**Лабораторная работа №3**

**Цель работы:**

Закрепить теоретический материал и практически освоить основные

возможности по использованию базовых алгоритмов растеризации отрезков и кривыx:

• пошаговый алгоритмов

• алгоритм ЦДА

• алгоритм Брезенхема

• алгоритм Брезенхема(окружность)

**Задачи работы:**

• Создать удобный и понятный пользовательский интерфейс

• Реализовать пошаговый алгоритм

• Реализовать алгоритм ЦДА

• Реализовать алгоритм Брезенхема

• Реализовать алгоритм Брезенхема для окружности

* Анализ результатов работы алгоритма

**Использованные средства разработки:**

Язык Python 3.12 и библиотека tkinter

**Временные характеристики:**

Были введены следующие входные данные для отрезка:

x1 = -150 y2 = 150

y1 = 50 x2 = -50

Для окружности :

x0 = -150

y0 = 150

R = 50

|  |  |  |
| --- | --- | --- |
| Алгоритм | Время (с поясняющей информацией) | Время (без поясняющей информации) |
| Пошаговый алгоритм | 15 мс | 15 мс |
| ЦДА | 10 мс | 10 мс |
| Алгоритм Брезенхема | 3 мс | 3 мс |
| Алг. Брезенхема для окружности | 15 мс | 15 мс |

Разницы между алгоритмом Брезенхема и пошаговым алгоритмом крайне мало, хотя алгоритм Брезенхема не использует дробную арифметику. Можно

предположить, что это из-за эмуляции отрисовки пикселей на экране, а их число в этих алгоритмах сопоставимое. То есть отрисовка пикселей занимает

большую часть времени, так как реализована не аппаратно.

**Вывод:**

В ходе выполнения данной работы я:

• создал приложение, позволяющее проводить растеризацию отрезков и

кривых базовыми алгоритмами

• закрепил полученные лекционные знания по различным алгоритмам

растеризации

• получил дополнительный опыт по проектировке приложений

• углубил знания Python

• получил дополнительный опыт работы с системой контроля версий Git