**Введение**

На курсовое проектирование была поставлена задача разработать программный продукт на тему: «Развиваем внимательность».

Цель курсового проекта заключается в разработке программного продукта, который позволит массовому пользователю улучшить внимательность, концентрацию и когнитивные способности.

Создаваемое приложение будет рассчитано для любого рода пользователей. Применять его смогут как дети, так и взрослые.

Далее 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. 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 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 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 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 «Список использованных источников» будет приведен список используемых при разработке источников.

В приложении 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

**Наименование задачи:** «Разработка программного продукта «Развиваем внимательность».

**Цель разработки:** создание программного продукта, который рассчитан на людей, заинтересованных в развитии различных качеств, таких как: повышение качества внимания, улучшение памяти и когнитивных функций, а также снижение уровня стресса. Игра может быть использована в образовательных и развивающих целях, как инструмент для повышения когнитивных способностей у детей и взрослых. Она также может быть применена в реабилитационных программах для людей, восстанавливающихся после травм головного мозга или страдающих от когнитивных нарушений.

**Назначение:** для развития способности к фокусировке внимания на задаче, поставленном задании и увеличения продуктивности своей деятельности, для повышения качества внимания, улучшения памяти и когнитивных функций, а также снижения уровня стресса

**Предметная область:** это программный продукт, который поможет развивать и тренировать внимательность и память пользователей. Игра будет представлять собой игровое поле с ячейками, которые случайным образом заполняются числами. Основная задача игрока — как можно быстрее и в правильном порядке назвать или указать последовательность чисел. Этот процесс способствует улучшению когнитивных функций, таких как концентрация внимания, скорость обработки анализируемой информации и развитие кратковременной памяти.

**Периодичность использования:** зависит от нужд потребителя, может использоваться ежедневно.

**Источники и способы получения данных:** продукт будет содержать 5 уровней сложности, представляющих собой игровое поле с ячейками, которые случайным образом заполняются числами, игровой процесс будет иметь различное визуальное и звуковое оформление, также после прохождения уровней пользователь сможет посмотреть статистику.

Разрабатываемый программный продукт позволит выполнить следующие действия:

* осуществить игровую сессию;
* возможность самостоятельно выбирать визуальное и звуковое оформление;
* осуществить просмотр статистики прохождения различных уровней;
* ознакомиться с правилами игры.

# 1.2 Инструменты разработки

Для разработки данного проекта выбрана среда Delphi (Delphi 10), так как это среда объектно-ориентированного программирования, относящаяся к классу RAD – (Rapid Application Development – «Средство быстрой разработки приложений»), реализованная на Object Pascal. Используется для разработки визуализированного представления программного обеспечения.

Также для разработки программы необходимы:

* Google – программа-браузер для нахождения информации, графических изображений и аудиофайлов;
* Word 2019 – для написания пояснительной записки, создания html-страниц;
* Power Point 2010 – для создания отчётной презентации;
* Help and Manual– для справочной системы;
* Smart Install Maker – для создания инсталятора.

При разработке данного программного продукта был использован компьютер со следующими характеристиками:

Процессор: 12th Gen Intel(R) Core(TM) i5-12450H 2.00 GHz;

* ОЗУ: 16Gb;
* память: SDD 512Gb;
* ОС – Windows 10.

# 1.3 Требования к приложению

На этапе исследования предметной области был установленный целый ряд требований, предъявляемые к разрабатываемому программному продукту.

При моделировании форм следует учесть такие моменты:

* интуитивно понятный интерфейс и управление;
* небольшое количество информации на экране;
* небольшие окна формы;
* визуально понятный и приятный глазу интерфейс;
* наличие информации о функционале приложения;
* небольшие кнопки.

Минимальные системные требования:

* 1.8 GHz процессор;
* 1 GB RAM;
* от 500 MB доступного места на жестком диске.

Рекомендуемые системные требования:

* двухъядерный AMD Athlon или аналогичный процессор Intel (или более производительный);
* 1 GB RAM;
* от 300 MB доступного места на жестком диске.

При разработке интерфейса приложения использованы различные цвета, а также для пользователя доступен выбор фона и музыки для комфортного пользования. Основные разделы приложения доступны с первой страницы. Разработан понятный пользователю интерфейс, фрагменты текста располагаются на экране так, чтобы пользователь не испытывал какого-либо дискомфорта в плане восприятия информации, отображённой на экране.

Ошибки программы вследствие некорректного взаимодействия пользователя при работе с программой через графический интерфейс не должны влиять на работоспособность.

Надёжность программы должна быть на высоком уровне, не допуская ситуаций программных ошибок и некорректной работы приложения.

# 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 два вида данных.

Первым видом являются данные, которые будут введены разработчиком на этапе реализации задачи. Сюда можно отнести изображения (фоновые изображения, кнопки), аудиофайлы, файлы для просмотра статистики.

Вторым видом данных является результат программы – например, вывод сообщения о результате прохождения игры.

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

# 2.2 Процессы

Согласно всем перечисленным требованиям и указаниям, которые были рассмотрены в разделе «Анализ задачи», было определено, чем конкретно должна заниматься разрабатываемая программа. Главной ее задачей будет возможность прохождения уровней игры.

Для реализации задач будут использоваться процедуры. С помощью процедур, например, будет осуществляться загрузка текстовой информации в MessageBox, выбор фона или музыки, поочередный доступ и реализация уровней различной сложности, а также просмотр статистики, переход с одной формы на другую.

# 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.

Особое внимание следует уделить интерфейсу. Разработчик должен так организовать внешний вид своей программы, что бы пользователь понял, что от него требуется.

Для организации эффектной 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.

Таким образом, для успешной работы всего проекта в целом следует обеспечить интуитивно понятный интерфейс с приятными цветами и шрифтами.

Структура навигации по проекту представлена на рисунке 2.3.

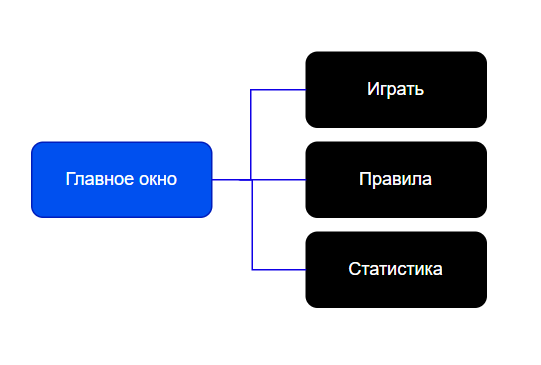
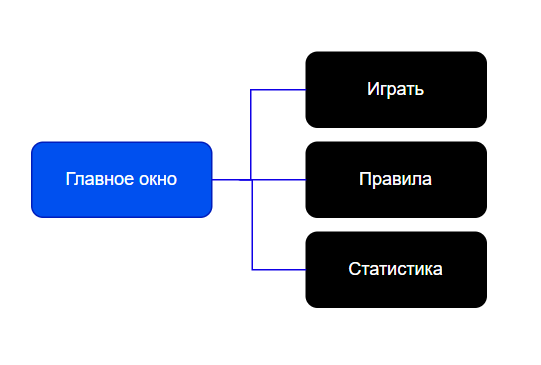


Рисунок 2.1 – Навигация между окнами программы

**3 Реализация**

**3.1 Структура программы**

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

* модуль Unit2 – главное меню приложения;
* модуль Unit1 – загрузочный экран
* модуль Unit3, Unit4 – отвечают за прохождение уровней игры, последовательный доступ, а также за сохранение результатов прохождения в файлы;
* модуль Unit5 – отвечает за отображение статистики;

**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 элементом управления закреплена | Назначение |
| 1 | 2 | 3 | 4 |
| procedure Timer1Timer(Sender: TObject); | Unit1.pas | Timer1 | Осуществляет загрузку ProgressBar1 |
| procedure FormCreate(Sender: TObject); | Unit1.pas | Form1 | Создание формы Form1 |
| procedure ComboBox1Change(Sender: TObject); | Unit2.pas | ComboBox1 | Выбор фона |
| procedure FormClose(Sender: TObject; var Action: TCloseAction); | Unit2.pas | Form2 | Закрытие программы |
| procedure FormCreate(Sender: TObject); | Unit2.pas | Form2 | Создание формы Form2 |
| procedure N3Click(Sender: TObject); | Unit2.pas | N3 | Закрытие программы |
| procedure N2Click(Sender: TObject); | Unit2.pas | N2 | Вывод правил в MessageBox |
| procedure N5x5Click(Sender: TObject); | Unit2.pas | N5x5 | Переход на Form3, запуск уровня 5х5 |
| procedure N6x6Click(Sender: TObject); | Unit2.pas | N6x6 | Переход на Form3, запуск уровня 6х6 |
| procedure N7x7Click(Sender: TObject); | Unit2.pas | N7x7 | Переход на Form3, запуск уровня 7х7 |
| Продолжение таблицы 1 |  |  |  |
| 1 | 2 | 3 | 4 |
| procedure N9x9Click(Sender: TObject); | Unit2.pas | N9x9 | Переход на Form3, запуск уровня 9х9 |
| procedure CreateParams(var params:TcreateParams); | Unit2.pas | Form2 | Создание иконки приложения в панели инструментов |
| procedure Image2Click(Sender: TObject); | Unit2.pas | Image2 | Вывод правил в MessageBox |
| procedure Label2Click(Sender: TObject); | Unit2.pas | Label2 | Вывод правил в MessageBox |
| procedure Image3Click(Sender: TObject); | Unit2.pas | Image3 | Переход на форму Form3 |
| procedure Label3Click(Sender: TObject); | Unit2.pas | Label3 | Переход на форму Form3 |
| procedure Image4Click(Sender: TObject); | Unit2.pas | Image4 | Переход на форму Form5 |
| procedure Label4Click(Sender: TObject); | Unit2.pas | Label4 | Переход на форму Form5 |
| procedure Image5Click(Sender: TObject); | Unit2.pas | Image5 | Закрытие программы |
| procedure Label5Click(Sender: TObject); | Unit2.pas | Label5 | Закрытие программы |
| procedure FormActivate(Sender: TObject); | Unit2.pas | Form2 | Активация формы |
| procedure ComboBox2Change  (Sender: TObject); | Unit2.pas | ComboBox2 | Выбор музыки |
| procedure StringGrid1DrawCell  (Sender: TObject; ACol, ARow: Integer;Rect: TRect; State: TGridDrawState); | Unit3.pas | StringGrid1 | Отрисовка клеток игрового поля |
| procedure StringGrid1MouseDown(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer); | Unit3.pas | StringGrid1 | Выбор правильной ячейки |
| procedure FormDestroy(Sender: TObject); | Unit3.pas | Form3 | Уничтожить форму |
| procedure Timer1Timer(Sender: TObject); | Unit3.pas | Timer1 | Работа с таймером |
| procedure FormActivate(Sender: TObject); | Unit3.pas | Form3 | Активация формы |
| procedure ComboBox1Change  (Sender: TObject); | Unit3.pas | ComboBox1 | Выбор уровня |
| procedure N1Click(Sender: TObject); | Unit3.pas | N1 | Переход на форму Form2 |
| procedure N2Click(Sender: TObject); | Unit3.pas | N2 | Закрытие программы |
| Продолжение таблицы 1 |  |  |  |
| 1 | 2 | 3 | 4 |
| procedure N3Click(Sender: TObject); | Unit3.pas | N3 | Вывод правил в MessageBox |
| procedure Image2Click(Sender: TObject); | Unit3.pas | Image2 | Переход на форму Form2 |
| procedure Label1Click(Sender: TObject); | Unit3.pas | Label1 | Переход на форму Form2 |
| procedure Image3Click(Sender: TObject); | Unit3.pas | Image3 | Остановка таймера |
| procedure Label2Click(Sender: TObject); | Unit3.pas | Label2 | Остановка таймера |
| procedure Image4Click(Sender: TObject); | Unit3.pas | Image4 | Продолжение таймера |
| procedure Label3Click(Sender: TObject); | Unit3.pas | Label3 | Продолжение таймера |
| procedure StartTimer(); | Unit3.pas | Timer1 | Работа с таймером |
| procedure SetDesItems(); | Unit3.pas |  | Установить доступные и недоступные уровни в ComboBox |
| procedure CreateParams(var params:TcreateParams); override; | Unit3.pas | Form3 | Создание иконки приложения в панели инструментов |
| procedure FormCreate(Sender: TObject); | Unit3.pas | Form3 | Создание формы |
| procedure FormClose(Sender: TObject; var Action: TCloseAction); | Unit3.pas | Form3 | Закрытие формы Form3 |
| procedure ccGreenClick(Sender: TObject); | Unit3.pas | ccGreen | Подсвечивание |
| procedure RepaintGreen(); | Unit3.pas | ccGreen | Отмечает клетки зеленым цветом |
| procedure SaveTimeInLeaderBoard(time:string; num:integer); | Unit4.pas | - | Cохраняет переданное время в лидерборд |
| function LoadLeaderBoard(num:integer): arr; | Unit4.pas | - | Dозвращает все строки из лидерборда |
| procedure SaveLevel(num:integer); | Unit4.pas | - | Cохраняет переданный уровень, если он выше чем текущий записанный |
| function LoadMaxLevel():integer; | Unit4.pas | - | Возвращает максимальный уровень |
| Продолжение таблицы 1 |  |  |  |
| 1 | 2 | 3 | 4 |
| procedure Createmaxlevel(); | Unit4.pas | - | Создаёт файл для хранения уровня |
| procedure LoadRes(num:integer); | Unit5.pas | Memo1 | Вывод статистики |
| procedure FormActivate(Sender: TObject); | Unit5.pas | Form5 | Активация формы |
| procedure CreateLeaderBoard(); | Unit4.pas | - | Создаёт файлы для хранения данных лидерборда |
| function SortStrings(const Strings: arr): TStringList; | Unit4.pas | - | Сортирует строки лидерборда по возрастанию |
| function GetBestResult(num:integer):string; | Unit4.pas | - | Получить лучший результат пользователя |
| procedure Image2Click(Sender: TObject); | Unit5.pas | Image2 | Вывод статистики уровня 5х5 |
| procedure Label1Click(Sender: TObject); | Unit5.pas | Label1 | Вывод статистики уровня 5х5 |
| procedure Image3Click(Sender: TObject); | Unit5.pas | Image3 | Вывод статистики уровня 6х6 |
| procedure Label2Click(Sender: TObject); | Unit5.pas | Label2 | Вывод статистики уровня 6х6 |
| procedure Image4Click(Sender: TObject); | Unit5.pas | Image4 | Вывод статистики уровня 7х7 |
| procedure Label3Click(Sender: TObject); | Unit5.pas | Label3 | Вывод статистики уровня 7х7 |
| procedure Image5Click(Sender: TObject); | Unit5.pas | Image5 | Вывод статистики уровня 8х8 |
| procedure Label4Click(Sender: TObject); | Unit5.pas | Label4 | Вывод статистики уровня 8х8 |
| procedure Image6Click(Sender: TObject); | Unit5.pas | Image6 | Вывод статистики уровня 9х9 |
| procedure Label5Click(Sender: TObject); | Unit5.pas | Label5 | Вывод статистики уровня 9х9 |
| procedure Image7Click(Sender: TObject); | Unit5.pas | Image7 | Переход на форму Form2 |
| procedure Label7Click(Sender: TObject); | Unit5.pas | Label7 | Переход на форму Form2 |

**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 |
| 1 | 2 | 3 |
| TMainMenu | Form2, Form3, | Используется для создания меню проекта |
| TImage | Form1, Form2, Form3, Form5 | Используется как фон, кнопки |
| TLabel | Form1, Form2, Form3, Form5 | Отображение надписей на форме |
| TTimer | Form1, Form3 | Используется для определения длительности загрузочного экрана, контроля времени в игре |
| TComboBox | Form2, Form3 | Используется для выбора музыки, выбора фона, выбора уровня |
| TMediaPlayer | Form2 | Используется для воспроизведения, паузы и остановки аудиофайлов |
| TRadioButton | Form3 | Используется для окрашивания ячеек |
| TStringGrid | Form3 | Используется в качестве игрового поля |
| TProgressBar | Form2 | Используется для отображения процесса загрузки |
| TMemo | Form5 | Используется для вывода статистики |

**3.2 Специфика программы**

Точное название проекта и его состав приводится в Таблице 3.

Таблица 3 – Спецификация программы

|  |  |  |
| --- | --- | --- |
| Имя файла | Назначение | |
| 1 | 2 | |
| Proоect1.dpr | Файл проекта, связывает все файлы, из которых состоит приложение | |
| Project1.exe | Исполняемый файл проекта | |
| Unit1.pas | Файл программного модуля для формы Form1 | |
| Unit2.pas | Файл программного модуля для формы Form2 | |
| Unit3.pas | Файл программного модуля для формы Form3 | |
| Unit4.pas | Файл программного модуля для формы Form3 | |
| Unit5.pas | Файл программного модуля для формы Form5 | |
| Unit1.dfm | Форма Form1 | |
| Unit2.dfm | Форма Form2 | |
| Unit3.dfm | Форма Form3 | |
| Unit5.dfm | Форма Form5 | |
| Продолжение таблицы 3 | |  |
| 1 | 2 | |
| Справка.chm | Справка | |
| Project1.res | Файл описания ресурсов | |
| Unit1.dcu | Файл с откомпилированным кодом модуля Unit1 | |
| Unit2.dcu | Файл с откомпилированным кодом модуля Unit2 | |
| Unit3.dcu | Файл с откомпилированным кодом модуля Unit3 | |
| Unit4.dcu | Файл с откомпилированным кодом модуля Unit4 | |
| Unit5.dcu | Файл с откомпилированным кодом модуля Unit5 | |
| Project1.dproj | Файл проекта, связывает все файлы, из которых состоит приложение. | |
| Setup.exe | Установщик | |
| \*.txt | Текстовые файлы | |
| \*.mp3 | Аудиофайлы | |
| \*.bmp | Картинки | |

**4 Тестирование**

Отчет о результатах тестирование представлен в таблице 4.

Таблица 4 – Отчет о результатах тестирования.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Тест | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Проверка пункта меню  «Играть» | Откроется форма Form3 | Открытие формы Form3 | Выполнено |
| 2 | Проверка пункта меню «Выйти» | Программа закроется | Закрытие программы | Выполнено |
| 3 | Проверка пункта меню «Правила» | Откроется MessageBox с правилами игры | Открытие MessageBox с правилами игры | Выполнено |
| 4 | Проверка пункта меню «Справка» | Откроется Справка | Открытие справки | Выполнено |
| 5 | Проверка кнопки «Правила» | Откроется MessageBox с правилами игры | Открытие MessageBox с правилами игры | Выполнено |
| 6 | Проверка кнопки «Справка» | Откроется Справка | Открытие справки | Выполнено |
| 7 | Проверка кнопки «Статистика» | Откроется форма Form5 | Открытие формы Form5 | Выполнено |
| 8 | Проверка кнопки «Выйти» | Программа закроется | Закрытие программы | Выполнено |
| 9 | Проверка ComboBox1(Выбрать музыку) | Начнет воспроизводится выбранная музыка | Воспроизведение выбранной музыки | Выполнено |
| 10 | Проверка ComboBox2(Выбрать фон) | Установится выбранный фон | Установка выбранного фон | Выполнено |
| 11 | Проверка кнопки «На главную» | Откроется форма Form2 | Открытие формы Form2 | Выполнено |
| 12 | Проверка lblTime | Отобразиться идущий таймер | Отображение идущего таймер | Выполнено |
| 13 | Проверка ComboBox1(Выбрать уровень) | Откроется выбранный уровень только при условии, если он доступен | Открытие выбранного уровня если он доступен | Выполнено |
| 16 | Проверка кнопки «Остановить» | Приостановка таймера | Приостановка таймера | Выполнено |
| 17 | Проверка кнопки «Продолжить» | Возобновление таймера если ранее был остановлен | Ввозобновление таймера если ранее был остановлен | Выполнено |
| Продолжение таблицы 4 | |  |  |  |
| 1 | 2 | 3 | 4 | 5 |
| 18 | Проверка кнопки «5х5» | Отображение статистики по данному уровню | Отображение статистики по данному уровню | Выполнено |
| 19 | Проверка кнопки «6х6» | Отображение статистики по данному уровню | Отображение статистики по данному уровню | Выполнено |
| 20 | Проверка кнопки «7х7» | Отображение статистики по данному уровню | Отображается статистика по данному уровню | Выполнено |
| 21 | Проверка кнопки «8х8» | Отображение статистики по данному уровню | Отображается статистика по данному уровню | Выполнено |
| 22 | Проверка кнопки «9х9» | Отображение статистики по данному уровню | Отображается статистика по данному уровню | Выполнено |
| 14 | Проверка работы StringGrid1 | При открытии формы Form3 во всех ячейках должны располагаться числа в рандомном порядке, после нажатия на все ячейки в порядке возрастания должен выводиться MessageBox, при нажатии на неправильную ячейку должен прозвучать звуковой сигнал | При открытии формы Form3 во всех ячейках располагаются числа в рандомном порядке, после нажатия на все ячейки в порядке возрастания выводится MessageBox, при нажатии на неправильную ячейку слышен звуковой сигнал | Выполнено |
| 15 | Проверка кнопки ccGreen | При нажатии на кнопку все правильно нажатые ячейки должны закрашиваться в зеленый цвет, на форме должно отображаться число, которое следует нажать следующим | При нажатии на кнопку все правильно нажатые ячейки закрашиваются в зеленый цвет, на форме отображается число, которое следует нажать следующим | Выполнено |

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 разработке программного продукта, который рассчитан на людей, заинтересованных в развитии различных качеств, таких как: повышение качества внимания, улучшение памяти и когнитивных функций, а также снижение уровня стресса. Игра может быть использована в образовательных и развивающих целях, как инструмент для повышения когнитивных способностей у детей и взрослых. Она также может быть применена в реабилитационных программах для людей, восстанавливающихся после травм головного мозга или страдающих от когнитивных нарушений.

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 разных классах ЭВМ и работать с данной программой было комфортно. Программа разработана на ПК со следующими характеристиками:

* процессор 12th Gen Intel(R) Core(TM) i5-12450H, 2,00 ГГц, ядер: 8, логических процессоров: 12
* объем ОЗУ 16Гб;
* графический адаптер NVIDIA GeForse RTX 3050 Laptop GPU
* операционная система Windows 10.

**5.2 Инсталляция**

Для того, чтобы установить программу необходимо запустить файл Setup.exe. Появится окно установки приложения «Развиваем внимательность».

Затем достаточно следовать приведенной инструкции установки приложения.

**5.3 Выполнение программы**

**5.3.1 Запуск программы**

Данную программу можно запустить различными способами.

* + запуск из каталога, в который устанавливалось приложение (по умолчанию C:\Desktop\КП\Проект1\Win32\Debug\Project.exe).
  + запуск с помощью ярлыка на рабочем столе.

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 Инструкция по работе с программой**

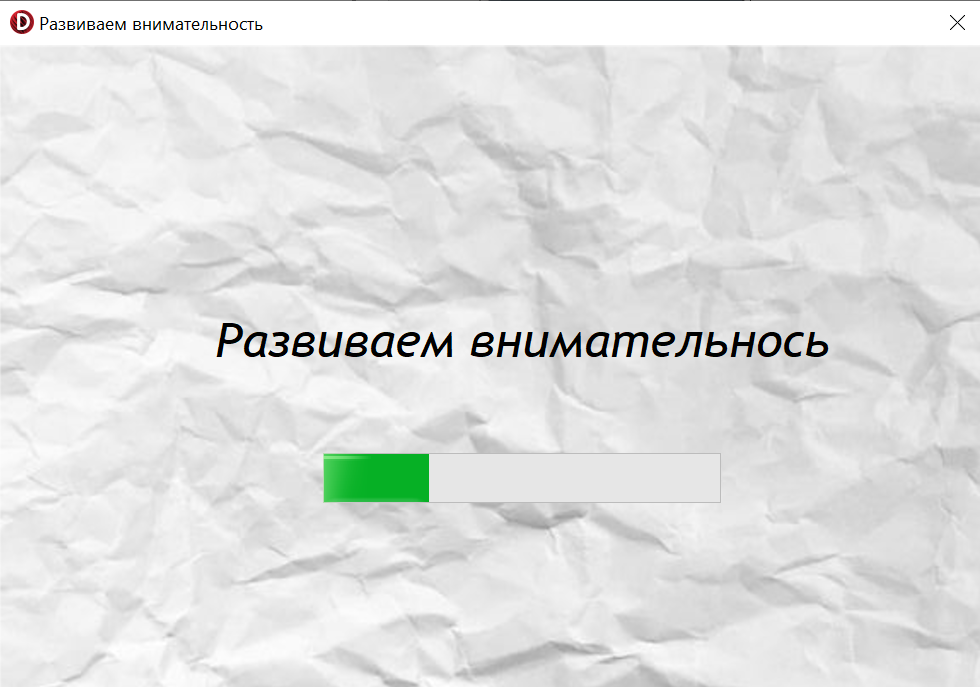
После запуска приложения нас встречает экран загрузки (рисунок 5.1).

Рисунок 5.1 – Загрузочный экран

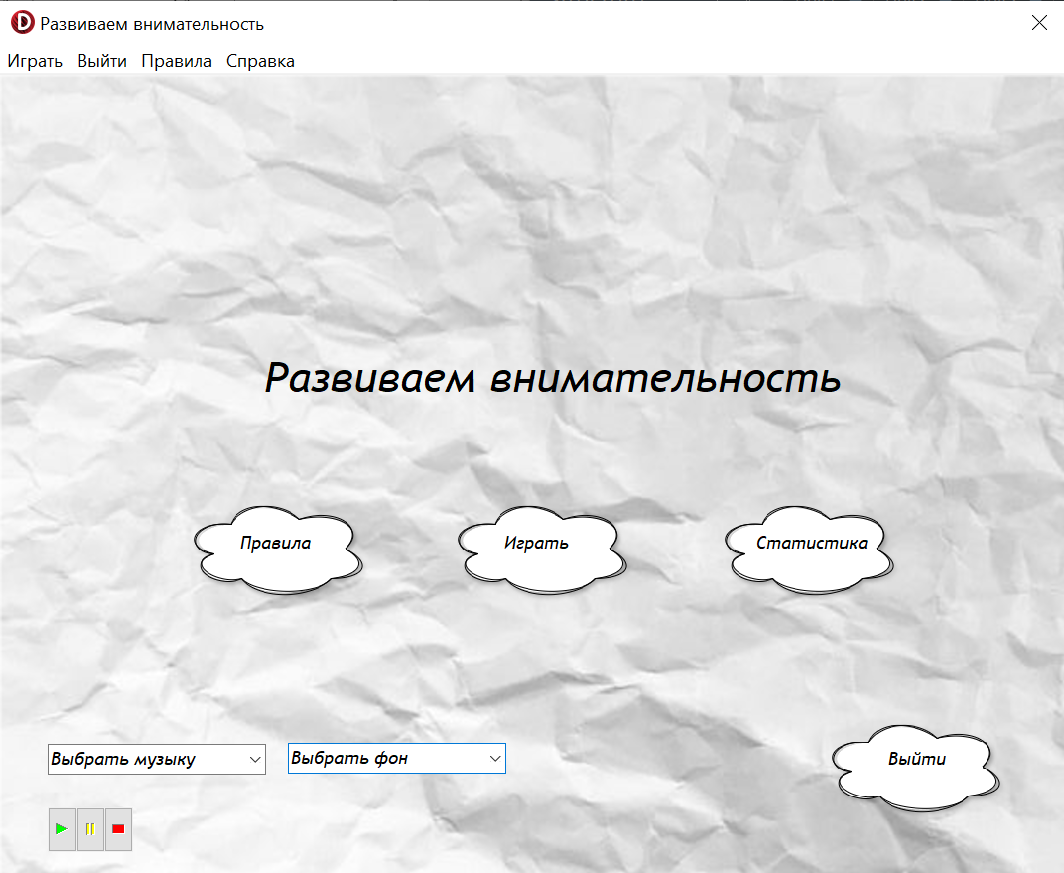
Далее автоматически открывается главная форма (рисунок 5.2).

Рисунок 5.2 – Главная форма

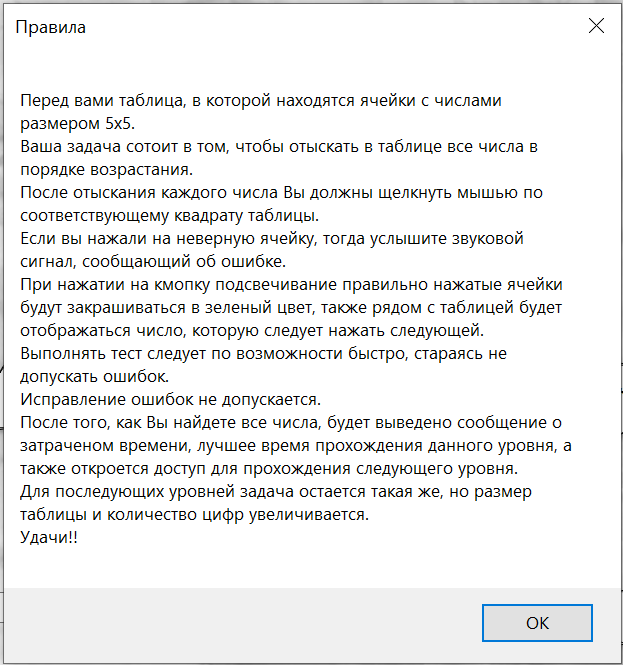
При нажатии на кнопку «Правила» будет отображаться MessageBox с подробным объяснением как проходить игру (рисунок 5.3).

Рисунок 5.3 – Правила

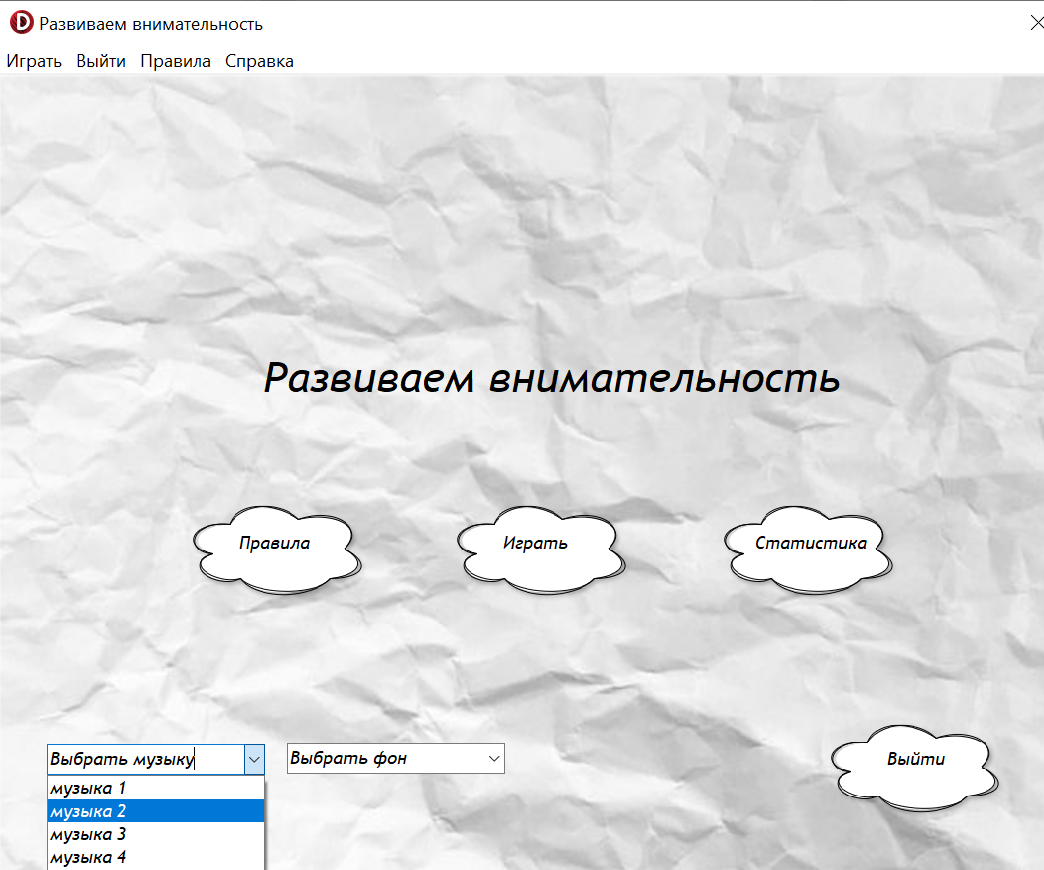
При нажатии на кнопку «Выбор музыки» пользователю будет предоставлен выбор из 5 аудиофайлов для комфортного пользования (рисунок 5.4).

Рисунок 5.4 – Выбор музыки

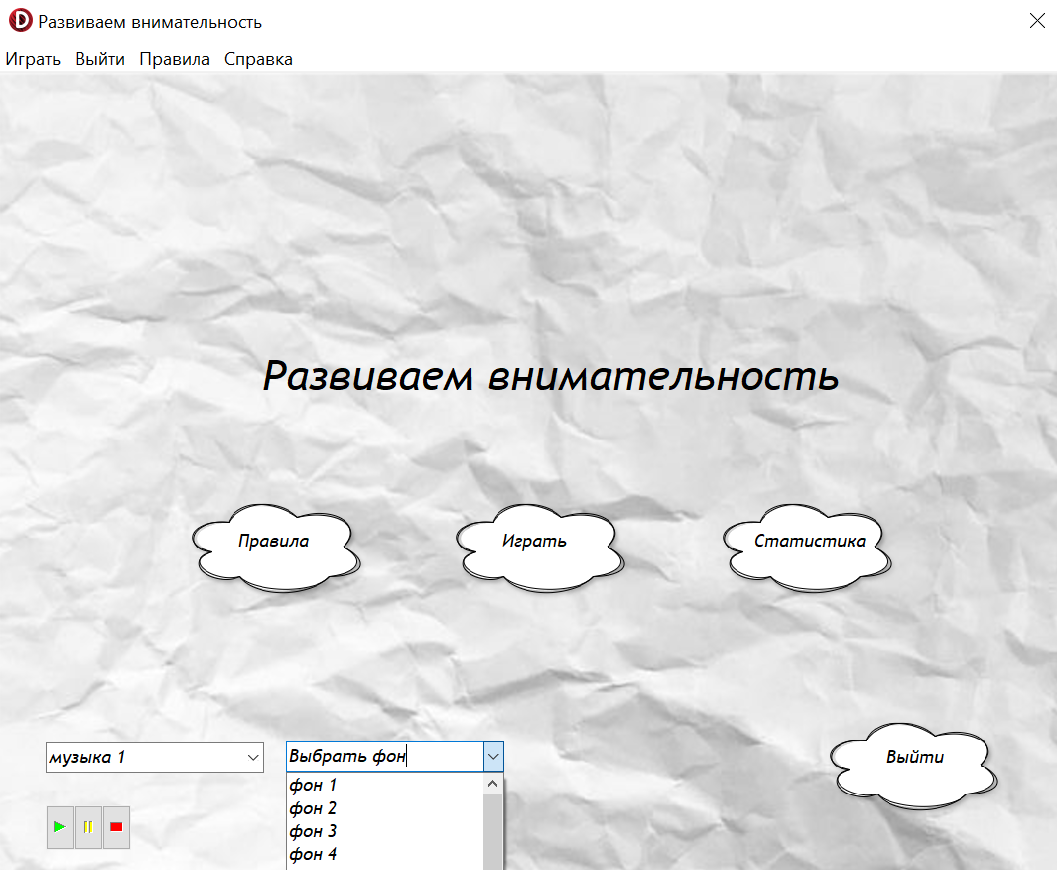
При нажатии на кнопку «Выбор фона» пользователю будет предоставлен выбор из 10 различных фонов для комфортного пользования (рисунок 5.5).

Рисунок 5.5 – «Выбор фона»

При нажатии на кнопку «Статистика» Будет осуществлен переход на форму со статистикой (рисунок 5.6). При нажатии на одну из кнопок уровней в Memo будет отображаться статистика по данному уровню. Также на этой форме есть кнопка «На главную», которая осуществляет переход на главную форму.

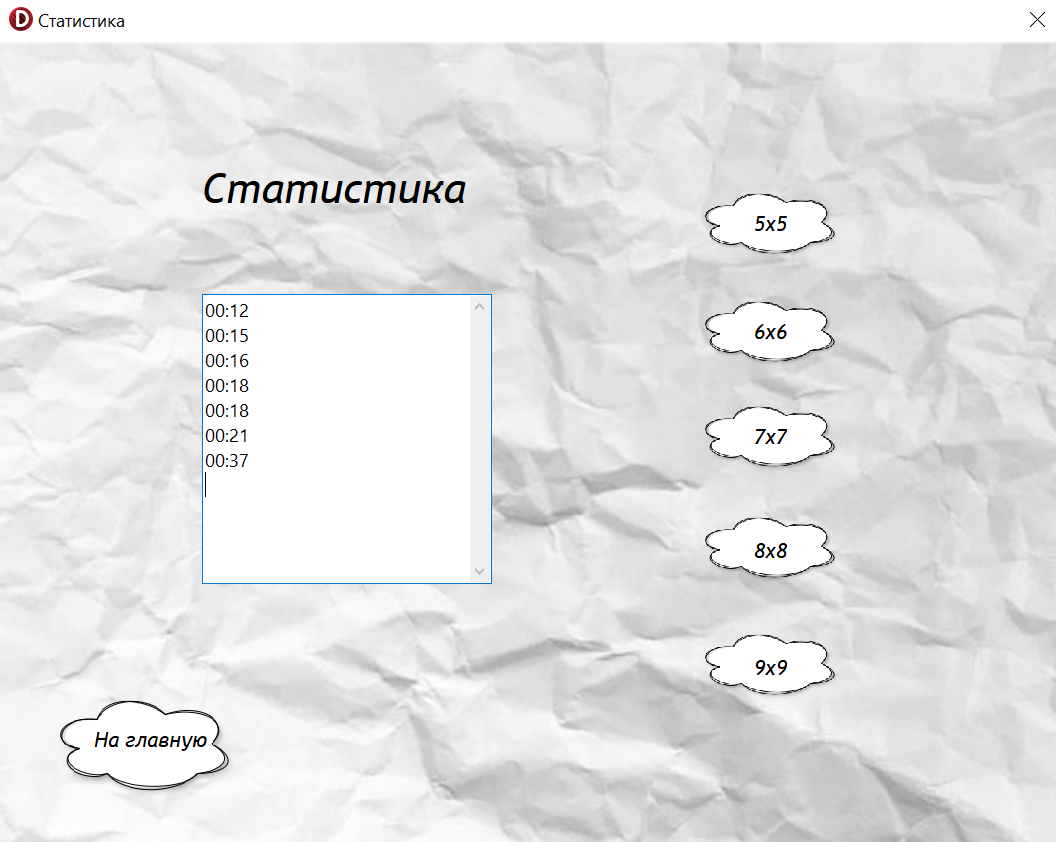


Рисунок 5.6 – Форма «Статистика»

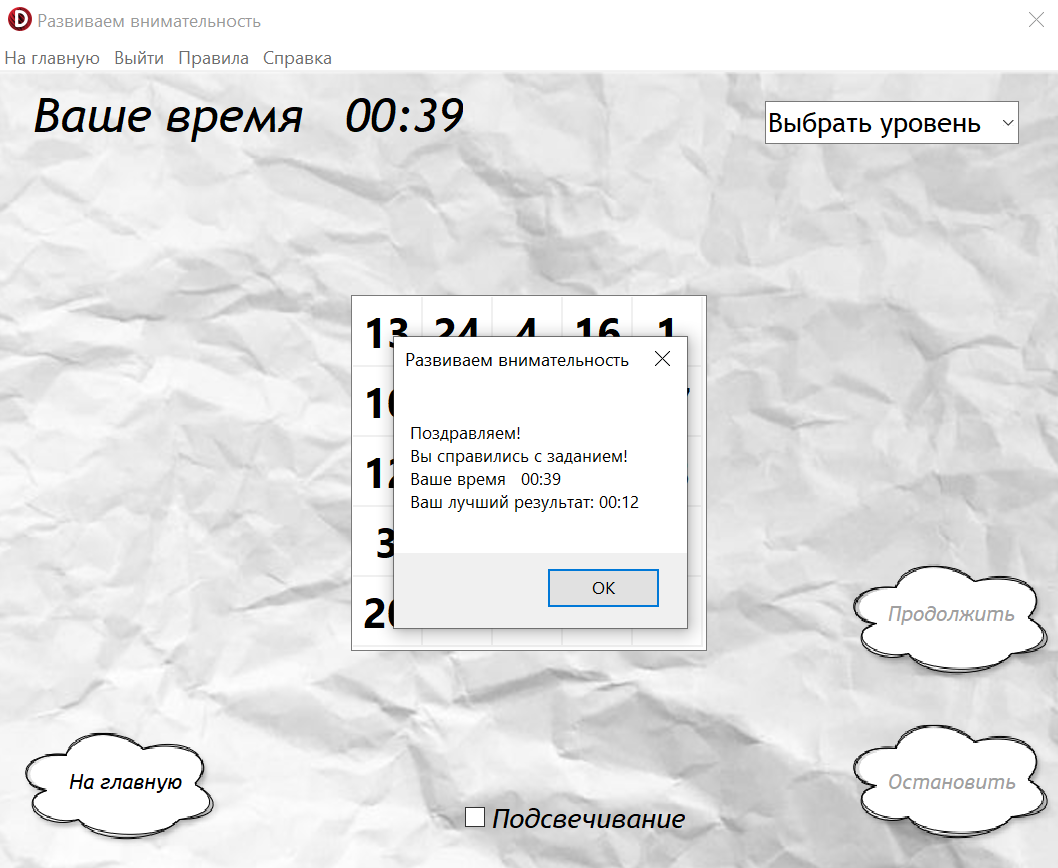
На главной форме при нажатии на кнопку «играть» будет осуществлен переход на форму с игрой. На ней располагается игровое поле с ячейками, которые в рандомном порядке заполняются числами. При нажатии на все ячейки с числами в правильном порядке появляется MessageBox с результатом прохождения уровня и лучшим результатом прохождения данного уровня, а также открывается доступ к следующему уровню (рисунок 5.7).

Рисунок 5.7 – Прохождение игры

При нажатии на кнопку «Подсвечивание» правильно нажатые ячейки закрашиваются и возле игрового поля появляется следующее число, которое нужно нажать (рисунок 5.8).



Рисунок 5.8 – Реализация «Подсвечивания»

При нажатии на кнопку «Остановить» происходит приостановка таймера. При нажатии на кнопку «Продолжить» приостановленный таймер возобновляется. При нажатии на кнопку «На главную» происходит переход на Главную форму.

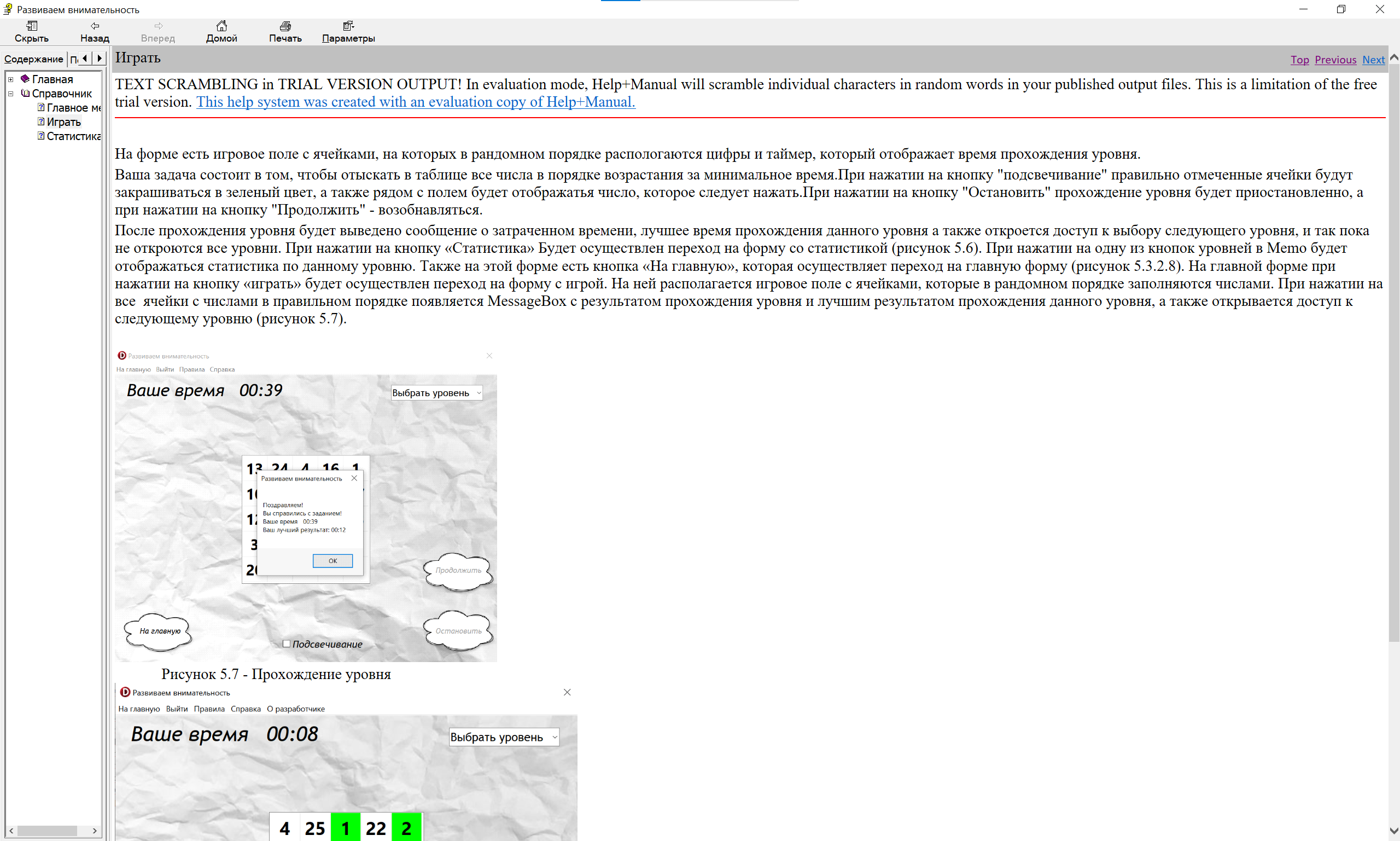
Также, на главной форме на главном меню реализована кнопка «Справка», нажав на которую открывается справочная система (рисунок 5.9).

Рисунок 5.9 - Справка

**5.3.3 Использование системы справочной информации**

Справочную систему можно запустить с помощью пункта меню «Справка» (рисунок 5.10).

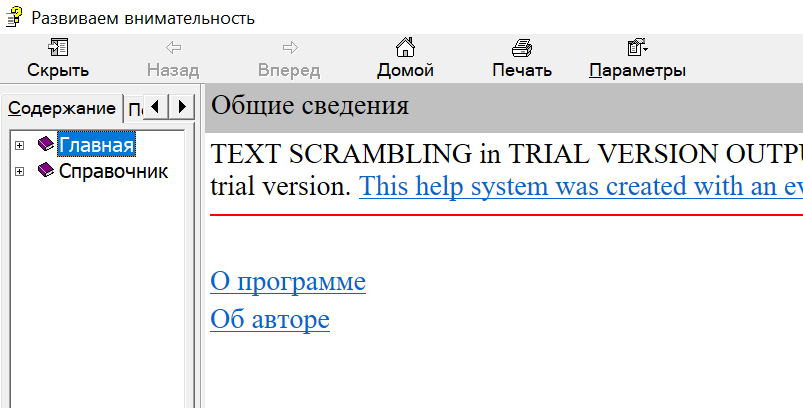


Рисунок 5.10 – Справка

**Заключение**

Цель данного проекта заключается в разработке программного продукта, который рассчитан на людей, заинтересованных в развитии различных качеств, таких как: повышение качества внимания, улучшение памяти и когнитивных функций, а также снижение уровня стресса. Игра может быть использована в образовательных и развивающих целях, как инструмент для повышения когнитивных способностей у детей и взрослых. Она также может быть применена в реабилитационных программах для людей, восстанавливающихся после травм головного мозга или страдающих от когнитивных нарушений.

В ходе выполнения данной курсовой работы было разработано и протестировано приложение «Развиваем внимательность» с графическим интерфейсом Приложение содержит 5 уровней сложности, представляющих собой игровое поле с ячейками, которые случайным образом заполняются числами, игровой процесс иметь различное визуальное и звуковое оформление, также после прохождения уровней пользователь может посмотреть статистику.

При разработке программного продукта использовалась объектно-ориентированная технология, как наиболее подходящая. Использование среды разработки Delphi(Delphi10.4) при программной реализации, позволило создать качественное современное программное обеспечение, обладающее высокой скоростью работы, удовлетворяющее всем уровням требований, предъявляемых к интерфейсу пользователя, обеспечивающее эффективность и гибкость работы. В процессе работы над проектом были изучены новые компоненты, благодаря которым программа стала выгладить лучше.

В заключении можно сказать, что данный программный продукт является законченной, полнофункциональной программой.

**Список использованных источников**

1. Валерий Рубанцев «Большой самоучитель по дэлфи» [Электронный ресурс] –– Режим доступа: https://drive.google.com –– Дата доступа: 20.06.2024
2. «PascalABC.NET: Введение в современное программирование» Осипов А. В. [Электронный ресурс] – Электронные данные. – Режим доступа: https://pascalabc.net. Дата доступа: 27.05.2024
3. Программирование на языке Delphi [Электронный ресурс] –– Режим доступа: [https://www.bsuir.by.](https://www.bsuir.by/) – Дата доступа: 27.05.2024

unit Unit1;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ExtCtrls, Vcl.ComCtrls, Vcl.StdCtrls, Unit2, unit4;

type

TForm1 = class(TForm)

ProgressBar1: TProgressBar;

Timer1: TTimer;

Image1: TImage;

Label1: TLabel;

procedure Timer1Timer(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

const

nazv='Развиваем внимательность';

var

Form1: TForm1;

implementation

{$R \*.dfm}

// создание файлов и значений по умолчанию

procedure TForm1.FormCreate(Sender: TObject);

begin

Createmaxlevel;

CreateLeaderBoard;

form1.Caption:=nazv;

end;

// реализация загрузки и открытие главной формы по её окончанию

procedure TForm1.Timer1Timer(Sender: TObject);

begin

if ProgressBar1.Position < ProgressBar1.Max then

ProgressBar1.Position:=ProgressBar1.Position+7

else begin Timer1.Enabled:=False;

form2.show;

form1.hide;

end;

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.ExtCtrls,

Vcl.Buttons, Vcl.MPlayer, Vcl.Menus, unit5, Vcl.Imaging.pngimage, ShellAPI;

type

TForm2 = class(TForm)

ComboBox1: TComboBox;

Image1: TImage;

Label1: TLabel;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

Image2: TImage;

Label2: TLabel;

Image3: TImage;

Label3: TLabel;

Image4: TImage;

Label4: TLabel;

Image5: TImage;

Label5: TLabel;

ComboBox2: TComboBox;

MediaPlayer1: TMediaPlayer;

procedure ComboBox1Change(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure FormCreate(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure CreateParams(var params:TcreateParams); override;

procedure Image2Click(Sender: TObject);

procedure Label2Click(Sender: TObject);

procedure Image3Click(Sender: TObject);

procedure Label3Click(Sender: TObject);

procedure Image4Click(Sender: TObject);

procedure Label4Click(Sender: TObject);

procedure Image5Click(Sender: TObject);

procedure Label5Click(Sender: TObject);

procedure FormActivate(Sender: TObject);

procedure ComboBox2Change(Sender: TObject);

procedure N1Click(Sender: TObject);

procedure N4Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

var lvlnum: Integer;

end;

const

nazv='Развиваем внимательность';

var

Form2: TForm2;

backgroundPath:string;

acttimes:integer;

implementation

{$R \*.dfm}

uses unit3, unit1;

// открытие формы со статистикой

procedure TForm2.Image4Click(Sender: TObject);

begin

Form2.Visible:=False;

Form5.Visible:=True;

end;

// открытие формы со статистикой

procedure TForm2.Label4Click(Sender: TObject);

begin

Form2.Visible:=False;

Form5.Visible:=True;

end;

//ВЫБОР ФОНА

procedure TForm2.ComboBox1Change(Sender: TObject);

begin

backgroundPath := ExtractFilePath(ParamStr(0))+'files/f' + inttostr(ComboBox1.ItemIndex+1) + '.bmp';

image1.Picture.LoadFromFile(backgroundPath);

end;

// установка заднего фона и прочих значений по умолчанию

procedure TForm2.FormCreate(Sender: TObject);

begin

acttimes := 0;

backgroundPath := ExtractFilePath(ParamStr(0))+'files/f1.bmp';

image1.Picture.LoadFromFile(backgroundPath);

form2.Caption:=nazv;

end;

//показать ПРАВИЛА

procedure TForm2.Image2Click(Sender: TObject);

var

f: textfile;

tf: TStringList;

begin

tf := TStringList.Create;

tf.LoadFromFile(ExtractFilePath(ParamStr(0))+'files/rules.txt'); // указываете путь к файлу с текстом

MessageBox(0, PChar(tf.Text), 'Правила', MB\_OK); // выводим текст из TStringList

end;

//показать ПРАВИЛА

procedure TForm2.Label2Click(Sender: TObject);

var

f: textfile;

tf: TStringList;

begin

tf := TStringList.Create;

tf.LoadFromFile(ExtractFilePath(ParamStr(0))+'files/rules.txt'); // указываете путь к файлу с текстом

MessageBox(0, PChar(tf.Text), 'Правила', MB\_OK); // выводим текст из TStringList

end;

//показать ПРАВИЛА

procedure TForm2.N2Click(Sender: TObject);

var

f: textfile;

tf: TStringList;

begin

tf := TStringList.Create;

tf.LoadFromFile(ExtractFilePath(ParamStr(0))+'files/rules.txt'); // указываете путь к файлу с текстом

MessageBox(0, PChar(tf.Text), 'Правила', MB\_OK); // выводим текст из TStringList

end;

//ВЫБОР МУЗЫКИ

procedure TForm2.ComboBox2Change(Sender: TObject);

begin

case ComboBox2.ItemIndex of

1: begin

MediaPlayer1.FileName := 'files/dinamic1.mp3';

MediaPlayer1.Open;

MediaPlayer1.Play;

end;

0: begin

MediaPlayer1.FileName := 'files/dinamic2.mp3';

MediaPlayer1.Open;

MediaPlayer1.Play;

end;

2: begin

MediaPlayer1.FileName := 'files/calm1.mp3';

MediaPlayer1.Open;

MediaPlayer1.Play;

end;

4: begin

MediaPlayer1.FileName := 'files/calm2.mp3';

MediaPlayer1.Open;

MediaPlayer1.Play;

end;

3:begin

MediaPlayer1.FileName := 'files/zvuki-gitary.mp3';

MediaPlayer1.Open;

MediaPlayer1.Play;

end;

end;

end;

// создание иконки приложения в панели инструментов

procedure TForm2.CreateParams(var params:TcreateParams);

begin

inherited CreateParams(params);

Params.ExStyle := params.ExStyle or WS\_EX\_APPWINDOW;

params.WndParent := GetDesktopWindow;

end;

//ВЫХОД

procedure TForm2.N3Click(Sender: TObject);

begin

form1.Close;

end;

// вывод справки

procedure TForm2.N4Click(Sender: TObject);

begin

ShellExecute(0, PChar('Open'),PChar(ExtractFilePath(ParamStr(0))+'files/справка.chm'),nil,nil,SW\_SHOW);

end;

//ВЫХОД

procedure TForm2.Label5Click(Sender: TObject);

begin

form1.Close;

end;

//ВЫХОД

procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);

begin

form1.Close;

end;

//ВЫХОД

procedure TForm2.Image5Click(Sender: TObject);

begin

form1.Close;

end;

//АКТИВАЦИЯ

procedure TForm2.FormActivate(Sender: TObject);

begin

if acttimes <=0 then begin

acttimes := acttimes + 1;

MediaPlayer1.FileName := 'files/dinamic2.mp3';

MediaPlayer1.Open;

MediaPlayer1.Play;

end;

end;

//ИГРАТЬ

procedure TForm2.Image3Click(Sender: TObject);

begin

form2.Visible := false;

form3.Visible:=true;

end;

//ИГРАТЬ

procedure TForm2.Label3Click(Sender: TObject);

begin

form2.Visible := false;

form3.Visible:=true;

end;

//ИГРАТЬ

procedure TForm2.N1Click(Sender: TObject);

begin

form2.Visible := false;

form3.Visible:=true;

end;

end.

unit Unit3;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ExtCtrls, Vcl.StdCtrls, shellapi,

Vcl.Buttons, Vcl.Grids, Vcl.Menus, Vcl.Imaging.pngimage, DateUtils, unit4;

type

TForm3 = class(TForm)

Image1: TImage;

lblTime: TLabel;

StringGrid1: TStringGrid;

Timer1: TTimer;

ccGreen: TCheckBox;

ComboBox1: TComboBox;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

Image2: TImage;

Label1: TLabel;

Image3: TImage;

Label2: TLabel;

Image4: TImage;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

procedure StringGrid1DrawCell(Sender: TObject; ACol, ARow: Integer;

Rect: TRect; State: TGridDrawState);

procedure StringGrid1MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure FormDestroy(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

procedure FormActivate(Sender: TObject);

procedure ComboBox1Change(Sender: TObject);

procedure N1Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure Image2Click(Sender: TObject);

procedure Label1Click(Sender: TObject);

procedure Image3Click(Sender: TObject);

procedure Label2Click(Sender: TObject);

procedure Image4Click(Sender: TObject);

procedure Label3Click(Sender: TObject);

procedure StartTimer();

procedure SetDesItems();

procedure CreateParams(var params:TcreateParams); override;

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure ccGreenClick(Sender: TObject);

procedure RepaintGreen();

procedure N4Click(Sender: TObject);

private

{ Private declarations }

Bitmap: TBitmap;

procedure Prepare(n: integer);

procedure SetNumbers;

public

{ Public declarations }

end;

const

nazv='Развиваем внимательность';

max\_n=9; //макс. размер поля

type TStatusCell=(scWhite, scGreen);//статус клетки - не выбрана/выбрана

type TGameStatus=(STOP, PLAY);//статус игры - остановлена/запущена

type TCell= Record

num: integer;// - число в ячейке

StatusCell: TStatusCell

end;

var

Form3: TForm3;

//всего чисел на поле:

AllNumbers: integer;

//выбрано чисел на поле:

ReadyNumbers: integer;

//массив, в котором хранится информация о поле:

masPole: array[0..max\_n, 0..max\_n] of TCell;

//

time,time0, stopTime, starttime: TDateTime;

lastTimeLikeStr : string;

//переключатель цвета чисел:

switch: integer= 1;

//статус игры:

GameStatus: TGameStatus= STOP;

lvlnum:integer;

implementation

{$R \*.dfm}

uses unit2, unit1;

//ОТМЕЧАТЬ КЛЕТКИ ЗЕЛЁНЫМ ЦВЕТОМ

procedure TForm3.RepaintGreen();

begin

if ccGreen.Checked then begin

StringGrid1.Repaint;

StringGrid1.Invalidate;

label4.Caption := inttostr(ReadyNumbers+1);

label4.Visible := true;

label5.Visible := true;

end

else

begin

label4.Visible := false;

label5.Visible := false;

end;

end;

procedure TForm3.ccGreenClick(Sender: TObject);

begin

RepaintGreen();

end; //ПОДСВЕЧИВАНИЕ

//ВЫБОР УРОВНЯ

procedure TForm3.ComboBox1Change(Sender: TObject);

begin

label4.Caption := '1';

lvlnum:=(ComboBox1.ItemIndex+5);

if LoadMaxLevel >= lvlnum then

Prepare(lvlnum);

GameStatus:= PLAY;

RepaintGreen();

end; //ВЫБОР УРОВНЯ

//Установить доступные и недоступные уровни в ComboBox

procedure TForm3.SetDesItems();

begin

for var i := 0 to ComboBox1.GetCount-1 do begin

if i+5 > LoadMaxLevel then

ComboBox1.Items[i] := ComboBox1.Items[i][1] + ' недоступен'

else

ComboBox1.Items[i] := ComboBox1.Items[i][1] + 'x' + ComboBox1.Items[i][1];

end;

end;

//АКТИВАЦИЯ ФОРМЫ

procedure TForm3.FormActivate(Sender: TObject);

begin

lvlnum:=(ComboBox1.ItemIndex+6);

RepaintGreen;

SetDesItems();

GameStatus := Play;

label4.Visible := False;

label4.Caption := '1';

time0:=Now;

time := Now - Now;

image1.Picture.LoadFromFile(unit2.backgroundPath);

form3.Caption:=nazv;

StartTimer;

Prepare(5);

end;

//ЗАКРЫТИЕ

procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction);

begin

form2.Show;

end;

//Создание иконки приложения в панели инструментов

procedure TForm3.CreateParams(var params:TcreateParams);

begin

inherited CreateParams(params);

Params.ExStyle := params.ExStyle or WS\_EX\_APPWINDOW;

params.WndParent := GetDesktopWindow;

end;

//УНИЧТОЖИТЬ ФОРМУ

procedure TForm3.FormDestroy(Sender: TObject);

begin

Bitmap.Free;

Timer1.Enabled:= False;

end;

//ПЕРЕХОД НА ГЛАВНУЮ ФОРМУ

procedure TForm3.Image2Click(Sender: TObject);

begin

form2.Visible := true;

form3.visible:=false;

ccGreen.Checked := false;

end;

procedure TForm3.Label1Click(Sender: TObject);

begin

form2.Visible := true;

form3.visible:=false;

ccGreen.Checked := false;

end;

procedure TForm3.N1Click(Sender: TObject);

begin

form2.Visible := true;

form3.visible:=false;

ccGreen.Checked := false;

end;

//ОСТАНОВИТЬ

procedure TForm3.Image3Click(Sender: TObject);

begin

//выключить таймер:

stopTime := now;

Timer1.Enabled:= False;

GameStatus:= STOP;

end;

procedure TForm3.Label2Click(Sender: TObject);

begin

//выключить таймер:

stopTime := now;

Timer1.Enabled:= False;

GameStatus:= STOP;

end;

//ВЫХОД

procedure TForm3.N2Click(Sender: TObject);

begin

form1.close;

end;

//ПРАВИЛА

procedure TForm3.N3Click(Sender: TObject);

var

f: textfile;

tf: TStringList;

begin

tf := TStringList.Create;

tf.LoadFromFile(ExtractFilePath(ParamStr(0))+'files/rules.txt'); // указываете путь к файлу с текстом

MessageBox(0, PChar(tf.Text), 'Правила', MB\_OK); // выводим текст из TStringList

end;

// вызов справки

procedure TForm3.N4Click(Sender: TObject);

begin

ShellExecute(0, PChar('Open'),PChar(ExtractFilePath(ParamStr(0))+ 'files/справка.chm'),nil,nil,SW\_SHOW);

end;

//ПРОДОЛЖИТЬ

procedure TForm3.Image4Click(Sender: TObject);

begin

if GameStatus= PLAY then exit;

GameStatus:= PLAY;

//включить таймер:

Timer1.Enabled:= True;

//засечь время начала тестирования:

time0 := time0 + now - stoptime - 1;

StartTimer();

end;

procedure TForm3.Label3Click(Sender: TObject);

begin

if GameStatus= PLAY then exit;

GameStatus:= PLAY;

//включить таймер:

Timer1.Enabled:= True;

//засечь время начала тестирования:

time0 := time0 + now - stoptime - 1;

StartTimer();

end;

//ТАЙМЕР

procedure TForm3.StartTimer();

begin

if GameStatus= PLAY then exit;

GameStatus:= PLAY;

//включить таймер:

Timer1.Enabled:= True;

end;

//ГОТОВИМСЯ К ИГРЕ

procedure TForm3.Prepare(n: integer);

var

w, h, lw: integer;

begin

//размер клетки в пикселях:

timer1.Enabled := true;

label3.Enabled := true;

image4.Enabled := true;

label2.Enabled := true;

image3.Enabled := true;

time0 := now;

time := now;

stopTime := now;

w:= StringGrid1.DefaultColWidth;

h:= StringGrid1.DefaultRowHeight;

//толщина линий:

lw:= StringGrid1.GridLineWidth;

//размеры игрового поля в клетках:

StringGrid1.ColCount:= n;

StringGrid1.RowCount:= StringGrid1.ColCount;

//размеры игрового поля в пикселях:

StringGrid1.Width:= 11 + (w + lw)\* n;

StringGrid1.Height:= 11 + (h + lw)\* n;

StringGrid1.Left:= (ClientWidth-StringGrid1.Width) div 2;

StringGrid1.Top:= (ClientHeight-StringGrid1.Height) div 2;

//всего чисел на поле:

AllNumbers:= n \* n;

//ни одно число не выбрано:

ReadyNumbers:= 0;

//расставить числа в сетке

SetNumbers;

caption:= nazv;

switch:= 1;

StringGrid1.Invalidate;

end;

//ОТРИСОВАТЬ КЛЕТКУ ИГРОВОГО ПОЛЯ

procedure TForm3.StringGrid1DrawCell(Sender: TObject; ACol, ARow: Integer;

Rect: TRect; State: TGridDrawState);

var

sNum: string;

n: integer;

begin

//цвет клетки: белый, если число ещё не выбрано,

//зелёный - если выбрано и разрешено отмечать клетки

if masPole[ACol, ARow].StatusCell= scWhite then

StringGrid1.Canvas.Brush.Color:= clWhite

else

if ccGreen.Checked then StringGrid1.Canvas.Brush.Color:= RGB(0,255,0);

StringGrid1.Canvas.FillRect(Rect);

//вывести число в клетке (ACol, ARow) соответствующим цветом:

n:= masPole[ACol, ARow].num;

StringGrid1.Canvas.Font.Size:= 21;

StringGrid1.Canvas.Font.Style:= StringGrid1.Font.Style+ [fsBold];

StringGrid1.Canvas.Font.Color:= clBlack;

sNum:= inttostr(n);

with Rect, StringGrid1.Canvas do

textrect(Rect, left+(right-left-textwidth(sNum)) div 2,

top+(bottom-top-textheight(sNum)) div 2, sNum);

end;

//ВЫБРАТЬ ЧИСЛО

procedure TForm3.StringGrid1MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

var

ACol,ARow: integer;

n: integer;

begin

if GameStatus<> PLAY then exit; //- уже играем!

//координаты мыши:

StringGrid1.MouseToCell(x,y,ACol,ARow);

//число в клетке:

n:= masPole[ACol,ARow].num;

if (n <> succ(ReadyNumbers)) then //- не следующее число!

MessageBeep(MB\_ICONASTERISK)

else begin

masPole[ACol,ARow].StatusCell:= scGreen;

//нашли ещё одно число:

inc(ReadyNumbers);

if ReadyNumbers < allnumbers then

label4.Caption := inttostr(ReadyNumbers+1)

end;

end;

//РАБОТА С ТАЙМЕРОМ

procedure TForm3.Timer1Timer(Sender: TObject);

var

s,p: string;

min, sec: string;

i, j: integer;

flg: boolean;

begin

if (time0 <> 0) then begin

//время игры:

time:=(now-time0);

sec:= inttostr(SecondOf(time));

min:= inttostr(MinuteOf(time));

if MinuteOf(time) > 59 then begin

Timer1.Enabled:= False;

s:= 'Ваше время закончилось!'#13#10'Попробуйте снова';

GameStatus:= STOP;

Application.MessageBox(pChar(s), nazv, IDOK);

exit

end;

if strtoint(sec) < 10 then sec:= '0' + sec;

if strtoint(min) < 10 then min:= '0' + min;

//не решена ли задача?

if ReadyNumbers >= allNumbers then begin

image4.Enabled := false;

label3.Enabled := false;

label2.Enabled := false;

image3.Enabled := false;

sec := inttostr(strtoint(sec) - 1);

GameStatus:= STOP;

Timer1.Enabled:= False;

SaveLevel(lvlnum+1);

SaveTimeInLeaderBoard(lastTimeLikeStr, lvlnum);

SetDesItems();

p:= 'Поздравляем!'#13#10'Вы справились с заданием! '#13#10''+'Ваше время '+ lasttimelikestr+''#13#10'Ваш лучший результат: ' + GetBestResult(lvlnum);

MessageBox(Application.Handle, pChar(p), nazv, MB\_OK);

end;

end;

lblTime.caption:= 'Ваше время ' + min + ':' + sec;

lasttimelikestr := min + ':' + sec;

end;

//РАССТАВИТЬ ЧИСЛА

procedure TForm3.SetNumbers;

var

i, j, n, m: integer;

iMas: array[1..max\_n \* max\_n] of boolean;

flg: boolean;

begin

//"перемешать" числа:

Randomize;

//размер поля:

m:= StringGrid1.ColCount;

//ни одно число не поставлено в сетку:

for i:= 1 to max\_n \* max\_n do iMas[i]:= false;

//расставляем числа:

for j:= 0 to m-1 do begin

for i:= 0 to m-1 do begin

flg:= false; //- число не выбрано

repeat

//случайное число:

n:= Random(AllNumbers)+ 1;

if iMas[n]= false then begin //- такого числа не было

iMas[n]:= True; //- число вышло

masPole[i, j].num:= n; //- занести его в массив

flg:= True; //- число выбрано

//клеткa не выбрана:

masPole[i, j].StatusCell:= scWhite;

end;

until flg= True; //повторять, пока не будет выбрано число

end;

end;

end;//SetNumbers

end.

unit Unit4;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ExtCtrls, Vcl.StdCtrls,

Vcl.Buttons, Vcl.Grids, Vcl.Menus, Vcl.Imaging.pngimage, DateUtils;

type arr = array[1..20] of string;

procedure SaveTimeInLeaderBoard(time:string; num:integer);

function LoadLeaderBoard(num:integer): arr;

procedure SaveLevel(num:integer);

function LoadMaxLevel():integer;

procedure Createmaxlevel();

procedure CreateLeaderBoard();

function SortStrings(const Strings: arr): TStringList;

function GetBestResult(num:integer):string;

implementation

// получить лучший результат пользователя

function GetBestResult(num:integer):string;

begin

var lb := LoadLeaderBoard(num);

var sorted := SortStrings(lb);

for var i := 0 to sorted.Count-1 do begin

if sorted[i] <> '00:00' then begin

Result := sorted[i];

exit

end;

end;

end;

// сохраняет переданное время в лидерборд

procedure SaveTimeInLeaderBoard(time:string; num:integer);

var F: TextFile;

data: arr;

begin

AssignFile(F, ExtractFilePath(ParamStr(0))+'files\'+inttostr(num)+'leaderboard.txt');

data := LoadLeaderBoard(num);

rewrite(F);

writeln(F, time);

for var i := 1 to 19 do

writeln(f, data[i]);

Close(F);

end;

// возвращает все строки из лидерборда

function LoadLeaderBoard(num:integer): arr;

var F: TextFile;

s: string;

data: arr;

begin

AssignFile(F, ExtractFilePath(ParamStr(0))+'files\'+inttostr(num)+'leaderboard.txt');

Reset(F);

for var i := 1 to 20 do begin

readln(F, s);

data[i] := s;

end;

Close(F);

Result := data;

end;

// сортирует строки лидерборда (и не только) по возрастанию

function SortStrings(const Strings: arr): TStringList;

var

StringList: TStringList;

i: Integer;

begin

StringList := TStringList.Create;

StringList.AddStrings(Strings);

StringList.Sort;

Result := StringList;

end;

// сохраняет переданный уровень, если он выше чем текущий записанный

procedure SaveLevel(num:integer);

var F: TextFile;

begin

AssignFile(F, ExtractFilePath(ParamStr(0))+'files\max\_level.txt');

var slikei := LoadMaxLevel;

if slikei < num then begin

Rewrite(f);

writeln(f, inttostr(num));

Close(f);

end;

end;

// возвращает максимальный уровень

function LoadMaxLevel():integer;

var F: TextFile;

begin

AssignFile(F, ExtractFilePath(ParamStr(0))+'files\max\_level.txt');

Reset(f);

var s:='';

readln(f, s);

Close(F);

Result := strtoint(s);

end;

// создаёт файл для хранения уровня

procedure Createmaxlevel();

var F: TextFile;

begin

if not FileExists(ExtractFilePath(ParamStr(0))+'files\max\_level.txt') then begin

AssignFile(F, ExtractFilePath(ParamStr(0))+'files\max\_level.txt');

Rewrite(F);

writeln(F, '5');

Close(F);

end;

end;

// создаёт файлы для хранения данных лидерборда

procedure CreateLeaderBoard();

var F: TextFile;

var name:string;

begin

for var i:= 5 to 9 do begin

if not FileExists(ExtractFilePath(ParamStr(0))+'files\'+inttostr(i)+'leaderboard.txt') then begin

AssignFile(F, ExtractFilePath(ParamStr(0))+'files\'+inttostr(i)+'leaderboard.txt');

Rewrite(F);

for var j:=1 to 20 do

writeln(F, '00:00');

Close(F);

end;

end;

end;

end.

unit Unit5;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, unit4, Vcl.ExtCtrls,

Vcl.Imaging.pngimage;

type

TForm5 = class(TForm)

Memo1: TMemo;

Image1: TImage;

Image2: TImage;

Image3: TImage;

Image4: TImage;

Image5: TImage;

Image6: TImage;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Image7: TImage;

Label7: TLabel;

procedure LoadRes(num:integer);

procedure FormActivate(Sender: TObject);

procedure Image2Click(Sender: TObject);

procedure Label1Click(Sender: TObject);

procedure Image3Click(Sender: TObject);

procedure Label2Click(Sender: TObject);

procedure Image4Click(Sender: TObject);

procedure Label3Click(Sender: TObject);

procedure Image5Click(Sender: TObject);

procedure Label4Click(Sender: TObject);

procedure Image6Click(Sender: TObject);

procedure Label5Click(Sender: TObject);

procedure Image7Click(Sender: TObject);

procedure Label7Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form5: TForm5;

implementation

{$R \*.dfm}

uses unit2;

//ВЫВОД СТАТИСТИКИ

procedure TForm5.LoadRes(num:integer);

begin

memo1.Clear;

var leaderboard := LoadLeaderBoard(num);

var sorted := SortStrings(leaderboard);

for var i := 0 to sorted.Count-1 do

if sorted[i] <> '00:00' then

memo1.Lines.Add(sorted[i]);

if memo1.Lines.count = 0 then

memo1.Lines.Add('Пройдите уровень чтобы посмотреть статистику');

end;

//5X5

procedure TForm5.Image2Click(Sender: TObject);

begin

LoadRes(5);

end;

procedure TForm5.Label1Click(Sender: TObject);

begin

LoadRes(5);

end;

//6X6

procedure TForm5.Image3Click(Sender: TObject);

begin

LoadRes(6);

end;

procedure TForm5.Label2Click(Sender: TObject);

begin

LoadRes(6);

end;

//7X7

procedure TForm5.Image4Click(Sender: TObject);

begin

LoadRes(7);

end;

procedure TForm5.Label3Click(Sender: TObject);

begin

LoadRes(7);

end;

//8X8

procedure TForm5.Image5Click(Sender: TObject);

begin

LoadRes(8);

end;

procedure TForm5.Label4Click(Sender: TObject);

begin

LoadRes(8);

end;

//9X9

procedure TForm5.Image6Click(Sender: TObject);

begin

LoadRes(9);

end;

procedure TForm5.Label5Click(Sender: TObject);

begin

LoadRes(9);

end;

//АКТИВАЦИЯ

procedure TForm5.FormActivate(Sender: TObject);

begin

image1.Picture.LoadFromFile(unit2.backgroundPath);

end;

//НА ГЛАВНУЮ

procedure TForm5.Image7Click(Sender: TObject);

begin

form2.Visible := true;

form5.visible:=false;

end;

procedure TForm5.Label7Click(Sender: TObject);

begin

form2.Visible := true;

form5.visible:=false;

end;

end.