**СОПРОВОДИТЕЛЬНАЯ ДОКУМЕНТАЦИЯ**

**Лабораторная работа 2**

**«Базовые растровые алгоритмы»**

**Цель работы:**

Цель лабораторной работы заключается в разработке приложения для визуализации базовых растровых алгоритмов. Алгоритмы включают в себя:

- Цифровой дифференциальный анализатор (ЦДА)

- Пошаговый алгоритм

- Алгоритм Брезенхема для окружности

- Алгоритм Брезенхема для линии

**Использованные средства разработки**:

Для разработки приложения был выбран язык программирования Python, а в качестве библиотеки для визуализации результатов - Matplotlib. Также были использованы элементы интерфейса Matplotlib Widgets.

**Реализация алгоритмов:**

1. ЦДА (Цифровой дифференциальный анализатор):

def draw\_dda\_line(x1, y1, x2, y2, color="b."):

Данный алгоритм использует алгоритм ЦДА для отрисовки отрезка между двумя точками. Интерфейс предоставляет слайдеры для ввода координат точек и кнопку "Обновить" для визуализации.

1. Пошаговый алгоритм:

def draw\_steps\_line(x, k, b, steps, color="b."):

Пошаговый алгоритм рисует линию с использованием уравнения прямой. Интерфейс аналогичен предыдущему с добавлением соответствующих слайдеров.

1. Брезенхем для окружности:

def draw\_b\_circle(x1, y1, x2, y2, color="b."):

Этот алгоритм использует алгоритм Брезенхема для рисования окружности. Слайдеры используются для ввода координат центра и конечной точки радиуса.

1. Брезенхем для линии:

def draw\_b\_line(x1, y1, x2, y2, color="b."):

Алгоритм Брезенхема для отрисовки линии между двумя точками. Интерфейс аналогичен предыдущим.

**Интерфейс приложения:**

Интерфейс приложения включает слайдеры для ввода координат точек и параметров алгоритмов, а также кнопки "Обновить" для визуализации изменений. Для каждого графического алгоритма, такого как ЦДА, пошаговый, Брезенхем для окружности и линии, а также Кастл-Питвей, предусмотрено отображение времени выполнения алгоритма под соответствующим графиком. Это время измеряется с момента начала выполнения алгоритма до его завершения и выводится с точностью до четырех знаков после запятой. Благодаря нашей библиотеке, предоставляется возможность взаимодействия с графиками. Мы можем легко увеличивать изображение, менять его расположение и получать информацию о координатах, когда наводите мышью на график.

**Возможности взаимодействия:**

ЦДА (Цифровой дифференциальный анализатор):

- Ввод координат точек с помощью слайдеров.

- Обновление графика по нажатию кнопки "Обновить".

Пошаговый алгоритм:

- Ввод параметров уравнения прямой (k, b) и координаты точки с помощью слайдеров.

- Обновление графика по нажатию кнопки "Обновить".

Брезенхем для окружности:

- Ввод координат центра и конечной точки радиуса с помощью слайдеров.

- Обновление графика по нажатию кнопки "Обновить".

Брезенхем для линии:

- Ввод координат точек с помощью слайдеров.

- Обновление графика по нажатию кнопки "Обновить".

**Заключение:**

В ходе выполнения лабораторной работы было создано приложение на языке программирования Python, предназначенное для демонстрации базовых алгоритмов растеризации и сглаживания. В рамках данного приложения реализованы алгоритмы цифрового дифференциального анализатора (ЦДА), пошагового алгоритма, алгоритма Брезенхема для линий, а также алгоритма Брезенхема для окружностей и анализ времени выполнения каждого алгоритма.

Программа обладает удобным графическим интерфейсом, позволяющим в режиме реального времени изменять параметры линий и окружностей, а также просматривать результаты работы каждого алгоритма. Для этого использованы элементы управления библиотеки Matplotlib, такие как ползунки и кнопки, что делает взаимодействие с программой интуитивно понятным.