# **Лабораторная работа №4**

## **Цель работы**

Целью данной работы является разработка и визуализация базовых растровых алгоритмов для построения графических примитивов (линий и окружностей) с использованием HTML и JavaScript. Программа позволяет пользователю наглядно увидеть, как работают эти алгоритмы на уровне отдельных пикселей, что способствует лучшему пониманию их внутренней логики и особенностей.

**Задачи**

Для достижения поставленной цели были сформулированы следующие задачи:

1. **Создать интерфейс** для взаимодействия с пользователем, позволяющий выбирать различные растровые алгоритмы.
2. **Разработать и реализовать** несколько популярных растровых алгоритмов для построения линий и окружностей:
   1. Пошаговый алгоритм
   2. Алгоритм ЦДА (Цифровой Дифференциальный Анализатор)
   3. Алгоритм Брезенхема для отрезка
   4. Алгоритм Брезенхема для окружности
   5. Алгоритм Кастла-Питвея
   6. Алгоритм сглаживания линий (антиалиасинг)
3. **Организовать вывод** графики на канвас (canvas) с использованием JavaScript.
4. **Обеспечить удобство использования** программы через кнопки, которые запускают выполнение выбранного алгоритма.
5. **Обеспечить корректное отображение графики** на экране, включая возможность очистки области рисования.

## **Использованные средства разработки**

Для реализации программы были использованы следующие средства:

1. **HTML5** — для создания структуры веб-страницы и размещения элементов управления.
2. **CSS** — для стилизации страницы и элементов управления.
3. **JavaScript** — для реализации растровых алгоритмов и взаимодействия с графическим элементом <canvas>.
4. **Canvas API** — для отрисовки линий и окружностей непосредственно на холсте (canvas).

## **Ход работы**

1. **Создание структуры HTML-документа**

В структуре HTML-документа основными элементами являются:

* Заголовок страницы (<h1>), который сообщает пользователю о назначении приложения — визуализация растровых алгоритмов.
* Блок с кнопками (<div class="controls">) для управления различными алгоритмами. Каждая кнопка запускает соответствующий алгоритм с помощью JavaScript.
* Элемент <canvas> с заданными размерами, который используется для рисования графических примитивов.

**2. Стилизация страницы**

С помощью CSS была выполнена простая стилизация:

* Обеспечен общий стиль для текста и элементов, включая выравнивание по центру и выбор шрифта.
* Добавлен светло-серый фон для страницы и рамка для элемента <canvas>.

**3. Реализация растровых алгоритмов**

Основная часть работы была сосредоточена на реализации следующих алгоритмов:

**3.1 Пошаговый алгоритм рисования линии**

Этот алгоритм представляет собой пошаговое построение линии на основе вычисления ошибки отклонения. Он использует базовые шаги для достижения конечных координат и отрисовывает точку на каждом шаге.

**3.2 Алгоритм ЦДА (Цифровой Дифференциальный Анализатор)**

Алгоритм ЦДА вычисляет приращения по координатам (x и y) на каждом шаге и строит линию с точностью до пикселя. Этот алгоритм подходит для рисования линий с плавающей точкой.

**3.3 Алгоритм Брезенхема для отрезка**

Алгоритм Брезенхема является одним из самых известных и эффективных методов для построения отрезков на растровом экране. Он минимизирует использование операций с плавающей точкой, что делает его быстрым и точным для графических приложений.

**3.4 Алгоритм Брезенхема для окружности**

Этот алгоритм расширяет метод Брезенхема для построения окружностей. Он использует симметрию окружности для сокращения количества вычислений, рисуя сразу несколько точек на разных квадрантах окружности.

**3.5 Алгоритм Кастла-Питвея**

Этот алгоритм также используется для построения линий и является вариантом алгоритма с использованием целочисленных операций для повышения производительности.

**3.6 Алгоритм сглаживания линий (антиалиасинг)**

Антиалиасинг (сглаживание) уменьшает эффект "рваных" линий (aliasing), улучшая визуальное восприятие линий на экране. В данном случае, для каждой точки линии рассчитывается прозрачность (alpha) в зависимости от положения точки относительно ближайшего пикселя.

**4. Взаимодействие с холстом (canvas)**

Для отрисовки графики используется HTML5 API элемента <canvas>. Основные функции JavaScript взаимодействуют с контекстом рисования (2d-контекст), чтобы рисовать линии и окружности на основе вычисленных координат.

**5. Управление программой**

Пользователь взаимодействует с программой через набор кнопок, каждая из которых вызывает соответствующую функцию для отрисовки линии или окружности. После нажатия кнопки холст очищается, и начинается выполнение выбранного алгоритма.

## **Вывод**

Разработанная программа предоставляет пользователю возможность визуализации работы различных растровых алгоритмов. Это полезно для учебных целей, так как позволяет увидеть, как каждый алгоритм строит линии и окружности на уровне отдельных пикселей. Программа также демонстрирует разницу в подходах к построению графических примитивов, что важно для понимания эффективности и особенностей каждого алгоритма.

**Зачем нужна эта программа**

Данная программа может использоваться для следующих целей:

* **Образовательные цели**: Программа полезна для студентов и преподавателей, изучающих компьютерную графику. Она позволяет наглядно увидеть, как работают различные алгоритмы построения линий и окружностей.
* **Демонстрация алгоритмов**: Программа может быть использована для презентаций и лекций по компьютерной графике, чтобы показать, как работают растровые алгоритмы на практике.
* **Исследовательские цели**: Программа может использоваться для сравнения эффективности различных алгоритмов.

**Где можно использовать**

Программа может быть использована в следующих областях:

* В учебных заведениях для преподавания компьютерной графики.
* В исследовательских проектах, связанных с компьютерной графикой и разработкой алгоритмов.
* Для разработки приложений, где необходимо построение графических примитивов, например, в играх или системах автоматизированного проектирования (CAD)

**Подходящие компьютерные системы**

Программа реализована с использованием стандартных технологий веб-разработки (HTML, CSS, JavaScript), поэтому она может быть запущена на любой системе, поддерживающей современные браузеры. Это включает:

* **Операционные системы**: Windows, macOS, Linux.
* **Устройства**: Персональные компьютеры, ноутбуки, планшеты и даже некоторые смартфоны.
* **Браузеры**: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari и другие браузеры, поддерживающие HTML5 и Canvas API.

Таким образом, программа является кроссплатформенной и не требует установки дополнительных плагинов или программного обеспечения.