Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ  ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

                        Факультет      Информационных технологий и управления

                        Кафедра         Интеллектуальных информационных технологий

**ОТЧЁТ**

по лабораторной работе №1

по дисциплине “Графический интерфейс интеллектуальных систем”

Выполнил:

Чаюк Т.А. , гр. 221702

Проверил:

Жмыркова А.В.

Минск 2025

**Цель работы**: Разработать программу, реализующую построение отрезков с использованием следующих алгоритмов:

* Алгоритм ЦДА (Цифровой дифференциальный анализатор)
* Целочисленный алгоритм Брезенхема
* Алгоритм Ву.

**Задание:**

Разработать элементарный графический редактор, реализующий построение отрезков с помощью алгоритма ЦДА, целочисленного алгоритма Брезенхема и алгоритма Ву. Вызов способа генерации отрезка задается из пункта меню и доступно через панель инструментов «Отрезки». В редакторе кроме режима генерации отрезков в пользовательском окне должен быть предусмотрен отладочный режим, где отображается пошаговое решение на дискретной сетке.

**Алгоритмы рисования:**

**ЦДА:**

def dda\_line(x0, y0, x1, y1):

dx = x1 - x0

dy = y1 - y0

steps = int(max(abs(dx), abs(dy)))

if steps == 0:

yield (x0, y0)

return

x\_inc = dx / steps

y\_inc = dy / steps

x = x0

y = y0

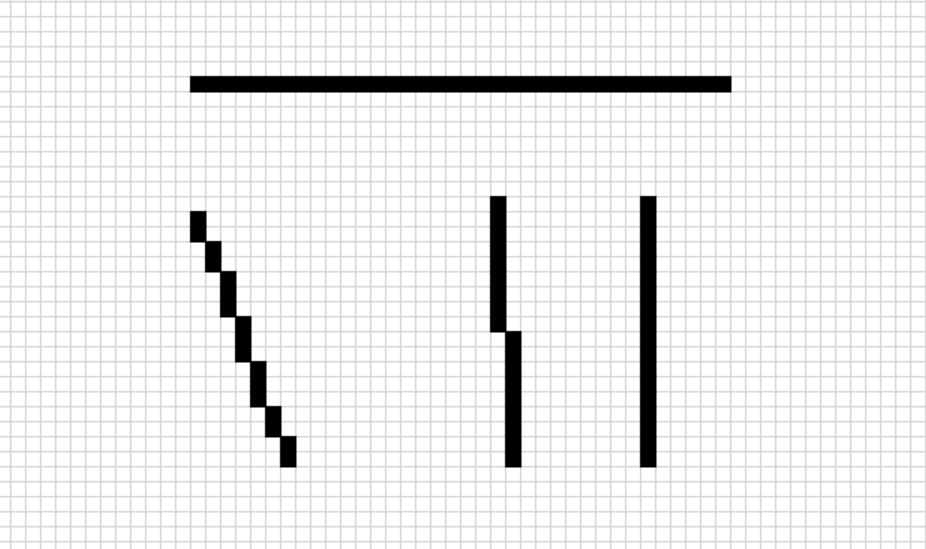
for i in range(steps + 1):

yield (round(x), round(y))

x += x\_inc

y += y\_inc

**Пример работы:**



*Рис. 1 - Пример работы алгоритма ЦДА*

**Алгоритм Брезенхема:**

def bresenham\_line(x0, y0, x1, y1):

dx = abs(x1 - x0)

dy = abs(y1 - y0)

sx = 1 if x0 < x1 else -1

sy = 1 if y0 < y1 else -1

err = dx - dy

while True:

yield (x0, y0)

if x0 == x1 and y0 == y1:

break

e2 = 2 \* err

if e2 > -dy:

err -= dy

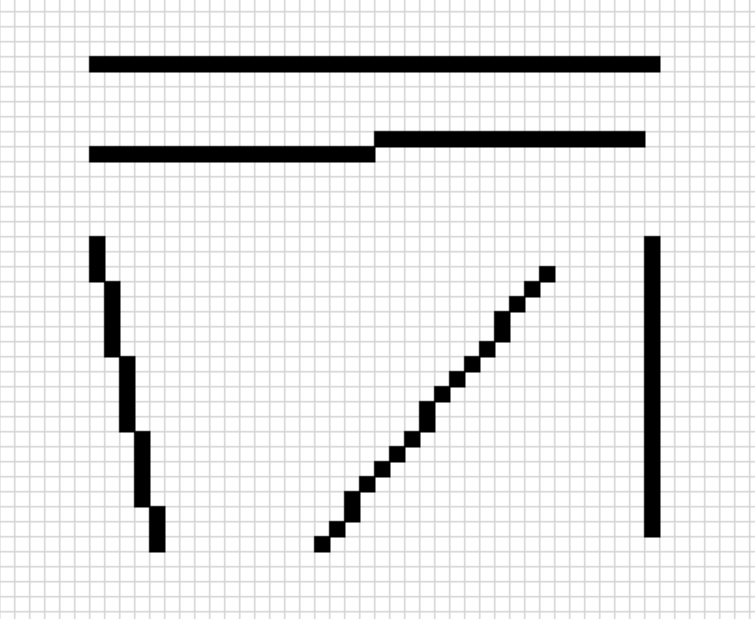
x0 += sx

if e2 < dx:

err += dx

y0 += sy

**Пример работы:**



*Рис. 2 - Пример работы алгоритма Брезенхема*

**Алгоритм Ву:**

def wu\_line(x0, y0, x1, y1):

def fpart(x):

return x - math.floor(x)

def rfpart(x):

return 1 - fpart(x)

steep = abs(y1 - y0) > abs(x1 - x0)

if steep:

# Меняем местами x и y

x0, y0 = y0, x0

x1, y1 = y1, x1

if x0 > x1:

x0, x1 = x1, x0

y0, y1 = y1, y0

dx = x1 - x0

dy = y1 - y0

gradient = dy / dx if dx != 0 else 1

xend = round(x0)

yend = y0 + gradient \* (xend - x0)

xgap = rfpart(x0 + 0.5)

xpxl1 = xend

ypxl1 = int(math.floor(yend))

if steep:

yield (ypxl1, xpxl1, rfpart(yend) \* xgap)

yield (ypxl1 + 1, xpxl1, fpart(yend) \* xgap)

else:

yield (xpxl1, ypxl1, rfpart(yend) \* xgap)

yield (xpxl1, ypxl1 + 1, fpart(yend) \* xgap)

intery = yend + gradient

xend = round(x1)

yend = y1 + gradient \* (xend - x1)

xgap = fpart(x1 + 0.5)

xpxl2 = xend

ypxl2 = int(math.floor(yend))

if steep:

yield (ypxl2, xpxl2, rfpart(yend) \* xgap)

yield (ypxl2 + 1, xpxl2, fpart(yend) \* xgap)

else:

yield (xpxl2, ypxl2, rfpart(yend) \* xgap)

yield (xpxl2, ypxl2 + 1, fpart(yend) \* xgap)

if steep:

for x in range(xpxl1 + 1, xpxl2):

y = int(math.floor(intery))

yield (y, x, rfpart(intery))

yield (y + 1, x, fpart(intery))

intery += gradient

else:

for x in range(xpxl1 + 1, xpxl2):

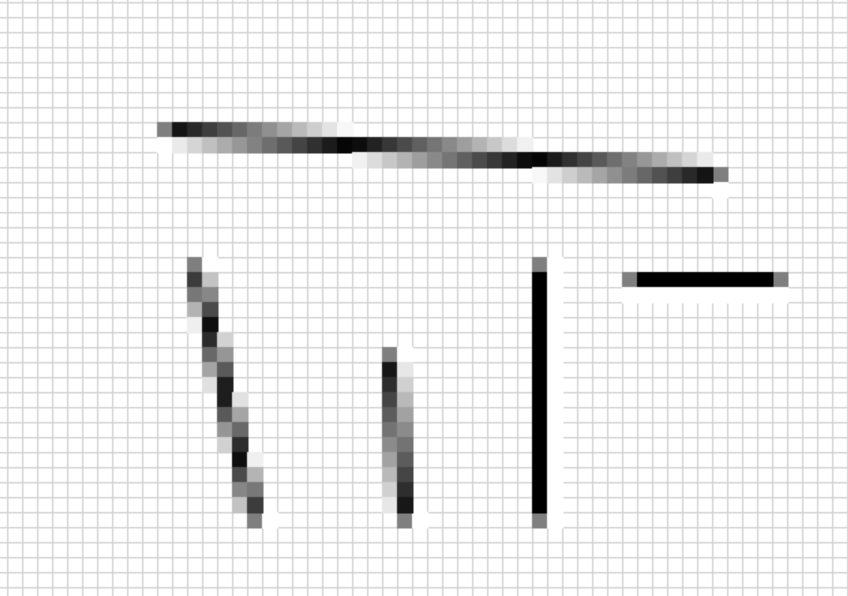
y = int(math.floor(intery))

yield (x, y, rfpart(intery))

yield (x, y + 1, fpart(intery))

intery += gradient

**Пример работы:**



*Рис. 3 - Пример работы алгоритма Ву*

**Вывод:** Алгоритм ЦДА, основанный на использовании вещественных чисел, обеспечивает простую и понятную реализацию, однако менее эффективен с точки зрения производительности из-за операций с дробными значениями.

Целочисленный алгоритм Брезенхема демонстрирует значительно более высокую производительность благодаря исключению операций с плавающей запятой, что делает его предпочтительным выбором для большинства практических задач растровой графики.

Алгоритм Ву отличается тем, что позволяет получать визуально более качественные и гладкие линии за счет использования полутоновых значений, хотя это требует дополнительных вычислений.