

# РАЗРАБОТКА ВЕКТОРНОГО ГРАФИЧЕСКОГО РЕДАКТОРА

И.Е. Мишутина, М.А. Кудрина

Самарский государственный аэрокосмический университет им. академика С.П.Королева  
(Национальный исследовательский университет)

Проведены проектирование и разработка векторного графического редактора, который реализует функции работы с графическими примитивами, включая задания их свойств (тип границы, цвет границы, тип фигуры, цвет заливки), а также действий над ними (группировка, порядок, копирование, вставка, отражение). В качестве алгоритмов растризации линии и эллипса выбран алгоритм Ву Сяолияня.

## **Введение**

С развитием компьютерной техники и технологий появилось множество способов построения графических объектов. Графический объект – это либо само графическое изображение или его часть. В зависимости от видов компьютерной графики под этим термином понимаются, как пиксели или спрайты (в растровой графике), так и векторные объекты, такие как круг, квадрат, линия, кривая и т.д. (в векторной графике). Существует специальная область информатики, изучающая методы и средства создания и обработки изображений с помощью программно-аппаратных вычислительных комплексов, – компьютерная графика. Компьютерная графика представляет собой одну из современных технологий создания различных изображений с помощью аппаратных и программных средств компьютера [1]. В зависимости от способа формирования изображений компьютерную графику принято подразделять на растровую и векторную. Говоря о графических редакторах, также необходимо отметить, что они, как и цифровые изображения, подразделяются на векторные и растровые.

В рамках данной работы был разработан векторный графический редактор, который позволяет пользователю выбирать графические примитивы, совершать над ними различные операции.

## **Постановка задачи**

Необходимо было разработать автоматизированную систему «Векторный графический редактор», которая должна реализовывать следующие функции: настройка параметров (выбор цвета, выбор толщины линии, выбор типа линии), рисование графических примитивов, масштабирование объектов, группировка объектов, повороты объектов, отражение одного типа объекта и группы объектов, заливка объекта цветом, градиентная заливка объекта, дополнительные функции (копирование, вставка, отмена последнего действия, возврат действия после отмены, добавление текста, добавление изображения), сохранение изображения в файл, загрузка изображения из файла, выдача справочной информации о системе.

## **Разработка проекта**

Диаграмма граничных классов проектируемой системы представлена на рисунке 1. На этой диаграмме определены основные экранные формы приложения, которые будут отображаться во время работы пользователя с векторным графическим редактором, а также основные методы, выполняемые для каждой из приведенных форм. На диаграмме граничных классов разрабатываемого векторного графического редактора представлены следующие классы:

- «Главная форма»;
- «Панель инструментов»;
- «Форма редактирования текста»;
- «Форма справки».

Класс «Главная форма» служит для отображения панели инструментов и рабочей области.

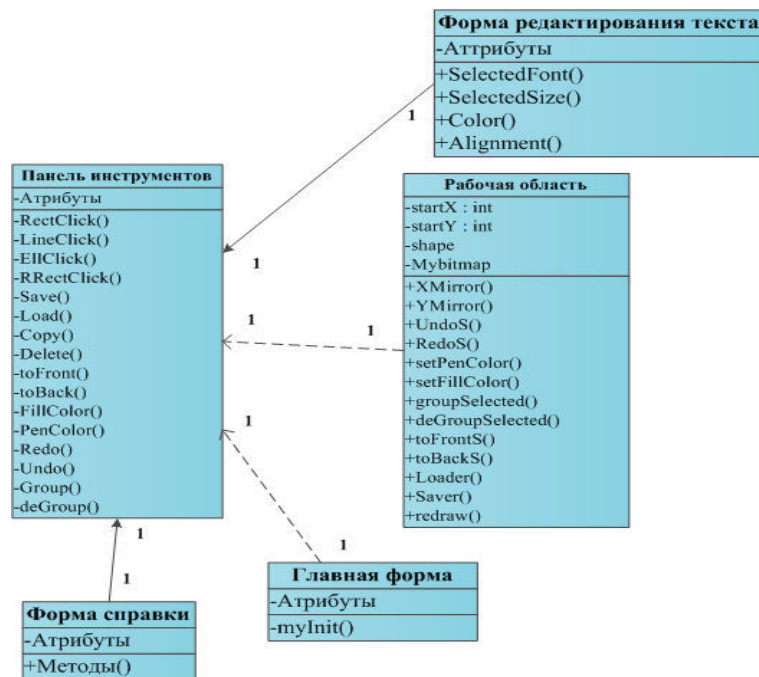


Рисунок 1 – Диаграмма граничных классов векторного графического редактора

Класс «Панель инструментов» предназначен для управления объектами. Содержит кнопки выбора фигур, их параметров и действий над ними, а так же меню сохранения, правки, справки. Класс «Форма редактирования текста» используется для редактирования текста. Включает в себя возможность выбора шрифта, его размера, цвет, а также выравнивания текста. Класс «Форма справки» предназначен для выдачи справочной информации о системе.

Диаграмма сущностных классов векторного графического редактора представлена на рисунке 2.

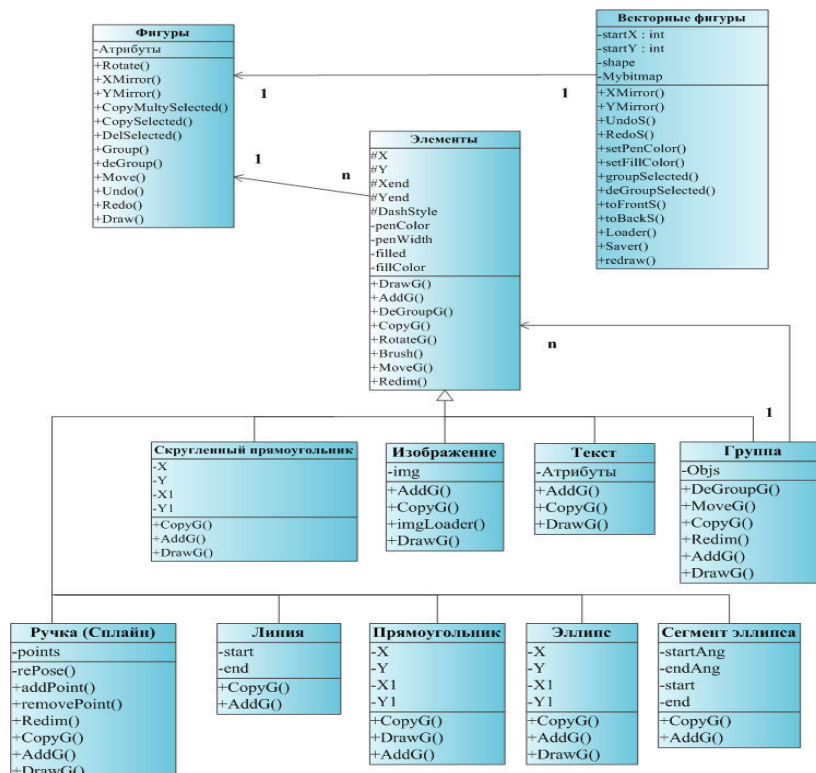


Рисунок 2 – Диаграмма сущностных классов векторного графического редактора

Здесь указаны основные сущности, определяющие предметную область векторного графического редактора и имеющие определенные атрибуты.

Класс «Векторные фигуры» хранит информацию о том, каким образом происходит отображение объектов и осуществления различных действий над ними.

Класс «Фигуры» хранит информацию об объектах, и их свойства через коллекцию «Элементы».

Класс «Элементы» хранит информацию о базовых свойствах и методах, наследуемых классами объектов: прямоугольник, линия, эллипс, скругленный прямоугольник, сегмент эллипса, ручка (сплайн), изображение, текст и группа.

### **Реализация алгоритмов обработки данных**

В системе реализованы следующие алгоритмы: рисования линии, эллипса, окружности, квадрата, скругленного квадрата, полигона, закрашки, градиентной закрашки, группировки, порядка наложения фигур, сохранения изображения или отдельных объектов в файл и загрузка их из файла, добавления текста RTF.

Было принято решение реализовать алгоритм Ву Сяолияня для рисования сглаженных линий. Алгоритм сочетает высококачественное устранение ступенчатости и скорость, близкую к скорости алгоритма Брезенхема без сглаживания. Горизонтальные и вертикальные линии не требуют никакого сглаживания, поэтому их рисование выполняется отдельно. Для остальных линий алгоритм Ву проходит их вдоль основной оси, подбирая координаты по неосновной оси аналогично алгоритму Брезенхема. Он отличается тем, что на каждом шаге ведётся расчёт для двух ближайших к прямой пикселей, и они закрашиваются с разной интенсивностью, в зависимости от удаленности. Например, если основной осью является X, то рассматриваются точки с координатами (x, y) и (x, y+1). В зависимости от величины ошибки, которая показывает как далеко ушли пиксели от идеальной линии по неосновной оси, распределяется интенсивность между этими двумя точками. Чем больше удалена точка от идеальной линии, тем меньше ее интенсивность. Значения интенсивности двух пикселей всегда дают в сумме единицу, то есть это интенсивность одного пикселя, в точности попавшего на идеальную линию. Такое распределение придаст линии одинаковую интенсивность на всем ее протяжении, создавая при этом иллюзию, что точки расположены вдоль линии не по две, а по одной [2].

Для растеризации эллипса также был использован алгоритм Ву (модификация метода для эллипса). Этот алгоритм позволяет нарисовать эллипс с использованием антиалиасинга, так что фигура не будет иметь резких краев. Также следует отметить, что этот алгоритм принимает не целые размеры эллипса, а вещественные. При изменении координат эллипса, нарисованный алгоритмом Брезенхема, перемещается резко, скачками. Эллипс по алгоритму Ву может перемещаться непрерывно. За счет этого можно обеспечивать плавную анимацию при рисовании движущихся изображений. Принцип работы алгоритма идентичен принципу работы алгоритма Ву для отрезка [3]. Для точки окружности выбираются два ближайших пикселя. Между ними пропорционально распределяется прозрачность цвета. При оптимизации данного алгоритма важно понимать, что более яркие пиксели, лежащие на внутренней части линии, придают вогнутость линии, а лежащие на внешней стороне – выпуклость.

Одним из самых простых алгоритмов закрашки является рекурсивный алгоритм с «затравкой». Однако, этот алгоритм приводит к большим затратам стековой памяти, поэтому его целесообразно использовать только для закрашки небольших фигур (не более 1 тыс. пикселей) [4].

### **Разработка и описание интерфейса пользователя**

При запуске программы перед пользователем появляется главная форма, которая предоставляет основные функции по управлению системой, приведенная на рисунке 3.

Она содержит панель инструментов, содержащую кнопки для выбора фигур, задания их свойств (тип границы, цвет границы, тип фигуры, цвет заливки), а также

действий над ними (группировка, порядок, копирование, вставка), меню и поле рисования. Слева от поля рисования находится список доступных фигур.

Атрибуты фигуры отображаются в специальном окне, которое также расположено на панели инструментов. С помощью меню программы и панели инструментов пользователь может произвести все действия, необходимые для создания изображения. Существует возможность выбора типа фигуры: с заливкой и без заливки, которая реализована с помощью соответствующей кнопки. При выборе типа фигуры с заливкой необходимо выбрать ее цвет. Также можно выбрать цвет и толщину линии, нажав кнопку «Граница». Тип линии (сплошная, штрих, штрих-пунктирная и т.д.) можно выбрать в окне свойств. Если пользователя не устраивает полученная фигура, он может удалить ее кнопкой «Удалить» или, наоборот, сделать копию фигуры с помощью соответствующей кнопки. Для удобства расположения объектов можно добавить сетку, нажав кнопку на панели инструментов. Ее размер также регулируется. Во вкладке меню «Файл» пользователь может сохранить изображение в файл, загрузить его из файла, помимо этого можно сохранять и загружать только выделенные объекты. В меню есть значок лупы, который позволяет выбрать масштаб рабочей области. Поменять масштаб также можно с помощью ползунка, расположенного под панелью инструментов.

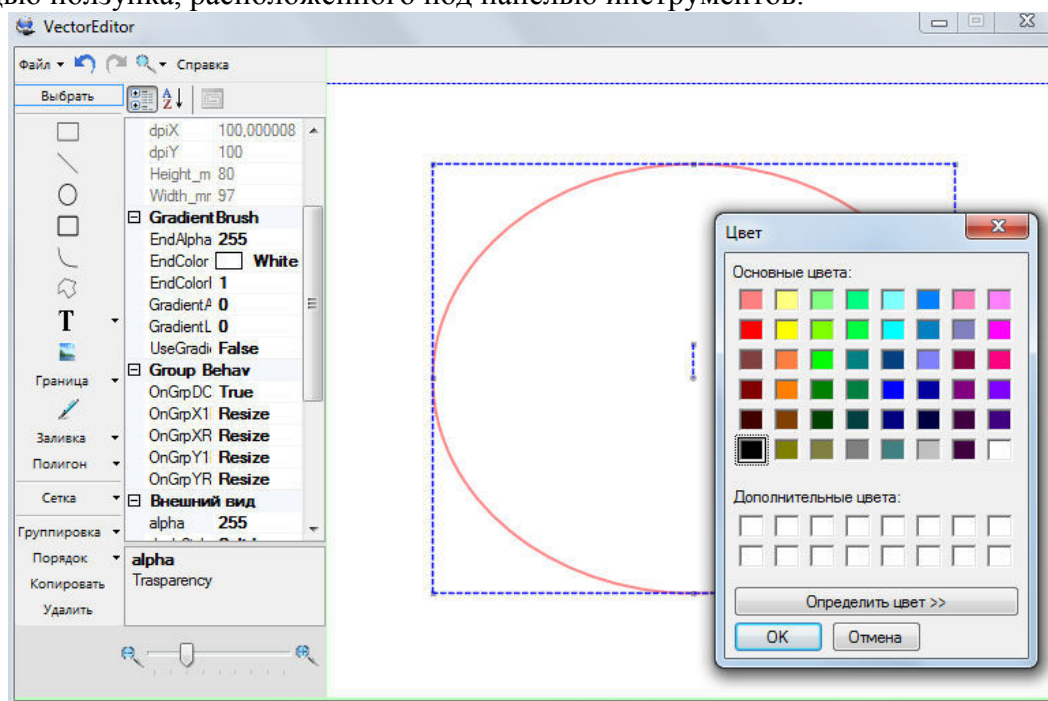


Рисунок 3 – Главная форма программы

## Заключение

В ходе проектирования данной работы был разработан векторный графический редактор. Разработанная система отвечает всем требованиям задания, учитывает особенности предметной области и реализует функции работы с графическими примитивами, включая задания их свойств (тип границы, цвет границы, тип фигуры, цвет заливки), а также действий над ними (группировка, порядок, копирование, вставка, отражение).

## Литература

1. Симонович, С. В. Информатика. Базовый курс: Учебник для вузов. 3-е изд. Стандарт третьего поколения. – СПб.: Питер, 2011 – 640с.:ил.
2. Басманов, Д. Брезенхам и У на страже диагоналей. [Электронный ресурс]/ Habrahabr. <http://habrahabr.ru/post/185086/> (Дата обращения 13.05.15)
3. Алгоритм Ву для растеризации эллипса с антиалиасингом. [Электронный ресурс]/ VinGrad. <http://base.vingrad.ru/view/2980-Algorithm-Vu-dlya-rasterizatsii-ellipsa-s-antialiasingom> (Дата обращения 17.05.15)
4. Кудрина, М.А. Компьютерная графика: учеб. / М.А. Кудрина, К.Е. Климентьев. – Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2013. – 138 с.