# **Отчёт по лабораторной работе №3 «Базовые растровые алгоритмы»**

## **Бань Мария**

**Цель работы**

Закрепление теоретического материала и практическое освоение основных возможностей по использованию базовых алгоритмов растеризации отрезков и кривых:

• пошаговый алгоритм

• алгоритм ЦДА

• алгоритм Брезенхема

• алгоритм Брезенхема для окружности

**Использованные средства разработки**

• JavaScript + HTML + CSS

**Задачи работы**

• Создать поле для отображения растеризованного отрезка на экране

• Создать поле для отображения пояснительной информации по ходу алгоритма на экране

• Создать удобный и понятный пользовательский интерфейс

• Реализовать пошаговый алгоритм

• Реализовать алгоритм ЦДА

• Реализовать алгоритм Брезенхема

• Реализовать алгоритм Брезенхема для окружности

**Временные характеристики**

Были введены наибольшие поддерживаемые входные данные для отрезка:

x0 = -300, y0 = -200

x1 = 300, y1 = 200

Для окружности:

x0 = 0, y0 = 0, r = 300

|  |  |  |
| --- | --- | --- |
|  | С поясняющей информацией | Без поясняющей информации |
| Пошаговый алгоритм | 3470 мс | 350 мс |
| Алгоритм ЦДА | 2100 мс | 350 мс |
| Алгоритм Брезенхема | 2640 мс | 350 мс |
| Алгоритм Брезенхема для окружности | 1400 мс | 550 мс |

Можно заметить, что вывод пояснительной информации занимает от 5 до 20 % времени работы программы. Разница в производительности между алгоритмом Брезенхема и пошаговым алгоритмом практически незаметна, несмотря на то что алгоритм Брезенхема не использует дробную арифметику. Это можно объяснить тем, что отрисовка пикселей на экране эмулируется, и количество пикселей, обрабатываемых в обоих алгоритмах, примерно одинаково. Таким образом, основное время затрачивается на отрисовку пикселей, что связано с тем, что эта операция не выполняется аппаратно.

**Вывод**

В ходе выполнения данной работы я:

• создал веб-приложение, позволяющее проводить растеризацию отрезков и кривых базовыми алгоритмами

• закрепил полученные лекционные знания по различным алгоритмам растеризации