Лабораторная работа №3

Цель работы:

Закрепить теоретический материал и практически освоить базовые алгоритмы растеризации отрезков и кривых:

• пошаговый алгоритмов

• алгоритм Брезенхема

Задачи работы:

• Создать приложение для отображения растеризованного отрезка на экране, для отображения пояснительной информации по ходу алгоритма на экране

• Создать удобный и понятный пользовательский интерфейс

• Реализовать пошаговый алгоритм

• Реализовать алгоритм Брезенхема

Использованные средства разработки:

• Фреймворк Qt и язык C++

Ход работы:

1. Создание PlotArea для отображения растеризованного отрезка на экране с координатной сетки и изменения масштаба.

2. Создание LogWidget для отображения поясняющей информации в ходе алгоритма.

3. Проектировка и создание удобного пользовательского интерфейса с возможностью выбора алгоритма, изменением масштаба, введением координат исходного отрезка

4. Реализация пошагового алгоритма в виде метода NaiveLine

5. Реализация алгоритма Брезенхема в виде метода BresenhamLine

6. Добавление поясняющих сообщений в ходе каждого алгоритма

7. Добавление поддержки измерения прошедшего времени для каждого алгоритма

8. Тестирование корректности работы алгоритмов и корректного отображения на сетке

Временные характеристики:

Были введены наибольшие поддерживаемые входные данные для отрезка:

x1 = -100 y1 = -75

y2 = 100 x2 = 75

Пошаговый алгоритм (Без отрисовки) – 3455 мс

Алгоритм Брезенхема (Без отрисовки) – 3295мс

Пошаговый алгоритм (С отрисовкой) – 4492 мс

Алгоритм Брезенхема (С отрисовкой) – 4300 мс

Разница между алгоритмом Брезенхема и пошаговым алгоритмом почти не заметна.

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

• получил дополнительный опыт работы с системой контроля версий Git