**Введение**

На курсовое проектирование была поставлена задача, разработки программы на тему: создание игрового приложения «Two At Ten»

Цель курсового проекта заключается в обобщении знаний и подготовки к квалификационному экзамену.

Приведем краткое описание разделов пояснительной записки.

Первый раздел носит название “Анализ задачи”. В нем вы сможете ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, определение ее организационно-экономической сущности. Также в этом разделе вы сможете узнать о том, как данная задача решается в настоящее время. Все входные и выходные данные тоже будут описаны в первом разделе. В подразделе “Инструменты разработки” будет рассмотрена среда, в которой создается данный курсовой проект. Здесь также будут установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачи.

В разделе “Проектирование задачи” будут рассмотрены основные аспекты разработки программного продукта. Здесь можно будет узнать об организации данных в контексте среды разработки. В данном разделе будет четко описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации, описана разработка системы справочной информации.

“Реализация задачи” – это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые будут использованы при реализации данного приложения. В этом разделе будут четко описаны функции пользователя и их структура. Здесь можно будет найти таблицу, в которой будет представлена полная аннотация файлов используемых в данном проекте.

Четвертый раздел – “Тестирование”. В нем будет описано полное и функциональное тестирование данной программы, т.е. будет оттестирован каждый пункт меню, каждая операция, которая выполняется приложением. Будут смоделированы все возможные действия пользователя при работе с программой, начиная от запуска до выхода.

В разделе “Применение” будет описано назначение, область применения, среда функционирования курсовой программы. Также в нем будет описано использование справочной системы.

“Заключение” будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В “Списке используемых источников” будет приведен список используемой литературы, нормативно- техническую и другую документацию.

В приложениях к пояснительной записке будет приведен листинг программы с необходимыми комментариями.

Схема работы системы будет представлена в графической части.

**1. Анализ задачи**

**1.1 Постановка задачи**

Темой данного курсового проекта является «Создание игрового приложения «Two At Ten»

В современном мире компьютерные игры стали популярным способом проведения досуга и развлечения. Они не только позволяют отдохнуть и развлечься, но и способствуют развитию некоторых навыков и умений. Разработка компьютерной игры, которая будет сочетать в себе элементы развлечения и умственного тренинга, представляет собой интересную задачу.

На рынке существует множество компьютерных игр, но не все из них обладают полезным содержанием или обучающими аспектами. Цель данного курсового проекта - создать игру, которая будет сочетать интересный геймплей с развитием когнитивных и логических навыков у игроков.

Аналогов для такой игры можно привести не так уж и много. В основном на рынке присутствуют развлекательные игры без обучающего компонента или обучающие программы, которые не предлагают интересного и захватывающего геймплея.

Игра будет предназначена для использования на персональных компьютерах и будет распространяться в виде установочного пакета, который пользователь сможет скачать и установить на свой компьютер. Игра должна быть доступной для широкой аудитории пользователей, включая людей без специальных знаний и навыков в области компьютерных игр.

Цель разработки данной игры - создать увлекательное и интересное игровое приложение, которое поможет игрокам развить свои когнитивные и логические способности. Игра должна иметь простой и интуитивно понятный интерфейс, чтобы пользователи могли легко освоить игровой процесс.

Игра предоставляет функционал для следующих видов пользователей: обычный игрок - основной пользователь игры, который будет играть и развиваться в игровом процессе.

**1.2 Инструменты разработки**

Для разработки данного проекта будет выбрана среда Delphi 11, так как

это самое удобная и доступная среда разработки на данный момент. Delphi 11-язык –программирования, относящийся к классу RAD- (Rapid Application Development – «Средство быстрой разработки приложений») средств CASE –тeхнологии. Delphi 11 сделал разработку приложений дляWindows быстрым иприятным процессом. Теперь разрабатывать сложные и интересные проекты можно только одним человеком, использующим Delphi 11

Интерфейс Windows обеспечивает полное перенесение CASE-технологии в интегральную систему поддержки работ по созданию прикладной системы на всех фазах жизненного цикла работы и проектирования системы.

Delphi 11 обладает широким набором возможностей, начиная от проектировщика форм и заканчивая поддержкой всех форматов популярных баз данных. Среда устраняет необходимость программировать такие компоненты Windows общего назначения, как метки, программы и даже диалоговые панели. Работая в Windows, можно видеть одинаковые «объекты» во многих разнообразных приложениях. Диалоговые панели (например, Choose File и Save File) являются примерами многократно-используемых компонентов, встроенных непосредственно в Delphi 11, который позволяет приспособить эти компоненты к имеющийся задаче, чтобы они работали именно так, как требуется создаваемому приложению. Также здесь имеются предварительно-определенные визуальные и не визуальные объекты.

Три основные части разработки интерфейса следующие: проектирование панели, проектирование диалога и представление окон. Для общего пользовательского доступа также должны учитываться условия применения архитектуры прикладных систем.

Сегодня появилась реальная возможность с помощью моделирования на современных многофункциональных средствах обработки и отображения информации таких как Delphi 11 конкретизировать тип и характеристики используемых информационных моделей, выявить основные особенности будущей деятельности операторов, сформулировать требования к параметрам аппаратно-программных средств интерфейса взаимодействия и т.д. Delphi 11 позволяет создать различные виды программ: консольные приложения, оконные приложения, приложения для работы с Интернетом и базами данных. То есть, Delphi 11 является не только средствами для работы с языком программирования Паскаль, но дополнительные инструменты, призванные для максимального упрощения и ускорения создание приложений.

К дополнительным инструментам можно отнести визуальный редактор форм, благодаря которому можно с легкостью создать полноценную программу, и другие визуальные составляющие разработки программного обеспечения. С Delphi вам не нужно вручную просчитывать расположение каждого элемента интерфейса пользователя, поэтому при разработке программы значительно экономится время.

Выгоды от проектирования в среде Windows с помощью Delphi 11:

* устраняется необходимость в повторном вводе данных;
* обеспечивается согласованность проекта и его реализации;
* увеличивается производительность разработки и переносимость программ.

Ни одно серьезное программное обеспечение не обходится без модуля справочной информации и руководства пользователя. Это придает программе законченный вид и показывает заботу о пользователе. DrExplain – легкий в использовании и функциональный инструмент, упрощающий создание справочных файлов Windows, печать справочных руководств и документации в целом. Программа имеет интуитивно понятный интерфейс. Все созданные проекты можно сохранить в различных форматах:HTML Help, Winhelp и MS Help 2.0 / Visual Studio Help, Browser-based Help, PDF и Word RTF, а также печатной документации при использовании одного и того же проекта. В основном окне программы содержатся оглавление (в виде древовидного списка) и текстовый редактор. Это дает возможность легко ориентироваться в оглавлении, редактировать или перемещать разделы справки без каких-либо проблем. Также утилита позволяет конвертировать help-файлы из одного формата в другой. Кроме приложений для работы с текстом в данном продукте содержатся утилиты для создания скриншотов и редактирования графических файлов.

Для создания инсталлятора будет использоваться мощное и удобное средство - Smart Install Maker. Программа обладает удобным и интуитивно понятным интерфейсом, а также полным набором необходимых функций для создания профессиональных инсталляторов с минимальным размером, высокой степенью сжатия файлов и приятным интерфейсом.

Помимо стандартного минимума, Smart Install Maker позволяет редактировать системный реестр и INI-файлы, создавать программные ярлыки, запускать ассоциируемые и исполняемые файлы, регистрировать новые шрифты и ActiveX компоненты, отображать тексты информации и лицензионного соглашения. Также, с помощью этой утилиты, можно создать мультиязыковые инсталляторы с поддержкой более 20-ти популярных языков мира.

Easy Paint tool SAI - это популярная среди художников программа для рисования (в частности для проработки интерфейса и игры «одевалка»). Удобна своей простотой и в то же время большим функционалом. Программа очень сильно распространена в использовании во всем мире и предпочитается рисующими людьми как один из самых удобных инструментов для создания рисунков. В нем можно сохранить файлы картинок во множестве вариантов (.png, .jpg, .bmp и так далее).

Microsoft Word 2019 – редактор текста для написания документации.

Разработка ведется на ноутбуке HP . У данного ноутбука следующие

параметры:

* процессор amd 4500m;
* объем ОЗУ 8 гб;
* объем места на HDD – 512 гб ;
* видеоподсистема 1024х768 точек с глубиной цвета 16 Bit;
* ОС – Windows 10 домашняя.

Как видно разрабатываемое приложение не очень требовательно к аппаратным ресурсам, что, является большим плюсом.

**1.3 Требования к приложению**

На этапе исследования предметной области были выявлены определенные требования, которые необходимо учесть при разработке нашей компьютерной игры.

Одним из ключевых требований является создание интерфейса, который будет интуитивно понятным и привлекательным для пользователей. Игра должна использовать приятную цветовую гамму и обладать четкой визуальной иерархией элементов интерфейса. Текстовая информация должна быть легко воспринимаемой, а объекты, такие как рисунки и символы, должны быть достаточного размера, чтобы пользователи могли комфортно их видеть.

Окна игры должны быть хорошо структурированы, без избытка информации, чтобы не перегружать пользователей лишними деталями. Каждое окно должно соответствовать тематике игры и быть эстетически оформленным.

Важным требованием является надежность игры. Хотя нет специальных требований к надежности, необходимо учесть возможные сбои и отказы. Игра должна быть способна быстро восстановиться после сбоя электропитания или сбоя операционной системы, чтобы минимизировать время простоя и обеспечить плавный переход к игровому процессу.

В целом, наша игра должна предоставлять надежный и удобный пользовательский интерфейс, который будет соответствовать тематике игры и обеспечивать комфортное игровое взаимодействие для всех пользователей.

**2. Проектирование задачи**

**2.1 Организация данных**

Проектирование задачи – это очень важный и ответственный этап в разработке любого приложения.

Важным является он вследствие того, что методы, по средствам которых пользователь управляет формами, построены на высокой степени специализации каждого из компонентов.

Необходимым условием при разработке данного приложения является описание организации данных, т.е. логическая и физическая структура данных

в контексте среды разработки. В разрабатываемой программе будут использоваться три вида данных.

Первым видом являются данные, которые будут введены разработчиком на этапе реализации задачи. Сюда можно отнести изображения (иконки), описание, изображения.

Вторым видом данных, используемых в программе, являются действия пользователя, он может беспрепятственно перемещаться меду формами, взаимодействовать с игровыми элементами.

Третьим видом данных является результат программы – удаление игровых элементов если пользователь правильно выполнил условие.

Таким образом, организация данных является важной задачей при разработке данной и любой программы.

**2.2 Процессы**

С учетом всех требований и указаний, которые были рассмотрены в разделе "Анализ задачи", мы определили, чем будет заниматься разрабатываемая нами игра. Главной задачей игры будет предоставление пользователю возможности насладиться увлекательным геймплеем и развлечением.

Для реализации этой задачи мы будем использовать различные процессы в игре. С помощью этих процессов будет осуществляться управление игровыми событиями, взаимодействие с игровыми объектами и отображение графических элементов. Все процессы в игре будут спроектированы таким образом, чтобы обеспечить плавный и приятный игровой опыт для всех пользователей.

**2.3 Описание внешнего пользовательского интерфейса**

Важным при выполнении курсового проекта является организация диалога между, пользователем и самой программой. Во многом это зависит от того, как программист разработает данную программу, какие компоненты будут использованы и какие методы будут автоматизированы. Во-первых, особое внимание следует уделить интерфейсу. Разработчик должен так организовать внешний вид своей программы, чтобы пользователь понял, что от него требуется.

Для организации эффективной работы пользователя нужно создать целостное приложение данной предметной области, в которой все компоненты приложения будут сгруппированы по функциональному назначению. При этом необходимо обеспечить удобный графический интерфейс пользователя. Приложение должно позволить пользователю решать задачи, затрачивая значительно меньше усилий, чем при работе с разрозненными объектами.

Ниже на рисунке 1 представлена система меню и организация навигации между окнами программы:

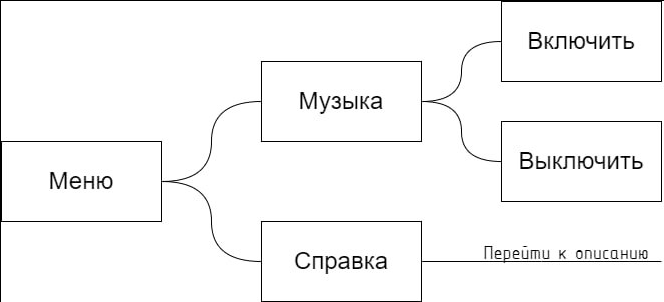


Рисунок 1 – Схема навигация между окнами программы

**3. Реализация**

**3.1 Реализация проекта**

Данный курсовой проект содержит 4 модулей. Далее рассмотрим назначение каждого модуля:

Unit 5- является заставкой в курсовом проекте;

Unit 1- главное меню игры;

Unit 2- содержит правила игры;

Unit 3- содержит игровое поле игры

**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 |
| 1 procedure TForm1.Image1Click(Sender: TObject); | Unit 1 | Image1 | | Переход на форму с игрой |
| 2 procedure TForm1.Image2Click(Sender: TObject); | Unit 1 | Image2 | | Переход на форму с правилами |
| 3 procedure TForm1.Image3Click(Sender: TObject); | Unit 1 | Image3 | | Закрытие приложение |
| 4 procedure TForm1. N1Click(Sender: TObject); | Unit 1 | MainMenu1 | | Открытие справки |
| 5 procedure TForm2.N1Click(Sender: TObject); | Unit 2 | MainMenu1 | Переход на форму с главным меню | |
| 6 procedure TForm2. FormCreate(Sender: TObject); | Unit 2 | Form2 | Помещение формы по центру экрана | |
| 7 procedure TForm3. FormCreate(Sender: TObject); | Unit 3 | Form3 | Помещение формы по центру экрана | |
| 8 procedure TForm3. Image1Click(Sender: TObject); | Unit 3 | Image1 | Создание игрового поля | |

|  |  |  |  |
| --- | --- | --- | --- |
| Продолжение таблицы 1 | | | |
| 1 | 2 | 3 | 4 |
| 9 procedure TForm3. procedure Click; override; | Unit 3 | Memo1..30 | Проверка выбранных ячеек на количество и соседство. Удаление выбранных ячеек, если они соответствуют условиям. Сброс флага Connected для удаленных ячеек. |
| 10 procedure TForm5. FormCreate(Sender: TObject); | Unit 5 | Form 5 | Помещение формы по центру экрана |
| 11 procedure TForm5. Timer1Timer(Sender: TObject); | Unit 5 | Timer1 | Определяет время появления заставки |
| 12 procedure TForm1. Onoff1Click(Sender: TObject); | Unit 1 | MainMenu1 | Выключает музыки на фоне |
| 13 procedure TForm1. N3Click(Sender: TObject); | Unit 1 | MainMenu1 | Включение музыки на фоне |
| 14 procedure TForm1. N3Click(Sender: TObject); | Unit 2 | MainMenu1 | Включение музыки на фоне |
| 15 procedure TForm1. N4Click(Sender: TObject); | Unit 2 | MainMenu1 | Выключает музыки на фоне |
| 16 procedure TForm3. Image2Click(Sender: TObject); | Unit 3 | Image2 | Переход в главное меню |
| 17 procedure TForm3. N2Click(Sender: TObject); | Unit 3 | MainMenu1 | Включение музыки на фоне |
| 18 procedure TForm3. N3Click(Sender: TObject); | Unit 3 | MainMenu1 | Выключает музыки на фоне |
| 18 procedure TForm3. FormClose(Sender: TObject; var Action: TCloseAction); | Unit 3 | Form3 | используется для остановки проигрывания медиафайла |

**3.1.2 Описание использованых компонентов**

Описание использованных для разработки приложения компонентов приводится в таблице 2.

Таблица 2- Использованые компоненты

|  |  |  |
| --- | --- | --- |
| Компонент | На какой форме  расположен | Назначение |
| TMainMenu | Form2, Form1,Form3 | Используется для создания главного меню проекта |
| TImage | Form1, Form2, Form3, Form5, | Используется как фон, выполняет функции кнопок |
| TTimer | Form5 | Используется для определения длительности заставки |
| TLabel | Form2, Form1 | Отображение надписей на форме |
| TMemo | Form3 | Используется в качестве ячеек |
| MediaPlayer | Form1 | Используется для воспроизведения звука |

**3.2 Спецификация программы**

Точное название проекта и его состав приводится в таблице 3.

Таблица 3 – Спецификация программы

|  |  |  |
| --- | --- | --- |
| Имя файла | Назначение | |
| 1 | 2 | |
| Two At Ten.exe | Исполняемый файл проекта, используется для запуска программы на выполнение. | |
| Two At Ten.dproj | Файл проекта, связывает все файлы из которых состоит приложение. | |
| Unit1.pas | Файл программного модуля главного меню | |
| Unit2.pas | Файл программного модуля теоретической части проекта | |
| Unit3.pas | Файл программного модуля игровой части проекта | |
| Unit5.pas | Файл программного модуля заставки проекта | |
| Unit1.dfm | Форма с главным меню | |
| Unit2.dfm | Форма с теоретическим материалом | |
| Unit3.dfm | Форма с игрой | |
| Продолжение таблицы 3 | | | | |
| 1 | | | 2 | |
| Unit5.dfm | | | Форма с заставкой | |
| Кнопка выйти из игры.png | | | Изображение | |
| Кнопка в главное меню.png | | | Изображение | |
| Кнопка заполнить игровое поле.png | | | Изображение | |
| Кнопка играть.png | | | Изображение | |
| Кнопка правила игры.png | | | Изображение | |
| Фон (игра).jpg | | | Изображение | |
| Заставка.png | | | Изображение | |
| Фон (главное меню) | | | Изображение | |
| Правила игры.htm | | | Страница с правилами игры | |
| Setup.exe | | | Установочный пакет программы | |
| Spravka.chm | | | Файл-справка | |
| Правила игры.htm | | | Страница с правилами игры | |

**4. Тестирование**

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения этапа написания программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы 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 кнопки  «Играть» | Открытие формы с игрой | Открытие формы с игрой | Выполнено |
| 3 | Проверка главного меня на кнопку «Выйти из игры» | Закрытие приложения | Закрытие приложения | Выполнено |
| 4 | Проверка кнопки  «Правила игры» | Открытие формы с правилами игры | Открытие формы с правилами игры | Выполнено |
| 5 | Проверка главного меню на кнопку «В главное меню» | Открытие формы главного меню | Открытие формы главного меню | Выполнено |
| 6 | Проверка главного меню на кнопку «Справка» | Открытие справки | Открытие справки | Выполнено |
| 7 | Проверка главного меню на кнопку «Музыка» | Раскрытие списка кнопок | Раскрытие списка кнопок | Выполнено |
| 8 | Проверка главного меню на кнопку «Выключить» | Выключение музыки | Выключение музыки | Выполнено |
| 9 | Проверка главного меню на кнопку «Включить» | Включение музыки | Включение музыки | Выполнено |

Продолжение таблицы 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 10 | Проверка главного меню на кнопку «Включить» | Включение музыки | Включение музыки | Выполнено |
| 11 | Проверка главного меню на кнопку «В главное меню» | Открытие главного меню | Открытие главного меню | Выполнено |
| 12 | Проверка кнопки «В главное меню» | Открытие главного меню | Открытие главного меню | Выполнено |
| 13 | Проверка кнопки «Заполнить игровое поле» | Появление ячеек на поле | Появление ячеек на поле | Выполнено |
| 15 | Проверка нажатия на ячейки с одинаковыми числами, либо дающими в сумме 10, стоящими рядом | Удаление ячеек | Удаление ячеек | Выполнено |
| 16 | Проверка нажатия на ячейки с неодинаковыми числами, либо дающими в сумме, нестоящими рядом | Ничего не происходит | Ничего не происходит | Выполнено |

При разработке программного продукта было решено множество проблем, связанных с ячейками, такие как: не удаление ячеек при выполнении условий удаления, не создание ячеек при нажатии на кнопку «Заполнить игровое поле».

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 данного проекта заключается в создании игры, которая обобщит знания и подготовит учащихся к квалификационному экзамену.

Игра легко запускается и функционирует на различных компьютерах. Быстродействие игры зависит от характеристик выбранного персонального компьютера, таких как процессор, объем оперативной памяти и графический адаптер. Тем не менее, вся функциональность игры оптимизирована для обеспечения плавного и комфортного игрового процесса на разных конфигурациях компьютеров.

Игра была тестирована на различных компьютерах разных классов, и она успешно функционировала на всех тестируемых машинах. Программа разработана и протестирована на персональном компьютере со следующими характеристиками:

- Процессор AMD 4500m 4 ядра, 4 потока

- Объем оперативной памяти: 8 ГБ;

- Графический адаптер: графическое ядро процессора

- Операционная система: Windows 10 домашняя.

**5.2 Инсталляция**

Для того, чтобы установить программу необходимо запустить файл Установщик.exe. Появится окно установки приложения “Two and Ten”.

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 «Two At Ten»

Вторым способом является запуск из каталога, в который устанавливалось приложение (по умолчанию C:\Program Files (x86)\ «Two At Ten»).

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 2.



Рисунок 2 – Заставка

Когда проходит заставка загружается форма с главным меню игры, на котором находятся кнопки играть, правила игры, выйти из игры, а также кнопки главного меню справка и музыка, кнопки показаны на рисунке 3

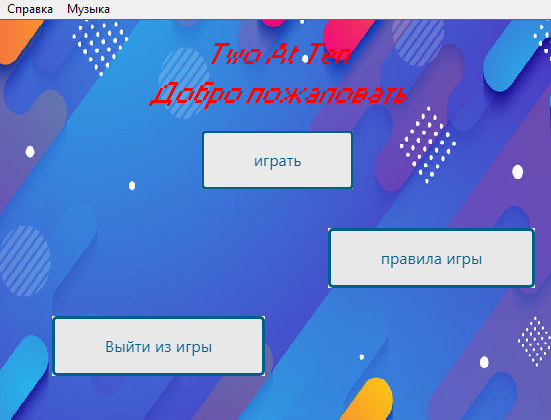


Рисунок 3 – Главное меню игры

При нажатии кнопки компонента главного меню справка, будет открываться окно с информацией об авторе и самом проекте

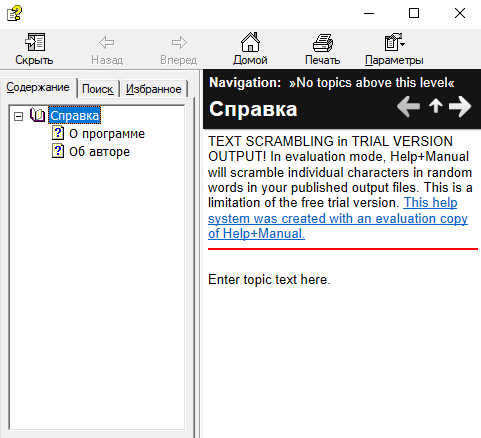


Рисунок 4 – Справка

Если в главном меню нажать кнопку правила игры, то откроется форма с правилами игры, эта форма содержит кнопки компонента главного меню, которыми можно включать/выключать музыку, а также выйти в главное меню, кнопки показаны на рисунке 5

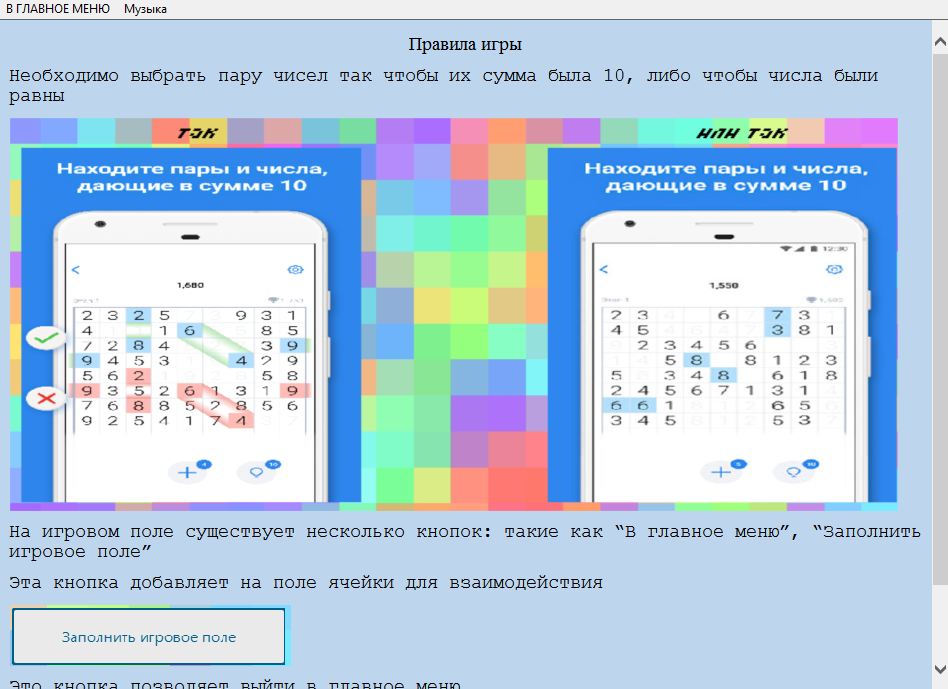


Рисунок 5 – Правила игры

При нажатии кнопки играть в главном меню игры откроется форма с игрой, на которой находятся кнопка заполнить игровое поле, при нажатии на которую на поле будут создаваться ячейки с которыми можно взаимодействовать, кнопка выйти в главное меню, а также кнопка главного меню позволяющая включать /выключать музыку

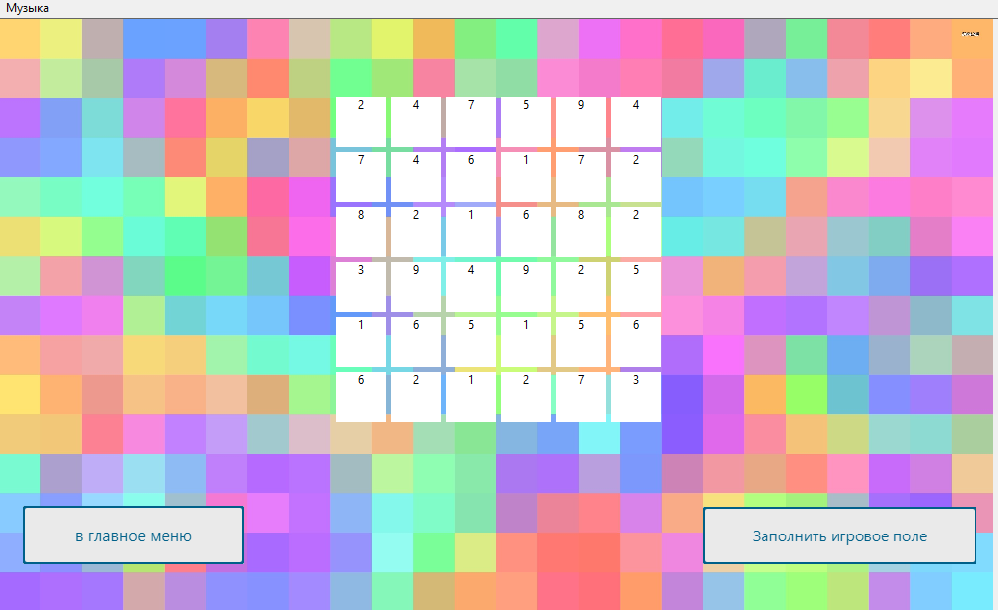


Рисунок 6 – Форма с игрой

**5.3.3 Завершение работы программы**

Для завершения работы программы необходимо в главном меню нажать кнопку «выйти из игры»

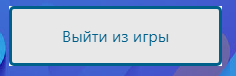


Рисунок 7 – Кнопка для выхода из игры

**Заключение**

Целью данного проекта было создание игры с целью обобщения знаний и подготовки к квалификационному экзамену. В рамках этой задачи был разработан простой и интуитивно понятный пользовательский интерфейс игры.

В ходе разработки данного программного продукта я приобрел навыки создания динамических компонентов игрового интерфейса, исследовал их взаимодействие и улучшил свои навыки создания собственных функций и процедур. Также я освоил обработку исключительных ситуаций, связанных с игровым процессом.

**Список использованных источников**

1. Григорьев, Д.А., Кузьменко, С.А. Delphi: Основы программирования. - Санкт-Петербург: БХВ-Петербург, 2019. - 576 с.

2. Шилдт, Г. Delphi. Программирование для Windows, OS X, iOS и Android. - Москва: ДМК Пресс, 2018. - 1248 с.

3. Марченко, А.В. Delphi. Программирование для Windows, OS X, iOS и Android. - Киев: Диалектика, 2017. - 496 с.

4. Райт, Р. Delphi. Программирование на языке высокого уровня. - Санкт-Петербург: Питер, 2016. - 848 с.

5. Будников, С.А. Delphi. Программирование для профессионалов. - Москва: БХВ-Петербург, 2015. - 864 с.

6. Кузьменко, С.А., Мальцев, А.В. Delphi. Компонентный подход. - Санкт-Петербург: БХВ-Петербург, 2014. - 560 с.

7. Архангельский, А. Я. Delphi 7. Справочное пособие. - Москва: Бином-Пресс, 2014. - 1024 с.

8. Чернышов, М.А. Delphi. Программирование баз данных. - Санкт-Петербург: БХВ-Петербург, 2013. - 688 с.

9. Франк, Х. Delphi. Создание клиент-серверных приложений. - Москва: ДМК Пресс, 2012. - 640 с.

10. Симдянов, И.А. Delphi. Программирование для Windows, Linux и OS X. - Санкт-Петербург: Питер, 2011. - 848 с.

**Приложение А**

Листинг программы

unit Unit5;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ExtCtrls, Vcl.Imaging.pngimage;

type

TForm5 = class(TForm)

Image1: TImage;

Timer1: TTimer;

procedure FormCreate(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

private

x: Integer;

{ Private declarations }

public

{ Public declarations }

end;

var

Form5: TForm5;

implementation

{$R \*.dfm}

uses Unit1;

procedure TForm5.FormCreate(Sender: TObject);

begin

Form5.Left:= (Screen.WorkAreaWidth - Form5.Width) div 2;

Form5.Top:= (Screen.WorkAreaHeight - Form5.Height) div 2;

end;

procedure TForm5.Timer1Timer(Sender: TObject);

begin

begin

if AlphaBlendValue = 255 then

x := x + 1;

if x = 0 then

AlphaBlendValue := AlphaBlendValue + 5

else if x = 100 then

AlphaBlendValue := AlphaBlendValue - 5;

if AlphaBlendValue = 0 then

begin

Form1.Show;

Timer1.Enabled := False;

Hide;

end;

end;

end;

end.

unit Unit1;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Imaging.pngimage,

Vcl.ExtCtrls, Vcl.Imaging.jpeg, Vcl.Menus,ShellAPI, Vcl.MPlayer;

type

TForm1 = class(TForm)

Label2: TLabel;

Image1: TImage;

Image3: TImage;

Label1: TLabel;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

MediaPlayer1: TMediaPlayer;

Onoff1: TMenuItem;

N3: TMenuItem;

Image5: TImage;

Image4: TImage;

Image2: TImage;

procedure Image1Click(Sender: TObject);

procedure Image3Click(Sender: TObject);

procedure N1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure Onoff1Click(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure Image5Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

uses Unit2, Unit3;

procedure TForm1.FormCreate(Sender: TObject);

begin

MediaPlayer1.FileName:='von.mp3';

MediaPlayer1.Open;

MediaPlayer1.Play;

end;

procedure TForm1.Image1Click(Sender: TObject);

begin

Form1.Hide;

Form3.show;

end;

procedure TForm1.Image3Click(Sender: TObject);

begin

Mediaplayer1.Destroy;

Close;

end;

procedure TForm1.Image5Click(Sender: TObject);

begin

Form1.Hide;

Form2.Show;

Form2.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'Правила игры.htm');

end;

procedure TForm1.N1Click(Sender: TObject);

begin

ShellExecute(0,PChar('open'),PChar('Spravka.chm'),nil,nil,SW\_SHOW);

end;

procedure TForm1.N3Click(Sender: TObject);

begin

MediaPlayer1.Play;

end;

procedure TForm1.Onoff1Click(Sender: TObject);

begin

MediaPlayer1.Stop;

end;

end.

unit Unit2;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Imaging.jpeg,

Vcl.Imaging.pngimage, Vcl.ExtCtrls, Vcl.OleCtrls, SHDocVw, Vcl.Menus;

type

TForm2 = class(TForm)

WebBrowser1: TWebBrowser;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

procedure N1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure N4Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

{$R \*.dfm}

uses Unit1;

procedure TForm2.FormCreate(Sender: TObject);

begin

Form2.Left:= (Screen.WorkAreaWidth - Form2.Width) div 2;

Form2.Top:= (Screen.WorkAreaHeight - Form2.Height) div 2;

end;

procedure TForm2.N1Click(Sender: TObject);

begin

form2.Hide;

form1.show;

end;

procedure TForm2.N3Click(Sender: TObject);

begin

form1.MediaPlayer1.Play;

end;

procedure TForm2.N4Click(Sender: TObject);

begin

form1.MediaPlayer1.Stop;

end;

end.

unit Unit2;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Imaging.jpeg,

Vcl.Imaging.pngimage, Vcl.ExtCtrls, Vcl.OleCtrls, SHDocVw, Vcl.Menus;

type

TForm2 = class(TForm)

WebBrowser1: TWebBrowser;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

procedure N1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure N4Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

{$R \*.dfm}

uses Unit1;

procedure TForm2.FormCreate(Sender: TObject);

begin

Form2.Left:= (Screen.WorkAreaWidth - Form2.Width) div 2;

Form2.Top:= (Screen.WorkAreaHeight - Form2.Height) div 2;

end;

procedure TForm2.N1Click(Sender: TObject);

begin

form2.Hide;

form1.show;

end;

procedure TForm2.N3Click(Sender: TObject);

begin

form1.MediaPlayer1.Play;

end;

procedure TForm2.N4Click(Sender: TObject);

begin

form1.MediaPlayer1.Stop;

end;

end.