**Отчет по лабораторной работе №4 «Базовые Алгоритмы Растеризации»**

**Цель:** закрепление теоретического материала и практическое освоение основных возможностей по использованию базовых алгоритмов растеризации отрезков и кривых.

**Задачи:** Написать приложение/веб-приложение, иллюстрирующее работу базовых растровых алгоритмов (4 алгоритма):

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

**Использованные языки и среды программирования:**

* C++
* Qt Creator Использовался комплект "Qt 6.2.4 for macOS"

**Функционал:**

**1)**Метод InitializeAlgorithms()

* **Описание: Метод для инициализации группы действий (**QActionGroup**), представляющей алгоритмы растеризации, а также связывание элементов интерфейса с соответствующими слотами.**
* **Вызовы методов и функций:**
  + QActionGroup::addAction()
  + QAction::setCheckable()
  + connect()

**2)**Слот on\_comboBox\_currentIndexChanged(int index)

* **Описание**: Слот, вызываемый при изменении выбора в Combobox. Определяет текущий выбранный алгоритм и обновляет интерфейс в соответствии с выбором.
* **Параметры**:
  + index: Индекс текущего элемента в Combobox.
* **Вызовы методов и функций**:
  + setVisible()
  + setText()

**3)**Событие **paintEvent(QPaintEvent \*)**

* **Описание**: Обработчик события отрисовки, отвечающий за рисование сетки и вызов соответствующих методов для выбранного алгоритма растеризации.

**4)**Метод **StepByStepLine(double x0, double y0, double x1, double y1)**

* **Описание**: Метод StepByStepLine реализует пошаговый алгоритм растеризации отрезка. Этот алгоритм работает на основе уравнения прямой, используя простые операции числами для определения следующей точки отрезка на плоскости. В каждом шаге, выбирается либо пиксель выше, либо пиксель ниже текущей позиции, исходя из значения функции прямой.
* **Параметры**: x0, y0, x1, y1: Координаты начальной и конечной точек отрезка.
* **Возвращаемое значение**: Вектор точек, представляющих растеризованный отрезок.

**5)**Метод **BresenhamLine(double x0, double y0, double x1, double y1)**

* **Описание**: Метод BresenhamLine реализует алгоритм Брезенхема для растеризации отрезка. Этот алгоритм представляет собой инкрементальный метод, который основан на использовании целочисленных операций и позволяет эффективно определить следующий пиксель на линии. Алгоритм минимизирует использование операций с плавающей точкой, что делает его более быстрым по сравнению с алгоритмами, основанными на вещественной арифметике.
* **Параметры**: x0, y0, x1, y1: Координаты начальной и конечной точек отрезка.
* **Возвращаемое значение**: Вектор точек, представляющих растеризованный отрезок.

**6)**Метод **DDALine(double x0, double y0, double x1, double y1)**

* **Описание**: Метод DDALine реализует алгоритм Цифрового Дифференциального Анализатора (ЦДА или DDA) для растеризации отрезка. Алгоритм основан на использовании операций с плавающей точкой и предназначен для равномерной дискретизации линии вдоль оси X или Y. Алгоритм ЦДА может быть предпочтителен при работе с отрезками, имеющими произвольный угол наклона и когда точная дискретизация по координатам пикселей не является первостепенной задачей.
* **Параметры**: x0, y0, x1, y1: Координаты начальной и конечной точек отрезка.
* **Возвращаемое значение**: Вектор точек, представляющих растеризованный отрезок.

**7)** Метод **BresenhamCircle(double x0, double y0, double R)**

* **Описание**: Метод BresenhamCircle реализует алгоритм Брезенхема для растеризации окружности. Этот алгоритм позволяет эффективно определить координаты пикселей на окружности.Этот метод предоставляет возможность получить четыре части окружности за одну итерацию, минимизируя количество вычислений и обеспечивая высокую производительность.
* **Параметры**:
  + x0, y0: Координаты центра окружности.
  + R: Радиус окружности.
* **Возвращаемое значение**: Вектор векторов точек, представляющих растеризованную окружность для каждой четверти.

**8)** Слоты **recalc()** и **on\_doubleSpinBox\_editingFinished(), on\_doubleSpinBox\_2\_editingFinished(), on\_doubleSpinBox\_3\_editingFinished(), on\_doubleSpinBox\_4\_editingFinished()**

* **Описание**: Слоты и методы для перерасчета отображаемого изображения при изменении значений в SpinBox'ах, связанных с координатами отрезка или окружности.