***Лабораторная работа 3***

***Базовые растровые алгоритмы.***

***Учащейся 4 группы, 3 курса, Дец Анна***

# Задача:

Написать приложение/веб-приложение, иллюстрирующее работу базовых растровых алгоритмов

* пошаговый алгоритм;
* алгоритм ЦДА;
* алгоритм Брезенхема;
* алгоритм Брезенхема (окружность).

# Время для построения линии с координатами (0, 0, 100, 100):

* алгоритм Брезенхема 450500нс
* пошаговый алгоритм 965100нс

# Средняя скорость рисования:

* алгоритм Брезенхема 140 / 450500 нс = 3.1 \* 10^(5) пикселей в секунду
* пошаговый алгоритм 140 / 965100 нс = 1.45 \* 10^(5) пикселей в секунду

# Сравнение алгоритмов Брезенхема и пошагового алгоритма рисования линии:

## **Пошаговый алгоритм:**

• ***Принцип***:

Этот алгоритм рисует линию, двигаясь пошагово от начальной точки к конечной. Он использует линейное уравнение y = kx + b, чтобы вычислить координаты каждой следующей точки.

• ***Преимущества***:

- Прост в реализации.

- Легко понять принцип работы.

• ***Недостатки***:

- Требует операций с плавающей запятой (для вычисления y = kx + b), что может быть медленнее для некоторых процессоров.

- Могут быть неточности при округлении координат.

• ***Применение*:** В некоторых случаях может быть предпочтительным, если требуется простота и легкость понимания алгоритма.

## **Алгоритм ЦДА (Цифровой дифференциальный анализатор):**

• ***Принцип:***

Это инкрементный алгоритм, который попиксельно рисует прямую линию, используя приращения по осям x и y. Он основывается на уравнении прямой y = mx + b и итерационном подходе, вычисляя координаты каждого пикселя, последовательно двигаясь по линии.

• ***Преимущества:***

- Простой и эффективный.

- Точность и универсальность.

- Экономия памяти.

• ***Недостатки:***

- Проблемы с крутыми углами: при крутых углах линия может быть недостаточно гладкой, так как алгоритм ЦДА основан на приращениях по осям x и y.

- Ограниченная гибкость: предназначен для построения прямых линий и не может быть легко адаптирован для других геометрических фигур.

• ***Применение:***

- Компьютерная графика: используется в компьютерной графике для построения прямых линий в играх, приложениях для рисования и других программах.

- Визуализация: для визуализации данных в научных и инженерных приложениях.

- Обработка изображений: для построения прямых линий в процессе обработки изображений, например, для выделения линейных объектов.

## **Алгоритм Брезенхема:**

• ***Принцип*:**

Этот алгоритм использует инкрементные алгоритмы, чтобы найти точки линии, избегая операций с плавающей запятой.

**• *Преимущества*:**

- Высокая скорость выполнения.

- Точный алгоритм, не требует округления координат.

- Не использует операции с плавающей запятой.

**• *Недостатки*:**

- Более сложный в реализации, чем пошаговый алгоритм.

- Требует больше операций сравнения.

• ***Применение*:** используется в большинстве графических систем для рисования линий, особенно когда требуется высокая скорость отрисовки.

## **Алгоритм Брезенхема для окружности:**

**• *Принцип***:

Это алгоритм инкрементного построения, который попиксельно определяет, какие пиксели нужно закрасить, чтобы получить наиболее точное изображение окружности. Он использует алгебраическое уравнение окружности и итерационный подход для вычисления координат каждого пикселя.

**• *Преимущества***:

- Простой и эффективный.

- Точность и универсальность.

- Экономия памяти: не требует дополнительных структур данных.

**• *Недостатки***:

- Не идеальная гладкость: не может обеспечить идеально гладкую окружность.

- Ограниченная гибкость: нельзя легко адаптировать для других геометрических фигур.

**• *Применение***:

- Компьютерная графика: построение окружностей в играх, приложения для рисования и в других программах.

- Визуализация: визуализация данных в научных и инженерных приложениях.

- Обработка изображений.

**Вывод:**

Алгоритм Брезенхема является более предпочтительным для рисования линий, так как он обеспечивает высокую скорость выполнения и точность. Пошаговый алгоритм может быть полезен, если требуется простота реализации и легкость понимания алгоритма.