**Введение**

На курсовое проектирование была поставлена задача, разработать программу на тему: Разработка графического редактора для создания анимации.

Цель курсового проекта заключается в создании анимации

Создаваемая программа будет рассчитана на любого рода пользователей. Применить данную программу смогут не только люди, работающие в сфере создания анимации, но и просто люди, заинтересовавшиеся в данной теме.

Далее приведём 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 создаётся данный курсовой проект. Здесь также будут установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачей.

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 чётко 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 чётко описаны функции пользователя и их структура. Здесь можно будет найти таблицу, в которой будет представлена полная аннотация файлов используемых в данном проекте.

Четвёртый 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 “Литературе” будет приведён список используемых при разработке источников.

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

Схема работы системы будет представлена в графической части.

**Анализ задачи**

**1.1 Постановка задачи**

Наименование задачи: **“**Разработка графического редактора для создания анимации”

Цель разработки:создание простых анимации для дальнейшего их использования, рисование кадров в графическом редакторе.

Назначение: данный программный продукт разрабатывается

для людей любого возраста, желающим создать простейшую анимацию.

Периодичность использования: зависит от нужд потребителя, может

использоваться ежедневно.

Разрабатываемый программный продукт должен позволять выполнить

следующие действия:

- рисование кадров;

-создание анимации с помощью кадров.

-скачивать анимации в формате gif.

**1.2 Инструменты разработки**

Для разработки данного проекта будет выбрана среда Delphi (Rad Studio 11.2) так как это одна из мощнейших сред для разработки приложений такого уровня. Delphi — это среда объектно-ориентированного программирования, относящаяся к классу RAD – (Rapid Application Development – «Средство быстрой разработки приложений»), основанная на языке Object Pascal. Используется для разработки и поддержки программного обеспечения.

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

- Процессор: i3 9300h 2.40 GHz;

- ОЗУ: 8Gb;

- Память: HDD 1TB;

- ОС – Windows 10.

Также для разработки программы необходимы:

- Google-браузер – нужен для нахождения информации и картинок;

- Word 2016 – нужен для написания пояснительной записки;

- PowerPoint 2016 – нужен для создания отчётной презентации;

- Photoshop 2021 – создание изображений;

- Smart install maker – Инсталлятор для программы

**1.3 Требования к приложению**

На этапе исследования предметной области был установленный целый

ряд требований, которые предъявляют к разрабатываемой задаче.

При моделировании форм следует учесть такие моменты:

- интуитивно понятный интерфейс

- небольшое количество информации на каждой форме.

Так же следует обратить внимание на такие моменты:

- визуально понятный и приятный глазу интерфейс;

- небольшие кнопки.

Минимальные системные требования:

- 1.8 GHz процессор (или более быстрый);

- 1 GB RAM;

- от 40 MB доступного места на жестком диске.

Рекомендуемые:

- 2 GB RAM ;

- от 40 MB доступного места на жестком диске.

**Требования к интерфейсу:** при разработке приложения должны быть использованы преимущественного красно-жёлтые и черные цвета. Основные разделы приложения должны быть доступны с первой страницы. Грамотный пользовательский интерфейс. Следовательно, каждое окно должно иметь ясную визуальную иерархию своих элементов. Фрагменты текста должны располагаться на экране так, чтобы пользователя было просто и понятно принимать информацию.

Пользователь не должен испытывать какого-либо дискомфорта в плане восприятия информация, отображённой на экране. Объекты (рисунки и символы) не должны быть слишком мелкие. Отказы программы вследствие некорректных действий пользователя при взаимодействии с программой через графический интерфейс не должны влиять на конечный результат.

**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.

Основным средством хранения информации в приложении являются bmp - для хранения изображения, которое рисует пользователь, фоновых изображений и текстур для кнопок.

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

Система справочной информации представлена файлом справки, который содержит информацию о приложении и правилах его использования.

Основными функциями приложения являются:

- рисование на холсте (PaintBox);

- сохранение и открытие рисунков для дальнейшего использования;

- анимация нарисованных кадров;

**2.2 Процессы**

Согласно всем перечисленным требованиям и указаниям, которые были рассмотрены в разделе «Анализ задачи», было определено, чем конкретно должна заниматься разрабатываемая курсовая программа. Главной её задачей будет являться: рисование кадров для дальнейшей анимации. Весь данный процесс будет заключаться в следующем.

На форме с графическим редактором, пользователь рисует кадры, после чего сохраняет их в удобной для него папке. Далее, после того, как пользователь перейдёт на форму для анимации кадров, он сможет открыть файлы с нарисованными картинками. В поле с кадрами пользователь может копировать и вставлять кадры для создания нужной ему анимации. В поле под кнопкой «Анимировать» пользователь вводит время через которое будет сменяться каждый кадр. При нажатии на кнопку «Анимировать» будет воспроизводиться анимация.

Было бы важным на этапе проектирования рассмотреть ещё состав и внешний вид используемой формы. Хотя проектирование формы заранее не так принципиально. Форму в любой момент времени достаточно просто можно изменить и отредактировать.

Однако хорошо организованная форма улучшает внешний вид программы. Поэтому эти вопросы более подробно будут рассмотрены ниже.

**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 представлена система меню и организация навигации между окнами программы (Рисунок 1):



Рисунок 1 – Навигация между окнами программы

**Прототип** – это наглядная модель пользовательского интерфейса. В сущности это «черновик», созданный на основе представления разработчика о потребностях пользователя. **Итоговое отображение программы может отличаться от прототипа.** Ниже можно наблюдать прототипы:

1. Логотипа (Рисунок 2)

2. Заставки (Рисунок 3)

3. Окна с графическим редактором (Рисунок 4)

4. Окна с редактором для анимации (Рисунок 5)

5. Информационного окна (Рисунок 6)



Рисунок 2 – Прототип логотипа



Рисунок 3 – Прототип заставки (Splash screen)

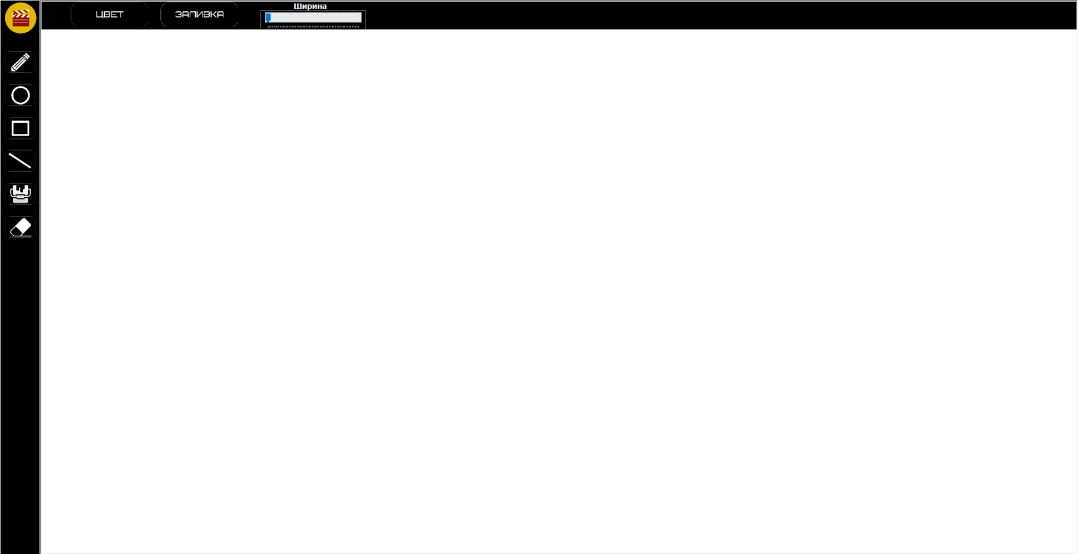
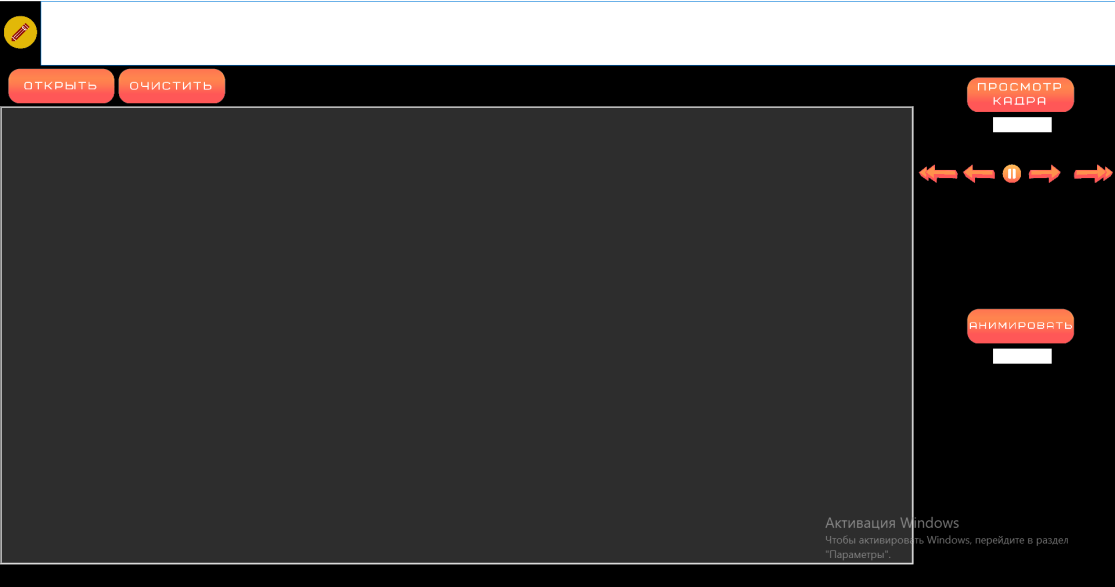
 Рисунок 4 – Окна с графическим редактором

Рисунок 5 – Окна с редактором для анимации

**

Рисунок 6 – Прототип информационного окна

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

**3. Реализация**

**3.1 Структура программы**

Данный курсовой проект содержит 4 модуля. Далее будет описано назначение каждого из них.

Модуль Unit2 является модулем, за которым закреплена главная форма с которых можно перейти на другие формы с загрузкой форм.

Модуль Unit1 является модулем, за которым закреплена форма с рисованием кадров, также с этой формы можно переходить на форму с анимацией кадров.

Модуль Unit3 является модулем, за которым закреплена форма с анимацией кадров, также с этой формы можно переходить на форму с рисованием кадров.

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

Описание разработанных процедур и функций приводится в таблице 1.

Таблица 1 - Процедуры и функции

|  |  |  |  |
| --- | --- | --- | --- |
| Имя процедуры (функции) | В каком модуле находится | За каким элементом управления закреплена | Назначение |
| 2 | 3 | 4 | 5 |
| procedure FormClose(Sender: TObject; var Action: TCloseAction); | Во всех | TForm | Закрытие программы |
| procedure FormShow(Sender: TObject); | Unit1  Unit2  Unit3 | TForm | Подгрузка всех нужных файлов/картинок для формы |
| procedure TForm2.Image1ClickTimer(Sender: TObject); | Unit2 | Timer1 | Отвечает за работу таймера, который загружает форму с рисованием |
| procedure TForm2.Image2ClickTimer(Sender: TObject); | Unit2 | Timer2 | Отвечает за работу таймера, который загружает форму с анимированием |
| procedure N3Click(Sender: TObject); | Unit3  Unit2 | TMainMenu | Открытие справки |
| procedure TForm1.Image13Click(Sender: TObject); | Unit1 | Image13 | Переход на форму с аниммированием |
| procedure TForm1.Image14Click(Sender: TObject); | Unit1 | Image14 | Выбор цвета для обода и карандаша |
| procedure TForm1.Image16Click(Sender: TObject); | Unit1 | Image16 | Выбор цвета для  заливки |
| procedure TForm1.Image1Click(Sender: TObject); | Unit1 | Image1 | Выбор  инструмента для рисования “карандаш” |
| procedure TForm1.Image2Click(Sender: TObject); | Unit1 | Image2 | Выбор  инструмента для рисования  “круг” |
| procedure TForm1.Image3Click(Sender: TObject); | Unit1 | Image3 | Выбор  инструмента для рисования  “прямоугольник” |
| procedure TForm1.Image4Click(Sender: TObject); | Unit1 | Image4 | Выбор  инструмента для рисования  “прямая линия ” |

Продолжение таблицы

|  |  |  |  |
| --- | --- | --- | --- |
| procedure TForm1.Image5Click(Sender: TObject); | Unit1 | Image5 | Выбор  инструмента для рисования  “заливка” |
| procedure TForm1.Image6Click(Sender: TObject); | Unit1 | Image6 | Выбор  инструмента для рисования  “ластик” |
| procedure TForm1.N2Click(Sender: TObject); | Unit1 | Opendialog | Открытие файла (рисунок) и загрузка его в paintbox |
| procedure TForm1.N3Click(Sender: TObject); | Unit1 | savedialog | Сохранение файла (рисунок) на компьютер |
| procedure TForm1.N7Click(Sender: TObject); | Unit1  Unit3 | Tmainmenu | Открытие файла справки |
| procedure TForm1.N8Click(Sender: TObject); | Unit1  Unit3 | Tmainmenu | Открытие информационного окна |
| procedure TForm1.paintbox1MouseDown(Sender: TObject; Button: TMouseButton;  Shift: TShiftState; X, Y: Integer); | Unit1 | PaintBox | Загрузка  Из buffer в Paintbox.  Проверка нажатых кнопок для рисования.  Загрузка  нарисованного объекта в переменную buffer. |
| procedure TForm1.paintbox1MouseMove(Sender: TObject; Shift: TShiftState; X,  Y: Integer); | Unit1 | PaintBox | Проверка опущена ли кнопка мыши, если да, то рисует объект |
| procedure TForm3.Image10Click(Sender: TObject); | Unit3 | Image10 | Отвечает за открытие первого кадра |
| procedure TForm3.Image11Click(Sender: TObject); | Unit3 | Image11 | Отвечает за открытие предыдущего кадра |
| procedure TForm3.Image12Click(Sender: TObject); | Unit3 | Image12 | Отвечает за открытие следующего кадра |
| procedure TForm3.Image13Click(Sender: TObject); | Unit3 | Image13 | Отвечает за открытие последнего кадра |
| procedure TForm3.Image18Click(Sender: TObject); | Unit3 | Image18 | Отвечает за работу таймера, который останавливает анимацию |
| procedure TForm3.Image19Click(Sender: TObject); | Unit3 | Image19 | Отвечает за работу таймера, который продолжает  анимацию |
| procedure TForm3.Image20Click(Sender: TObject); | Unit3 | Image20 | Отвечает за открытие формы с рисованием |

Продолжение таблицы

|  |  |  |  |
| --- | --- | --- | --- |
| procedure TForm3.Image21Click(Sender: TObject); | Unit3 | Image20 | Отвечает за открытие python файла, который сохраняет gif анимацию на компьютер |
| procedure TForm3.Image2Click(Sender: TObject); | Unit3 | Image2  Savedialog | Отвечает за очистку линий memo1 |
| procedure TForm3.Image4Click(Sender: TObject); | Unit3 | Image4  Opendialog | Отвечает за открытие файлов (кадры) и загрузка их пути в memo1 |
| procedure TForm3.Image6Click(Sender: TObject); | Unit3 | Image6  Edit1 | Отвечает за открытие кадра под номером в edit1 |
| procedure TForm3.Image8Click(Sender: TObject); | Unit3 | Image8  Edit2 | Отвечает за таймера, который отвечает за анимацию с интервалом в edit2 (по умолчанию 1000мс) |

**3.1.2 Описание использованных компонентов**

Описание использованных при разработке приложения компонентов приводиться в таблице 2

Таблица 2 – Использованные компоненты

|  |  |  |
| --- | --- | --- |
| Компонент | На какой форме расположено | Назначение |
| TMaimMenu | From1,Form3 | Используется для создания справки |
| TImage | На всех формах | Картинки на формах |
| Timer | Form3, Form2 | Анимирование, загрузочный экран |
| Gauge1 | Form2 | Для загрузочного экрана |
| Openidalog | Form1, Form3 | Для загрузки файлов с компьютера |
| Savedialog | Form1, Form3 | Для сохранения фалов на компьютер |

**3.2 Спецификация программы**

Точное название проекта и его состав приводится в таблице 3.

Таблица 3 – Спецификация программы

| Имя файла | Назначение |
| --- | --- |
| АниматоR34.exe | Исполняемый файл проекта, используется для запуска программы на выполнение. |
| АниматоR34.dproj | Файл проекта, связывает все файлы из которых состоит приложение. |
| Unit2.pas | Файл программного модуля главной формы |
| Unit1.pas | Файл программного модуля для рисования кадров |
| Unit3.pas | Файл программного модуля для анимации кадров |
| Unit4.pas | Файл программного модуля просмотра информационного окна |
| Unit2.dfm | Главная форма |
| Unit1.dfm | Форма с графическим редактором |
| Unit3.dfm | Форма с редактором анимации |
| Unit4.dfm | Форма с информационным окном |
| АниматоR34.chm | Справочная информация |
| animate.txt | Текстовый файл с информацией сохранения gif-анимации |
| Gifanimate.py | Файл python, который конвертирует картинки в gif и сохраняет его |
| Setup.exe | Установочный пакет |
| Icon instruments | Папка с картинками, задействованными программой для инструментов |
| Main icon | Папка с остальными картинками, задействованными программой |

**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 представлены в таблице 4

Таблица 4 - Отчет о результатах тестирования.

|  |  |  |  |
| --- | --- | --- | --- |
| Тест | Ожидаемый результат | Фактический результат | Результат тестирования |
| Проверка кнопки на форме меню «Графический редактор» | Переход на форму с графическим редактором | Переход на форму | Выполнено |
| Проверка кнопки на форме меню «Редактор анимации» | Переход на форму с графическим редактором | Переход на форму | Выполнено |
| Проверка кнопки «карандаш» на форме с графическим редактором | Рисование кривых линий | Кривые линии | Выполнено |
| Проверка кнопки «круг» на форме с графическим редактором | Рисование круга | Круг | Выполнено |
| Проверка кнопки «прямоугольник» на форме с графическим редактором | Рисование прямоугольника | Прямоугольник | Выполнено |
| Проверка кнопки «линия» на форме с графическим редактором | Рисование прямой линии | линия | Выполнено |
| Проверка кнопки «заливка» на форме с графическим редактором | Заливка закрытой плоскости выбранным цветом | Заливка закрытой плоскости | Выполнено |
| Проверка кнопки «ластик» на форме с графическим редактором | Рисование кривых линий белым цветом | Кривые линии белым цветом | Выполнено |
| Проверка кнопки «цвет» на форме с графическим редактором | Выбор цвета для рисования | Рисует выбранным цветом | Выполнено |
| Проверка кнопки «цвет заливки» на форме с графическим редактором | Выбор цвета заливки для рисования | Заливка выбранным цветом | Выполнено |
| Проверка TrackBar на форме с графическим редактором | Изменение толщины линий | Изменяется толщина линий | Выполнено |
| Проверка кнопки «Открыть»  На форме с графическим редактором | Открытие рисунка в программе | Открытие рисунка | Выполнено |
| Проверка кнопки переход с графического редактора на редактор анимации | Переход на форму с редактором анимации | Переход на форму | Выполнено |

Продолжение таблицы

|  |  |  |  |
| --- | --- | --- | --- |
| Проверка кнопки «сохранить» на форме с графическим редактором | Сохранение рисунка на компьютер | Сохранение рисунка | Выполнено |
| Проверка кнопки «открыть» на форме с редактором анимации | Открытие кадров в программе | Открытие кадров | Выполнено |
| Проверка кнопки «очистить» на форме с редактором анимации | Очистка линий в компоненте memo | Очистка линий | Выполнено |
| Проверка кнопки «просмотр кадра» на форме с редактором анимации | Открытие кадра, указанного в edit1 | Открытие кадра | Выполнено |
| Проверка кнопки «первый кадр» на форме с редактором анимации | Открытие первого кадра | Открытие кадра | Выполнено |
| Проверка кнопки «предыдущий кадр» на форме с редактором анимации | Открытие предыдущего кадра | Открытие кадра | Выполнено |
| Проверка кнопки «следующий кадр» на форме с редактором анимации | Открытие следующего кадра | Открытие кадра | Выполнено |
| Проверка кнопки «последний кадр» на форме с редактором анимации | Открытие последнего кадра | Последний кадр | Выполнено |
| Проверка кнопки «анимировать» на форме с редактором анимации | Анимирование кадров с интервалом, указанным в edit 2 | Анимация кадров | Выполнено |
| Проверка кнопки «стоп» на форме с редактором анимации | Остановка анимации на кадре | Остановка анимации | Выполнено |
| Проверка кнопки «пуск» на форме с редактором анимации | Продолжение анимации с текущего кадра | Продолжение анимации | Выполнено |
| Проверка кнопки «сохранить gif» на форме с редактором анимации | Сохранение анимации под указанным именем, за счёт python файла | Сохранение анимации | Выполнено |
| Проверка кнопки «Справка» | Открытие справки программы | Открытие справки | Выполнено |
| Проверка кнопки «О программе» | Открытие информационного окна | Открытие информационного окна | Выполнено |

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

При использовании данного электронного средства обучения пользователь не столкнется с ошибками системы

**5 Применение**

**5.1 Общие сведения о программном продукте**

Разработанная программа является графическим редактором для создания анимации

Данная программа предназначена для рисования и дальнейшей анимации кадров.

Основными функциями приложения являются:

- рисование кадров в графическом редакторе.

- просмотр анимации в редакторе анимации.

- сохранение рисунков и анимации на компьютер.

- изучение справочной информации про функционал программы.

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

Тестирование проводилось на различных классах ЭВМ и работать с данной программой было комфортно. Но даже на системе следующего типа работать было комфортно:

Разработка происходила на ПК со следующими характеристиками:

- Процессор: i3-9400h 2.40 GHz;

- ОЗУ: 8Gb;

- Память: SSD 250GB;

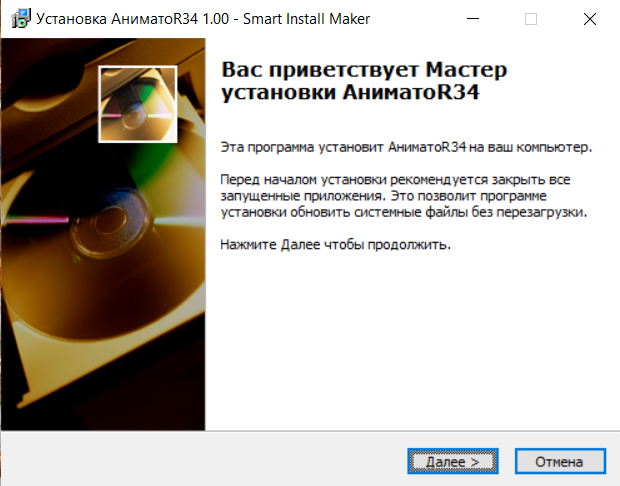
- ОС – Windows 10.

**5.2 Инсталляция**

Для того что бы установить программу необходимо запустить файл Setup.exe. Появится окно установки графического редактора для созданий анимации “АниматоR34”(Рисунок 8).

Рисунок 8 – Установка обучающего приложения АниматоR34

После нажатия на кнопку далее появится окно выбора каталога установки (Рисунок 7).



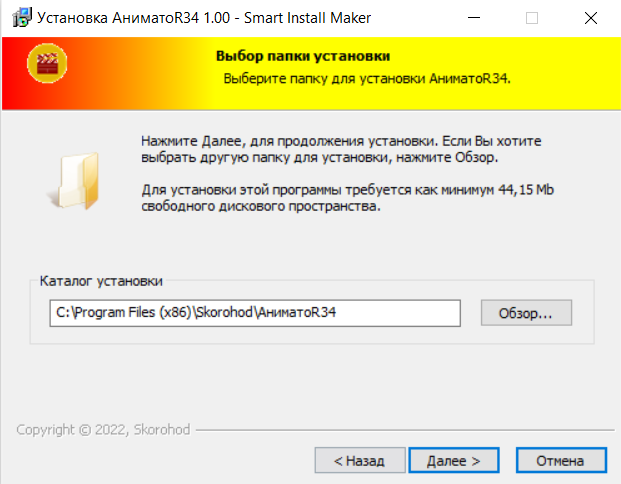


Рисунок 7 –Выбор каталога установки

После выбора каталога, нажимаем «Далее», после нам показывают каталог установки. Если же все устраивает, нажимаем «Установить» (Рисунок 8).

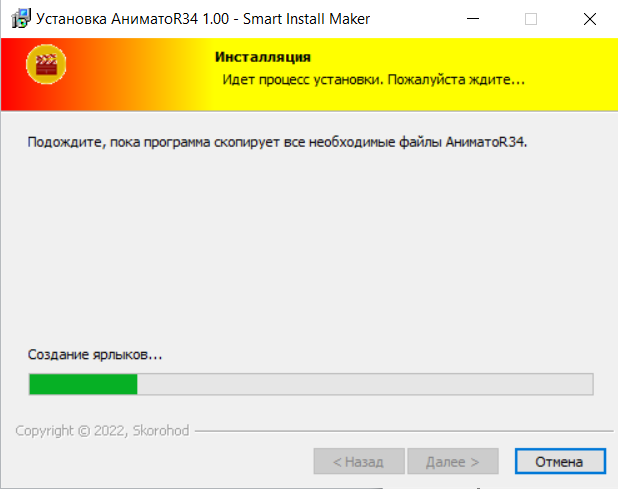


Рисунок 8 – Установка файлов

После установки, можно найти каталог с скачанными файлами (Рисунок 9):

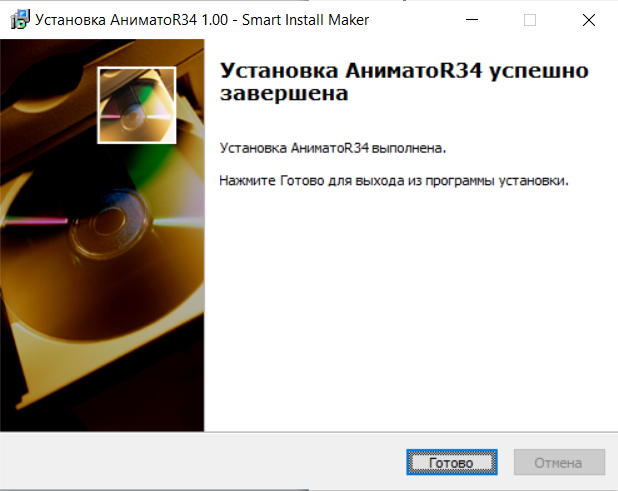


Рисунок 9 – Файлы установлены

**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 клавишей мыши на ярлык под названием АниматоR34.exe (Рисунок 10):



Рисунок 10 – Ярлык программы

Вторым способом является запуск из каталога, в который установилась программа.

**5.3.2 Инструкции по работе с программой**

После запуска программы на экране появляется загрузочный экран на котором можно перейти в графический редактор или редактор анимации (Рисунок 11).



Рисунок 11 – Загрузочный экран

После нажатия на кнопку с карандашом, нас переносит на окно с графическим редактором (Рисунок 12).



Рисунок 12 – Окно с графический редактор

На этой форме производится рисование кадров пользователем.

Работа с файлами: добавлена возможность открывать (для редактирования) и сохранять (для дальнейшего редактирования или создания анимации) файлы.

Рисование:

Главный инструмент для рисования – мышь.

Кнопки рисования располагаются на панели, очень простой и понятный интерфейс.

Пояснение кнопок для редактирования:

* Pen(ручка) – для рисования кривых линий.
* Circle(круг или эллипс) – для рисования окружностей и эллипсов.
* Rectangle(квадрат или прямоугольник) – для рисования квадратов и прямоугольников.
* Line(линия) – для рисования прямых линий.
* Filling(заливка) – для закраски закрытых областей.
* Eraser(ластик) – для закраски ненужных объектов белым цветом.
* Color(цвет) – выбор цвета, действует на (Pen, Circle (обводка фигуры), Rectangle (обводка фигуры), Line)
* Fill(цвет заливки) – выбор цвета, действует на (Circle (внутреннее пространство), Rectangle(внутреннее пространство), Filling)
* Ползунок – отвечает за толщину линий.

В главном меню есть все кнопки редактирования, описанные выше.

Затем мы можем нажать на кнопку в верхнем левом углу и перейти на окно с редактированием анимации. (Рисунок 13).

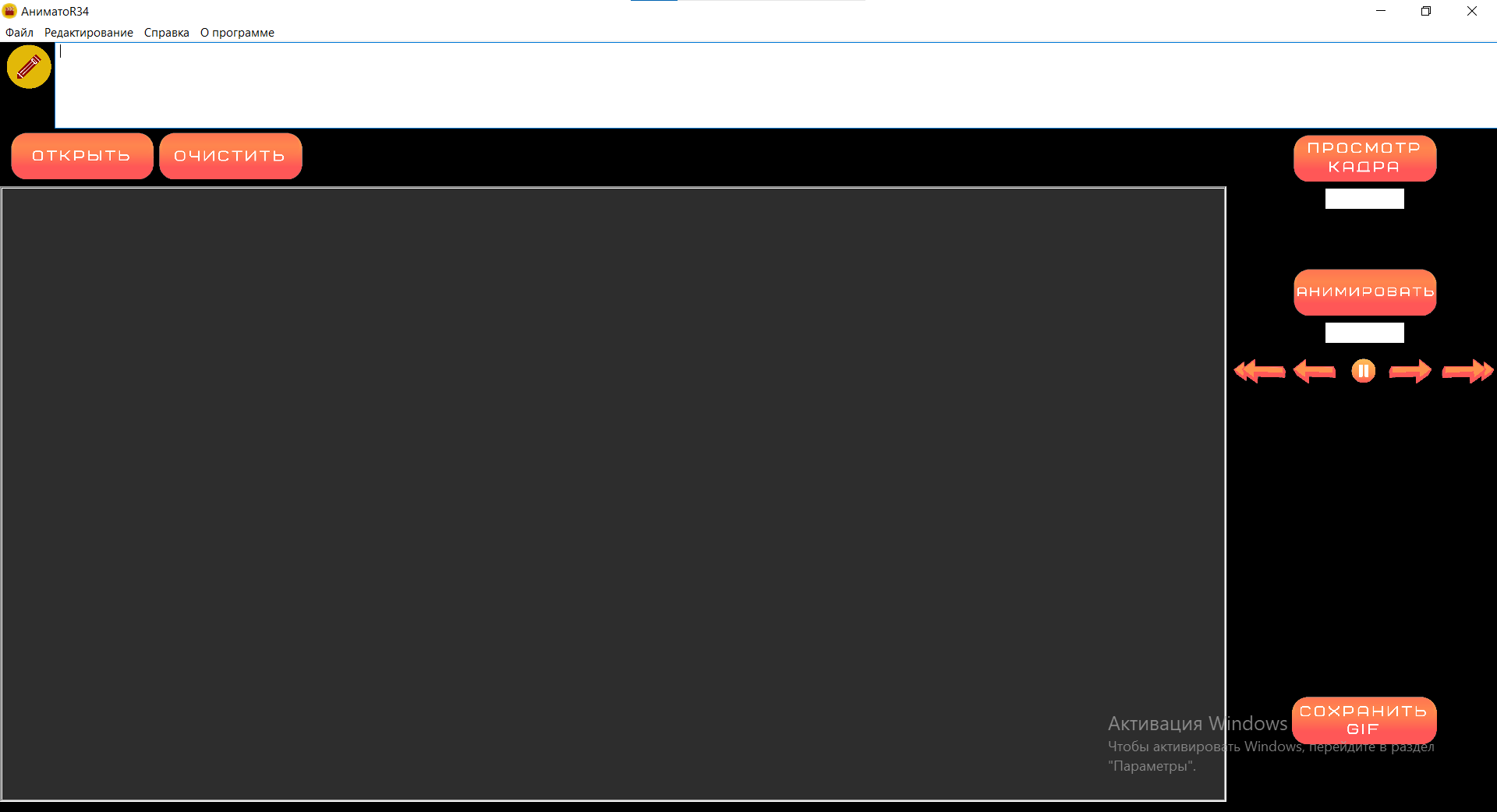


Рисунок 13 – Окно с редактором анимации

На этой форме производится выбор кадров на компьютере, анимирование кадров с настрокой интервала, скачивание полученной анимации в формате GIF-изображения.

Работа с файлами: добавлена возможность открывать кадры (для их прокрутки) и сохранять анимацию (в формате GIF).

Прокрутка кадров в программе:

после открытия файлов (кнопка "Открыть"), данные фалов (пути к файлам) записываются в поля Memo. В компоненте Memo можно копировать, вставлять и удалять пути к фалам (кадры).

Анимация в программе производится загрузкой файлов из полей Memo в компонент Image, через промежуток времени, который можно установить в поле ниже кнопки "Анимировать".

При необходимости в программе можно просматривать кадры. В поле ниже кнопки "Просмотр кадра" можно написать кадр для открытия его в Image. Также добавлены стрелки "На первый слайд", "На один кадр назад", "На один кадр вперёд", "На последний кадр" для перемещения между кадрами.

 Сохранение анимации:

При нажатии на кнопку "Сохранить GIF" в текстовый файл animate.txt записываются данные интервала, путь сохранения GIF, и пути кадров.

После чего вызывается Python файл, который считывает с файла animate.txt все данные и конвертирует их в GIF.

**5.3.3 Завершение работы с программой**

Завершить работу с программой можно 2 способами: из окон графического редактора и редактора анимации, нажав на кнопку «Выйти» или в главном меню нажать на кнопку «Выход» (Рисунок 14).

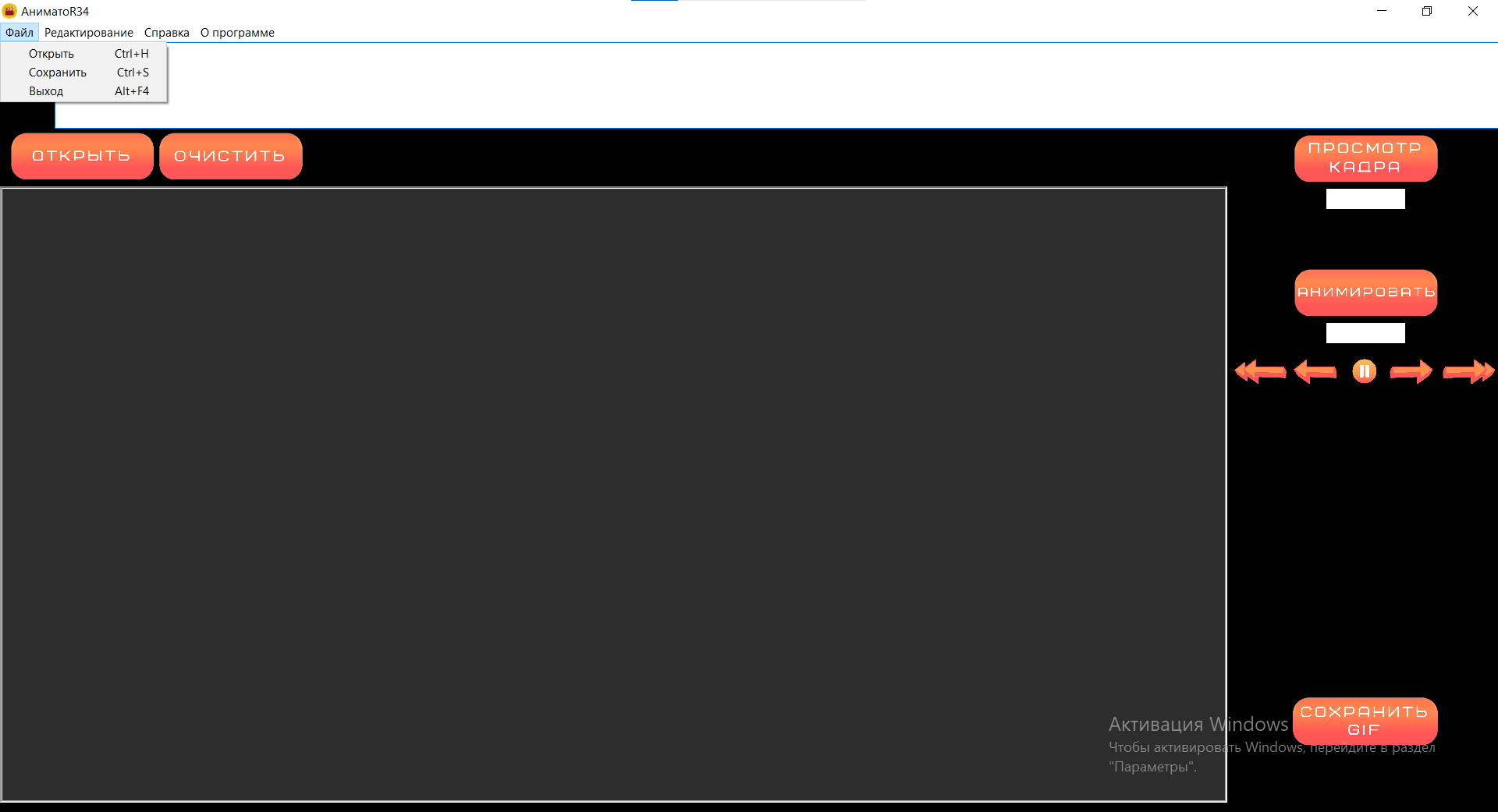


Рисунок 14– Выход из программы

**5.4 Использование системы справочной информации**

Справочную систему можно запустить с помощью главного меню «Справка» или нажав на клавишу F1 (Рисунок 15).



Рисунок 15 – Справка

После нажатия появиться окно со справочной информацией о программе (Рисунок 16).

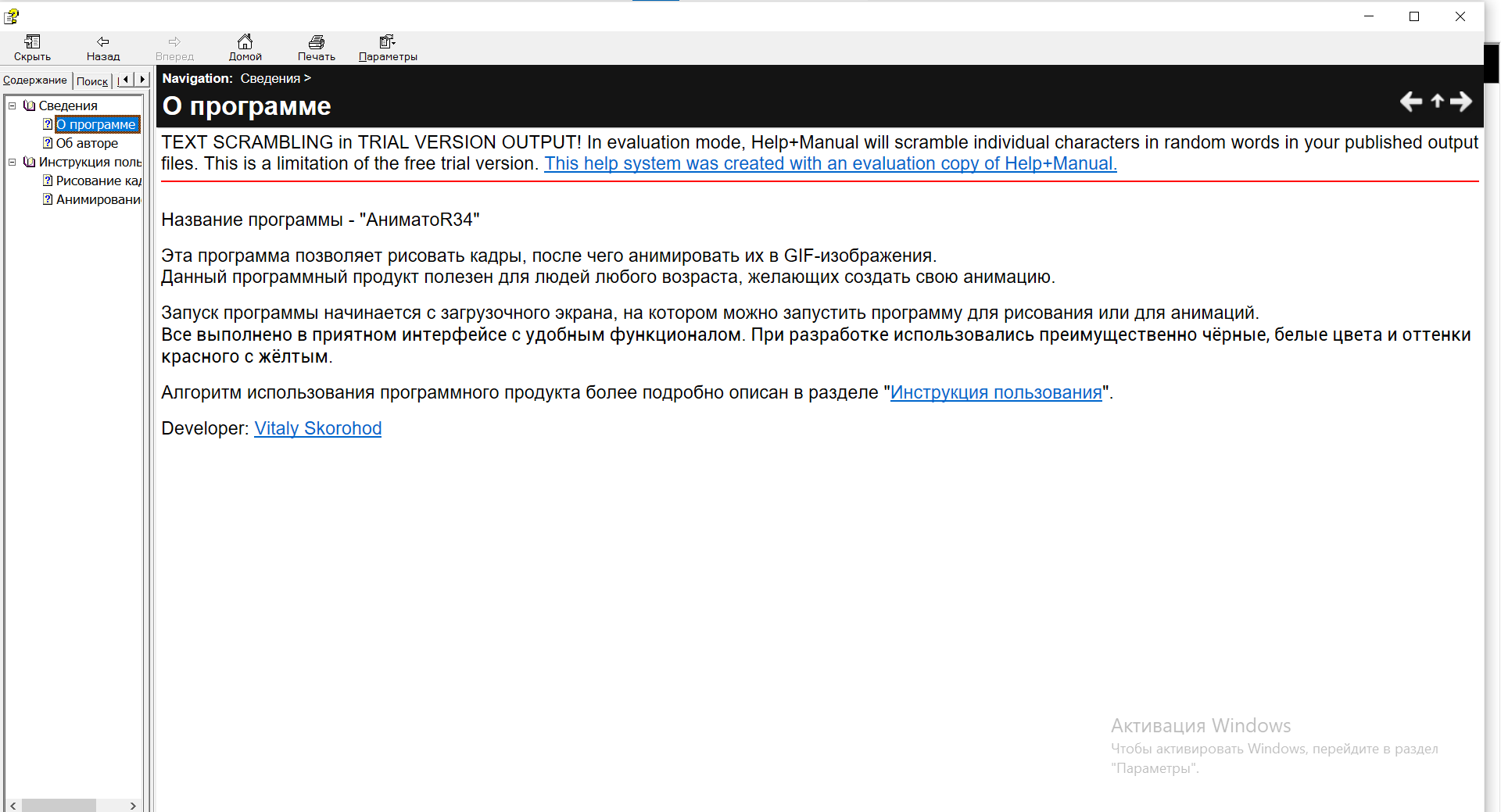


Рисунок 16 – Справочная информация

Информационное окно можно открыть с помощью главного меню «О программе» (рисунок 17)



Рисунок 17 – информационное окно

**Заключение**

Цель данного проекта была реализована – создан графический редактор для создания анимации.

Данное приложение будет интересно людям всех возрастов, желающих создать анимации.

Был реализован простой и понятный пользовательский интерфейс, который позволяет использовать приложение пользователю, не обладающему дополнительными знаниями ЭВМ.

В ходе реализации поставленной задачи был укреплён пройденный курс программирования в среде Delphi, а также получено много дополнительной информации о её возможностях.

После тщательного тестирования приложения были выявлены некоторые недоработки, которые были частично исправлены на стадии проектирования, и полностью исключены на стадии тестирования программы. В целом при реализации программы, были выполнены все условия, перечисленные в предыдущих разделах пояснительной записки.

Таким образом, можно сказать, что поставленная задача была реализована успешно.

**Список используемых источников**

1. Киберфорум «Рисование на форме» – Режим доступа: <https://www.cyberforum.ru/delphi-multimedia/thread75538.html> – Дата доступа 18.06.2022.

2. Форум профессиональных и начинающих программистов. [Электронный ресурс]. – Режим доступа: <https://www.programmersforum.ru/>. – Дата доступа 20.06.2022.

3. Иллюстрированный самоучитель по Delphi [Электронный ресурс]. – Режим доступа: [samoychiteli.ru/document27136.htm](file:///C:\Users\User\Downloads\samoychiteli.ru\document27136.htm). – Дата доступа 22.06.2022.

4. Электронная книга по Delphi [Электронный ресурс]. – Режим доступа: <https://lp.embarcadero.com/DelphiBookInRussia>. –Дата доступа 24.06.2022.

5. Создание относительных путей к файлам/картинкам[  
Электронный ресурс]. – Режим доступа: <http://delphibasics.ru/ExtractFilePath.php> - Дата доступа 24.06.2022.

6. Программирование в Delphi [видео ресурс] – Режим доступа: <https://www.youtube.com/watch?v=6xtI8sxfbSI> – Дата доступа 25.06.2022

Приложение A

Листинг программы

From2 – загрузочный экран

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

begin

Application.Terminate;

end;

procedure TForm2.FormCreate(Sender: TObject);

begin

//загрузка картинок

image1.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'main icons/applic.png');

image2.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'main icons/iconstartpani.png');

image3.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'main icons/iconstartpain.png');

//загрузка иконки приложения

form2.Icon.LoadFromFile(ExtractFilePath(Application.ExeName)+'main icons/AниматоR34\_Icon1.ico');

end;

procedure TForm2.Image2Click(Sender: TObject);

begin

//при нажатии на кнопку запускает таймер

timer2.Enabled:=true;

Gauge1.Progress:=Gauge1.Progress+1; //прибавляет gauge 1% каждые 20 мс

image1.Cursor:=crAppstart;

image2.Cursor:=crAppstart;

image3.Cursor:=crAppstart;

if(Gauge1.Progress=100) then //если gauge=100, то переходит на форму с редактором анимации

begin

Timer2.Enabled:=False;

Form2.Hide;

Form3.Show;

end;

end;

procedure TForm2.Image2Clicktimer(Sender: TObject);

begin

// рип нажатии на кнопку запускает таймер

timer1.Enabled:=true;

Gauge1.Progress:=Gauge1.Progress+1; //приибавляет gauge 1% каждые 20 мс

image1.Cursor:=crAppstart;

image2.Cursor:=crAppstart;

image3.Cursor:=crAppstart;

if(Gauge1.Progress=100) then //если gauge=100, то переходит на форму с графическим редактором

begin

Timer1.Enabled:=False;

Form2.Hide;

Form1.Show;

end;

end;

end.

Form1 – графический редактор

procedure TForm1.circle1Click(Sender: TObject);

begin

speedbutton2.down:= true;

end;

procedure TForm1.circle2Click(Sender: TObject);

begin

speedbutton3.down:= true;

end;

procedure TForm1.Color1Close(Sender: TObject);

begin

image14.Visible:=true;

end;

procedure TForm1.Color2Close(Sender: TObject);

begin

image16.Visible:=true;

end;

procedure TForm1.ctrz1Click(Sender: TObject);

begin

//загружает последний шаг из buffer в img

img.Assign(buffer2);

timer1.Enabled:=true;

end;

procedure TForm1.eraser1Click(Sender: TObject);

begin

speedbutton5.down:= true;

end;

procedure TForm1.eraser2Click(Sender: TObject);

begin

speedbutton6.down:= true;

end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);

begin

application.Terminate;

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

//загрузка картинок

image1.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/penfalse.jpg');

image2.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/circlefalse.jpg');

image3.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/rectfalse.jpg');

image4.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/linefalse.jpg');

image5.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/fillfalse.jpg');

image6.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/eraserfalse.jpg');

image7.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/pentrue.jpg');

image8.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/circletrue.jpg');

image9.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/recttrue.jpg');

image10.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/linetrue.jpg');

image11.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/filltrue.jpg');

image12.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/erasertrue.jpg');

image13.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'main icons/iconstartpani.png');

image14.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/цветfalse.png');

image15.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/цветtrue.png');

image16.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/заливкаfalse.png');

image17.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments/заливкаtrue.png');

//загрузка иконки

form1.Icon.LoadFromFile(ExtractFilePath(Application.ExeName)+'main icons/AниматоR34\_Icon1.ico');

//создаёт TBitmap

Img:=TBitmap.Create;

buffer:=TBitmap.Create;

buffer2:=TBitmap.Create;

//присваитвает размер PaintBox

img.Width:=PaintBox1.ClientWidth;

buffer.Width:=PaintBox1.ClientWidth;

buffer2.Width:=PaintBox1.ClientWidth;

img.Height:=PaintBox1.ClientHeight;

buffer.Height:=PaintBox1.ClientHeight;

buffer2.Height:=PaintBox1.ClientHeight;

//выбирается инструмент для рисования (карандаш)

nowdrawing:=sPen;

// переменной dwn присваивает значение false

dwn:=false;

// цвет заливки по умолчанию -- белый

col2:=clwhite;

end;

procedure TForm1.Image13Click(Sender: TObject);

begin

// при нажатии кнопки переходит на форму с редактором анимации

form3.show;

form1.hide;

end;

procedure TForm1.Image14Click(Sender: TObject);

begin

image14.Visible:=false;

//открывает ColorDialog и выбирается цвет пользователем

if Color1.Execute then

col1:= Color1.Color;

end;

procedure TForm1.Image16Click(Sender: TObject);

begin

image16.Visible:=false;

//открывает ColorDialog и выбирается цвет заливки пользователем

if Color2.Execute then

col2:= Color2.Color;

end;

procedure TForm1.Image1Click(Sender: TObject);

begin

speedbutton1.down:=true;

if speedbutton1.down then image1.Visible:=false;

if speedbutton1.down = false then image1.Visible:=true;

if speedbutton2.down = false then image2.Visible:=true;

if speedbutton3.down = false then image3.Visible:=true;

if speedbutton4.down = false then image4.Visible:=true;

if speedbutton5.down = false then image5.Visible:=true;

if speedbutton6.down = false then image6.Visible:=true;

end;

procedure TForm1.Image2Click(Sender: TObject);

begin

speedbutton2.down:=true;

if speedbutton2.down then image2.Visible:=false;

if speedbutton1.down = false then image1.Visible:=true;

if speedbutton2.down = false then image2.Visible:=true;

if speedbutton3.down = false then image3.Visible:=true;

if speedbutton4.down = false then image4.Visible:=true;

if speedbutton5.down = false then image5.Visible:=true;

if speedbutton6.down = false then image6.Visible:=true;

end;

procedure TForm1.Image3Click(Sender: TObject);

begin

speedbutton3.down:=true;

if speedbutton3.down then image3.Visible:=false;

if speedbutton1.down = false then image1.Visible:=true;

if speedbutton2.down = false then image2.Visible:=true;

if speedbutton3.down = false then image3.Visible:=true;

if speedbutton4.down = false then image4.Visible:=true;

if speedbutton5.down = false then image5.Visible:=true;

if speedbutton6.down = false then image6.Visible:=true;

end;

procedure TForm1.Image4Click(Sender: TObject);

begin

speedbutton4.down:=true;

if speedbutton4.down then image4.Visible:=false;

if speedbutton1.down = false then image1.Visible:=true;

if speedbutton2.down = false then image2.Visible:=true;

if speedbutton3.down = false then image3.Visible:=true;

if speedbutton4.down = false then image4.Visible:=true;

if speedbutton5.down = false then image5.Visible:=true;

if speedbutton6.down = false then image6.Visible:=true;

end;

procedure TForm1.Image5Click(Sender: TObject);

begin

speedbutton5.down:=true;

if speedbutton5.down then image5.Visible:=false;

if speedbutton1.down = false then image1.Visible:=true;

if speedbutton2.down = false then image2.Visible:=true;

if speedbutton3.down = false then image3.Visible:=true;

if speedbutton4.down = false then image4.Visible:=true;

if speedbutton5.down = false then image5.Visible:=true;

if speedbutton6.down = false then image6.Visible:=true;

end;

procedure TForm1.Image6Click(Sender: TObject);

begin

speedbutton6.down:=true;

if speedbutton6.down then image6.Visible:=false;

if speedbutton1.down = false then image1.Visible:=true;

if speedbutton2.down = false then image2.Visible:=true;

if speedbutton3.down = false then image3.Visible:=true;

if speedbutton4.down = false then image4.Visible:=true;

if speedbutton5.down = false then image5.Visible:=true;

if speedbutton6.down = false then image6.Visible:=true;

end;

procedure TForm1.N2Click(Sender: TObject);

begin

//открывает OpenDialog с выбором рисунка для открытия

if Open1.Execute then

begin

img:=TBitmap.create;

img.loadfromfile(Open1.FileName); //загружает выбранные файл в PaintBox

paintbox1.width:=img.width;

paintbox1.height:=img.height;

end

else ShowMessage('Вы не выбрали файл')

end;

procedure TForm1.N3Click(Sender: TObject);

var name:string;

begin

// открывает SaveDialog

if Save1.Execute then

begin

//сохранияет рисунок под именем выбранным пользователем и добавляет .bmp

name:= Save1.FileName+'.bmp';

img.SaveToFile(name);

end

else ShowMessage('Вы не созранили файл');

end;

procedure TForm1.N4Click(Sender: TObject);

begin

// закрыввает программу

close;

end;

procedure TForm1.N6Click(Sender: TObject);

begin

speedbutton1.down:= true;

end;

procedure TForm1.N7Click(Sender: TObject);

begin

//загружает справку

ShellExecute(0,PChar('Open'),PChar('AниматоR34help.chm'),nil,nil,SW\_SHOW);

end;

procedure TForm1.N8Click(Sender: TObject);

begin

//загружает форму информационного окна

form4.show;

end;

procedure TForm1.paintbox1MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

buffer.Assign(img); //загружает img в buffer

//если нажата левая кнопка мыши, то загружает buffer в img и принимает положение мыши

if button=mbLeft then begin

img.assign(buffer);

x0:=x; y0:=y;

//проверка какая кнопка для рисования нажата

if SpeedButton1.Down then

begin

nowdrawing:=sPen;

img.canvas.MoveTo(x,y);

end else

if SpeedButton2.Down then

nowdrawing:=sEllipse else

if SpeedButton3.Down then

nowdrawing:=sRect else

if SpeedButton4.Down then

nowdrawing:=sPoly else

if SpeedButton5.Down then

nowdrawing:=sFill;

dwn:=true;

img.Canvas.Pen.Color:=col1;

img.Canvas.Brush.Color:=col2;

//если нажата кнопка "ластик", то принимает значение кнопки "карандаш", но с белым цветом

if SpeedButton6.Down then

begin

nowdrawing:=sPen;

img.Canvas.MoveTo(x,y);

img.Canvas.Pen.Color:=clWhite;

end;

// ширина для рисования принимает значение позиции TrackBar

img.Canvas.pen.Width:=TrackBar1.Position;

if nowdrawing=sFill then

begin

img.Canvas.FloodFill(x0,y0,img.Canvas.Pixels[x,y],fsSurface);

dwn:=false;

end

end else

begin

if (dwn)and(nowdrawing=sPoly) then begin

x0:=x;

y0:=y;

buffer.Assign(img);

end;

end;

// загружает img в buffer2

buffer2.Assign(img);

paintbox1.Canvas.CopyRect(bounds(0,0,img.Width,img.Height),

img.Canvas,bounds(0,0,img.Width,img.Height));

end;

procedure TForm1.paintbox1MouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

begin

//если dwn опущена то не рисует

if not dwn then exit;

img.assign(buffer);

case nowdrawing of

//рисует в зависимости, какое значение принимает nowdarawing

sPen:begin

img.Canvas.LineTo(x,y);

buffer.Assign(img);

end;

sRect:begin

img.Canvas.Rectangle(x0,y0,x,y);

end;

sEllipse:begin

img.Canvas.Ellipse(x0,y0,x,y);

end;

sPoly:begin

img.Canvas.MoveTo(x0,y0);

img.Canvas.LineTo(x,y);

end;

sFill:begin

end;

end;

paintbox1.Canvas.CopyRect(bounds(0,0,img.Width,img.Height),

img.Canvas,bounds(0,0,img.Width,img.Height));

end;

procedure TForm1.paintbox1MouseUp(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

//если кнопка на Paintbox отпустилась, переменной dwn присваивает значение false

if button=mbLeft then dwn:=false;

end;

procedure TForm1.paintbox1Paint(Sender: TObject);

begin

paintbox1.Canvas.CopyRect(bounds(0,0,img.Width,img.Height),

buffer.Canvas,bounds(0,0,img.Width,img.Height));

timer1.Enabled:=false;

end;

procedure TForm1.straightline1Click(Sender: TObject);

begin

speedbutton4.down:= true;

end;

end.

Form3 – редактор анимации

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

begin

application.Terminate;

end;

procedure TForm3.FormCreate(Sender: TObject);

begin

// загрузка картинок

image2.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\очиститьfalse.png');

image3.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\очиститьtrue.png');

image4.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\открытьfalse.png');

image5.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\открытьtrue.png');

image6.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\просмотр кадраfalse.png');

image7.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\просмотр кадраtrue.png');

image8.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\анимироватьfalse.png');

image9.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\анимироватьtrue.png');

image10.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\назадallfalse.png');

image11.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\назад1false.png');

image12.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\вперёд1false.png');

image13.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\вперёдallfalse.png');

image14.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\назадalltrue.png');

image15.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\назад1true.png');

image16.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\вперёд1true.png');

image17.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\вперёдalltrue.png');

image18.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\паузаfalse.png');

image19.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\паузаtrue.png');

image20.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'main icons/iconstartpain.png');

image21.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\сохранитьfalse.png');

image22.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'icons instruments\сохранитьtrue.png');

// загрузка иконки программы

form3.Icon.LoadFromFile(ExtractFilePath(Application.ExeName)+'main icons/AниматоR34\_Icon1.ico');

end;

procedure TForm3.Image10Click(Sender: TObject);

begin

//если линий memo не равно 0, то открывает 0 строку в image

if memo1.Lines.Count<>0 then

begin

image1.Picture.LoadFromFile(memo1.Lines[0]);

i1:=0;

end;

image10.Visible:=false;

timer2.Enabled:=true;

end;

procedure TForm3.Image11Click(Sender: TObject);

begin

image11.Visible:=false;

timer2.Enabled:=true;

while i1<0 do

i1:=i1+1;

//уменьшает значение переменной i1 на 1

i1:= i1-1;

i:=i1;

//елсли значение i1 не равно -1, то открывает линию под номером i1

if i1<>-1 then

image1.Picture.LoadFromFile(memo1.Lines[i1]);

end;

procedure TForm3.Image12Click(Sender: TObject);

begin

image12.Visible:=false;

timer2.Enabled:=true;

while i1>=memo1.Lines.Count do

i1:=i1-1;

//увеличивает значение переменной i1 на 1

i1:= i1+1;

i:=i1;

//елсли значение i1 не равно количеству лмний в memo, то открывает линию под номером i1

if i1<>memo1.lines.Count then

image1.Picture.LoadFromFile(memo1.Lines[i1]);

end;

procedure TForm3.Image13Click(Sender: TObject);

begin

//если значение линий memo не равно 0, то открывавет посленюю линию в image

if memo1.Lines.Count<>0 then

begin

image1.Picture.LoadFromFile(memo1.Lines[memo1.Lines.Count-1]);

i1:=memo1.Lines.Count-1;

end;

image13.Visible:=false;

timer2.Enabled:=true;

end;

procedure TForm3.Image18Click(Sender: TObject);

begin

//останавливает таймер1

timer1.Enabled:=false;

image18.Visible:=false;

end;

procedure TForm3.Image19Click(Sender: TObject);

begin

//запускает таймер 1 если значение линий memo не равно 0

if memo1.Lines.Count<>0 then

begin

timer1.Enabled:=true;

end;

image18.Visible:=true;

end;

procedure TForm3.Image20Click(Sender: TObject);

begin

// при нажатии на кнопку открывает форму с графическим редактором

form1.show;

form3.hide;

end;

procedure TForm3.Image21Click(Sender: TObject);

var

dir:string;

name:string;

line: TStringList;

begin

image21.Visible:=false;

//сохранияет директорию в dir

dir:=GetCurrentDir;

//открывает Savedialog и сохраняет имя в переменную name, прибавляя ".gif"

if Savedialog1.Execute then

begin

name:= savedialog1.FileName+'.gif'

end;

//добавляет в StringList интервал таймера, переменную name и линии memo

line:= TstringList.Create;

line.Add(intToStr(timer1.Interval));

line.Add(name);

line.Add(memo1.Lines.Text);

// после сохраняет линии StringList в файл animate.txt

line.SaveToFile(ExtractfilePath(Application.ExeName)+'animate.txt');

SetCurrentDir(dir); // возвращает значение dir

ShellExecute(Handle, nil,'gifanimate.py', nil, nil, SW\_SHOW);// запускает python файл

end;

procedure TForm3.Image2Click(Sender: TObject);

begin

//очичает линии memo

memo1.Lines.Clear;

image2.Visible:=false;

timer2.Enabled:=true;

end;

procedure TForm3.Image4Click(Sender: TObject);

var i: integer;

dir:string;

begin

i1:=0;

dir:=GetCurrentDir; // сохраняет директорию в переменную dir

image4.Visible:=false;

//открывает OpenDialog и добавляет пути к файлам в линии memo

if Open1.Execute then

begin

for i:=0 to open1.Files.Count-1 do begin

memo1.Lines.add(open1.Files[i])

end

end;

SetCurrentDir(dir); //возвращает значение dir

end;

procedure TForm3.Image6Click(Sender: TObject);

begin

image6.Visible:=false;

timer2.Enabled:=true;

//если значение линий memo не равно 0, то открывает картинку под номером edit1

if memo1.Lines.Count<>0 then

begin

image1.picture.LoadFromFile(memo1.Lines[strtoint(edit1.text)-1]);

i1:=strtoint(edit1.text)-1

end

else showmessage('Нет кадров');

end;

procedure TForm3.Image8Click(Sender: TObject);

begin

//если значение edit2 не пустое, то интервалу таймеру1 присваивается значение edit2

if edit2.text<>'' then

timer1.Interval:=strtoint(edit2.text);

image8.Visible:=false;

image18.Visible:=true;

timer2.Enabled:=true;

//если значение линий memo не равно 0, то запускает таймер1

if memo1.lines.Count<>0 then

begin

timer1.Enabled:=True;

i:=0;

end

else showmessage('Нет кадров');

end;

procedure TForm3.N10Click(Sender: TObject);

begin

image11.OnClick(self);

end;

procedure TForm3.N11Click(Sender: TObject);

begin

image12.OnClick(self);

end;

procedure TForm3.N12Click(Sender: TObject);

begin

image10.OnClick(self);

end;

procedure TForm3.N13Click(Sender: TObject);

begin

image11.OnClick(self);

end;

procedure TForm3.N14Click(Sender: TObject);

begin

form4.show;

end;

procedure TForm3.N3Click(Sender: TObject);

begin

//при нажатии кнопки запускает спрвку

ShellExecute(0,PChar('Open'),PChar('AниматоR34help.chm'),nil,nil,SW\_SHOW);

end;

procedure TForm3.N4Click(Sender: TObject);

begin

image4.OnClick(self);

end;

procedure TForm3.N5Click(Sender: TObject);

begin

image21.OnClick(self);

end;

procedure TForm3.N6Click(Sender: TObject);

begin

form3.close;

end;

procedure TForm3.N7Click(Sender: TObject);

begin

image2.OnClick(self);

end;

procedure TForm3.N8Click(Sender: TObject);

begin

image6.OnClick(self);

end;

procedure TForm3.N9Click(Sender: TObject);

begin

image8.OnClick(self);

end;

procedure TForm3.Open1Close(Sender: TObject);

begin

image4.Visible:=true;

end;

procedure TForm3.SaveDialog1Close(Sender: TObject);

begin

image21.Visible:=true;

end;

procedure TForm3.Timer1Timer(Sender: TObject);

begin

i1:=i;

//заружает пути из memo1 в image

image1.picture.LoadFromFile(memo1.lines[i]);

i:=i+1;

//если значение i>=линий memo, то останавливает таймер

if i>=memo1.Lines.Count then

timer1.Enabled:=false;

end;

procedure TForm3.Timer2Timer(Sender: TObject);

begin

image2.Visible:=true;

image6.Visible:=true;

image8.Visible:=true;

image10.Visible:=true;

image11.Visible:=true;

image12.Visible:=true;

image13.Visible:=true;

timer2.Enabled:=false;

end;

end.

Form4 – информационное окно

procedure TForm4.FormCreate(Sender: TObject);

begin

//загружает картинки

image1.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'main icons/infoscreen.png');

//загружает иконку

form4.Icon.LoadFromFile(ExtractFilePath(Application.ExeName)+'main icons/AниматоR34\_Icon1.ico');

end;