Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

Игровое средство «Memory»

БГУИР КП 1-40 01 01 115 ПЗ

Студент: гр. 851006 Коловайтис Н.А.

Руководитель: Данилова Г.В.

Минск 2019

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc482467313)

[1 АНАЛИЗ АНАЛОГОВ ПРОГРАММНОГО СРЕДСТВА 4](#_Toc482467314)

[1.1 «Fun Easy Learn» 4](#_Toc482467315)

[1.2 «Дуолинго» 5](#_Toc482467316)

[1.3 «Lingualeo» 5](#_Toc482467317)

[2 ЦЕЛИ И ЗАДАЧИ 6](#_Toc482467318)

[2.1 Цель курсовой работы 6](#_Toc482467319)

[2.2 Задачи курсовой работы 6](#_Toc482467320)

[2.3 Необходимые ресурсы для курсовой работы 7](#_Toc482467321)

[3 РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА 8](#_Toc482467322)

[3.1 Структура программы 8](#_Toc482467323)

[3.2 Интерфейс программного средства 9](#_Toc482467324)

[3.3 Загрузка словаря 11](#_Toc482467325)

[3.4 Разработка игры «Словарь» 12](#_Toc482467326)

[3.5 Разработка игры «Easy-спринт» 13](#_Toc482467327)

[3.5.1 Генерирование рандомного расположения слов 14](#_Toc482467328)

[3.5.2 Создание таймера 15](#_Toc482467329)

[4 ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА 16](#_Toc482467330)

[5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 17](#_Toc482467331)

[5.1 Системные требования 17](#_Toc482467332)

[5.2 Установка программы 17](#_Toc482467333)

[5.3 Инструкция по использовани 17](#_Toc482467334)

[5.3.1 Запуск программы 17](#_Toc482467335)

[5.3.2 Начало работы 17](#_Toc482467336)

[5.3.3 Игра «Словарь» 20](#_Toc482467337)

[5.3.4 Игры «Слово-перевод» и «Перевод-слово» 21](#_Toc482467338)

[5.3.5 Игра «Прослушать и выбрать» 21](#_Toc482467339)

[5.3.6 Игра «Конструктор слов» 22](#_Toc482467340)

[5.3.7 Игра «Easy-спринт» 22](#_Toc482467341)

[ЗАКЛЮЧЕНИЕ 23](#_Toc482467342)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc482467343)

[ПРИЛОЖЕНИЕ А Исходный код программы 25](#_Toc482467344)

[ПРИЛОЖЕНИЕ Б. Блок-схема процедуры Fall 25](#_Toc482467345)

# ВВЕДЕНИЕ

Кратковременная память позволяет нам без подготовки отвечать на сложные вопросы, импровизировать и быстро оценивать ситуацию. Раньше считалось, что у каждого человека заложена определённая ёмкость кратковременной памяти, однако последние исследования в области когнитивных наук и психологии доказали, что мы можем её натренировать.

Когда нам нужно без подготовки ответить на какой-то вопрос, кратковременная память позволяет нам одновременно говорить и мысленно составлять план следующих высказываний. Она также используется, когда мы читаем. Чтобы понимать суть материала, нам необходимо помнить то, что мы уже прочли, и связывать это с тем, что мы прочитаем дальше. Кроме того, кратковременная память помогает нам игнорировать лишнюю информацию, в том числе всё, что нас отвлекает. А ведь способность фокусироваться на задаче очень важна, особенно в наше время.

Все мы сейчас страдаем от переизбытка информации, социальные сети и различные оповещения днём и ночью требуют нашего внимания. Мозгу приходится прилагать немалые усилия, чтобы определить, какую информацию запомнить, а какую нет. Всё это вызывает тревогу и стресс и ещё больше сокращает возможности нашей памяти.

Есть два выхода.

Во-первых, уменьшить число отвлекающих факторов и тем самым понизить уровень стресса. Это положительно скажется не только на памяти, но и на всех жизненных сферах.

Но иногда это просто невыполнимо. Мы не можем предвидеть, когда начальник потребует от нас срочный отчёт или когда заболеет кто-то из близких. Да и, говоря по правде, обычно нам даже нравится информационное разнообразие, которое сейчас представлено онлайн.

Значит, остаётся второй вариант — развивать и укреплять свою кратковременную память. Именно поэтому, целью данной курсовой работы является разработка игрового средства «Memory», которое с легкостью поможет натренировать кратковременую память. В пояснительной записке рассмотрены такие разделы, как: разработка программы, тестирование программы, руководство по использованию.

# АНАЛИЗ АНАЛОГОВ ПРОГРАММНОГО СРЕДСТВА

В настоящее время существует огромное количество способов для тренировки кратковременной памяти. Одним из самых распространенных является карточная игра «Memory», известная также под названиями «Concentration», «Match», «Match», «Up», «Pelmanism», «Shinkei-suijaku», «Pexeso» или просто «Пары». Суть игры заключается в следующем. На стол рубашкой вверх выкладывается четное количество карт. При этом каждая карта имеет едиинственную пару среди остальных карт. Игрок выбирает две карты и переворачивает их. Если карты совпали, то они остаются лежать открытыми, иначе эти карты закрываются. Цель игры – открыть все карты. Уже существует множество компьютерных реализаций данной игры, немного отличающихся между собой правилами.

## «Memory Box»

«Memory Box» - одна из реализаций данной игры. Есть возможность выбрать тип значков на карточках. Также можно выбрать тип игры («Classic», «Search», «Triplet») и игровой режим («Casual», «Challange», «Blockade», «Bombs»). Кроме данных расширений, немного изменены правила. А именно, перед игрой пользователю показывают, где лежат все карточки, и дают несколько секунд, чтобы их запомнить. Есть возможность переключения вида поля с 2d на 3d. Из минусов можно отметить слишком частую рекламу и отсутствие справки, поясняющей отличие игровых режимов друг от друга.



Рисунок 1.1 – Интерфейс приложения «Memory Box»

* 1. **«Math&Match»**

В данной программе на скрытой стороне карточек находятся не картинки, а числа. Отличием от аналогов является дополнительный режим, при котором каждой карточке соответствует не точно такая же карточка, а карточка с математическим выражением, которое численно равно значению карточки. Т.е. карточке «4» будет соответствовать, например, карточка «2+2». Это дает возможность тренировать не только память, но и навык считать в уме.

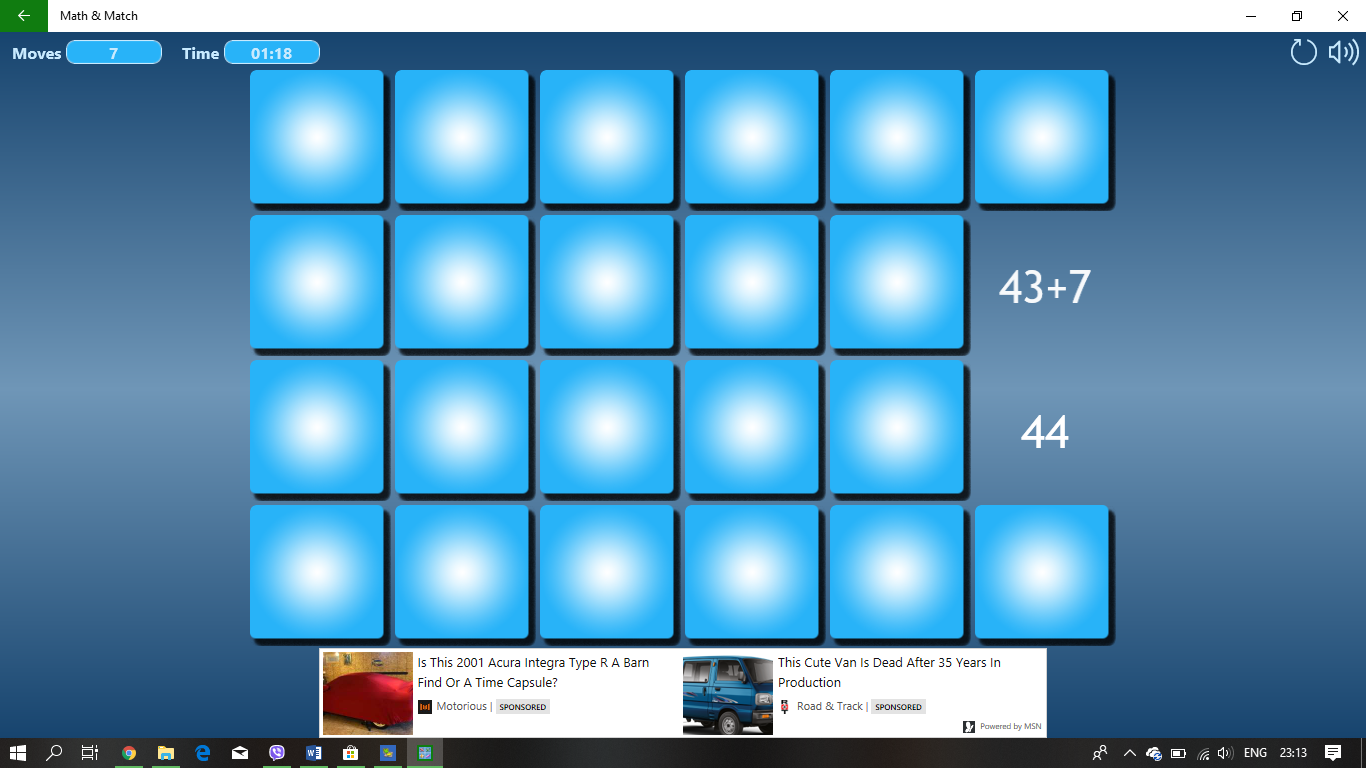


Рисунок 1.2 – Интерфейс сервиса «Math&Match»

* 1. **«Masha and The Bear Memory»**

Как можно догадаться из названия, игра оформлена в стиле мультфильма «Маша и медведь». В данной вариации карточной игры присутствует режим игры на двух игроков. Игрокам предлагается по очереди выполнять ход. Если игрок смог найти подходящую пару карточек, то он получает очко. Побеждает тот игрок, который по окончанию игры набрал больше очков. В программе очень много долгой анимации, что портит впечатление от игры.

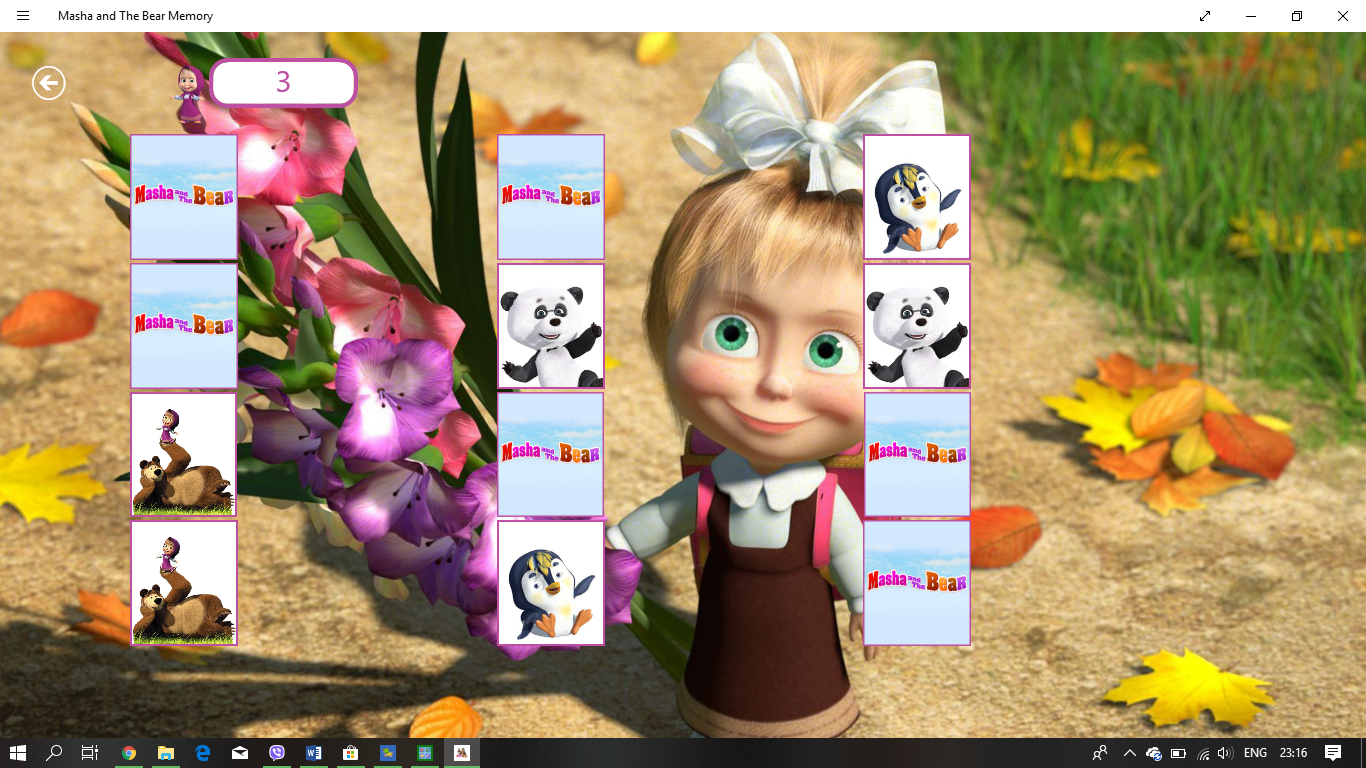


Рисунок 1.3 – Интерфейс приложения «Masha and The Bear Memory»

1. **ЦЕЛИ И ЗАДАЧИ**
   1. **Цель курсовой работы**

Целью данной курсовой работы является создание игрового обучающего приложения «Memory», направленного на развитие кратковременной памяти.

* 1. **Задачи курсовой работы**

В результате сравнения аналогов программного средства и анализа предметной области в данном курсовом проекте поставлены следующие задачи:

* разработать игровое обучающее приложение «Memory»;
* создать игровое поле с карточками;
* реализовать переворот карточки;
* разработать алгоритмы для проверки совпадения карт и определения конца игры;
* разработать алгоритмы для перемешивания карт;
* разработать возможность изменения размера игрового поля;
* предусмотреть возможность замены стандартных карт на другие;
* разработать режим игры для нескольких пользователей;
* реализовать таблицу рекордов;
* создать стартовое меню;
* создать стартовое меню;

Задача пользователя состоит в том, чтобы натренировать свою память запоминать значение и расположение объектов. У пользователя есть стимул научиться быстро и безошибочно угадывать карточки в одиночной игре, для того, чтобы в будущем побеждать товарищей в режиме мультиплеера.

* 1. **Необходимые ресурсы для курсовой работы**

Для разработки программного средства использован язык программирования Delphi. Среда разработки – RAD Studio 10. Использование данной среды разработки дает множество возможностей в создании оконных приложений Windows. С помощью данной среды можно очень легко и быстро разработать качественное оконное приложение, обеспечить удобные средства для ввода, редактирования, сохранения и отображения данных.

1. **РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА**
   1. **Структура программы**

В данном приложении используется 7 модулей:

* MainFrame используется для загрузки главного меню, а также окна с описанием правил игры;
* ChoiceTheThemeFrame предназначен для загрузки окна выбора тем;
* ChoiceTheGameFrame используется для загрузки окна выбора игр;
* DictionaryLevelFrame — модуль игры «Словарь»;
* MemoryLevelFrame — модуль игры «Easy-спринт»;
* ConstructionWordsLevelFrame — модуль игры «Конструктор слов»;
* TheWordTranslationLevelFrame — модуль, предназначенный для реализации 3 игр: «Слово-перевод», «Перевод-слово», «Прослушать и выбрать»

Листинг всех процедур и функций, используемых в модуле TheWordTranslationLevelFrame:

//*процедура, предназначенная для загрузки ресурсов в зависимости от выбранной игры*

* procedure InitializationTheWordTranslationLevel;

//*процедура, осуществляющая открытие формы*

* procedure FormShow(Sender: TObject);

//*процедура, активирующаяся при перемещении фокуса на форму*

* procedure FormActivate(Sender: TObject);

//*процедура, отвечающая за создание изображений*

* procedure CreateImage;

*//процедура, обрабатывающая перемещение курсора по фоновому изображению на форме*

* procedure MenuImageMouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

*//процедура, отвечающая за нажатие по форме*

* procedure MenuImageMouseDown(Sender: TObject; Button:TMouseButton;

Shift: TShiftState; X, Y: Integer);

*//процедура вывода текста на картинку*

* procedure OutputText(number: Byte; text: string);

*//процедура, предназначенная для вывода возможных вариантов слов*

* procedure OutputWords(answer, temp1, temp2, temp3: string);

*//процедура, отвечающая за генерирование нового слова*

* procedure TheGame;

*//процедура обработки клика по изображению*

* procedure ImageClick(Sender: TObject);

*//процедура, осуществляющая произношение слова*

* procedure ListenButtonClick(Sender: TObject);

*//процедура, отвечающая за проверку корректности овета*

* procedure CheckAnswer(case1: Byte; text: string);

*//процедура, выполняющая генерирование возможных ответов*

* procedure RandomizeImage(answer: Byte; var temp1, temp2, temp3: WordTranslationGame);

*//процедура, предназначенная для очистки списка*

* procedure ListFree;
  1. **Интерфейс программного средства**

Проанализировав палитру компонентов среды Delphi, был сделан вывод, что имеющиеся визуальные компоненты не подходят для организации работы приложения из-за отсутствия возможности детальной настройки и несоответствия стилю программы. Поэтому появилась необходимость разработать собственный способ визуального представления данных. Наиболее целесообразным стало использование TImage и методов TCanvas. Это позволило создать более эстетичный внешний вид игрового приложения.

Вышеописанный подход не лишен недостатков. Использование для ввода данных полей, которые отличаются от стандартных элементов управления, используемых в Windows, вносит дисбаланс во внешний вид программного средства. В связи с этим возникает идея спроектировать дизайн необходимых элементов управления таких, как кнопки, вручную с использованием графического редактора. Несмотря на высокую трудоемкость, этот способ позволяет избежать шаблонности дизайна, а также приобрести опыт разработки внешнего вида элементов управления с нуля.

Для создания собственных элементов управления используется графический редактор Adobe Photoshop CS6, предоставляющий широкие возможности для работы с графикой.

Любая кнопка имеет три состояния: она может быть неактивна, зажата, на нее может быть наведен указатель мыши. Реакция на изменение состояния требует перехвата следующих событий: наведение указателя мыши на кнопку, перемещение указателя за ее границу, нажатие и отпускание кнопки мыши. К сожалению, компонент TImage не обладает событиями OnMouseEnter и OnMouseLeave, которые необходимы для обеспечения визуального отклика на наведение указателя мыши на кнопку. Так, в модуле ConstructionWordsLevelFrame создана процедура, предназначенная для создания французского алфавита, который используется для выбора букв, необходимых для построения слова. Алфавит представляет собой набор компонентов типа TImage. Ниже приведен фрагмент кода, отвечающий за динамическое создание данного компонента.

with TImage.Create(Self) do

begin

// Расчет положения компонента

Left := (j \* widthPanel) + x;

Top := (i \* heigthPanel + y) + 15;

// Загрузка фонового изображения

Picture.LoadFromFile(IMAGE\_PANEL);

// Очистка фона у свойства Brush

Canvas.Brush.Style := bsClear;

Canvas.Font.Size := 15;

Name := str[temp];

// Вывод текста с буквой на TImage

Canvas.TextOut(12,5,AnsiUpperCase(name));

// Установка размера компонента

Width := widthPanel;

Height := heigthPanel;

// Обработка события при нажатии на изображение

OnMouseDown := Image1MouseDown;

// Обработка события при отпускании кнопки мыши

OnMouseUp := Image1MouseUp;

Tag := 1;

Parent := ConstructionWordsLevel;

end;

При обработке событий нажатия либо отпускания кнопки мыши определяется активный компонент, а затем загружается соответствующее изображение. Ниже приведен отрывок обработки события, отвечающего за нажатие на изображение. Если тип активного компонента является TImage, за исключением MenuImage, используемого для хранения фонового изображение, то выполняется следующий фрагмент кода:

if (Sender is TImage) and ((Sender as TImage) <> MenuImage) then

begin

// Вызов процедуры, отвечающей за загрузку изображения

SetState((Sender as TImage) ,IM\_DOWN);

(Sender as TImage).Canvas.Brush.Style := bsClear;

(Sender as TImage).Canvas.Font.Size := 15;

// Перезаписывание надписи на активной кнопке

(Sender as TImage).Canvas.TextOut(12,5,AnsiUpperCase

(Sender as TImage).name));

…

end;

end;

* 1. **Загрузка словаря**

В программе реализована возможность выбора одной из 6 тем, после чего загружается список слов в соответствии с выбранной темой. Из-за особенностей французского языка, который помимо основных букв латинского алфавита имеет специальные символы, ANSI-кодировка файла не подходит, в следствии чего необходимо использовать кодировку UNICODE. Чтение текстового файла осуществляется при помощи процедуры, код которой приведен ниже. Вначале информация считывается в строку в формате Unicode, которая затем конвертируется в ANSI-строку.

procedure LoadFromFileUnicode(fileName: string);

var

f: File;

WStrBuff : WideString;

strResult: string;

CountWideChar : Integer;

begin

AssignFile(f, fileName);

// Открытие файл в режиме "только чтение". Размер

считываемых блоков равен 2 байта

Reset(f, SizeOf(WideChar));

// Расчет размера файла, исчисляемый в количестве целых

двубайтных блоков

CountWideChar := FileSize(f);

// Установка длины WideChar строки для того, чтобы в неё

поместились данные из файла

SetLength(WStrBuff, CountWideChar);

// Помещение данных из файла в WideChar строку

BlockRead(f, PWideChar(WStrBuff)^, CountWideChar);

// Преобразование WideChar строки в ANSI строку

strResult := WideCharLenToString(PWideChar(WStrBuff), CountWideChar);

//Освобождение памяти, выделенной для WideChar строки

Finalize(WStrBuff);

CloseFile(f);

// Вызов процедуры, разреляющей строку на слова

DivisionOfWords(strResult);

end;

* 1. **Разработка игры «Словарь»**

Для хранения слов используется двунаправленный кольцевой список. Двунаправленные списки, в отличие от однонаправленных, позволяют от каждого звена двигаться по списку в любом направлении. Каждое звено такой структуры содержит два поля ссылочного типа. Значением одного поля является ссылка на последующее звено списка. Значением другого — ссылка на предыдущее звено списка. Информационным элементом списка является структура данных Запись, которая имеет поля, отвечающие за хранение изображения, слова на русском и французском языках, аудиозаписи. Ниже приведено описание данной структуры:

type

DictionaryGame = ^TDictionaryGame;

TDictionaryGame = record

word: record

imgWord: TPngImage;

nameRusWord, nameFrWord: string;

audio: string;

end;

next, prev: DictionaryGame;

end;

Для отображения словаря используются динамически создаваемые компоненты типа TImage, на которых отображается слово и его перевод, а также динамически созданные кнопки типа TBitBtn для переключения слов «Вперед», «Назад» и для воспроизведения слова. Преимуществом данного типа кнопок является возможность установить на ее поверхности изображение, что позволяет создать более привлекательный интерфейс. Отрывок кода, отображающий создание кнопки «Вперед», приведен ниже:

NextButton := TBitBtn.Create(Self);

with NextButton do

begin

Parent := Self;

// Установка количества изображений

NumGlyphs := 3;

// Загрузка растрового рисунка кнопки

Glyph.LoadFromFile('images\Dictionary\Next.bmp');

// Задание положения кнопки

Top := 205;

Left := 504;

// Установка размеров кнопки

Height := 30;

Width := 30;

// Описание события клика на кнопку

OnClick := NextButtonClick;

end;

Свойство **Glyph** типа TBitmap определяет растровый рисунок кнопки. По умолчанию свойство Glyph имеет значение None, т. е. кнопка не содержит рисунок. Рисунок может содержать до трех отдельных изображений (глифов). Какое именно изображение выводится на кнопке, зависит от ее текущего состояния:

* первое изображение отображается, если кнопка не нажата (по умолчанию);
* второе изображение отображается, если кнопка неактивна и не может быть выбрана;
* третье изображение отображается, когда кнопка нажата (выполнен щелчок).



Рисунок 3.1 – Растровый рисунок кнопки «Вперед»

Для проигрывания аудиозаписей подключается стандартный модуль MMSystem, а для загрузки изображений в формате PNG — Imaging.pngimage.

* 1. **Разработка игры «Easy-спринт»**

Игра представляет собой поле из 12 картинок, которые находятся в перевернутом состоянии. При запуске игры первоначально программно генерируются компоненты TImage, на которые загружаются изображения перевернутых картинок.



Рисунок 3.2 – Изображение перевернутой картинки

Затем за каждым компонентом закрепляется текст: слово на французском языке либо его перевод на русский. Игра состоит в том, чтобы отыскать правильное соответствие между словом и переводом.

* + 1. **Генерирование рандомного расположения слов**

Для того, чтобы слова каждый раз располагались в различном порядке, необходимо использовать функцию Random. Первоначально генерируется номер слова и его перевода, которые записаны в массиве, служащем для хранения всех слов из темы, выбранной пользователем. Затем эти слова в рандомном порядке записываются пустой в массив, строкового типа, из которого они в дальнейшем будут помещаться на картинку. Нижеприведенный отрывок кода описывает данную процедуру.

count := COUNT\_WORD;

i := 1;

repeat

// Выбор слова из общего списка

repeat

check := True;

Randomize;

element := random (COUNT\_WORD) + 1;

// Проверка для избежания повторения слов

for j := 1 to 6 do

if (number[j] = element) then

check := False;

until check;

number[i] := element;

// Рандомное определение положения слова на поле

repeat

key := random (COUNT\_WORD) + 1;

until TCheckImage[key] = '';

// Запись в массив под выбранным номером

TCheckImage[key] :=formatfloat('00',i)+rusWord[element];

repeat

key := random (COUNT\_WORD) + 1;

until TCheckImage[key] = '';

TCheckImage[key] :=formatfloat('00',i)+frWord[element];

Dec(count,2);

Inc(i);

// Выполнение алгоритма до того момента, пока не будут

// расставлены все слова

until count = 0;

* + 1. Создание таймера

Для отслеживания времени прохождения игры реализована процедура создания таймера. Время отображается при помощи компонента StatusBar в формате ЧЧ:ММ:СС. Ниже приведена процедура подсчета времени.

procedure TMemoryLevel.Stopwatch(Secondes: Cardinal);

const

TEXT = 'Время выполнения: %0.2d:%0.2d:%0.2d';

var

h, m, s: Integer;

begin

h := Secondes div 3600;

m := (Secondes div 60) mod 60;

s := Secondes mod 60;

TimeStatusBar.SimpleText := Format(TEXT, [h, m, s]);

end;

Работа таймера завершается, когда пользователь правильно соединит все пары, после чего производится удаление всех ранее созданных картинок и возращение к окну выбора следующей игры.

На данном этапе проектирование и разработка основных функций игрового приложения закончены. Все основные функции реализованы.

1. **ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА**

Большинство ошибок, возникших в результате написания программного средства, были связаны с недочетами проектирования ПО и исправлены на стадии разработки. Однако в процессе тестирования первоначальной версии продукта был выявлен ряд случаев некорректной работы программы.

Одной из ошибок было неправильное поведение программы в игре «Easy-спринт» при многократном клике на одну и ту же картинку. Корректность соответствия слова и перевода определяется по имени компонента TImage, которое присваивается при его создании. Первые два символа в имени – это номер, который у верной пары одинаков. Могла возникнуть ситуация, когда на одной из картинок появлялся текст и не исчезал при некорректном выборе перевода из-за того, что картинка считала саму себя верной парой. При этом количество правильных ответов увеличивалось, что приводило к досрочному завершению игры. Проблема была решена отключением события OnClick при клике на первую картинку и обратном включении при выборе возможной пары.

Также в модулях, предназначенных для выбора темы и игры была выявлена следующая ошибка. При обработке события, отвечающего за движение мыши, исходя из координат курсора, в переменную целочисленного типа записывается номер, по которому, при обработке события, отвечающего за нажатие клавиши мыши, определяется, какая именно выбрана тема либо игра. Однако при клике на свободном участке экрана все же производилось открытие окна с темой либо игрой, фокус на которой был последним. Проблема была решена добавлением логической переменной. Только при наведении на область, содержащую картинку переменная принимала истинное значение. Теперь событие MoseDown срабатывало только при точном нажатии.

Остальные случаи возможного некорректного поведения программы были обнаружены на стадии проектирования и исправлены на стадии разработки.

1. **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

В данном разделе приводится вся необходимая информация, которая поможет пользователю в работе с приложением «EasyLingvo».

* 1. **Системные требования**

Для корректной работы приложения необходимо иметь операционную систему Windows версии XP или выше.

* 1. **Установка программы**

Для установки приложения следует запустить файл «Setup\_EasyLingvo.exe» и выполнить все инструкции программы.

* 1. **Инструкция по использовани**
     1. Запуск программы

Для запуска программы необходимо открыть исполняемый файл «EasyLingvo.exe».

Рисунок 5.1 – Иконка приложения «EasyLingvo»

* + 1. Начало работы

При запуске программы можно увидеть следующий интерфейс (рисунок 5.2). Перемещение указателя для выбора режима выполняется курсором мыши. Для выбора пункта меню необходимо нажать левую кнопку мыши. Указателем для выбора режима игры служит анимация изменения цвета кнопок с указаниями. Приведённые ниже пункты меню выполнют следующие функции:

* Играть – открывает окно с выбором темы.
* Об игре – запускает раздел описания данного игрового приложения, базовый инструктаж по игре, информацию об авторе.
* Выход – закрывает игру.



Рисунок 5.2 – Главное меню приложения

Запуск приложения осуществляется нажатием левой кнопки мыши по кнопке «Играть». В результате открывается окно с перечнем следующий тем: животные, цвета, эмоции, семья, еда, глаголы.

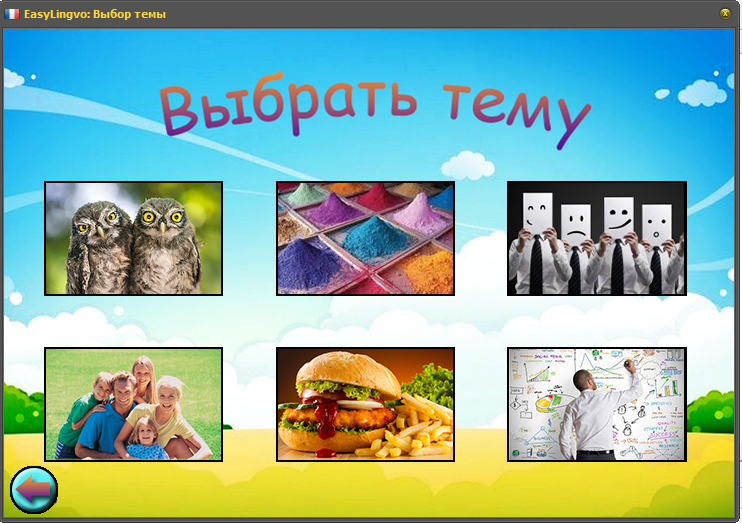


Рисунок 5.3 – Окно «Выбор темы»

Когда курсор мыши наводится на иконку темы, вокруг нее появляется красная рамка. При нажатии левой кнопки мыши, осуществляется переход к выбору заданий. Перед пользователем открывается окно «Выбрать игру», содержащее 6 способов запоминания слов: Словарь, Слово-перевод, Перевод-слово, Конструктор слов, Easy-спринт, Прослушать и выбрать.



Рисунок 5.4 – Окно «Выбор игры»

* + 1. Игра «Словарь»

При выборе игры «Словарь» появляется изображение и два поля ниже, содержащих слово, отображаемое на картинке, и его перевод. Картинки появляются в установленном порядке. При помощи кнопок «Вперед» и «Назад» пользователь имеет возможность просмотреть слова и их перевод. Появление картинки сопровождает аудио с правильным произношением слова.

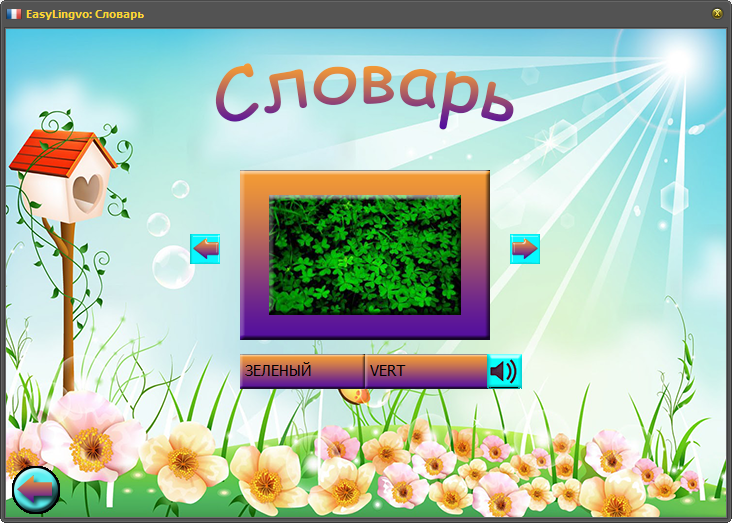


Рисунок 5.5 – Игра «Словарь»

* + 1. Игры «Слово-перевод» и «Перевод-слово»

Данные режимы игры имеют схожую структуру. При запуске любой из указанных игр вверху окна появляется слово на русском либо на французском языке. Справа предлагается 4 варианта ответа. При правильном соотношении слова к его переводу в нижней части окна отображается картинка, связанная с искомым словом, а правильный вариант ответа подсвечивается желтым цветом. Картинка сопровождается аудио, а затем следует переход к следующему слову. При неправильном выборе перевода слово подсвечивается красным цветом, а поле, где должна появиться картинка, остается пустым. Пользователь может повторить попытку.

Рисунок 5.6 – Игры «Слово-перевод» и «Перевод-слово»

* + 1. Игра «Прослушать и выбрать»

Структура этой игры напоминает игры, приведенные в пункте 5.3.4. Главным отличием является лишь то, что пользователь должен воспринять слово на слух. Вместо слова появляется кнопка, нажав на которую, пользователь может прослушать слово несколько раз. Принцип отображения корректности ответа остается тем же.

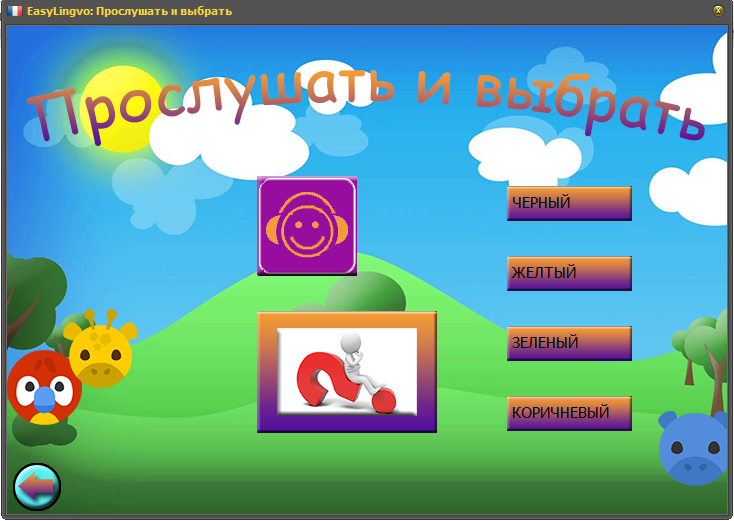


Рисунок 5.7 – Игра «Прослушать и выбрать»

* + 1. Игра «Конструктор слов»

Вверху экрана появляется слово на русском языке, а ниже приведен алфавит французского языка. Пользователю нужно записать перевод слова. При нажатии на букву, она появляется в области зеленого цвета, размер которой определяется длиной верного ответа. Если слово введено правильно, то появляется следующее слово, если совершена ошибка, то появляется изображение, предупреждающее о неверном переводе слова. Можно повторять попытку до того момента, пока слово не будет введено верно. Если пользователь не помнит слово, то можно его пропустить, нажав на синюю кнопку

* + 1. Игра «Easy-спринт»

Поле игры «Easy-спринт» состоит из 12 картинок. При нажатии на картинку появляется слово на русском либо французском языке. Цель игры состоит в том, чтобы отыскать пару: слово и его перевод. Если пара найдена верно, то текст на картинках остается, иначе неверно выбранные картинки переворачиваются. Время выполнения отображается внизу экрана.

**ЗАКЛЮЧЕНИЕ**

В современном развивающемся мире знание только одного языка является недостаточным. Процесс глобализации с каждым днём расширяет свои границы и охватывает большее количество стран. В связи с этим возникает необходимость знания иностранного языка. Люди придумывают свои собственные методики обучения, которые считают самыми действенными. Безусловно, лучшим способом изучения иностранного языка, является общение с носителями языка, полное погружение в языковую среду, однако очень немногие люди имеют такую возможность.

Целью курсовой работы было создание программы «EasyLingvo», которая обеспечит более быстрое изучение французского языка для начинающих. В данном приложении разработан красивый дизайн интерфейса, который создаёт рабочую атмосферу для занятия, присутствует обильное количество простых французских слов, которые разделены на темы для более легкого изучения. Также к каждому французскому и русскому словам добавлено произношение для того, чтобы пользователь мог воспринимать иностранный язык не только на письме, но и в речи, также это полезно пользователям, которые имеют слуховую память. Кроме этого, каждый обучающийся может повторять слова, которые он услышал. Для тех, кто имеет зрительную память, разработана специальная цветовая гамма, которая помогает более детально запомнить необходимую информацию. Для создания интерфейса были использованы навыки работы с приложением «Photoshop». В «EasyLingvo» присутствуют игры, которые во время серьёзного изучения французского языка помогают расслабиться.

В дальнейшем приложение будет разрабатываться и совершенствоваться, предлагая пользователю различные методики изучения французского языка. Также важным аспектом станет добавление других иностранных языков для того, чтобы приложение было полезно для большего количества людей. При дальнейшем совершенствовании будет добавлена функция «learn and teach», которая позволит пользователю вносить новые слова, заимствования, сленг. Количество игр будет расширяться.

Использование разработанного приложения полезно не только для тех, кто хочет серьёзно заниматься лингвистикой, но и для туристов. Отличительная черта «EasyLingvo» - доступность и понятность. В приложении может разобраться даже человек, не являющийся активным пользователем компьютера.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Delphi справочник (электронный ресурс). – Электронные данные. – Режим доступа: http://delphibasics.ru/Class.php;

2. Серебряная, Л.В. Марина, И.М. Структуры и алгоритмы обработки данных: учебно-метод. пособие для студ. спец. «Программное обеспечение информационных технологий» всех форм обуч. / Л. В. Серебряная, И. М. Марина. – Минск: БГУИР, 2012. – 49 с. : ил.;

3. Графические возможности Делфи (электронный ресурс). – Электронные данные. – Режим доступа: http://delphi7.gym5cheb.ru;

4. В.Фаронов Программирование в Delphi 6 (учебный курс). – М.: Издатель Молгачева С.В., 2001. – 672 с., ил.

**ПРИЛОЖЕНИЕ А Исходный код программы**

# ПРИЛОЖЕНИЕ Б. Блок-схема процедуры Fall