|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

**Лабораторная работа №5**

**по дисциплине «Компьютерная графика»**

|  |  |
| --- | --- |
| **Тема:** Реализация и исследование алгоритмов растрового заполнения сплошных областей.  **Студент:** Жигалкин Д.Р.  **Группа:** ИУ7-45Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель:** Куров А. В. |  |

Москва.

2020 г.

**Цель работы:**

Реализация и исследование одного из алгоритмов (по заданию преподавателя) растрового заполнения области.

**Задание:**

Реализовать **алгоритм заполнения со списком ребер и флагом.** Обеспечить ввод произвольной многоугольной области, содержащей произвольное количество отверстий.

Ввод (вершин многоугольника) производить с помощью мыши, при этом для удобства пользователя должны отображаться ребра, соединяющие вводимые вершины. Предусмотреть ввод горизонтальных и вертикальных ребер.

Пользователь должен иметь возможность задания цвета заполнения.

Работа программы должна предусматривать два режима – с задержкой и без задержки.

Режим с задержкой должен позволить проследить выполняемую последовательность действий.

(Задержку целесообразно выполнять после обработки очередной строки).

Обеспечить замер времени выполнения алгоритма (без задержки, с выводом на экран только окончательного результата).

**Теоретический материал:**

Одной из особенностей растровых дисплеев является возможность представления сплошных областей. Растровая развертка сплошных областей (заполнение многоугольников или заполнение контуров) - генерация сплошных областей на основе простых описаний ребер или вершин. Для решения этой задачи используются различные методы, которые обычно разделяют на две категории: растровая развертка и затравочное заполнение.

В растровых методах делается попытка определить в порядке сканирования строк принадлежность точки внутренней области контура или многоугольника. Эти алгоритмы обычно просматривают многоугольники (контуры) от верхней точки до нижней. Методы растровой развертки применимы обычно и к векторным дисплеям, в которых они используются для штриховки или закраски контуров.

**Алгоритм заполнения со списком ребер и флагом.**

Алгоритм со списком ребер и флагом - двухшаговый алгоритм. Во-первых, обрисовывается контур, ограничивающий область, в результате чего на каждой строке сканирования определяются пары ограничивающих пикселов, а во-вторых, активизируются пикселы, расположенные между вычисленными на предыдущем шаге ограничивающими пикселами.

Процесс реализации алгоритма со списком ребер и флагом состоит из следующих этапов:

1. Обработка ребер многоугольника, ограничивающего заполняемую область. Также, как в предыдущих алгоритмах, считается, что строки сканирования проходят через центр строк пикселов, то есть через середину интервала y. После определения пересечений вычисляем самый левый пиксел, расположенный правее точки пересечения, у которого абсцисса x больше абсциссы точки пересечения.

1. Заполнение области. Для каждой строки сканирования, имеющей точки пересечения с многоугольником, ограничивающим область, выполняется следующая последовательность действий:

а) f=0, (f-промежуточная переменная (флаг), показывающая расположение очередного пиксела: fl=0 - пиксел лежит вне области заполнения, fl=1 - пиксел лежит внутри заполняемой области);

x=xl (xl - левая граница);

б) пока (x<=xr ) (xr - правая граница) выполнить следующие действия:

если пиксел (x,y) имеет граничное значение , то проинвертировать значение переменной f (f=0, если было f=1; f=1, если было f=0);

если f=1, то присвоить пикселу (x,y) цвет многоугольника, в противном случае, присвоить пикселу (x,y) цвет фона;

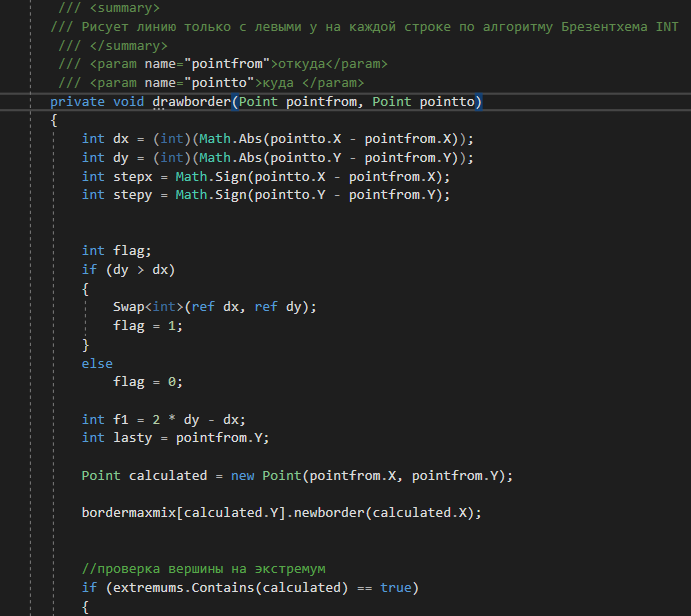
x=x+1.

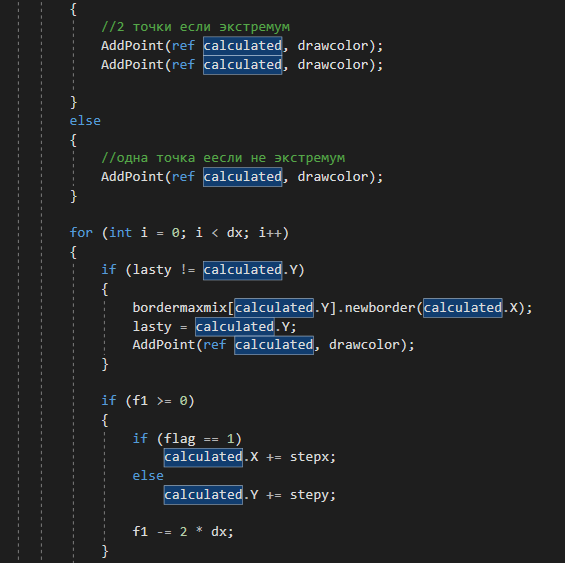
в) переход к следующей сканирующей строке y=y+1.

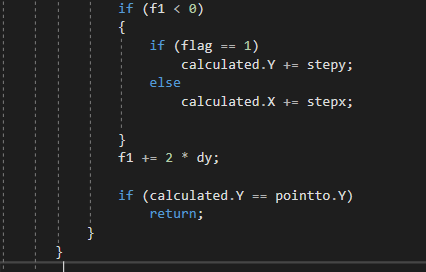
В рассмотренном алгоритме каждый пиксел обрабатывается только один раз, поэтому затраты на ввод/вывод значительно меньше, чем в алгоритме по ребрам или алгоритме с перегородкой. Ни один из этих трех алгоритмов (если они работают с буфером кадра) не требует построения, сортировки и поддержки каких-либо списков. При программной реализации алгоритм с упорядоченным списком ребер и алгоритм со списком ребер и флагом работают приблизительно с одной скоростью. Однако алгоритм со списком ребер и флагом пригоден для аппаратной или микропрограммной реализации, в результате чего он может выполняться на один-два порядка быстрее, чем алгоритм с упорядоченным списком ребер. Этот алгоритм можно даже использовать при создании динамических изображений.

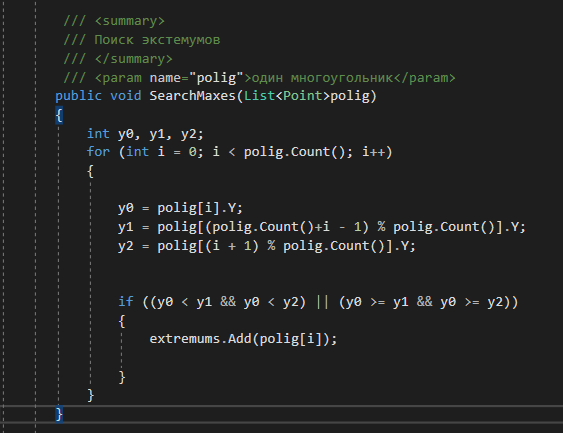
**Исходный код.**

**1.** Обрисовка контура

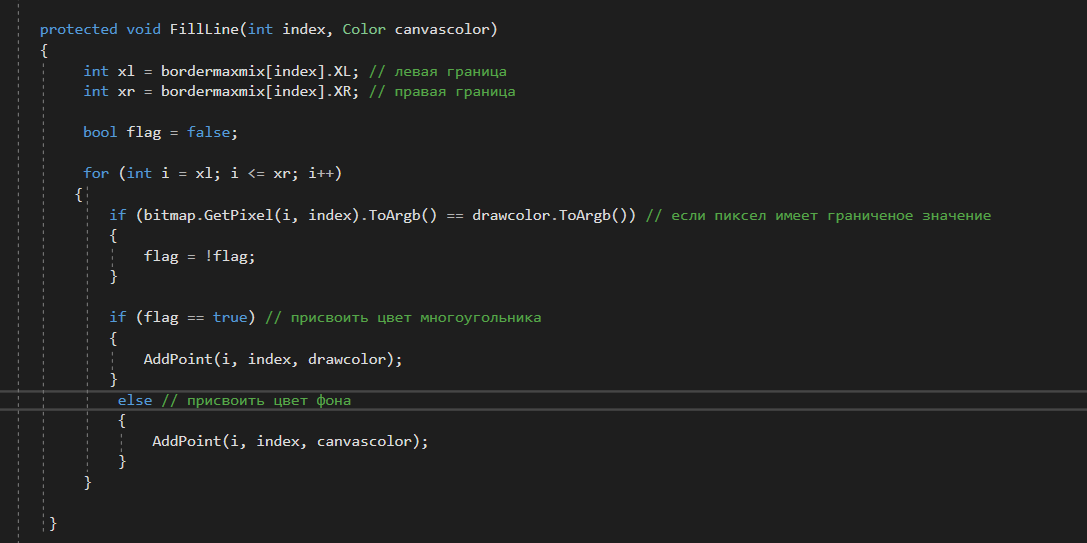
****

****

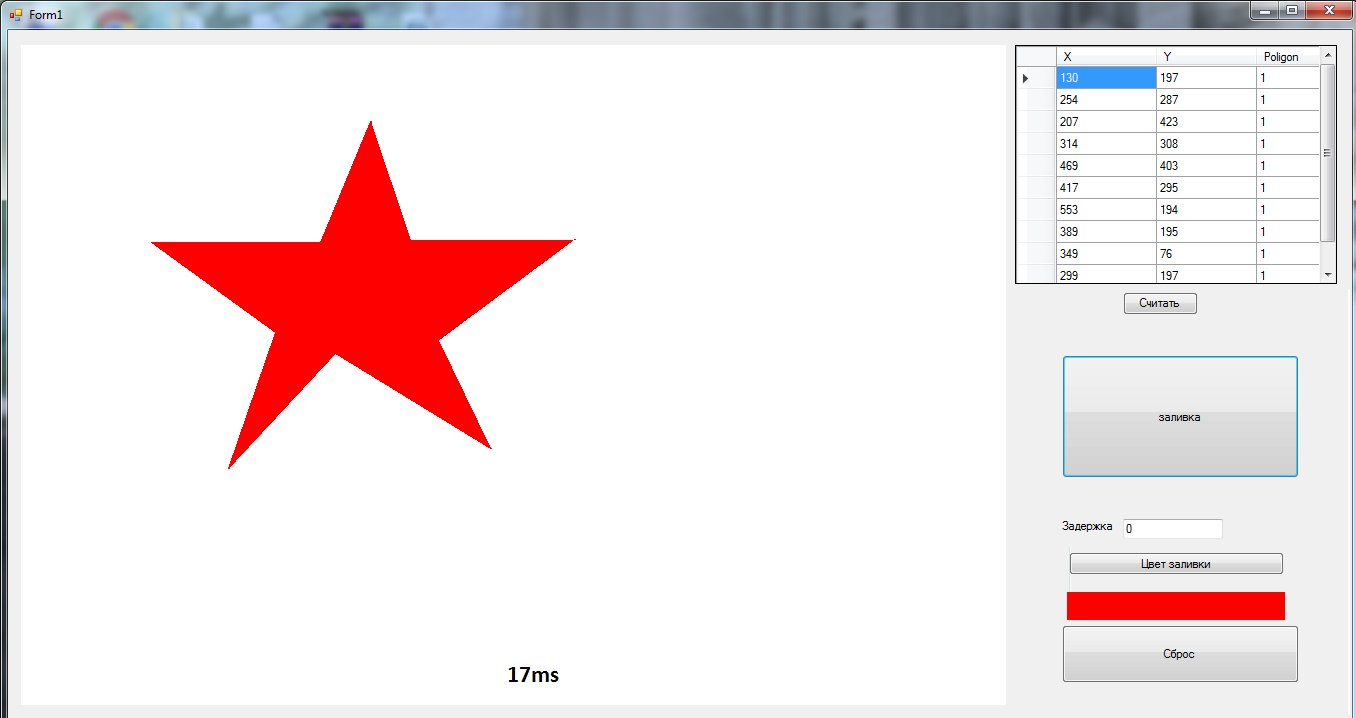
****

****

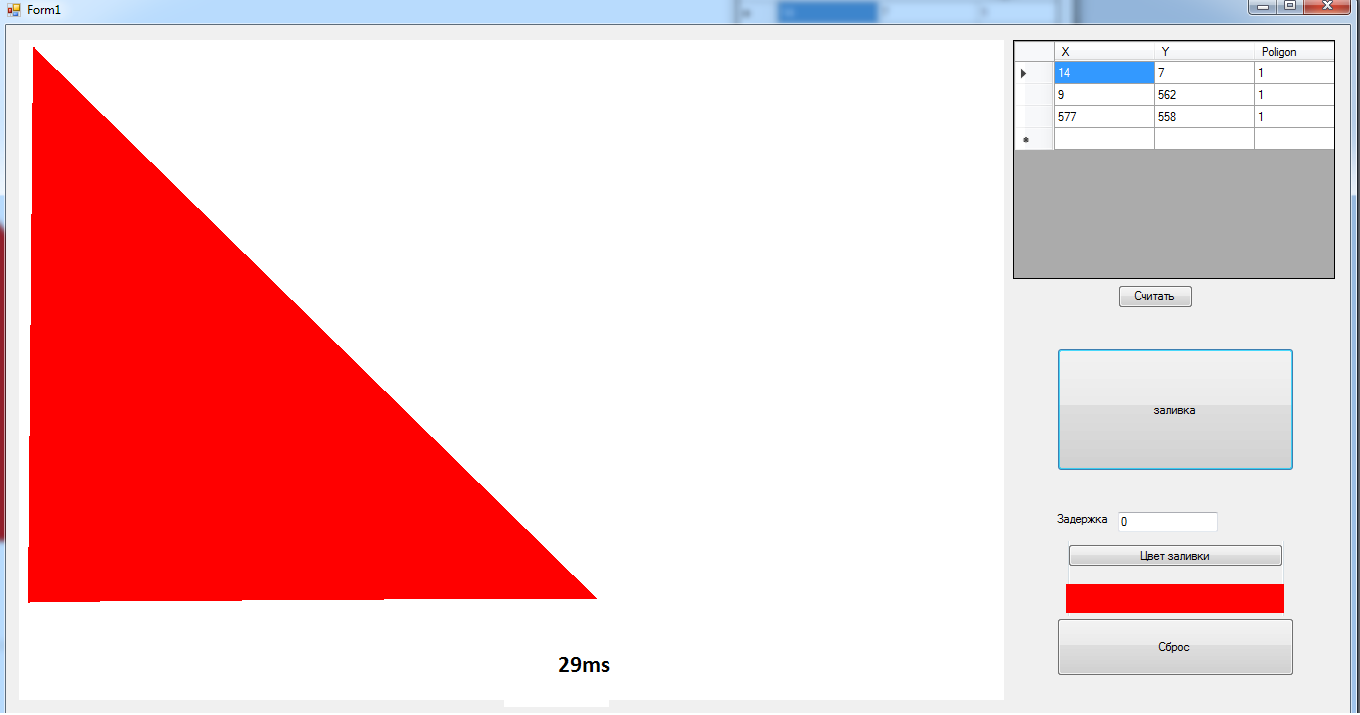
**2. Заполнение области.**

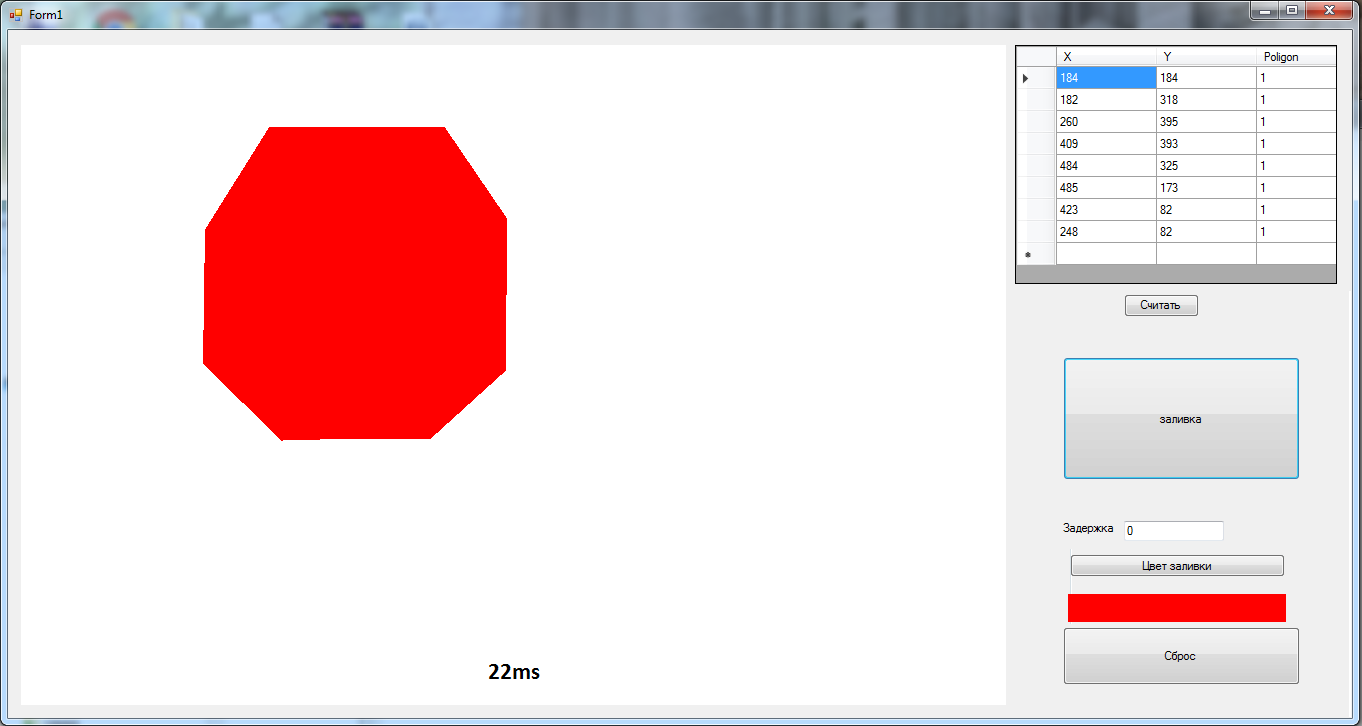
****

**Примеры работы программы.**









Пример работы с задержкой по ссылке(видео):  
<https://www.youtube.com/watch?v=-UC9AaB8JmM>

Пример работы с задержкой в виде фото:

