов и логическое мышление.

Быстродействие любой программы во многом зависит от характеристик выбранного персонального компьютера: рабочей частоты процессора, объема оперативной памяти и т.д. Несмотря на все реализованные в ней задачи, она легко запускается и функционирует на любых машинах.

Тестирование проводилось на моих друзьях, знакомых, младшей сестре, одногруппниках, работать с данной программой было комфортно. Программа разработана на ПК со следующими характеристиками:

* процессор AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz, ядер: 8, логических процессоров: 8, тип системы 64-разрядная операционная система, процессор x64
* объем ОЗУ 16Гб;
* графический адаптер NVIDIA GeForce RTX 3060 Laptop GPU;
* операционная система Windows 11.

**5.2 Инсталляция**

Для того, чтобы установить программу необходимо запустить файл Установщик.exe. Появится окно установки приложения «Игровое приложение виселица»

Затем достаточно следовать приведенной инструкции установки приложения.

**5.3 Выполнение программы**

**5.3.1 Запуск программы**

Данную программу можно запустить различными способами. Первым из них является запуск с помощью ярлыка на рабочем столе. Необходимо дважды щелкнуть левой кнопкой мыши на ярлыке с названием «Игровое приложение виселица»

Вторым способом является запуск из каталога, в который устанавливалось приложение (по умолчанию C:\Program Files (x86)\Игровове приложение виселица).

По подготовленным тестам будет осуществляться функциональное и полное тестирование программного продукта. Отчет о результатах тестирования будет представлен в 4 разделе пояснительной записки.

**5.3.2 Инструкции по работе с программой**

После запуска приложения на экране нас встречает заставка на рисунок 1.

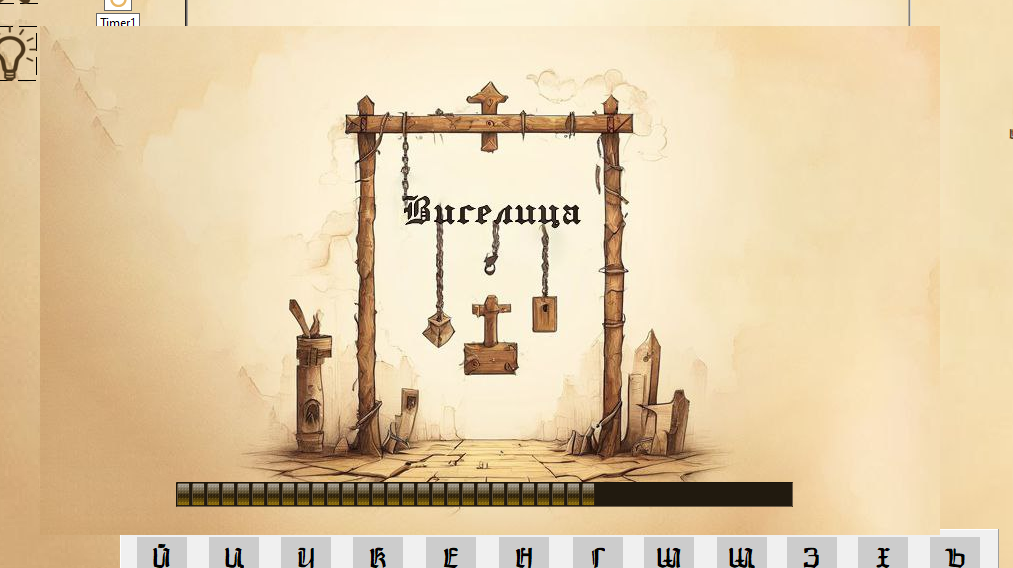


Рисунок 1 – заставка

Дальше мы видим появившееся главное меню, где оттуда уже и начинаем свой путь. Здесь мы можем перейти сразу к уровням игры нажав на кнопку «Начать игру», конечно советую для начала ознакомится с правилами игры , перейти к которым вы можете нажав на кнопку «Правила». Так же есть кнопка «Рекорды», где вы сможете наблюдать все свои рекорды сделанные на каждом уровне игры. Нажав на кнопку «Настройки» вы переходите к настройкам звука. И конечно соответственно кнопка «Выйти» для выхода из нашего игрового приложения, так же это можно сделать в верхнем меню. Главное меню на рисунке 2.

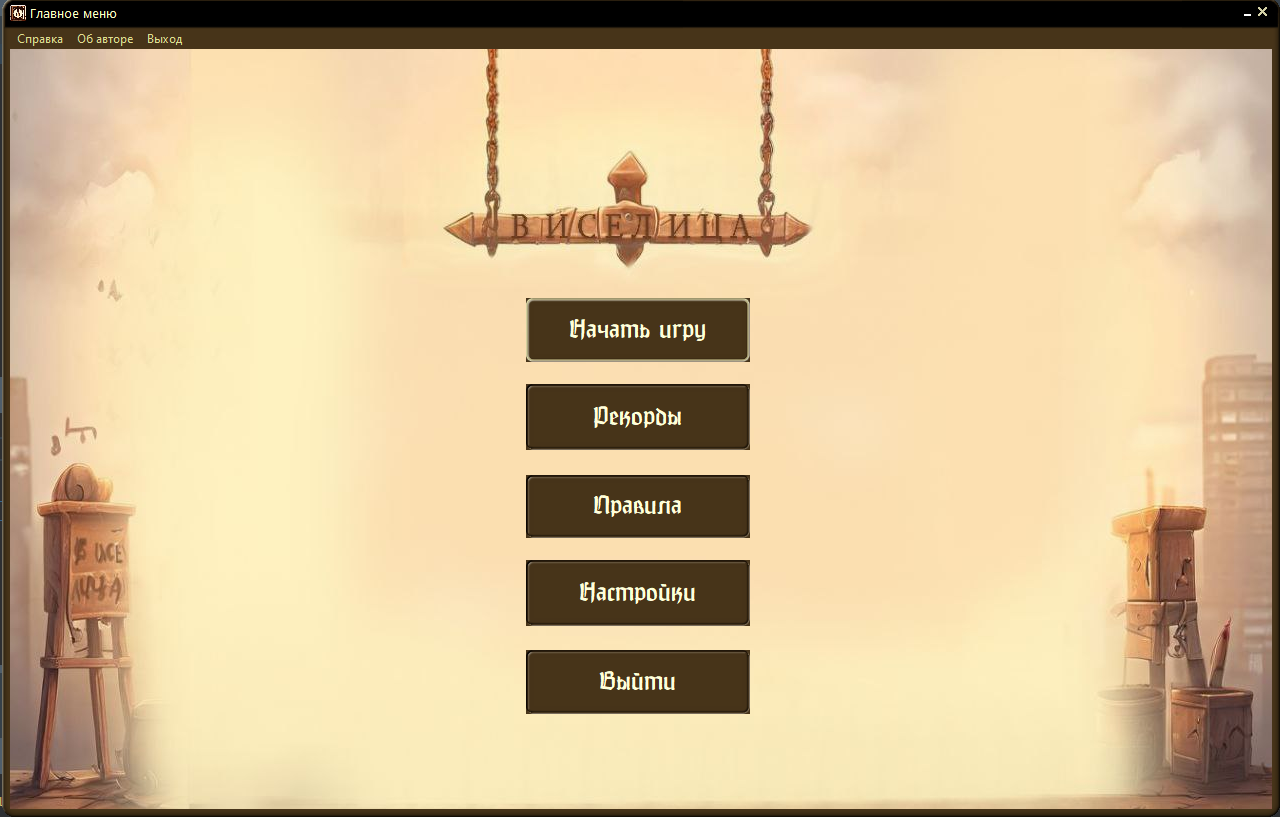


Рисунок 2 – главное меню

Как описано выше, при нажатии на кнопку «Правила» мы переходим к ним. Правила игры на рисунке 3.

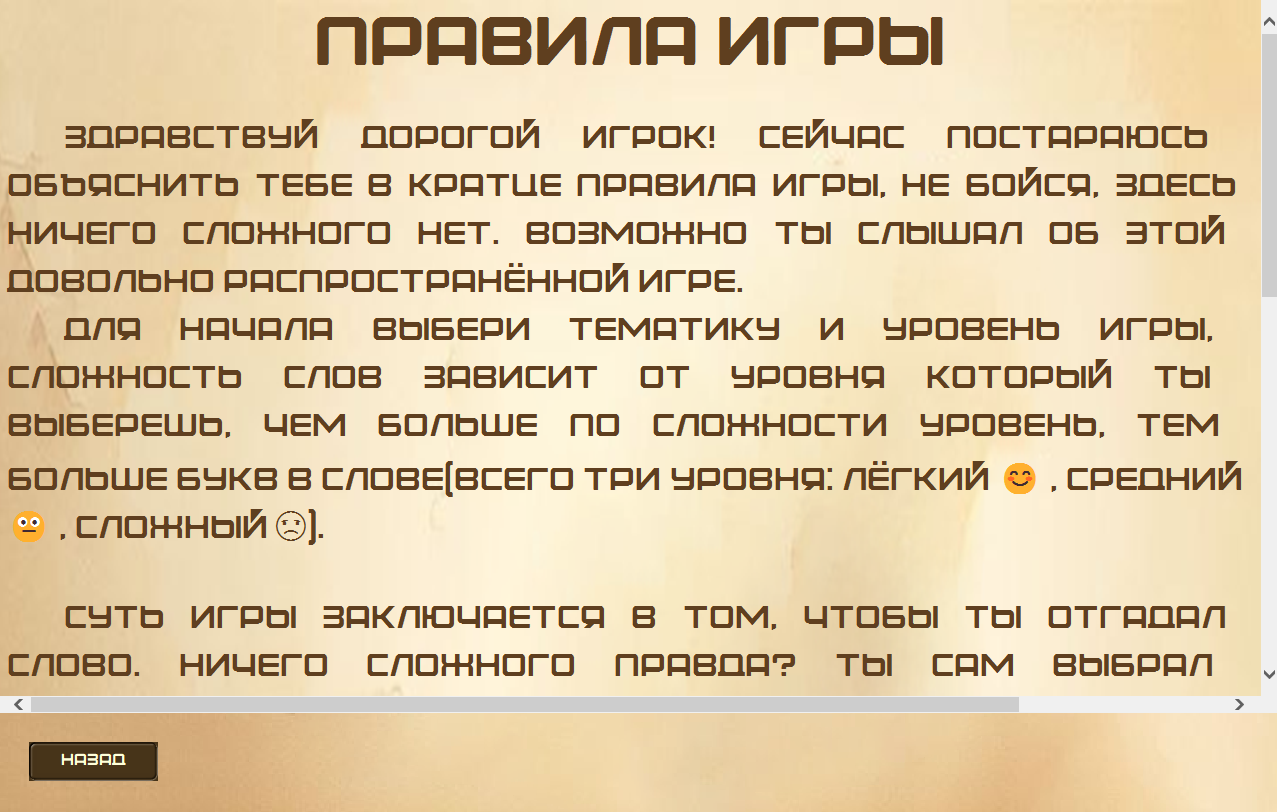


Рисунок 3 – правила игры

Настройки на рисунке 4. Здесь вы легко сможете отрегулировать громкость звука или же выключить полностью.

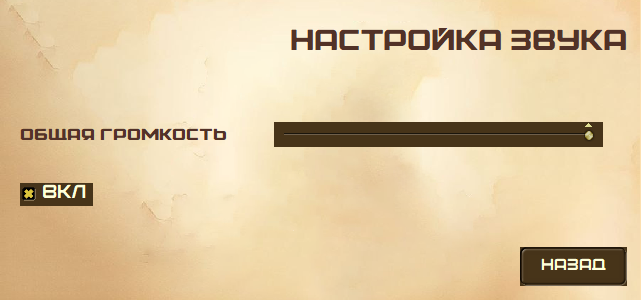


Рисунок 4 – настройки

Рекорды на рисунке 5. В таблице рекордов вы можете наблюдать свои результаты побед в игре. Хотите сравнить сегодня как проходили тот же уровень, что и вчера? Без проблем, вам всего лишь нужно на главном меню нажать на кнопку «рекорды» для перехода к таблице. Здесь мы так же можете очистить таблицу от этих рекордов.



Рисунок 5 – рекорды

Теперь вы можете спокойно перейти к самой игре, но для начала выберем уровень игры и тематику. Нажмите на «Начать игру», теперь вы перешли у меню уровней. Здесь вы должны ввести свой никнейм для того чтобы проще ориентироваться в таблице рекордов, если кто-то на вашем пк собирается тоже опробовать игровое приложение. Меню уровней на рисунке 6.

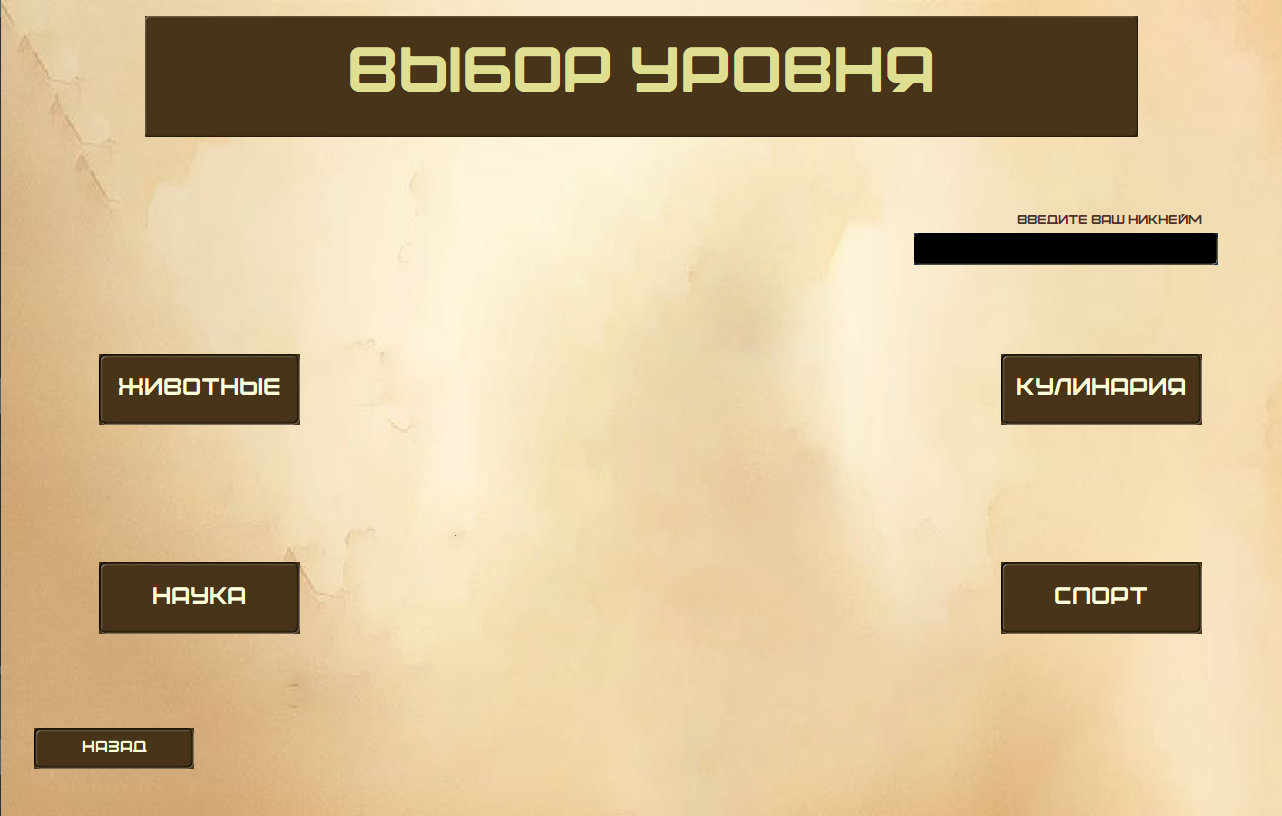
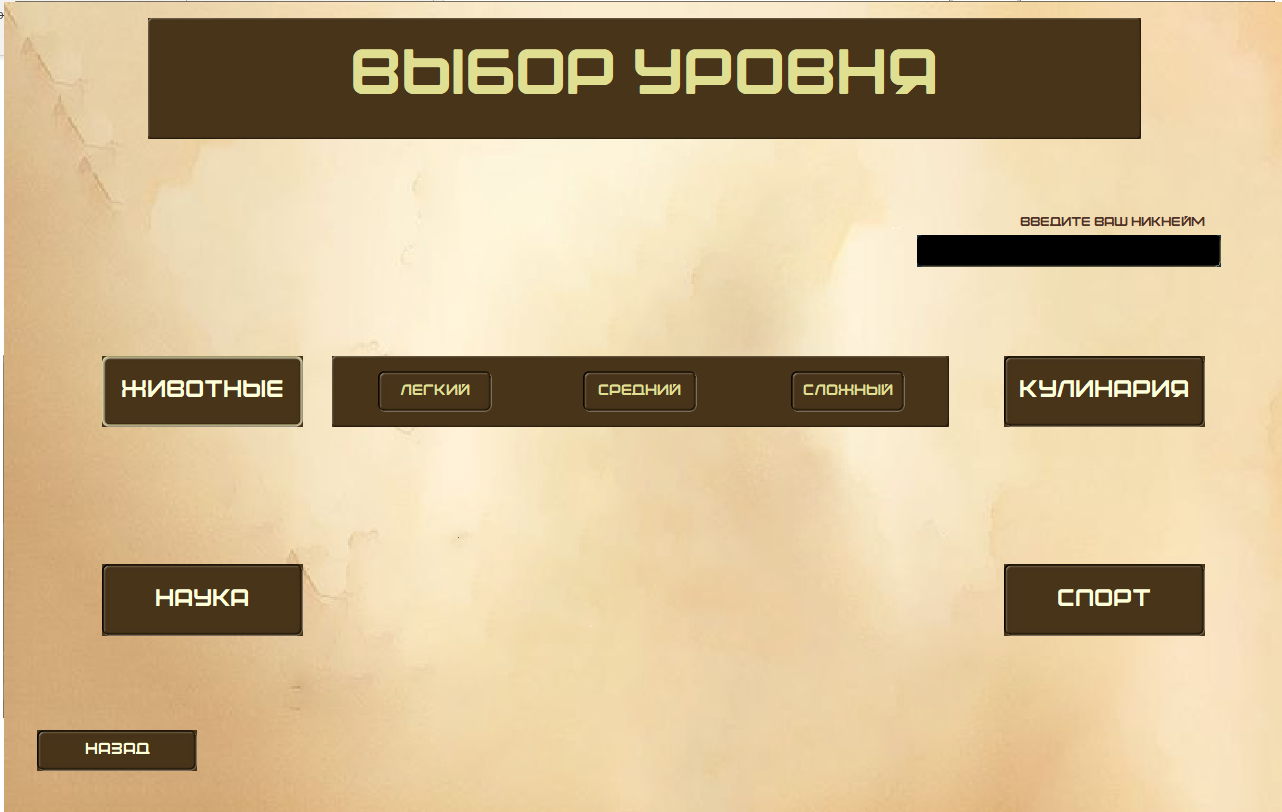


Рисунок 6 – меню уровней

Теперь при нажатии на тематику по которой вы хотите отгадывать слова вам откроются уровни сложности этих слов. Подробнее на рисунке 7.



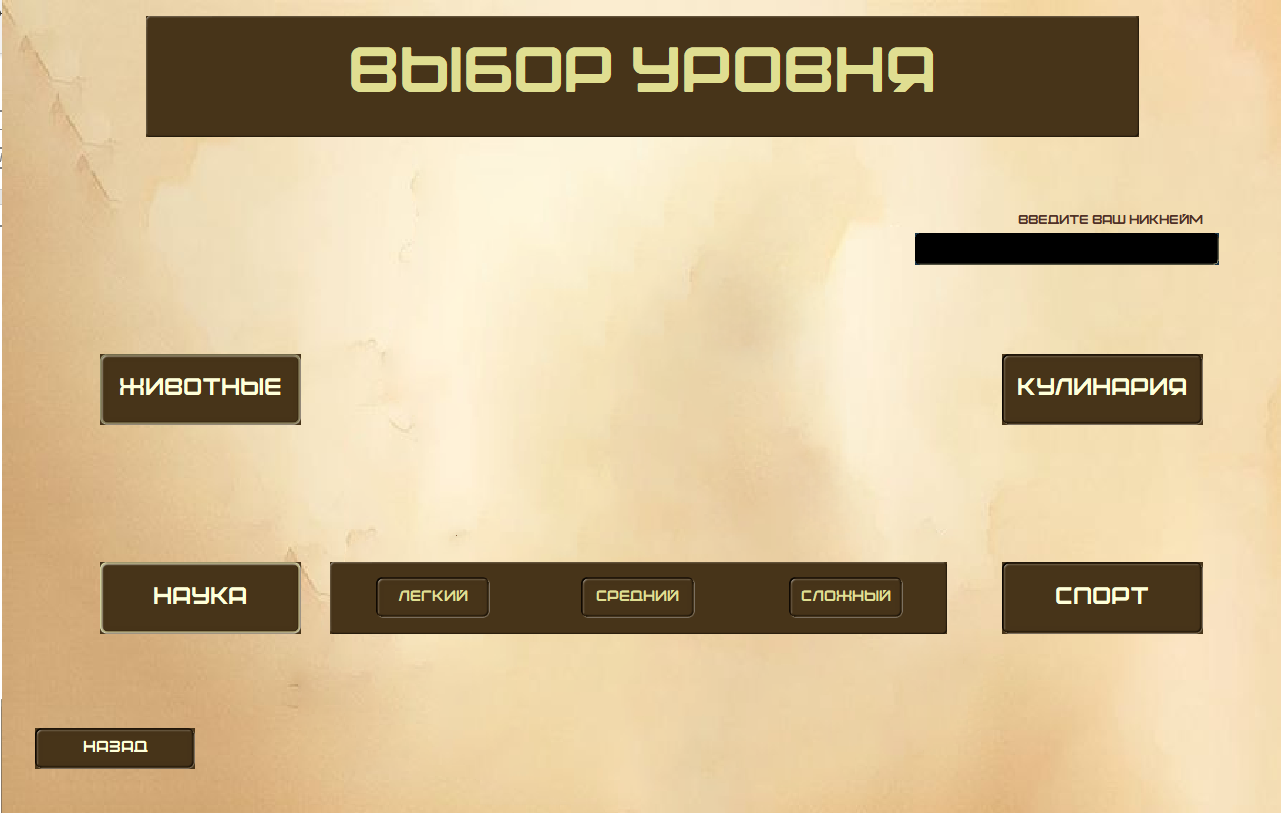


Рисунок 7 – развернутое меню уровней

При нажатии на уровень сложности вы перейдете к самой игре где уже можете начать игровой процесс, сейчас мы рассмотрим подробнее. Окно игры на рисунке 8.



Рисунок 8 – окно игры

Перед вами есть клавиатура на которой вы сможете выбирать буквы, которые считаете есть в слове, но она вам недоступна, пока не нажмете на кнопку «начать» (рисунок 9). При нажатии на кнопку мы видим начало игры (Рисунок 10).



Рисунок 9 – кнопка «начать»

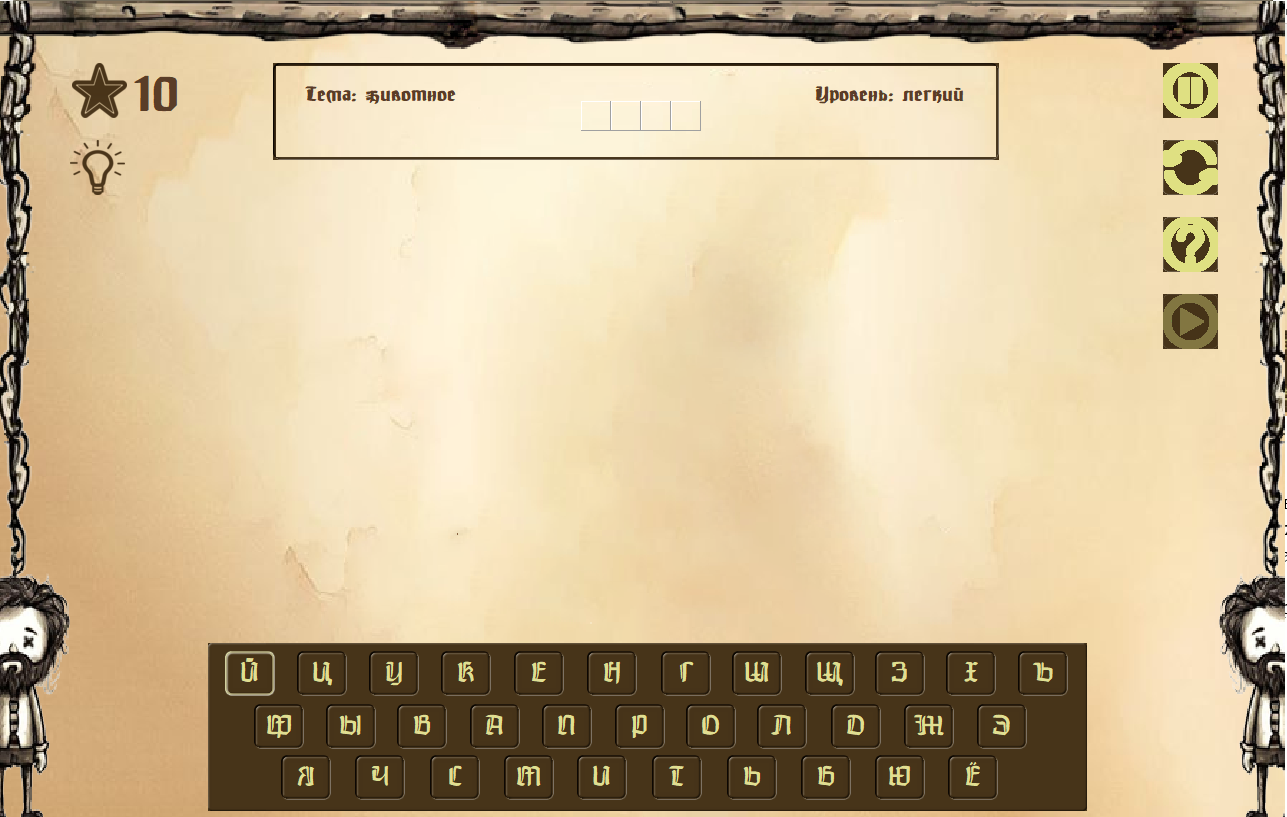


Рисунок 10 – окно игры при нажатии «начать»

Теперь вы видите появившиеся пустые ячейки и недоступную кнопку «начать» (если не справляетесь с этим слово, вы в любое время , пока не проиграли, начать заново нажав на кнопку, выберется новое рандомно слово для угадывания(кнопка «начать заново» на рисунке 11) )



Рисунок 11 – кнопка «Начать заново»

При нажатии на клавиатуре на букву, эта буква либо появится в пустых ячейках, если она есть в слове, либо если ее нет, то будет по частям появляться наш человечек отображающий ваши ошибки, с каждой неправильной буквой новая часть тела. Этот момент на рисунке 12.

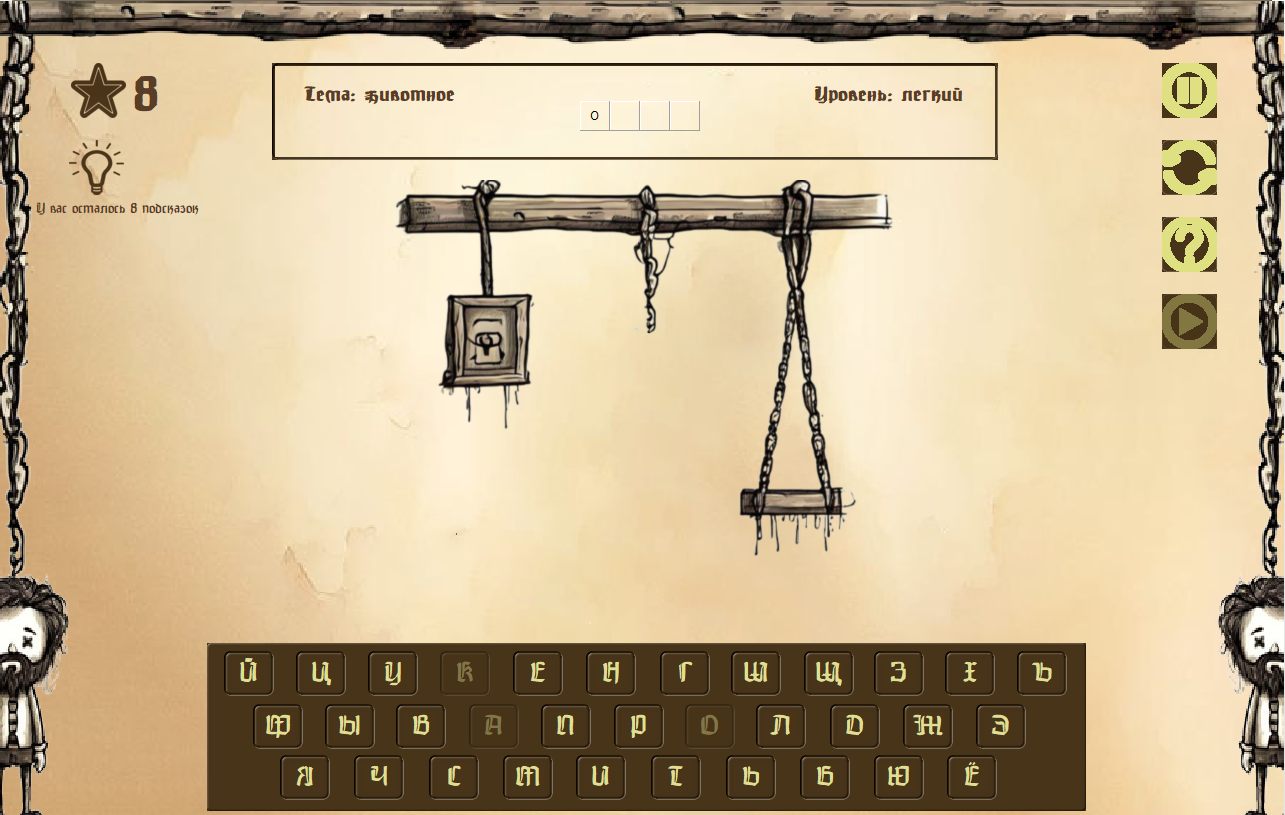


Рисунок 12 – окно игры при взаимодействии с клавиатурой

Как вы можете заметить , так же начали отниматься наши звезды, всего их 10, каждая неправильная буква равняется отнятой одной звездой, и ещё подсказки, при нажатии на подсказку(кнопка подсказка на рисунке 13) так же отнимается по одной звезде на подсказку, но зато вам открывается новая буква в слове, что облегчает игровой процесс. Отображение на данный момент количества подсказок на рисунке 14.



Рисунок 13 – кнопка «Подсказка»



Рисунок 14 – количество подсказок

Так же вы можете приостановить игру нажав на кнопку «пауза», в меню паузы вы сможете перейти к настройкам звука(кнопка в виде шестеренки), вернуться обратно к игре(кнопка в виде треугольника) или же выйти обратно к уровням выбрав новый уровень(кнопка в виде домика).Кнопка «пауза» и меню паузы на рисунке 15 и рисунке 16.



Рисунок 15 – кнопка «пауза»



Рисунок 16 – меню паузы

**Заключение**

Целью данного проекта было обобщение знаний и подготовка к квалификационному экзамену. В процессе работы над проектом были реализованы несколько ключевых аспектов, которые обеспечивают успешную реализацию и качественное выполнение поставленных задач.

Одной из основных задач проекта было создание простого и понятного пользовательского интерфейса. Удалось создать понятный интерфейс, который обеспечивает легкий доступ ко всем функциям приложения. Пользовательский интерфейс был тщательно продуман, чтобы быть удобным для использования как новичками, так и опытными пользователями. Это главное для себя такие элементы, как меню, заставка, меню уровней, настройка звука, рекорды, правила игры и игровой процесс.​

Для достижения высокого уровня удобства интерфейс был создан с учетом лучших практик в области проектирования интерфейсов. В результате навигация по приложению стала простой и понятной, что значительно повысило уровень пользовательского опыта. Это обеспечивается отсутствием замечаний со стороны тестировщиков по поводу сложности использования программы.​

Процесс тестирования проекта в своей тщательной оценке всех возможных особых ситуаций. Были разработаны скрипты, которые могли бы привести к сбоям или ошибкам в работе программы. Каждая из этих ситуаций была решена, что обеспечило надежную и стабильную работу приложения. В процессе тестирования было выявлено и устранено несколько возможных ошибок, которые обеспечивают отсутствие условий в надлежащей эксплуатации​​​.

В процессе разработки программного продукта я значительно расширила свои профессиональные навыки. Одним из важных достижений стало умение создавать динамические компоненты и прорабатывать их взаимодействие. Это решение в создании нормальных процедур и функций, а также более глубокое понимание обработки исключительных ситуаций. Эти навыки являются частью профессиональной деятельности разработчика и постепенно закрепляются в ходе работы над проектом.​

Проект состоял из нескольких модулей, каждый из которых имел свое уникальное назначение. Были успешно реализованы такие модули, как меню, заставка, меню уровней, настройка звука, записи, правила игры и сам игровой процесс. Каждый модуль был тщательно разработан и интегрирован в общее приложение, позволяющее создать цельную и функциональную программу.

К итогу проект достиг всех поставленных целей и задач, что обеспечило успешную реализацию всех модулей и отсутствие ошибок в работе программы. Программа не только способствует развитию бизнеса, но и имеет образовательную ценность, помогая пользователям расширить свой словарный запас и улучшить орфографические навыки с помощью программного процесса. Это делает приложение востребованным среди благоприятных условий​​.

В заключение можно с уверенностью сказать, что программа выполнена успешно. Она является результатом тщательной работы и глубокого понимания всех аспектов разработки программного обеспечения. Проект подтвердил приобретение знаний и навыков, а также готовность к сдаче квалификационного экзамена, демонстрируя высокие показатели.

**Список использованных источников**

1. Вощинская, Г.Э. Разработка компонентов в DELPHI. - Воронеж: ИПЦ ВГУ, 2007. - 57 с.
2. Культин, Н. Основы программирования в Delphi 7. - Санкт-Петербург: Питер, 2009. - 640 с.
3. Культин, Н. Delphi 6. Программирование на Object Pascal / Н. Культин. - М.: БХВ-Петербург, 2012. - 528 c.
4. Лишнер Р. Delphi. Справочник. - Пер. с англ. - СПб: Символ-Плюс, 2001. - 640 с.
5. Torry's Delphi Pages - Портал по программированию на Delphi, содержащий множество исходников и компонентов на Delphi
6. delphi-help - справочник по программированию, статьи.