

Учреждение образования
“Белорусский государственный университет
информатики и радиоэлектроники”

Факультет информационных технологий и управления
Кафедра интеллектуальных информационных технологий

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине «Графический интерфейс интеллектуальных систем»

Выполнил: студент гр. 121701

Смольник В.А.

Проверила:

Жмырко А.В.

Минск 2023

Тема

Алгоритмы построения отрезков

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.

Выполнение задания

Функция построения отрезка методом DDA:

```
Explain | Test | Document | Fix | Ask
1 usage
def build_by_dda(self, point1, point2):
    x1, y1 = point1
    x2, y2 = point2
    dx = x2 - x1
    dy = y2 - y1
    length = max(abs(dx), abs(dy))
    dx /= length
    dy /= length
    x = x1 + 0.5 * dx
    y = y1 + 0.5 * dy
    line = []
    for _ in range(int(length)):
        line.append(self.canvas.create_rectangle(x - 1, y - 1, x + 1, y + 1, fill="blue", outline="blue"))
        x += dx
        y += dy
    self.lines.append(line)
```

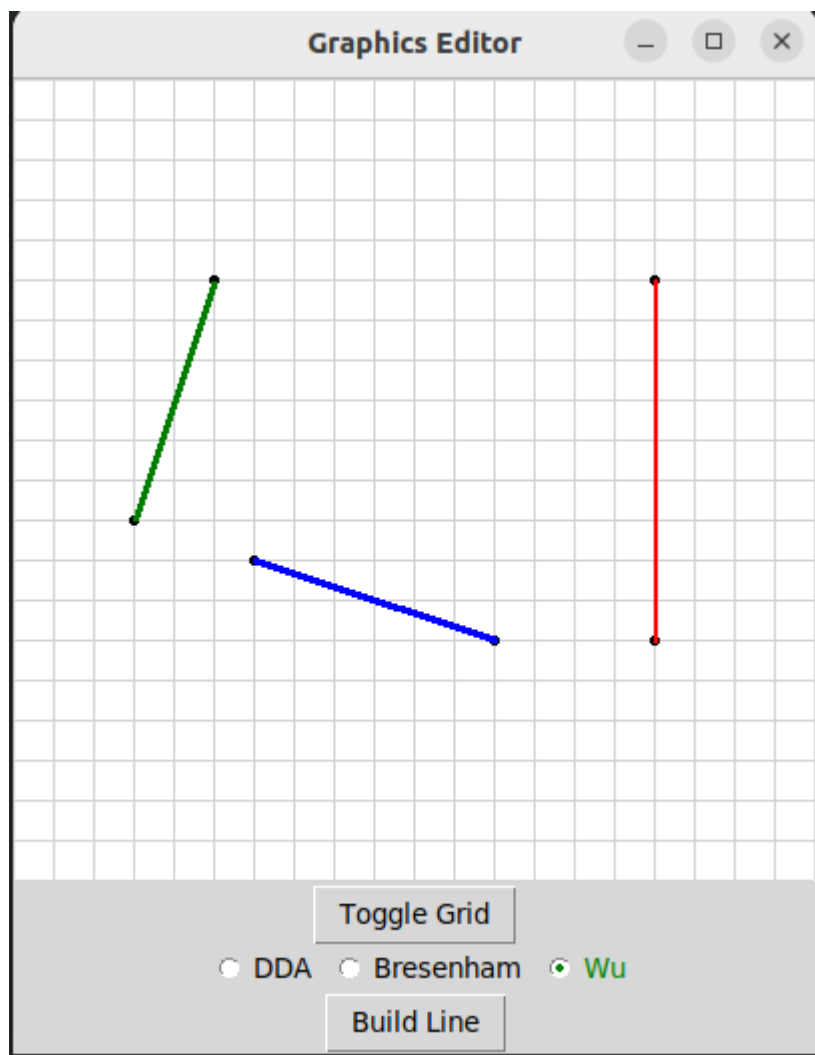
Функция построения отрезка методом Брезенхама:

```
def build_bresenham_line(self, point1, point2):
    x1, y1 = point1
    x2, y2 = point2
    dx = abs(x2 - x1)
    dy = abs(y2 - y1)
    sx = 1 if x1 < x2 else -1
    sy = 1 if y1 < y2 else -1
    err = dx - dy
    line = []
    while True:
        line.append(self.canvas.create_rectangle(x1, y1, x1 + 1, y1 + 1, fill="red", outline="red"))
        if x1 == x2 and y1 == y2:
            break
        e2 = 2 * err
        if e2 > -dy:
            err -= dy
            x1 += sx
        if e2 < dx:
            err += dx
            y1 += sy
    self.lines.append(line)
```

Функция построения отрезка методом WU:

```
def build_wu_line(self, point1, point2):
    x1, y1 = point1
    x2, y2 = point2
    dx = x2 - x1
    dy = y2 - y1
    if abs(dx) > abs(dy):
        if x2 < x1:
            x1, x2 = x2, x1
            y1, y2 = y2, y1
        gradient = dy / dx
        y = y1 + gradient
        for x in range(x1 + 1, x2):
            self.canvas.create_rectangle(x, int(y), x + 1, int(y) + 1, fill="green", outline="green")
            self.canvas.create_rectangle(x, int(y) + 1, x + 1, int(y) + 2, fill="green", outline="green")
            y += gradient
    else:
        if y2 < y1:
            x1, x2 = x2, x1
            y1, y2 = y2, y1
        gradient = dx / dy
        x = x1 + gradient
        for y in range(y1 + 1, y2):
            self.canvas.create_rectangle(int(x), y, int(x) + 1, y + 1, fill="green", outline="green")
            self.canvas.create_rectangle(int(x) + 1, y, int(x) + 2, y + 1, fill="green", outline="green")
            x += gradient
```

Графический интерфейс:



Вывод: во время выполнения данной лабораторной работы был освоен навык построения отрезков тремя способами. Написано десктопное приложение для построения отрезков тремя способами