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 функцию для отображения растеризованного отрезка на

экране

• Создать удобный и понятный пользовательский интерфейс

• Реализовать пошаговый алгоритм

• Реализовать алгоритм ЦДА

• Реализовать алгоритм Брезенхема

• Реализовать алгоритм Брезенхема для окружности

Использованные средства разработки:

Java

Ход работы:

1. Создание интерфейса RasterizationApp и функции draw\_shape и drawGrid для отображения

растеризованного отрезка на экране с поддержкой координатной сетки и

изменения масштаба.

2. Проектировка и создание удобного пользовательского интерфейса с

возможностью выбора алгоритма, введением координат исходного отрезка.

3. Реализация пошагового алгоритма в виде функций drawStepByStep,

stepByStepLineXBased и stepByStepLineYBased

4. Реализация алгоритма ЦДА в виде метода drawDDA

5. Реализация алгоритма Брезенхема в виде метода drawBresenham

6. Реализация алгоритма Брезенхема для окружности в виде метода

drawCircleBresenham и addCirclePixels

7. Добавление поддержки измерения прошедшего времени для

каждого метода

Временные характеристики:

Были введены большие входные данные для отрезка (хоть на экране они не

видны, алгоритмы всё равно высчитываются, а скорость их печати не

учитывается):

x0 = -1000 y0 = -1000

x1 = 1500 y1 = 1500

Для окружности:

x0 = -1000 y0 = -1000 R = abs(x1 – x0) = 2500

Пошаговый алгоритм 0.862 мс

Алгоритм ЦДА 0.534 мс

Алгоритм Брезенхема 0.412 мс

Алгоритм Брезенхема для

окружности

1.18 мс

Пошаговый алгоритм работает медленнее всего, так как при его работе на

каждой итерации идут операции умножения с вещественными числами, в то

время как в других алгоритмах, не используются операции умножения. В ЦДА

алгоритме идёт прибавление вещественных чисел и их округление. В

алгоритме Брезенхема идут операции сложения и вычитания с целыми

числами, что и обуславливает его быструю работу по сравнению с другими

алгоритмами.

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 языка Java