Министерство образования Республики Беларусь

УО «Белорусский Государственный Университет Информатики и Радиоэлектроники»

Кафедра информатики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по курсу

«Архитектура вычислительных систем»

Тема: «Графический редактор»

Выполнил:

студент группы 253505

Волчецкий А.М.

Проверил:

Богданов С.А.

Минск, 2015

СОДЕРЖАНИЕ

1. [Введение 3](#_Toc295107539)
2. [Основные сведения о графических редакторах 4](#_Toc295107540)

2.1 Растровые графические редакторы…………………………………6

2.2 Векторные графические редакторы………………………………...6

2.3 Форматы файлов для хранения графических изображений………8

1. [Графические редакторы,](#_Toc295107541)

используемые для создания изображений…………………………...10

3.1 Программы просмотра графических изображений……………....10

3.2 Редакторы растровой графики…………………………………….10

3.3 Редакторы векторной графики…………………………………….11

3.4 Понятие о цветовом пространстве. Основные цветовые модели (RGB, CMYK)…………………………………………………………..11

1. [Создание графического редактора](#_Toc295107542)…………………………………...13

4.1 Описание проектного решения…………………………………...13

4.2 Программная реализация…………………………………………13

4.3 Руководство пользователя………………………………………..16

1. [Заключение](#_Toc295107549)…………………………………………………………….18
2. [Список использованной литературы…………………](#_Toc295107548)………………19
3. [Приложение](#_Toc295107549)……………………………………………………………19
4. **Введение**

Представление данных на мониторе компьютера в графическом виде впервые было реализовано в середине 50-х годов для больших ЭВМ, применявшихся в научных и военных исследованиях. С тех пор графический способ отображения данных стал неотъемлемой частью подавляющего числа компьютерных систем, в особенности персональных.

Компьютерная графика - это специальная область информатики, изучающая методы и средства создания и обработки изображений с помощью программно-аппаратных вычислительных комплексов. Она охватывает все виды и формы представления изображений, доступных для восприятия человеком либо на экране монитора, либо в виде копии на внешнем носителе (бумага, киноплёнка, ткань и прочее).

Без компьютерной графики невозможно представить себе не только компьютерный, но и обычный, вполне материальный мир. На сегодняшний день компьютеры и компьютерная графика неотъемлемая часть жизни современного общества. Для примера назовём медицину (компьютерная томография), научные исследования (визуализация строения вещества, векторных полей и других данных), моделирование тканей и одежды, опытно-конструкторские разработки, рекламные щиты, цветные журналы, спецэффекты в фильмах - всё это в той или иной мере имеет отношение к компьютерной графике. Поэтому созданы программы для создания и редактирования изображений, то есть графические редакторы.

Компьютерной графикой в последнее время занимаются многие, что обусловлено высокими темпами развития вычислительной техники. Более 90% информации здоровый человек получает через зрение или ассоциирует с геометрическими пространственными представлениями. Компьютерная графика имеет огромный потенциал для облегчения процесса познания и творчества.

В настоящее время рынок программного обеспечения переполнен различными программами и редакторами, позволяющими обрабатывать и редактировать цифровые фото. Говоря о графических редакторах, прежде всего, необходимо отметить, что все цифровые изображения подразделяются на векторные и точечные. В первом случае изображения построены из различных геометрических элементов или примитивов (отрезков, треугольников, прямоугольников или окружностей). Поэтому векторная графика позволяет легко манипулировать масштабом изображения без каких бы то ни было геометрических искажений, а потому широко используется для построения шрифтов, рисованных изображений, в оформительской и издательской работе.

Графический редактор - программа (или пакет программ), позволяющая создавать и редактировать двумерные изображения с помощью компьютера.

Актуальность курсовой работы на тему «Графический редактор» обусловлена тем, что любой школьник, студент и преподаватель в настоящее время должен владеть компьютерными технологиями на высоком уровне.

Объектом курсовой работы служат современные компьютерные технологии с их аппаратным и программные обеспечением.

Предметом курсового исследования являются функциональные возможности системы графических редакторов, используемые в науке и образовании.

Целью курсовой работы является создание графического редактора, используя объектно-ориентированный язык программирования, описание его функциональных возможностей.

Гипотеза курсовой работы состоит в следующем: систематическое освоение и использование функциональных возможностей графического редактора способствует повышению уровня графической подготовки школьников и студентов, решающих учебные и научные задачи.

Задачами курсовой работы выделяется:

1) освоение полного спектра функциональных возможностей графических редакторов различных видов;

2) изучение программ для просмотра графических изображений, выделение области применения графического редактора;

3) создание графического редактора, используя объектно-ориентированный язык программирования, описание его функциональных возможностей.

**2. Общие сведения о графических редакторах**

**2.1 Растровые графические редакторы**

Растровый графический редактор - специализированная программа, предназначенная для создания и обработки изображений. Подобные программные продукты нашли широкое применение в работе художников-иллюстраторов и фотографов, при подготовке изображений к печати типографским способом или на фотобумаге, публикации в интернете.

Что бы охарактеризовать растровые графические редакторы, можно сравнить их с обычным рисованием, то есть с рисованием карандашом на обычной бумаге. Основное отличие между рисунками на бумаге и на экране монитора состоит в том, что на мониторе любое изображение дискретно, то есть оно состоит из конечного числа прямоугольных точек, каждая из которых может быть только одного цвета из фиксированной палитры. У рисунка на бумаге такую минимальную единицу, как точка, найти нельзя. На бумаге невозможно нарисовать два совершенно одинаковых рисунка. С помощью растрового редактора можно создать именно такой дискретный рисунок, состоящий из фиксированного числа точек, после чего можно изменять цвет каждой точки по отдельности. Именно поэтому любой редактор такого типа перед созданием рисунка просит указать его точные размеры и иногда палитру цветов. Такие редакторы позволяют строить линии и графические примитивы, заливать области определенным цветом, вбивать текст, рисовать различными инструментами - карандашами, кистями, распылителями. В зависимости от выбранного инструмента строятся линии с различными свойствами - полупрозрачные, с размытыми краями, заполненные текстурой. Всегда есть возможность увеличивать масштаб, для того чтобы можно было работать с отдельными точками.

Растровые графические редакторы позволяют пользователю рисовать и редактировать изображения на экране компьютера, а также сохранять их в различных растровых форматах. Например, JPEG и TIFF, позволяют сохранять растровую графику с незначительным снижением качества за счёт использования алгоритмов сжатия с потерями, PNG и GIF, поддерживают хорошее сжатие без потерь, и BMP, также поддерживает сжатие, но в общем случае представляет собой несжатое «попиксельное» описание изображения.

Растровые графические редакторы имеют как минимум два значительных недостатка:

1) масштабирование. Дело в том, что при масштабировании рисунка его четкость не увеличивается, и пикселей не становится больше.

2) размер файла. Файл растрового графического редактора сохраняет в себе информацию о каждой точке рисунка, ее положение, цвет. Причем это самая минимальная информация, хотя и она занимает немало памяти. В более мощных редакторах сохраняется по несколько слоев, и размер отдельных файлов может достигать десятков и сотен мегабайтов.

Что бы избавиться от проблемы с масштабированием, изобрели технологию интерполяции изображений. Эта технология состоит в том, что на картинку добавляются промежуточные пиксели, которые располагаются между основными пикселями и принимают цвет, средний для соседних ему. Однако такая технология не на много, более того, она почти не увеличивает качества изображения, а лишь увеличивает его размер.

**2.2 Векторные графические редакторы**

графический редактор изображение трехмерный

Основным логическим элементом векторной графики является геометрический объект. В качестве объекта принимаются простые геометрические фигуры (так называемые примитивы - прямоугольник, окружность, эллипс, линия), составные фигуры или фигуры, построенные из примитивов, цветовые заливки, в том числе градиенты.

Преимущество векторной графики заключается в том, что форму, цвет и пространственное положение составляющих ее объектов можно описывать с помощью математических формул а не описывать каждый пиксель в отдельности.

Важным объектом векторной графики является сплайн. Сплайн - это кривая, посредством которой описывается та или иная геометрическая фигура[1]. На сплайнах построены современные шрифты TryeType и PostScript.

У векторной графики много достоинств. Она экономна в плане дискового пространства, необходимого для хранения изображений: это связано с тем, что сохраняется не само изображение, а только некоторые основные данные, используя которые, программа всякий раз воссоздает изображение заново.

Объекты векторной графики легко трансформируются и модифицируются, что не оказывает практически никакого влияния на качество изображения. Масштабирование, поворот, искривление могут быть сведены к паре-тройке элементарных преобразований над векторами.

В тех областях графики, где важное значение имеет сохранение ясных и четких контуров, например, в шрифтовых композициях, в создании логотипов и прочее, векторные программы незаменимы.

Векторная графика может включать в себя и фрагменты растровой графики: фрагмент становится таким же объектом, как и все остальные (правда, со значительными ограничениями в обработке).

Однако, с другой стороны, векторная графика может показаться чрезмерно жесткой, "фанерной". Она действительно ограничена в чисто живописных средствах: в программах векторной графики практически невозможно создавать фотореалистические изображения.

В любом случае, если вы работаете с графикой, то неизбежно будете иметь дело с обеими ее формами - векторной и растровой. Понимание их сильных и слабых сторон позволит вам выполнить свою работу максимально эффективно.

Векторная графика описывает изображения с использованием прямых и изогнутых линий, называемых векторами, а также параметров, описывающих цвета и расположение. Например, изображение древесного листа описывается точками, через которые проходит линия, создавая тем самым контур листа. Цвет листа задается цветом контура и области внутри этого контура. В растровом редакторе лист дерева описывается конкретным расположением и цветом каждой точки сетки.

При редактировании элементов векторной графики Вы изменяете параметры прямых и изогнутых линий, описывающих форму этих элементов. Вы можете переносить элементы, менять их размер, форму и цвет, но это не отразится на качестве их визуального представления. Векторная графика не зависит от разрешения, т.е. может быть показана в разнообразных выходных устройствах с различным разрешением без потери качества.

Увеличение или уменьшение объектов производится увеличением или уменьшением соответствующих коэффициентов в математических формулах. К сожалению, векторный формат становится невыгодным при передаче изображений с большим количеством оттенков или мелких деталей (например, фотографий). Ведь каждый мельчайший блик в этом случае будет представляться не совокупностью одноцветных точек, а сложнейшей математической формулой или совокупностью графических примитивов, каждый из которых, является формулой. Это приводит к утяжелению файла.

**2.3 Форматы файлов для хранения графических изображений**

Способ кодирования информации с помощью последовательности байт называют форматом. Графический формат - это способ записи графической информации. Все графические форматы подразделяются на растровые и векторные.

Растровый формат характеризуется тем, что все изображение по вертикали и горизонтали разбивается на достаточно мелкие прямоугольники - так называемые элементы изображения, или пикселы (от английского pixel - picture element).

В файле, содержащем растровую графику, хранится информация о цвете каждого пиксела данного изображения. Размер изображения, хранящегося в файле, задается в виде числа пикселов по горизонтали и вертикали.

Наиболее распространенные графические форматы:

1) BMP - (Windows Bitmap) разрабатывался фирмой Microsoft как совместимый со всеми приложениями Windows. В формате BMP можно сохранять черно-белые, серые полутоновые, индексные цветные и цветные изображения системы RGB. Недостаток этих графических форматов: большой объем. Следствие - малая пригодность для Internet-публикаций;

2) GIF - поддерживает до 256 цветов, позволяет задавать один из цветов как прозрачный, дает возможность сохранения с чередованием строк (при просмотре сначала выводится каждая 8-я, затем каждая 4-я и т.д. Это позволяет судить об изображении до его полной загрузки). Способен содержать несколько кадров в одном файле с последующей последовательной демонстрацией (т.н. "анимированный GIF"). Уменьшение размера файла достигается удалением из описания палитры неиспользуемых цветов и построчного сжатия данных (записывается количество точек повторяющегося по горизонтали цвета, а не каждая точка с указанием ее цвета). Для сжатия файла используется высокоэффективный алгоритм Лемпела - Зива - Велча (LZW);

3) JPEG - миллионы цветов и оттенков, палитра не настраиваемая, предназначен для представления сложных фотоизображений. Разновидность progressive JPEG позволяет сохранять изображения с выводом за указанное количество шагов (от 3 до 5 в Photoshop'e) - сначала с маленьким разрешением (плохим качеством), на следующих этапах первичное изображение перерисовывается все более качественной картинкой. Анимация или прозрачный цвет форматом не поддерживаются. Уменьшение размера файла достигается сложным математическим алгоритмом удаления информации - чем заказываемое качество ниже, тем коэффициент сжатия больше, файл меньше. Главное, подобрать максимальное сжатие при минимальной потере качества. Последний идентифицирует и отбрасывает данные, которые человеческий глаз не в состоянии увидеть (незначительные изменения в цвете не различаются человеком, тогда как улавливается даже малейшая разница в интенсивности, поэтому JPEG меньше подходит для обработки черно-белых полутоновых изображений), что приводит к существенному уменьшению размера файла. Наиболее подходящий формат для размещения в Интернете полноцветных изображений. Вероятно, до появления мощных алгоритмов сжатия изображения без потери качества останется ведущим форматом для представления фотографий в Web;

4) PNG - создавался специально для Интернета и благодаря патентной политике Compuserve постепенно вытесняет GIF (см. выше). Позволяет выбирать палитру сохранения - серые полутона, 256 цветов, true color ("истинные цвета"). В зависимости от свойств изображения действительно иногда предпочтительнее GIF'a или JPG'a. Позволяет использовать "прозрачный" цвет, но, в отличие от GIF'a таких цветов может быть до 256. В отличие от GIF сжатие без потери качества производится и по горизонтали, и по вертикали (алгоритм собственный, параметры тоже не настраиваемые).

5) PDF (Portable Document Format) - это пример смешанного формата, предназначенного для хранения текста и графики одновременно. В формате PDF сохраняются данные текстовым редактором Adobe Acrobat. Для сжатия графики применяется метод LZW;

6) PSD - формат графического редактора Adobe Photoshop. Обладает очень большими возможностями. Хранит данные о различных палитрах цветов, о прозрачности, имеет возможность хранения послойных изображений. При этом отличается большим размером.

**3. Графические редакторы, используемые для создания изображений**

**3.1 Программы просмотра графических изображений**

Программа для просмотра изображений - тип программного обеспечения, предназначенного для просмотра мультимедийных данных, в первую очередь изображений.

Программа просмотра изображений и факсов Windows позволяет работать с изображениями, не открывая приложений редактирования изображений. Дважды щелкнув изображение, можно просмотреть его в программе просмотра изображений и факсов Windows, позволяющей делать следующее:

1) прокручивать изображения, находящиеся в папке;

2) увеличивать и уменьшать размер просматриваемого изображения;

3) просматривать изображение в полном размере или в формате, наиболее походящем для размеров окна;

4) управлять файлами изображений, а также печатать, сохранять, удалять и изменять сведения о файлах;

5) просматривать в виде демонстрации слайдов все изображения или отобранную группу изображений;

6) в случае необходимости открывать изображение в программе редактирования (заметьте, что при этом программа просмотра изображений и факсов Windows закрывается);

7) поворачивать изображения вправо и влево на 90 градусов.

**3.2 Редакторы растровой графики**

Paint - простейший редактор, входящий в стандартную поставку операционных систем Microsoft. Он обладает набором простейших функций (кисточка, карандаш, резинка и т.д.), которые позволяют создавать незамысловатые картинки. К сожалению, для обработки графики он практически не пригоден. Photoshop - на сегодняшний день это самый мощный пакет для профессиональной обработки растровой графики[5]. Это целый комплекс, обладающий многочисленными возможностями модификации растрового рисунка, имеющий огромный набор различных фильтров и эффектов, причем есть возможность подключать инструменты независимых производителей.

Пакет предлагает, например, средства для восстановления поврежденных изображений, ретуширования фотографий или создания самых фантастических коллажей, которые только может позволить себе наше воображение. В общем, потенциал этого пакета поистине огромен. Начиная с версии 5.5 в пакет включена программа Adobe ImageReady, предоставляющие огромные возможности по обработке графики под WEB (оптимизация изображений, создание анимированных gif, "разрезание" картинок на более мелкие и т.д.). Еще один не менее известный графический редактор—Corel Draw. Здесь также имеются все необходимые инструменты для обработки графики, разнообразные фильтры, текстуры. Разница лишь в удобстве работы, интерфейсе и скорости наложения фильтров - наложение происходит немного медленнее. редактор предоставляет великолепные возможности для эмуляции реальных инструментов рисования: графит, мел, масло и т.д. Также позволяет имитировать фактуру поверхности материалов, живопись, создавать анимацию. Очень удобен для разработки фоновых рисунков или Web-страниц в стиле живописи.

**3.3 Редакторы векторной графики**

Illustrator - пакет позволяет создавать, обрабатывать и редактировать векторную графику. По своей мощности он эквивалентен растровому редактору Adobe Photoshop: имеет аналогичный интерфейс, позволяет подключать различные фильтры и эффекты, понимает многие графические форматы, даже такие как .cdr (Corel Draw) и .swf (Flash).- безусловно, такой известный графический пакет не мог обойтись без средств для обработки векторной графики. Пакет по своей мощности практически не уступает Adobe Photoshop. Помимо обработки векторной графики, в этом пакете существует обработчик растровой графики (Photo Paint), трассировщик изображений, редактор шрифтов, подготовки текстур, а также огромные коллекции с изображениями (CorelGallery). Streamline - еще один продукт фирмы Adobe, предназначенный для трассировки (перевода) растровой графики в векторную. Это небольшой, но очень полезный и мощный продукт. Особенно полезен, если вы создаете Web-страницы с использованием векторной графики, например, технологии Flash.

**3.4 Понятия цветового пространства. Основные цветовые модели (RGB, CMYK)**

Цветовое пространство представляет собой модель представления цвета, основанную на использовании цветовых координат[6]. Цветовое пространство строится таким образом, чтобы любой цвет был представим точкой, имеющей определённые координаты, причём так, чтобы одному набору координат соответствовал один цвет.

Цветовые пространства описываются набором цветовых координат и правилами построения цветов. К примеру, RGB является трёхмерным цветовым пространством, где каждый цвет описан набором из трёх координат - каждая из них отвечает компоненте цвета в разложении на красный, зелёный и синий цвета. Количество координат задаёт размерность пространства. RGB(аббревиатура английских слов Red, Green, Blue - красный, зелёный, синий) - аддитивная цветовая модель, как правило, описывающая способ синтеза цвета для цветовоспроизведения. В российской традиции иногда обозначается как КЗС.

Цвета получаются путём добавления к черному. Иначе говоря, если цвет экрана, освещённого цветным прожектором, обозначается в RGB как (r1, g1, b1), а цвет того же экрана, освещенного другим прожектором, - (r2, g2, b2), то при освещении двумя прожекторами цвет экрана будет обозначаться как (r1+r2, g1+g2, b1+b2).

Цветовая модель RGB имеет по многим тонам цвета более широкий цветовой охват (может представить более насыщенные цвета), чем типичный охват цветов CMYK, поэтому иногда изображения, замечательно выглядящие в RGB, значительно тускнеют и гаснут в CMYK.

Четырёхцветная автотипия (CMYK: Cyan, Magenta, Yellow, Key color) - субтрактивная схема формирования цвета, используемая прежде всего в полиграфии для стандартной триадной печати. Схема CMYK, как правило, обладает сравнительно небольшим цветовым охватом.

По-русски эти цвета часто называют так: голубой, пурпурный, жёлтый; но профессионалы подразумевают cyan, magenta и yellow (о значении K см. далее). Печать четырьмя красками, соответствующими CMYK, также называют печатью триадными красками.

Так, исторически в разных странах сложилось несколько стандартизованных процессов офсетной печати. Сегодня это американский, европейский и японский стандарты для мелованной и немелованной бумаг[5]. Именно для этих процессов разработаны стандартизованные бумаги и краски. Для них же созданы соответствующие цветовые модели CMYK, которые используются в процессах цветоделения. Однако, многие типографии, в которых работают специалисты с достаточной квалификацией (или способные на время пригласить такого специалиста), нередко создают профиль, описывающий печатный процесс конкретной печатной машины с конкретной бумагой. Этот профиль они предоставляют своим заказчикам.

# **4. Разработка графического редактора**

**4.1 Описание проектного решения**

Целью данной работы является разработка простейшего графического редактора с помощью объектно-ориентированного языка программирования, с использованием его графических функций и методов.

Реализованный графический редактор позволяет создавать собственные с помощью различных возможностей таких как:

1) создание векторных объектов (круга, прямоугольника, линии) и их различные модификации (растягивание) в эллипс, прямоугольник.

2) создание рисунков с помощью линий (карандаша);

3) отменить последнее действие;

4) в любой момент изменить уже нарисованный объект;

Объектно-ориентированный анализ и проектирование нельзя определить ни как проектирование сверху вниз, ни как проектирование снизу вверх; его можно скорее назвать «возвратным проектированием», что подразумевает ступенчатый процесс разработки системы с постепенной модификацией различных, но, тем не менее, согласованных между собой логических и физических представлений о системе в целом.

Когда рисунок создан, можно осуществлять его редактирование, сохранение под новым именем либо в уже созданном файле. Для редактирования разработан ряд инструментов, осуществляющих различные функции:

1) карандаш;

2) рисование прямоугольников;

3) рисование окружностей;

4) рисование линий;

**4.2 Программная реализация**

Программа состоит из двух проектов: Helpers и DrawImage

Helpers является вспомогательным проектом, осуществляющим работу с файлами (сохранение, чтение), восстановление прежнего состояние окна программы.

Это проект состоит из следующих компонентов:

* 1. DocHelper. Осуществляет работу с файлами. Рисунки можно сохранять в файл и считывать оттуда. Для сохранения предоставляются два способа: Save и SaveAs. Первый пересохраняет уже открытый файл изображения. Второй способ позволяет создать новый файл.
  2. DragDropHelper. Упрощает пользователя работу с программой. Для открытия файла изображения достаточно просто перетащить файл в рабочую область.
  3. MruFilesHelper. MRU—most recently used files. Позволяет пользователю осуществить быстрый доступ к недавно открытым файлам с помощью меню программы. Для этого необходимо выбрать пункт меню File->Recent files.
  4. PersistWindowState. Сохраняет положения окна программы и его параметры—высоту, ширину. Во время следующего запуска эти параметры восстанавливаются.

Проект Helpers просто дополнять и модифицировать. В частности, не представляет никаких затруднений добавить новые компоненты.

DrawImage является основным проектом программы. В нем операции выполняются непосредственно с изображением.

Command является базовым классом для команд Undo (отменить действие) и Redo (восстановить отмененное действие). В нем определены методы, имеющие одноименные названия. Реализация этих методов происходит в дочерних классах: CommandAdd, CommandChangeState, CommandDelete, CommandDeleteAll. Они предназначен для добавления, изменение, удаление объекта и всего рисунка соответственно.

Похоже реализованы инструменты для рисования. Есть базовый класс Tool, в которым определены обработчики событий движения, нажатия, отпускания мыши.

Наследником класса Tool является ToolObject, который, в то же время, базовый класс для всех инструментов, создающих графические объекты. В нем реализована метод добавления новых объектов, а также переопределен метод обработки события мыши. Классы ToolLine, ToolPolygon, ToolRectangle, ToolEllipse наследуют функциональность ToolObject, определяют собственную реакцию на события мыши и позволяют в дальнейшем нарисовать линию, ломаную линию, прямоугольник и круг соответственно. Так же следует упомянуть о классе ToolPointer. Этот класс определяет курсор. Очень важный и полезный инструмент, позволяющий менять форму, перетаскивать и выделять фигуры

Сейчас будут описаны классы, отвечающие непосредственно за отрисовку объектов. Здесь базовым является класс DrawObject. Он обладает такими параметрами, как ширина и цвет линии. DrawLine, DrawPolygon, DrawRectangle, DrawEllipse расширяют функциональность DrawObject для отрисовки необходимых объектов.

Все отрисованные объекты находятся в пределах рабочей области. Рабочую область представляет класс GraphicsList. Так же в нем определены все методы, необходимые для отрисовки объектов:

public void Draw(Graphics g) –рисует объект, переданный через параметр

public bool Clear() –очищает рабочую область

public void Insert(int index, DrawObject obj)—вставляет объект в определенное место

public void Replace(int index, DrawObject obj)—заменяет один объект другим

public void SelectAll()—выделяет все объекты в рабочей области

public bool DeleteSelection()—удаляет выделенные объекты

public bool MoveSelectionToFront()—перемещает выделенные объекты на передний план

Интерфейс содержит набор кнопок и меню. При нажатии на какую-либо кнопку из предложенных обрабатываются события OnMouseDown, OnMouseMove, OnMouseUp. При нажатии кнопки мыши(OnMouseDown) начинается прорисовка графического объекта, при движении мыши(OnMouseMove) объект прорисовывается в соответствии с изменением координат указателя мыши, когда кнопку отпускают(OnMouseUp) прорисовка заканчивается.

Узнать предназначение кнопок пользователю помогут всплывающие подсказки.

**4.3 Руководство пользователя**

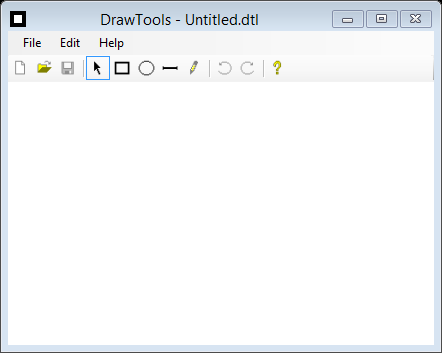


Рис. 1. Интерфейс программы

Программа поставляется в виде исполняемого ЕХЕ файла, запускается под управлением ОС Windows 7, 8, 8,1, не требует большого количества системных ресурсов. Имеет удобный пользовательский интерфейс и достаточно широкие возможности для рисования, просмотра и редактирования изображений.

Главное окно редактора носит имя редактируемого файла.

Меню состоит из трех пунктов:

* + 1. File
  1. New (создать новое изображение)
  2. Open (открыть файл)
  3. Save (пересохранить открытый файл)
  4. SaveAs (создать новый файл)
  5. Recent files (список последних файлов)
  6. Exit (выход из программы)
     1. Edit

2.1) Select All (выделить все объекты)

2.2) Unselect All (снять выделение)

2.3) Delete All (удалить все)

2.4) Delete (удалить выделенное)

2.5) Move to front (переместить на передний план)

2.6) Move to back (переместить на задний план)

3) Help

3.1) About (информация о программе)

Основные элементы панели инструментов выполняют следующие действия:

1. карандаш - рисование произвольной линии

2) окружность - рисование различных окружностей и эллипсов;

3) прямоугольник - рисование квадратов и прямоугольников;

4) линия – рисование прямой линии

**5. Заключение**

Данная курсовая работа потребовала изучения большого количества литературного материала, в результате чего были изучены основные типы графических редакторов, их достоинства и недостатки. Познакомившись с наиболее популярными в настоящее время программами для просмотра графических изображений, была достигнута поставленная цель курсовой работы: создан графический редактор.

В ходе выполнения курсового проекта закреплены понятие объектно-ориентированного программирования, объектно-ориентированного анализа и объектно-ориентированного подхода, навыки по программированию в объектно-ориентированной среде.

Интерфейс созданной программы удобен, прост, наглядно отображает ее возможности. Редактор снабжен системой всплывающих подсказок. Главное меню редактора содержит команды работы с файлами, команды занесения рисунка в буфер и вывода рисунка из буфера.

Тестирование подтвердило, что программа корректно выполняет обработку данных и демонстрацию результатов.

Всё это свидетельствует о работоспособности программы и позволяет сделать вывод о ее пригодности для создания и редактирования графических изображений.

**6. Список использованной литературы**

1. Петров М.Н. Компьютерная графика. - СПб.: Питер, 2007. - 478с.

2. Степанов А.Н. Информатика. - М.: Солон-Пресс, 2010. - 554с.

3. Симонович С.В. Компьютерная графика. - С-Пб: АСТ-Пресс, 2005. - 480с.

4. Гурский Ю.Л. Photoshop. - М: Корона-Принт, 2004. - 552с.

5. Мельниченко В.В., Легейда В.В. Настоящий самоучитель компьютерной графики. - М.: Век +, 2005. - 560с.

**7. Приложение**

Программа написанная на языке C#. Реализация основных функций программы.

Отрисовка ломаной линии

public override void Draw(Graphics g)

{

int x1 = 0, y1 = 0; // previous point

int x2, y2; // current point

g.SmoothingMode = SmoothingMode.AntiAlias;

var pen = new Pen(Color, PenWidth);

PointEnumerator enumerator = pointArray.GetEnumerator();

if (enumerator.MoveNext())

{

x1 = enumerator.Current.X;

y1 = enumerator.Current.Y;

}

while (enumerator.MoveNext())

{

x2 = enumerator.Current.X;

y2 = enumerator.Current.Y;

g.DrawLine(pen, x1, y1, x2, y2);

x1 = x2;

y1 = y2;

}

pen.Dispose();

}

Перемещение выделенных объектов на передний план

public bool MoveSelectionToFront()

{

var tempList = new DrawList();

int n = graphicsList.Count;

// Read source list in reverse order, add every selected item

// to temporary list and remove it from source list

for (int i = n - 1; i >= 0; i--)

{

if ((graphicsList[i]).Selected)

{

tempList.Add(graphicsList[i]);

graphicsList.RemoveAt(i);

}

}

// Read temporary list in direct order and insert every item

// to the beginning of the source list

n = tempList.Count;

for (int i = 0; i < n; i++)

{

graphicsList.Insert(0, tempList[i]);

}

return (n > 0);

}

Обработка события нажатия клавиши для курсора

public override void OnMouseDown(DrawArea drawArea, MouseEventArgs e)

{

commandChangeState = null;

wasMove = false;

selectMode = SelectionMode.None;

var point = new Point(e.X, e.Y);

// Test for resizing (only if control is selected, cursor is on the handle)

foreach (DrawObject o in drawArea.GraphicsList.Selection)

{

int handleNumber = o.HitTest(point);

if (handleNumber > 0)

{

selectMode = SelectionMode.Size;

// keep resized object in class member

resizedObject = o;

resizedObjectHandle = handleNumber;

// Since we want to resize only one object, unselect all other objects

drawArea.GraphicsList.UnselectAll();

o.Selected = true;

commandChangeState = new CommandChangeState(drawArea.GraphicsList);

break;

}

}

// Test for move (cursor is on the object)

if ( selectMode == SelectionMode.None )

{

int n1 = drawArea.GraphicsList.Count;

DrawObject o = null;

for ( int i = 0; i < n1; i++ )

{

if ( drawArea.GraphicsList[i].HitTest(point) == 0 )

{

o = drawArea.GraphicsList[i];

break;

}

}

if ( o != null )

{

selectMode = SelectionMode.Move;

// Unselect all if Ctrl is not pressed and clicked object is not selected yet

if ( ( Control.ModifierKeys & Keys.Control ) == 0 && !o.Selected )

drawArea.GraphicsList.UnselectAll();

// Select clicked object

o.Selected = true;

commandChangeState = new CommandChangeState(drawArea.GraphicsList);

drawArea.Cursor = Cursors.SizeAll;

}

}

// Net selection

if ( selectMode == SelectionMode.None )

{

// click on background

if ( ( Control.ModifierKeys & Keys.Control ) == 0 )

drawArea.GraphicsList.UnselectAll();

selectMode = SelectionMode.NetSelection;

}

lastPoint.X = e.X;

lastPoint.Y = e.Y;

startPoint.X = e.X;

startPoint.Y = e.Y;

drawArea.Capture = true;

drawArea.Refresh();

if ( selectMode == SelectionMode.NetSelection )

{

// Draw selection rectangle in initial position

ControlPaint.DrawReversibleFrame(

drawArea.RectangleToScreen(DrawRectangle.GetNormalizedRectangle(startPoint, lastPoint)),

Color.Black,

FrameStyle.Dashed);

}

}