**Лабораторная работа 4**

**Отчёт**

**Цель работы:**

Закрепление теоретического материала и практическое освоение базовых алгоритмов растеризации отрезков и кривых. Растеризация отрезков и кривых является важной задачей в компьютерной графике, и понимание их работы позволяет создавать и отображать различные геометрические фигуры на экране.

**Задача**:

Написать приложение/веб-приложение, иллюстрирующее работу четырех базовых алгоритмов растеризации: пошаговый алгоритм, алгоритм ЦДА, алгоритм Брезенхема для отрезков и алгоритм Брезенхема для окружностей.

Реализовать функции для каждого из алгоритмов растеризации, которые будут принимать координаты начальной и конечной точек (для отрезков) или центра и радиус (для окружностей), и возвращать список пикселей, которые должны быть заполнены для отображения фигуры на экране.

Создать интуитивный и удобный интерфейс приложения с возможностью выбора алгоритма растеризации, ввода координат и отображения результата на экране. Интерфейс должен включать систему координат, оси, линии сетки и подписи для наглядного представления графических объектов.

Также необходимо указать временные характеристики реализованных алгоритмов в своей реализации.

Предоставить исходный код приложения/веб-приложения, а также сопроводительную документацию.

**Средства разработки**:

Фреймворк Qt и язык Python 3.11\_3.11.1776.0\_x64.

**Ход работы**:

Импортируются необходимые библиотеки: matplotlib, numpy, PySide6, raster (это модуль, содержащий реализацию алгоритмов растеризации).

Определен класс Line, который представляет линию на графике. Класс имеет следующие атрибуты: start\_x, start\_y, end\_x, end\_y (координаты начальной и конечной точек) и color (цвет линии).

Определен класс ZoomableCanvas, который является подклассом FigureCanvas из библиотеки matplotlib. Он предоставляет холст для рисования графики и поддерживает масштабирование с помощью колеса мыши. Класс имеет методы reset (сброс настроек масштабирования) и calculate\_zoom (вычисление новых границ масштабирования). Также класс обрабатывает события прокрутки колесика мыши для изменения масштаба графика.

Определен класс MainWindow, который является главным окном приложения. Он наследуется от класса QMainWindow из фреймворка PySide6. В конструкторе класса создается графический интерфейс, включающий холст ZoomableCanvas, выпадающий список QComboBox для выбора алгоритма растеризации, кнопки QPushButton для рисования и очистки графика. Также класс содержит методы для обработки событий нажатия и отпускания кнопки мыши для рисования линий на графике. Методы также обрабатывают нажатия на кнопки "Draw" и "Clear" для выполнения растеризации и очистки графика соответственно.

В функции update\_filled\_pixels происходит выполнение растеризации для всех линий, заданных пользователем. Выбирается алгоритм растеризации на основе выбранного значения в выпадающем списке. Затем вызывается функция rasterize\_line из модуля raster для растеризации линии. Результы растеризации добавляются в список filled\_pixels. Затем происходит отображение растеризованных пикселей на графике с использованием метода fill холста ZoomableCanvas. После этого график обновляется с помощью метода draw.

Импортируются необходимые библиотеки: Decimal, numpy и math.

Определяется перечисление RasterizingAlgorithm, которое содержит четыре значения: Naive, DDA, Bresenham и Circle. Это перечисление используется для выбора алгоритма растеризации.

Определяется функция my\_round, которая округляет число до целого значения. Она использует функцию Decimal из библиотеки Decimal для точного округления.

Определяется функция rasterize\_line, которая выполняет растеризацию линии или окружности в зависимости от выбранного алгоритма.

Для каждого алгоритма растеризации реализован соответствующий код:

В алгоритме Naive проверяются координаты начальной и конечной точек, и в зависимости от их значений формируется список пикселей, которые должны быть заполнены для отображения линии на экране. Используется цикл for и функция np.arange для генерации пикселей.

В алгоритме DDA вычисляются шаги инкремента для координат x и y на основе разности между начальными и конечными координатами. Затем используется цикл for для генерации пикселей на основе инкрементов.

В алгоритме Bresenham реализована адаптация алгоритма Брезенхема для растеризации отрезков. Используется алгоритм, который работает с целыми числами и позволяет эффективно определить пиксели, которые должны быть заполнены для отображения линии на экране.

В алгоритме Circle вычисляются координаты пикселей, которые должны быть заполнены для отображения окружности на экране. Используется алгоритм Брезенхема для растеризации окружностей.

В конце кода выполняется пример использования функции my\_round. Цикл for проходит по значениям n от 0.5 до 7.0 с шагом 0.5 и выводит округленные значения n с использованием функции my\_round.

**Выводы**:

Данный код представляет собой реализацию графического интерфейса для визуализации растеризации линий и окружностей на координатной плоскости. Он использует библиотеку matplotlib для отображения графики и PySide6 для создания окна приложения.

Приложение позволяет пользователю рисовать линии на графике, выбирать алгоритм растеризации и выполнять растеризацию для всех нарисованных линий. Реализованы несколько алгоритмов растеризации, которые можно выбрать из выпадающего списка. После выполнения растеризации, результирующие пиксели отображаются на графике.

Класс ZoomableCanvas предоставляет возможность масштабирования графика с помощью колеса мыши, что позволяет пользователю легко изменять масштаб и просматривать детали графика.

Общий ход работы приложения следующий:

* Пользователь рисует линии на графике, щелкая на точки начала и конца линии.
* При нажатии на кнопку "Draw" выполняется растеризация всех нарисованных линий с использованием выбранного алгоритма.
* Растеризованные пиксели отображаются на графике с использованием метода fill холста.
* При нажатии на кнопку "Clear" график и список линий очищаются.

Эта лабораторная работа представляет пример использования библиотек matplotlib и PySide6 для создания интерактивной графической программы. Она также демонстрирует применение алгоритмов растеризации для визуализации геометрических объектов на плоскости.

Также был создан exe файл, который является самодостаточным.