|  |  |
| --- | --- |
| 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**

|  |  |
| --- | --- |
| **Тема** РЕАЛИЗАЦИЯ И ИССЛЕДОВАНИЕ АЛГОРИТМОВ РАСТРОВОГО ЗАПОЛНЕНИЯ СПЛОШНЫХ ОБЛАСТЕЙ  **Студент** Склифасовский Д. О.  **Группа ИУ 7-45**  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |  |

Москва.

2020 г.

**Цель работы:**

Реализация и исследование одного из алгоритмов (Алгоритм заполнения со списком ребер и флагом) растрового заполнения области.

**Техническое задание:**

В рамках данной работы необходимо реализовать один из алгоритмов заполнения сплошной области.

Необходимо 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).

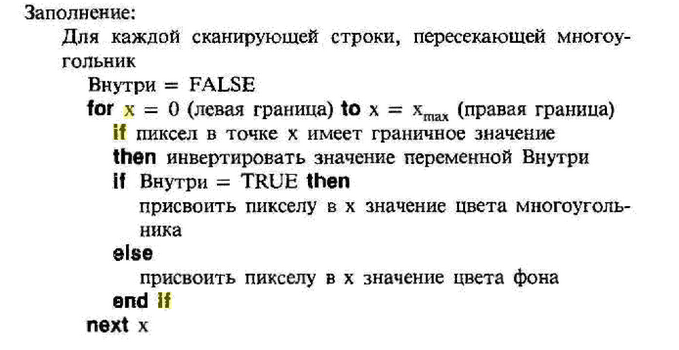
Обеспечить замер времени выполнения алгоритма (без задержки, с выводом на экран только окончательного результата).

**Теоретический материал:**

Все алгоритмы заполнения можно разделить на две группы – растровые и затравочные. В растровом алгоритме для каждой строки каждый раз решается задача определения, лежит пиксель внутри многоугольника или нет. Рассматриваются сканирующие строки, пересекающие многоугольник. Для пикселей каждой сканирующей строки определяется их принадлежность многоугольнику. Уменьшить трудоёмкость задачи можно если принять во внимание факт того, что пиксели, расположенные друг рядом с другом, сохраняют неизменным «свойство принадлежности».

**Алгоритм заполнения со списком ребер и флагом. (По Роджерсу)**

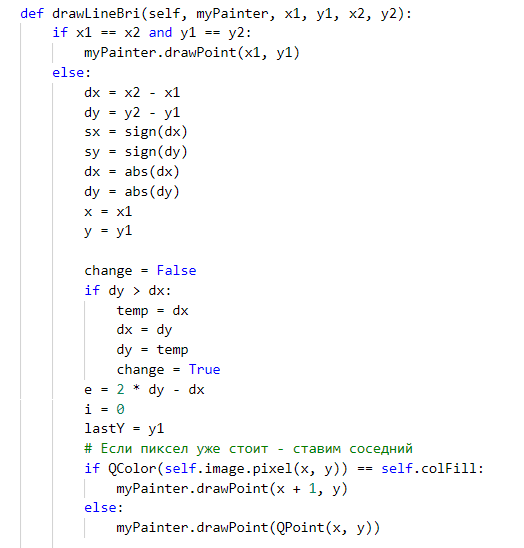
Алгоритм, использующий список ребер и флаг, является двухшаговым. Первый шаг состоит в обрисовке контура, в результате чего на каждой сканирующей строке образуются пары ограничивающих пикселов. Второй шаг состоит в заполнении пикселов, расположенных между ограничивающими. Более точно алгоритм можно сформулировать в следующем виде:

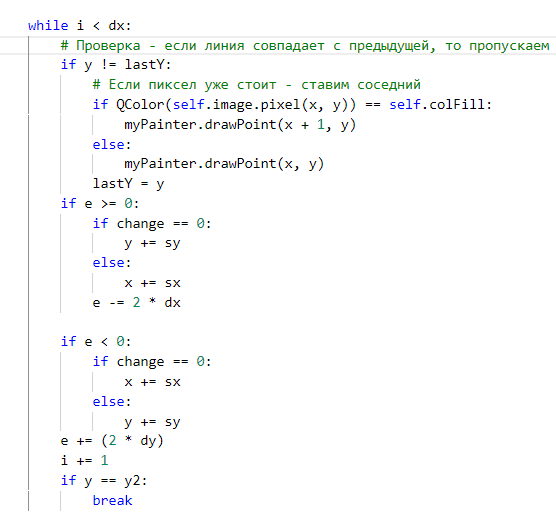


**Моя реализация:**

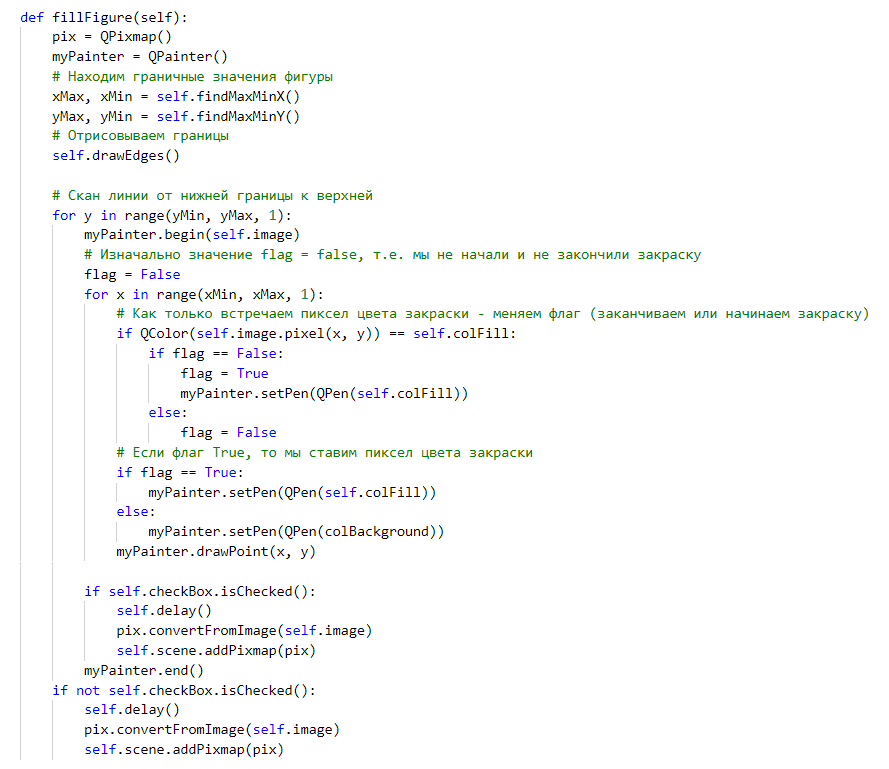
Основной проблемой для меня была отрисовка ребер. Изначально я использовал метод ЦДА для отрисовки каждого ребра, но в некоторых ситуациях случалось закрашивание фоновым цветом части многоугольника, так как пикселы ребра либо не отрисовывались (из-за округления) либо ставилось 2 соседних пиксела. В связи с этим я решил использовать целочисленный метод Брезенхема, так как он обрабатывает каждую строку. Так же, чтобы закраска начиналась сразу при встрече нужного пиксела, нужно было ставить только один пиксел на линии (ребра), вместо нескольких, поэтому я добавил проверку.

**Реализация метода Брезенхема:**





**Реализация алгоритма заполнения со списком ребер и флагом**



**Примеры:**

Сделал в качестве видео: <https://youtu.be/fj2wpb2Be4E>