**Отчет по лабораторной работе 3**

**Введение**

Цель лабораторной работы — изучить и реализовать основные алгоритмы компьютерной графики, применяемые для рисования линий, окружностей и антиалиасинга. В данной работе реализованы следующие алгоритмы: **Step-by-step**, **DDA**, **Брезенхем (линия и окружность)**, и **Wu (антиалиасинг линии)**. Пользовательский интерфейс создан с использованием **Java Swing**, что позволяет вводить данные и наблюдать результат работы алгоритмов в графической форме.

**Описание реализованных алгоритмов**

**1. Step-by-step (Пошаговый алгоритм)**

Этот алгоритм заключается в пошаговом вычислении промежуточных точек линии.

**Принцип работы**:

Разница координат по x и y используется для вычисления количества шагов.

На каждом шаге значения координат увеличиваются на постоянный шаг, пропорциональный наклону линии.

**Преимущества**:

Простота реализации.

**Недостатки**:

Производительность низкая из-за необходимости работать с дробными числами.

Подвержен ошибкам округления.

**2. DDA (Digital Differential Analyzer)**

DDA — это усовершенствованный пошаговый алгоритм, который автоматически вычисляет промежуточные точки для рисования линии.

**Принцип работы**:

Рассчитывается наклон линии (dx/dy).

Координаты точек обновляются по принципу инкремента.

**Время выполнения**: O(steps), где steps — длина линии в пикселях.

**Преимущества**:

Равномерное распределение пикселей на линии.

Меньше вычислений, чем в Step-by-step.

**Недостатки**:

Используются дробные числа, что может быть менее эффективно на старых процессорах.

**3. Брезенхем (Bresenham) для линии**

Алгоритм Брезенхема для линии — один из самых известных алгоритмов. Он опирается на целочисленную арифметику, что делает его быстрым.

**Принцип работы**:

Вычисляется ошибка смещения от идеальной линии (отличие реальной координаты от целочисленной).

Ошибка корректируется путём добавления или вычитания dx и dy.

**Время выполнения**: O(n), где n — длина линии.

**Преимущества**:

Исключает необходимость работать с дробными числами.

Высокая скорость и простота реализации.

**Недостатки**:

Может не подходить для сложных сценариев, где нужна высокая точность отображения.

**4. Брезенхем (Bresenham) для окружности**

Алгоритм Брезенхема для окружности основан на симметрии. Он строит пиксели только для одной восьмой части окружности, затем отображает их зеркально.

**Принцип работы**:

Рассчитывается ошибка по аналогии с алгоритмом для линий.

Координаты пикселей выбираются таким образом, чтобы минимизировать отклонение от идеальной окружности.

**Время выполнения**: O(radius), где radius — радиус окружности.

**Преимущества**:

Высокая скорость благодаря использованию симметрии.

Целочисленные вычисления.

**Недостатки**:

Подходит только для рисования окружностей.

**5. Wu (Антиалиасинг линии)**

Алгоритм Wu предназначен для сглаживания линий путём использования градиента цвета.

**Принцип работы**:

Рассчитывается наклон линии.

Пикселям вдоль линии присваиваются частичные значения интенсивности цвета, что создаёт эффект сглаживания.

**Время выполнения**: O(n), где n — длина линии.

**Преимущества**:

Высокая визуальная точность.

Линия выглядит сглаженной, без "зубчатости".

**Недостатки**:

Более сложная реализация.

Необходимость работы с полупрозрачными пикселями.

**Сравнение времени выполнения алгоритмов**

| **Алгоритм** | **Сложность** | **Особенности** |
| --- | --- | --- |
| Step-by-step | O(n) | Низкая производительность. Дробные числа. |
| DDA | O(n) | Улучшение Step-by-step. |
| Bresenham (линия) | O(n) | Высокая скорость. Целочисленные вычисления. |
| Bresenham (окружность) | O(radius) | Эффективен для окружностей. |
| Wu | O(n) | Эффект сглаживания линий. |

**Выводы**

1. **Bresenham для линии** — наиболее эффективный алгоритм для рисования прямых линий на сетке пикселей благодаря использованию целочисленной арифметики.
2. **Bresenham для окружности** идеально подходит для окружностей, благодаря учёту симметрии и оптимизации.
3. **Wu** подходит для случаев, где важен антиалиасинг, хотя его сложность немного выше из-за работы с интенсивностями.
4. **Step-by-step** и **DDA** менее эффективны, но полезны для понимания базовых принципов построения графики.