**МГТУ им. Н.Э. Баумана**

**КОМПЬЮТЕРНАЯ ГРАФИКА**

**Лабораторный практикум №5**

**по теме: «*реализация и исследование алгоритмов растрового заполнения сплошных областей*»**

***Студент: Нгуен Фыок Санг***

***Группa: ИУ7И-46Б***

***Работу проверил:***

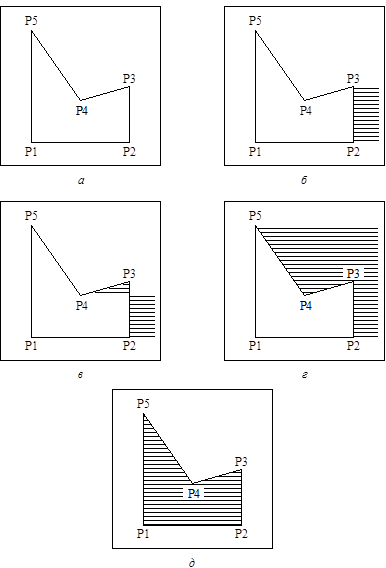
2020

**Цель работы: реализация и исследование одного из алгоритмов (по заданию преподавателя) растрового заполнения области.**

1. **Алгоритм заполнения по ребрам**

Вход: список ребер

* 1. ***Идея алгоритма:***



Для каждого ребра многоугольника AB, A(xA, yA), B(xB, yB)

1. Если ребро горизонтально (yA = yB): пропустить это ребро
2. Иначе:
   1. Использовать строки сканирования через середину (т.е. y + 1/2)
   2. Для каждой сканирующей строки, пересекающей ребро AB в (*x*1, *y*1), дополнить, т.е. активизировать, подсветить все пиксели, у которых центры лежат справа от (*x*1, *y*1), т.е. для (*x*, *y*), *x* + 1/2 > *x*1.
   3. ***Отзывы:***
      * К каждому ребру алгоритм применяется индивидуально, причем, порядок обработки ребер многоугольника не важен.
      * Рекомендуется использовать этот алгоритм совместно с дисплейным файлом.
      * Для сложного изображения каждый пиксель может обрабатываться много раз. Следовательно, эффективность алгоритма ограничена скоростью ввода/вывода.
      * Чтобы уменьшить количество обработок пикселя, рекомендуется окружить многоугольник прямоугольником (с ребрами, параллельными осям координат).
   4. ***Код программы:***

def filling\_by\_edges(window):

pix = QPixmap()

painter = QPainter()

painter.begin(window.image)

*#Найти самую правую точку области*

x\_max = max\_x(window.edges)

*# для каждого ребра*

for edge in window.edges: *# edge = [ [xA, yA], [xB, yB] ]*

*# если горизонтальное ребро(yA = yB) - пропустить это ребро*

if edge[0][1] == edge[1][1]:

continue

*#Если точка A выше, чем B, поменять местами*

if edge[0][1] > edge[1][1]:

edge[0][1], edge[1][1] = edge[1][1], edge[0][1]

edge[0][0], edge[1][0] = edge[1][0], edge[0][0]

dx = (edge[1][0] - edge[0][0])

dy = (edge[1][1] - edge[0][1])

*#смещение по х (при dy = 1)*

m = dx \* 1.0 / dy

y = edge[0][1]

y\_end = edge[1][1]

x\_st = edge[0][0] + m \* 1 / 2.0 *# T.(x\_st, y + 1/2) - пересечение*

while y < y\_end:

x = int(x\_st)

if (x + 0.5 <= x\_st):

x += 1

*# (x, y) - Самая левая точка находится справа от ребра*

*#активизировать, подсветить все пиксели,у которых*

*#центры лежат справа от Пересечения*

while x <= x\_max:

pixel\_color = QColor(window.image.pixel(x, y))

if pixel\_color == font\_color:

painter.setPen(QPen(fill\_color))

else:

painter.setPen(QPen(font\_color))

painter.drawPoint(x, y)

x += 1

*# перейти к следующей строке*

x\_st += m

y += 1

# с задержкой

if window.isDelay.isChecked():

delay()

pix.convertFromImage(window.image)

window.scene.addPixmap(pix)

delay()

painter.end()

# без задержки

if not window.isDelay.isChecked():

pix.convertFromImage(window.image)

window.scene.addPixmap(pix)

