Министерство образования и науки Российской Федерации Федеральное агентство по образованию

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №4 дисциплины

«Компьютерная графика»

Выполнил студент группы ИВТ-22 /Крючков И. С/ Проверил /Коржавина А. С./

Киров 2021

**Цель работы:**

закрепить лекционный материал по изучению базовых алгоритмов компьютерной графики алгоритмов закраски.

# Задание:

1. Написать на языке PASCAL программу, реализующую алгоритм заливки многоугольника любой формы.
2. Проверить правильность работы программы, нарисовав с помощью функций модуля GRAPH невыпуклый многоугольник (с дырами внутри), закрасьте его заданным цветом, указав координаты принадлежащей многоугольнику точки.
3. Написать и отладить программу, реализующую алгоритм построчного заполнения выпуклого многоугольника, заданного координатами вершин и цветом границы (можно воспользоваться алгоритмом заполнения треугольника, прочитанным на лекции).

**Теория:**

Возможность представления сплошных областей - это одна из уникальных характеристик растрового дисплея. Все возможные методы такого представления можно разделить на две категории: растровая развертка и затравочное заполнение.

В первой в порядке сканирования строк определяют, лежит ли точка внутри области. Такие алгоритмы применимы и к векторным дисплеям для штриховки контуров.

Во второй предполагают известной некоторую точку внутри замкнутого контура. Ее называют затравочной и ищут соседние с ней точки. Если найденная точка не внутри контура, значит она принадлежит границе, если внутри - она становится новой затравочной и поиск продолжается. Подобные алгоритмы применимы только к растровым дисплеям.

Многие замкнутые контуры либо многоугольники, либо их можно аппроксимировать простыми многоугольниками. Поэтому далее речь идет о заполнении многоугольников.

Растровая развертка многоугольников

Под многоугольником будем понимать фигуру на плоскости, ограниченную несамопересекающейся замкнутой ломаной. Основная идея растровой развертки многоугольников заключается в заполнении лишь одной строки в каждый данный момент. Алгоритмы, работающие по этому принципу называют алгоритмами построчного сканирования .Простым способом выделения пикселов сканирующей строки попадающих внутрь многоугольника, является поочередная проверка всех пикселов строки растра (экрана).Но обычно большая часть пикселов лежит вне многоугольника, поэтому данный метод неэффективен во времени. Его можно несколько улучшить, помещая многоугольник внутрь минимально объемлющего прямоугольника со сторонами, параллельными сторонам растра,и анализируя лишь точки внутри этого прямоугольника .Но можно найти более эффективный метод, если допустить с большой долей вероятности, что соседние пикселы в строке (столбце) имеют одинаковые характеристики. Это допущение основывается на понятии пространственной когерентности: при перемещении от точки к точке или от одной сканирующей строки к соседней рассматриваемый многоугольник в большинстве случаев остается постоянным. Можно искать только те точки, где когерентность строк (столбцов) нарушена.

Пусть многоугольник задан набором своих вершин, причем соседние точки в наборе - его смежные вершины. Тогда на каждой сканирующей строке внутренние пикселы можно закрасить так:

1. Найти точки пересечения сканирующей строки со всеми ребрами многоугольника, кроме горизонтальных.
2. Отсортировать найденные точки по возрастанию координаты х.
3. Закрасить все пикселы между парами точек пересечений. Алгоритм работает только в случае существенных пересечений и четного количества точек пересечений. Нечетное количество получается при пересечении сканирующей строки с локальными минимумами и максимумами, когда Y - координаты предыдущей и последующей вершин больше (меньше) чем у рассматриваемой. В этом случае такие пересечения учитываются как два. После такой корректировки алгоритм будет работать правильно при любой форме многоугольника.

Для выпуклых многоугольников алгоритм можно упростить и повысить его эффективность. Границу выпуклого многоугольника разбивают на две ломаные: "левую" и "правую" и, возможно, два ребра "верхнее" и "нижнее", тогда каждая ломаная будет иметь ровно одно пересечение с каждой сканирующей прямой. Используя алгоритм Брезенхема и одновременно генерируя растровое представление обеих ломаных, находят левый и правый пикселы границы многоугольника на каждой сканирующей прямой. Последовательно заполняя интервалы между этими пикселами от верхней строки к нижней, получают итоговую растровую развертку выпуклого многоугольника.

Алгоритмы заполнения с затравкой

Ищут и закрашивают связную компоненту области, содержащую затравочный пиксел. Под связностью понимают четырех или восьмисвязность в зависимости от постановки задачи. Каждый из пикселов четырехсвязной области достижим из любого другого пиксела этой области с помощью последовательности перемещений на пиксел из четырехэлементного набора горизонтальных и вертикальных направлений, восьмисвязной - из восьмиэлементного (добавляются еще и диагональные направления). Алгоритмы для восьмисвязных областей применимы к четырехсвязным, но не наоборот.

Процесс заполнения областей иногда называют заливкой, так как можно представить, что в точке затравки находится источник, заливающий всю область определенным цветом. Область может быть определена внутренне (все внутренние пикселы - одного цвета, внешние - другого) и гранично (все пикселы на границе или вне области - одного цвета не такого, как внутри).Далее будет рассмотрен гранично-заполняющий алгоритм для четырехсвязных областей (соответствующие внутренне-заполняющие алгоритмы можно получить аналогичным путем, а алгоритмы для четырехсвязной области легко переделать в алгоритмы для восьмисвязной).

Построчный алгоритм заливки

Применяется идея построчного сканирования и используется стек для хранения одного затравочного пиксела для любого непрерывного интервала на сканирующей строке. Непрерывный интервал - это группа примыкающих друг к другу пикселов, ограниченная уже закрашенными или граничными пикселами. Если непрерывный интервал пикселов принадлежит внутренней части области, пикселы на и под этим интервалом либо граничные, либо тоже находятся внутри области и могут служить затравочными для своих строк. Поэтому можно предложить следующую схему заполнения:

**Схемы алгоритмов:**

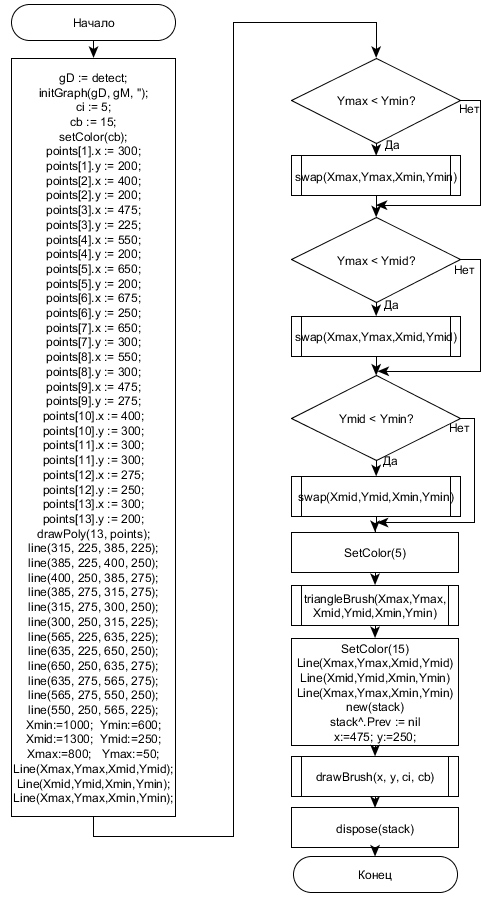


Рисунок 1 Основная программа

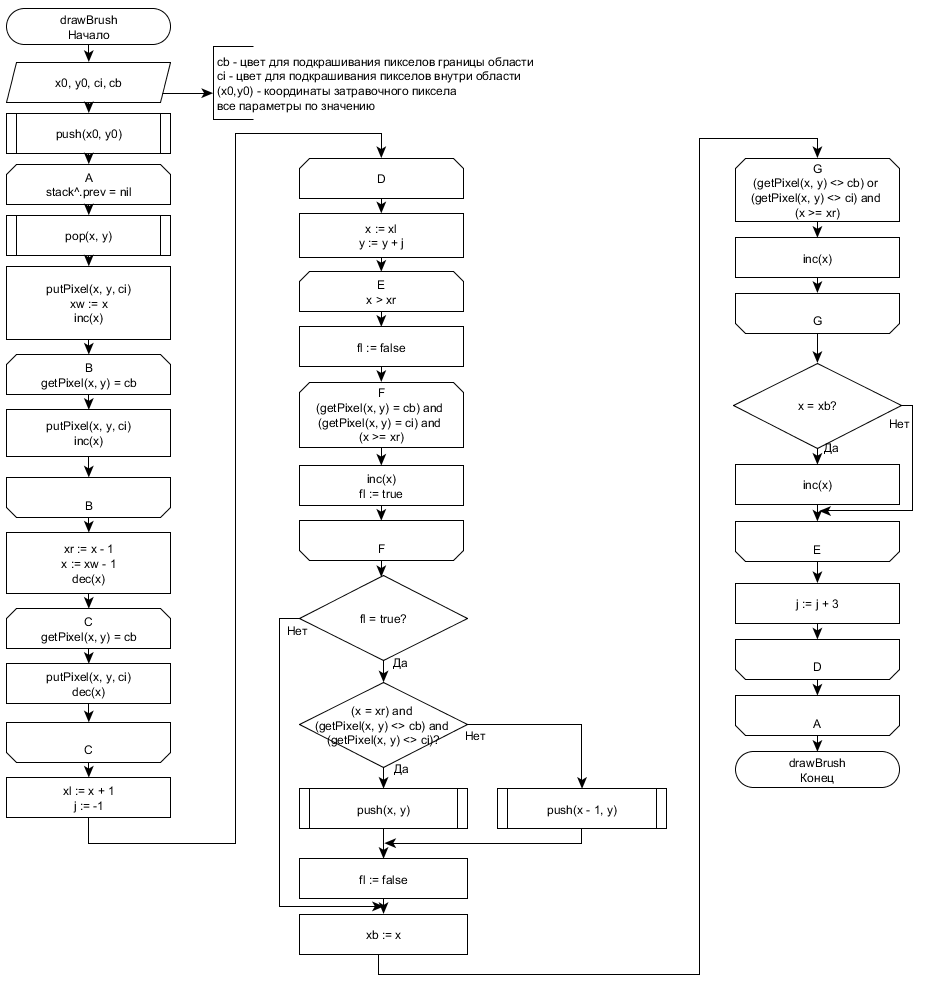


Рисунок 2 Алгоритм построчной заливки

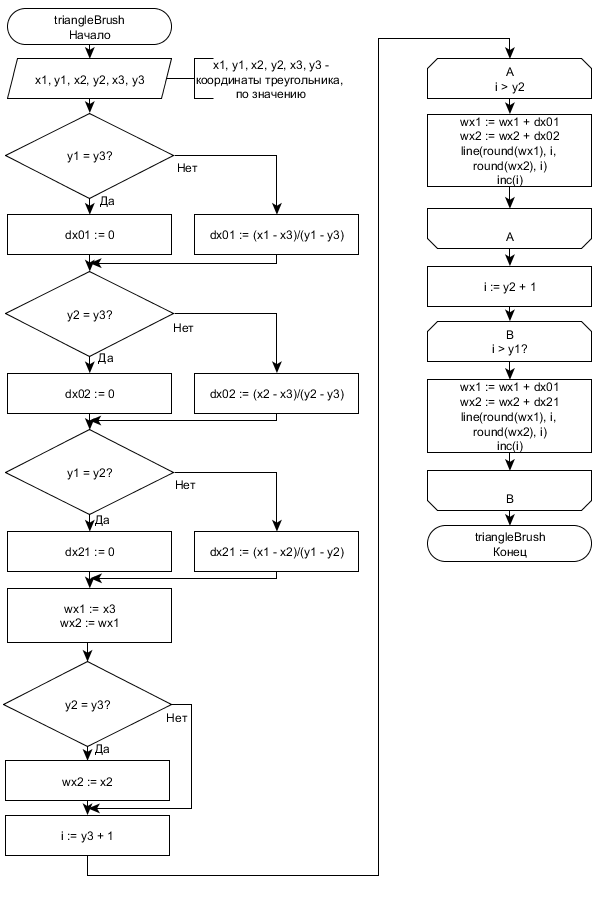


Рисунок 3 Алгоритм заполнения треугольника

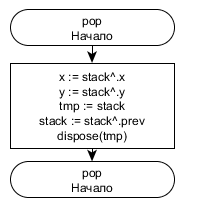


Рисунок 4 Алгоритм извлечения из стека

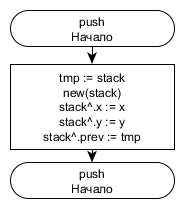
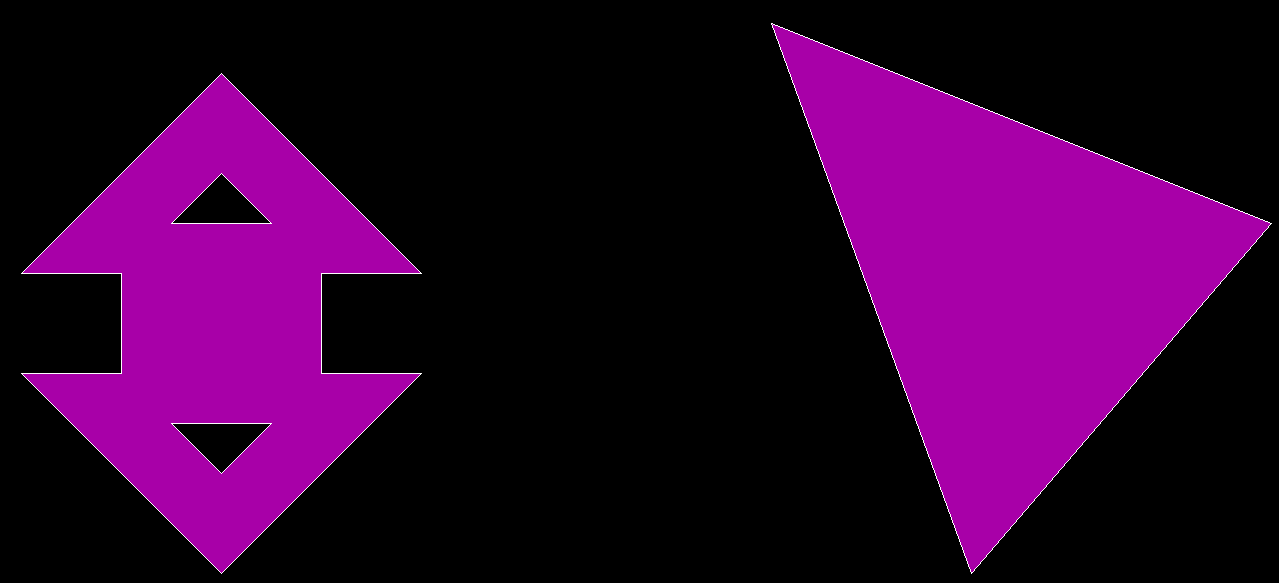


Рисунок 5 Алгоритм добавления в стек

**Экранная форма:**

****

**Вывод:** в ходе выполнения лабораторной работы был закреплен лекционный материал по изучению базовых алгоритмов закраски. Реализована программа выполняющая построчный алгоритм заливки и алгоритм заполнения треугольника.