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 (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

О Т Ч Е Т

по лабораторной работе № 7

Название: Реализация алгоритма отсечения отрезка регулярным отсекателем

Дисциплина: Компьютерная графика

Студент	ИУ7-43Б	27.05.2020	С.С. Кононенко
	(Группа)	(Подпись, дата)	(И.О. Фамилия)
Преподаватель			А.В. Куров
		(Подпись, дата)	(И.О. Фамилия)

Москва, 2020

1. Цель работы

Изучение и программная реализация алгоритма отсечения отрезка.

2. Техническое задание

Необходимо обеспечить ввод регулярного отсекателя — прямоугольника. Высветить его первым цветом. Также необходимо обеспечить ввод нескольких (до десяти) различных отрезков (высветить их вторым цветом). Отрезки могут иметь произвольное расположение: горизонтальные, вертикальные, имеющие произвольный наклон. Ввод осуществлять с помощью мыши и нажатия других клавиш. Выполнить отсечение отрезков, показав результат третьим цветом. Исходные отрезки не удалять.

3. Теоретический материал

Отсечение - это операция удаления изображения за пределами выделенной области, называемой окном. Чтобы выполнить данную операцию, необходимо прежде всего задать тип отсекателя.

Регулярным (стандартным) отсекателем на плоскости является прямоугольник со сторонами, параллельными координатным осям объектного пространства или экрана. Такое окно задается левым, правым, верхним и нижним двумерными ребрами. Для выполнения отсечения необходимо задать абсциссы $X_Л$, $X_П$ левого и правого ребер и ординаты $Y_н$,$Y_в$ нижнего и верхнего ребер. Цель отсечения будет состоять в определении точек, отрезков или их частей, которые лежат внутри отсекателя.

Отрезок целиком лежит внутри окна, если обе его концевые точки лежат внутри окна. Однако обратное утверждение, к сожалению, верно не всегда. Отрезок, концевые точки которого лежат вне окна, может быть как полностью невидимым, так и частично видимым. Полностью невидимым называется отрезок, целиком лежащий вне отсекателя. Частично видимым называется отрезок, одна часть которого лежит в пределах отсекателя, а другая - вне его. Если обе концевые точки отрезка невидимы, то он будет заведомо невидимым, если они (вершины отрезка) одновременно лежат левее или правее или ниже или выше окна.

Алгоритмы отсечения должны быстро отбирать отрезки, полностью лежащие внутри отсекателя (или же наоборот, полностью снаружи). Для этого используются коды концов отрезка. Код представляет из себя набор из 4-х единиц или нулей. Для первого бита ставится едиинца, если точка левее отсекателя. Для второго – если точка правее отсекателя. Для третьего – если точка ниже отсекателя, для четвертого – если точка выше отсекателя. В противных случаях ставится единица. Если побитовое произведение кодов концевых точек отрезка не равно нулю, то отрезок полностью не видим.

Алгоритм разбиения отрезка средней точкой

В алгоритме используются коды концевых точек отрезка и проверки, выявляющие полную видимость отрезков. Суть алгоритма заключается в разбиение отрезка средней точкой (рекурсивно), до тех пор, пока не будет найдено пересечение.

Формально, алгоритм можно разбить на три этапа:

1) Если концевая точка видима, то она будет наиболее удаленной от видимой точки. Процесс завершен.
2) Если отрезок тривиально характеризуется как невидимый, то выходная информация не формируется. Процесс завершен.
3) Грубо оцениваем наиболее удаленную видимую точку путем деления отрезка пополам. Применить п.1 к $P_1P_{ср}$ и $P_{ср}P_2$. Если $P_{ср}P_2$ тривиально не отвергается как невидимый, то средняя точка дает верхнюю оценку для наиболее удаленной видимой точки. Продолжаем процедуру с P_1P_m. Иначе, средняя точка дает оценку снизу для наиболее удаленной видимой точки. Продолжаем процедуру с P_2P_m. Так продолжаем до того момента, пока отрезок не станет меньше заданной точности.

4. Реализация алгоритма

```
def midpointcut(root, cut, dot_start, dot_end, eps):
    i = 1
    while True:
        code_start = set_code(dot_start, cut)
        code_end = set_code(dot_end, cut)

        if code_start == 0 and code_end == 0:
            root.draw_line(dot_start, dot_end, root.res_color)
            return

        if code_start & code_end:
            return

        if i > 2:
            root.draw_line(dot_start, dot_end, root.res_color)
            return

        dot_r = dot_start

        if code_end == 0:
            dot_start, dot_end = dot_end, dot_r
            i += 1
            continue

        while get_distance(dot_start, dot_end) >= eps:
            dot_middle = [(dot_start[0] + dot_end[0]) / 2, (dot_start[1] +
dot_end[1]) / 2]
            dot_tmp = dot_start
            dot_start = dot_middle

            code_start = set_code(dot_start, cut)
            code_end = set_code(dot_end, cut)

            if code_start & code_end:
                dot_start = dot_tmp
                dot_end = dot_middle

        dot_start, dot_end = dot_end, dot_r
        i += 1
```

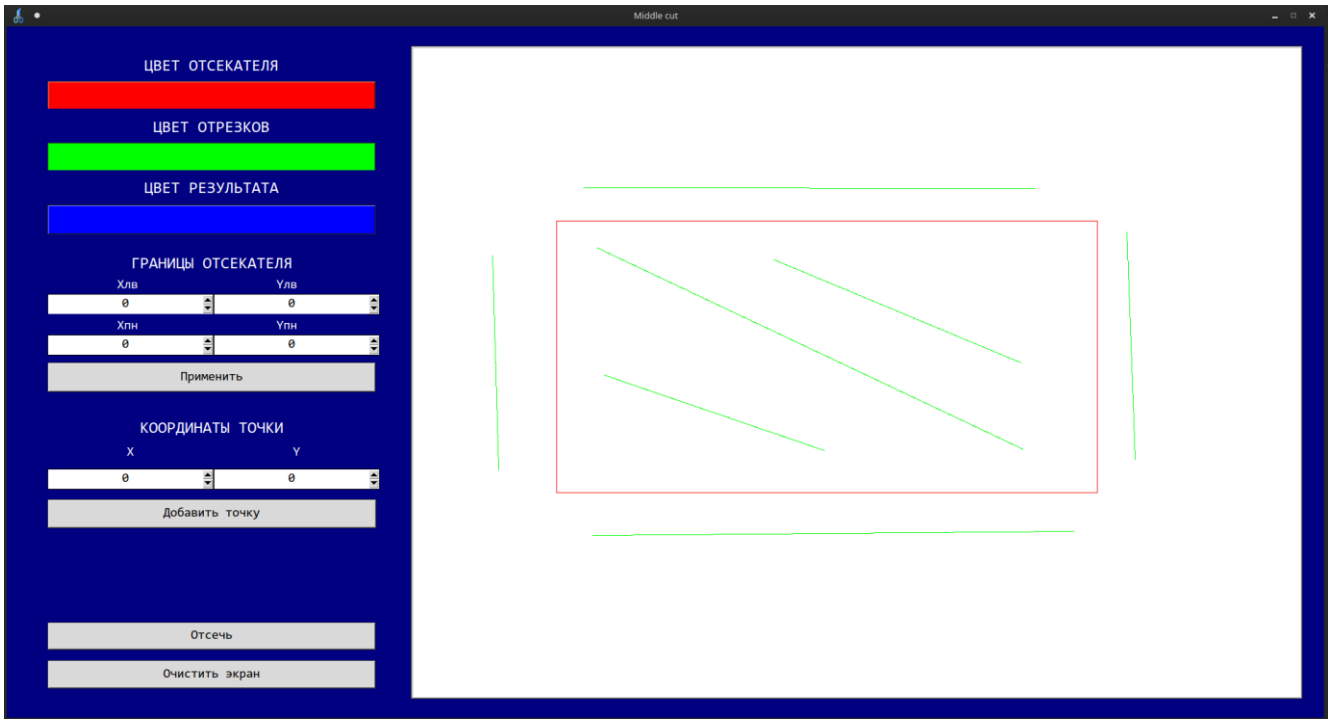
Реализация вспомогательных функций (нахождение кода конца отрезка, нахождение расстояния между точками):

```
LEFT = 0b0001
RIGHT = 0b0010
BOTTOM = 0b0100
TOP = 0b1000


def set_code(dot, cut):
    code = 0b0000
    if dot[0] < cut[0]:
        code += LEFT
    if dot[1] < cut[1]:
        code += TOP
    if dot[0] > cut[2]:
        code += RIGHT
    if dot[1] > cut[3]:
        code += BOTTOM

    return code


def get_distance(dot_start, dot_end):
    return sqrt((dot_start[0] - dot_end[0])**2 + (dot_start[1] - dot_end[1])**2)
```

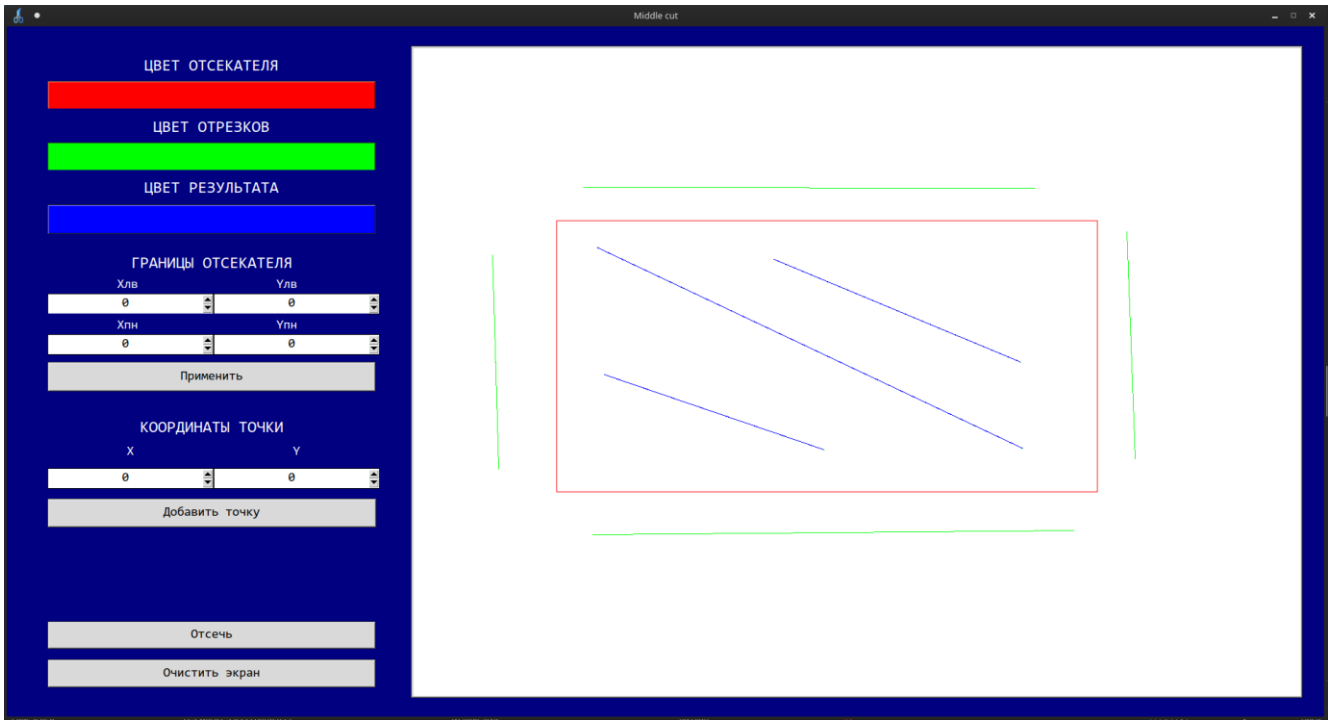
5. Демонстрация работы программы

Тривиальные случаи.

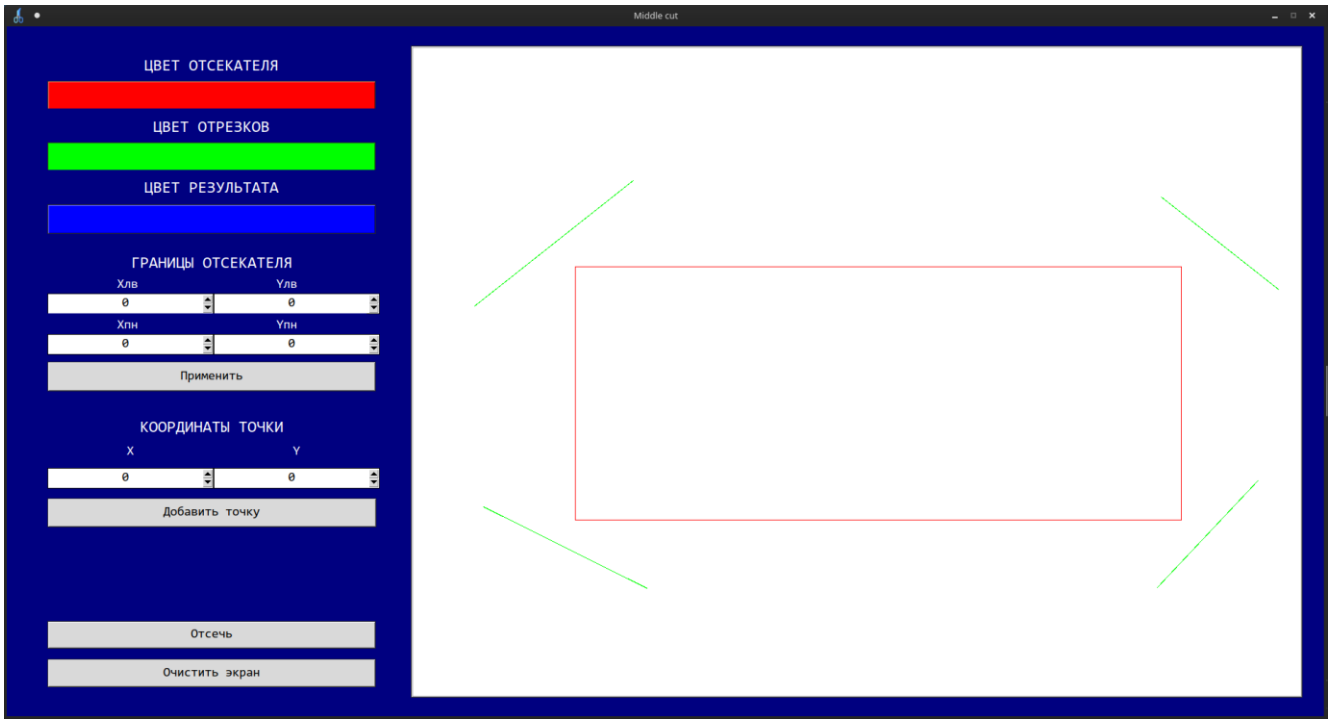
До:



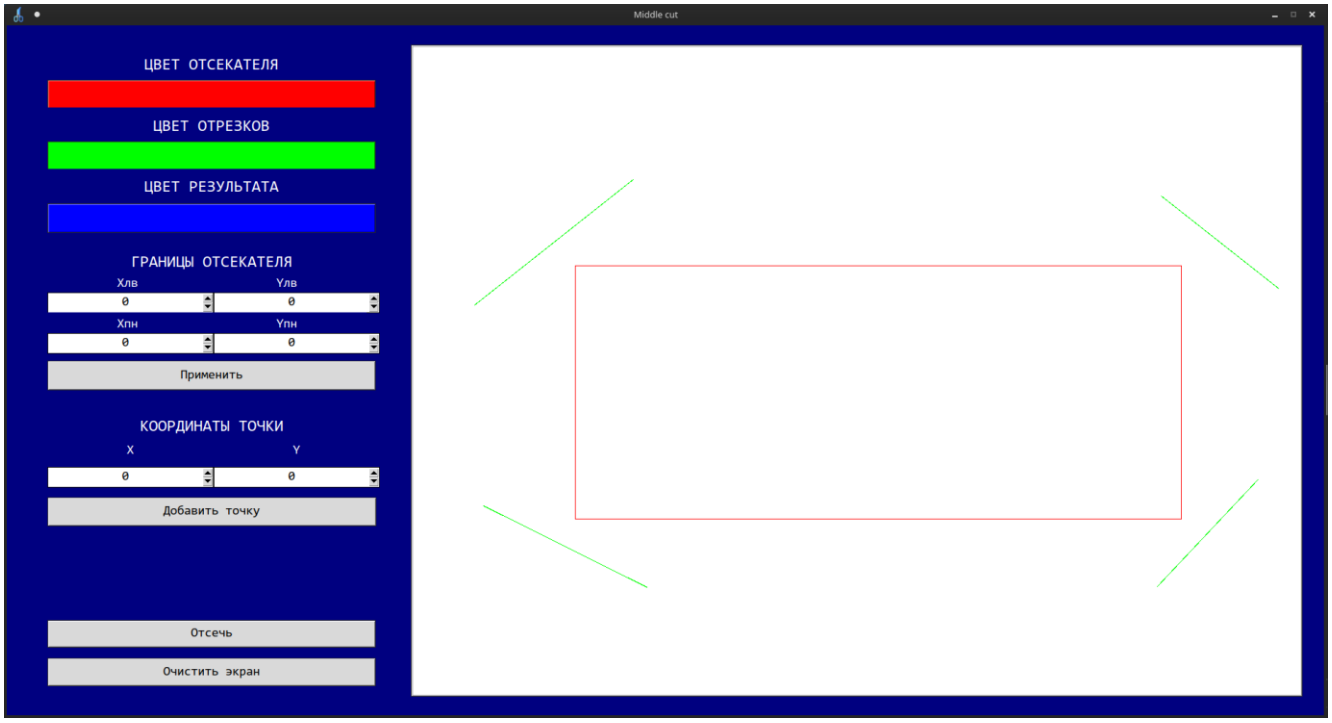
После:



До:

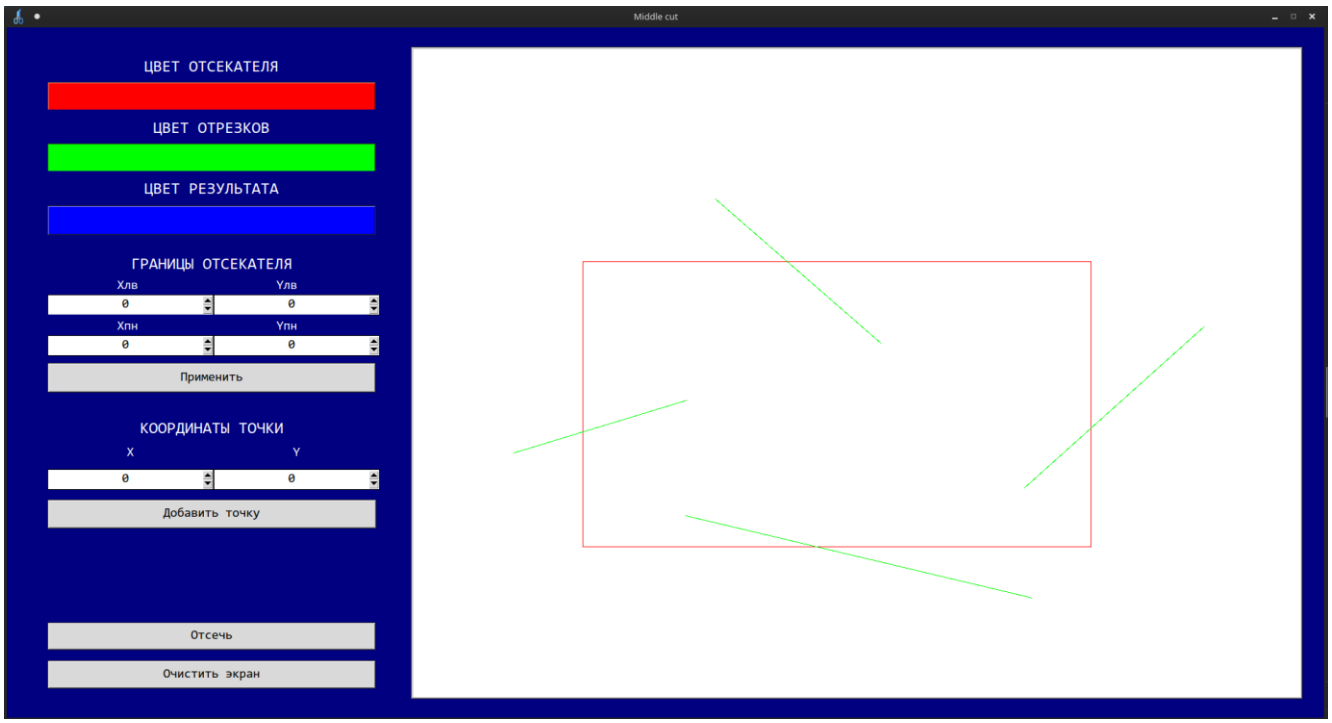

Middle cut
ЦВЕТ ОТСЕКАТЕЛЯ
ЦВЕТ ОТРЕЗКОВ
ЦВЕТ РЕЗУЛЬТАТА
ГРАНИЦЫ ОТСЕКАТЕЛЯ
Хлв
Улв
Хпн
Упн
0
Применить
КООРДИНАТЫ ТОЧКИ
X
Y
Добавить точку
Отсечь
Очистить экран

После:

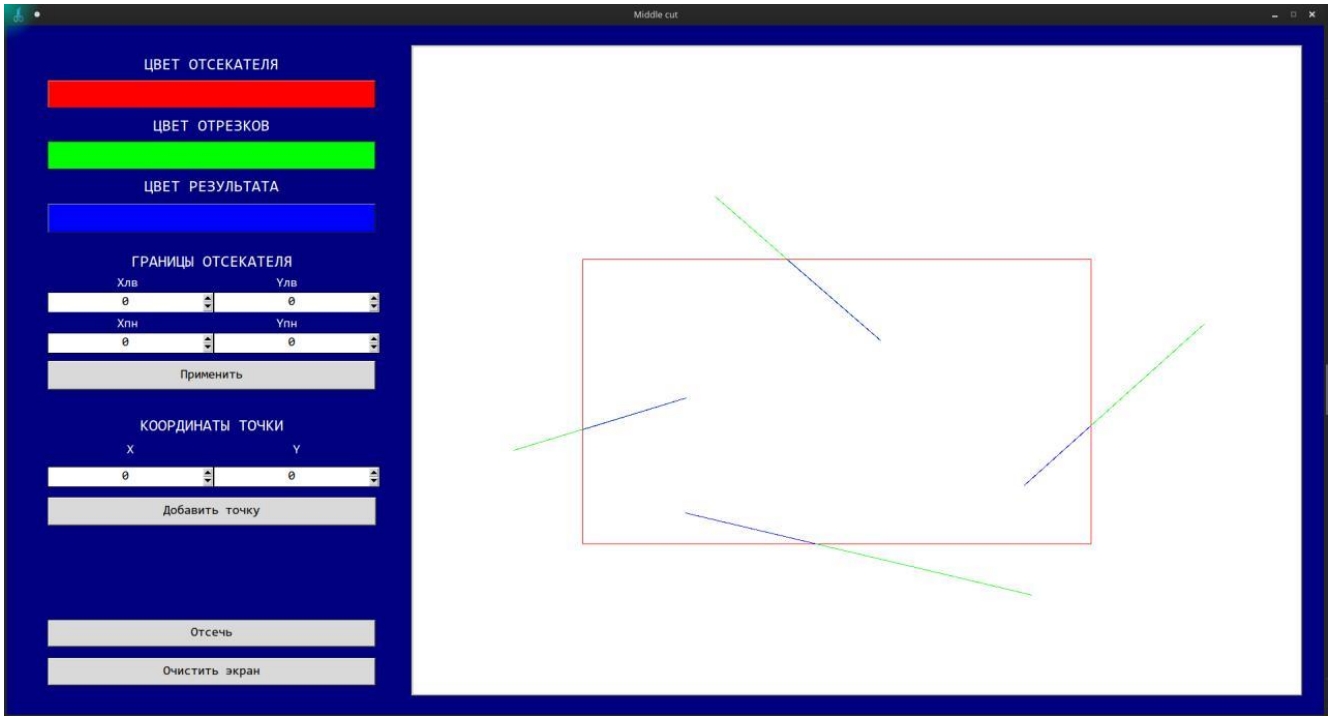

Middle cut
ЦВЕТ ОТСЕКАТЕЛЯ
ЦВЕТ ОТРЕЗКОВ
ЦВЕТ РЕЗУЛЬТАТА
ГРАНИЦЫ ОТСЕКАТЕЛЯ
Хлв
Улв
Хпн
Упн
0
Применить
КООРДИНАТЫ ТОЧКИ
X
Y
Добавить точку
Отсечь
Очистить экран

Пересечение отрезка и отсекателя в одной точке.

До:

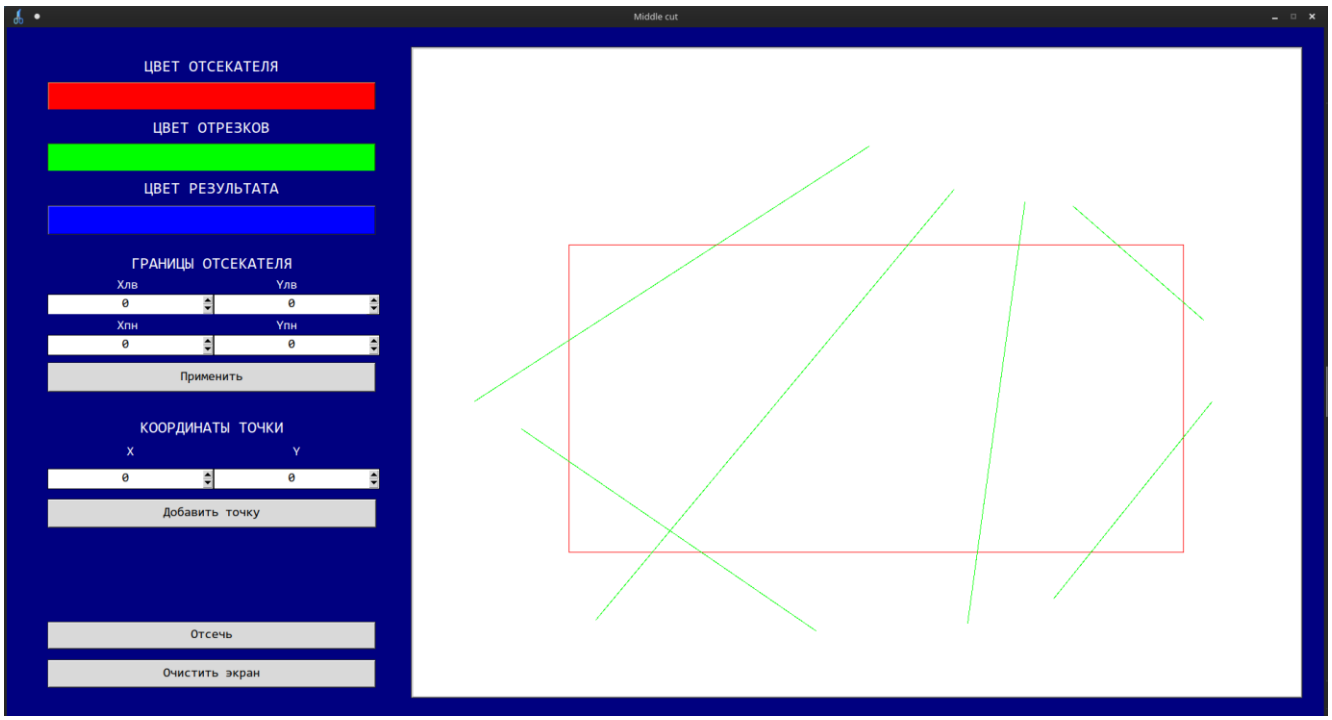


После:

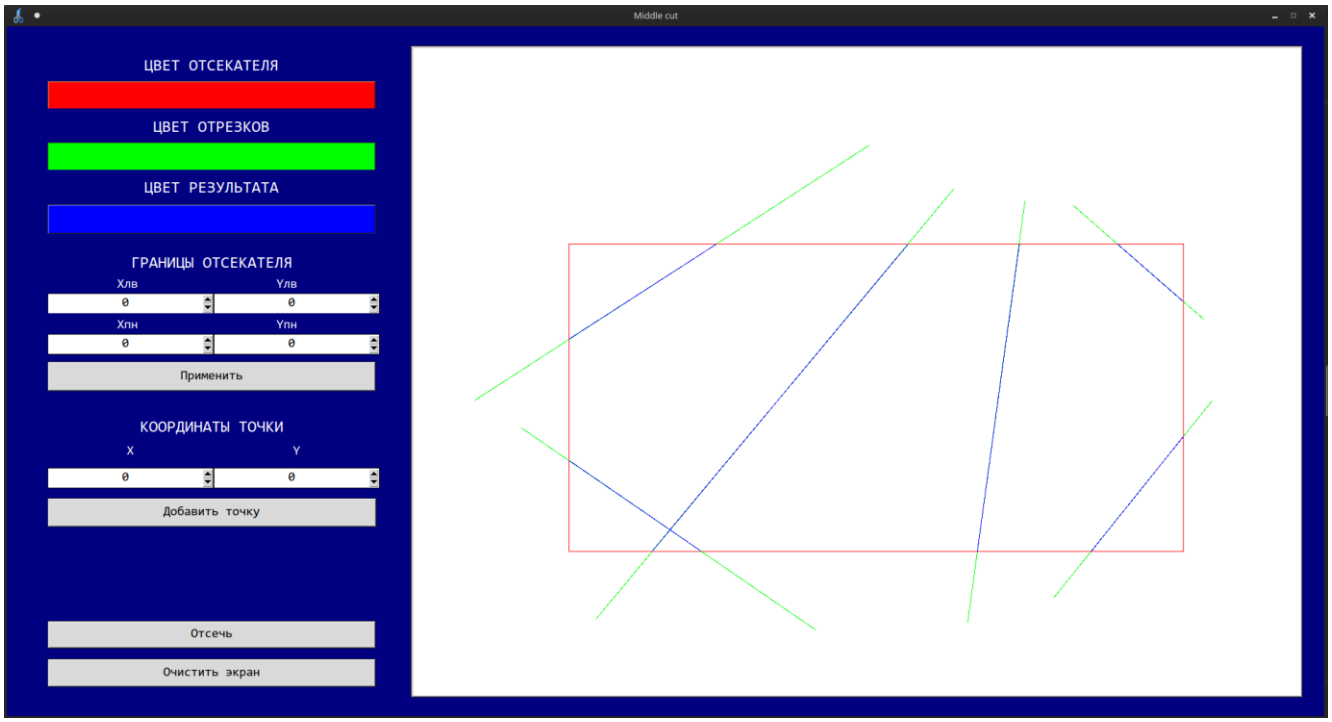


Пересечение отрезка и отсекателя в двух точках.

До:

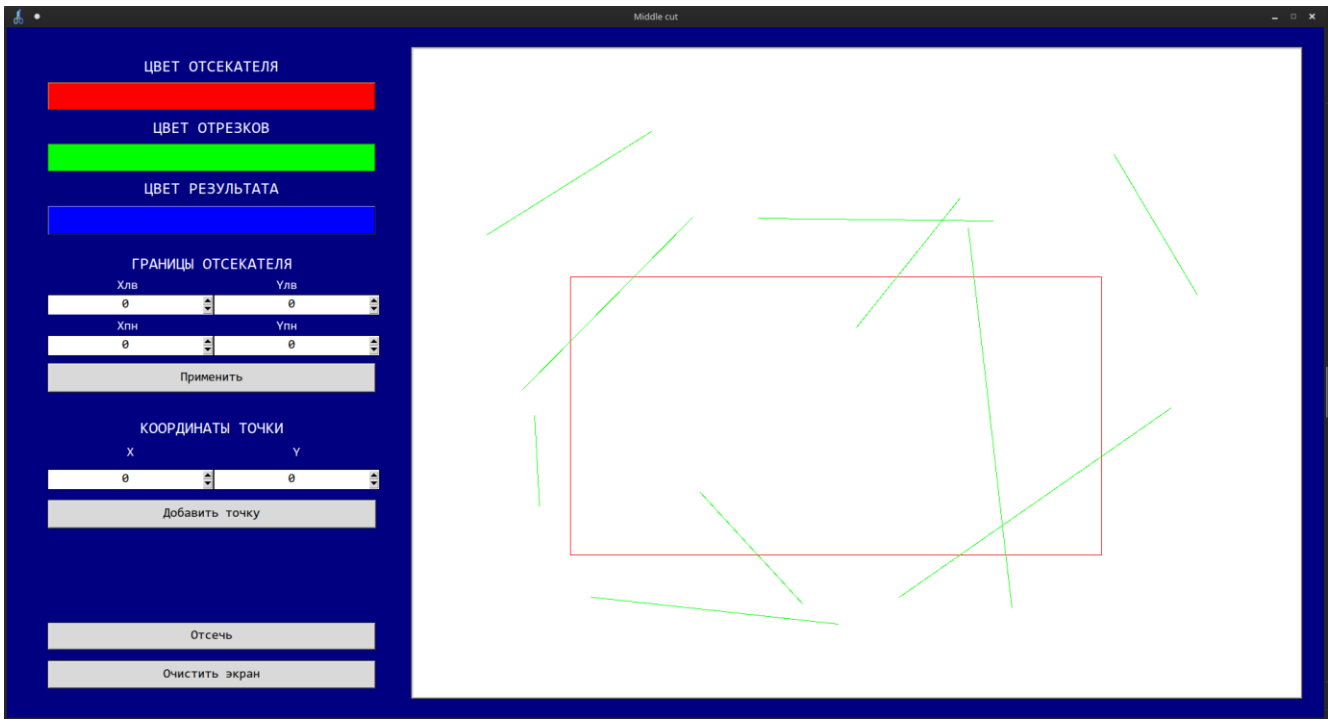


После:

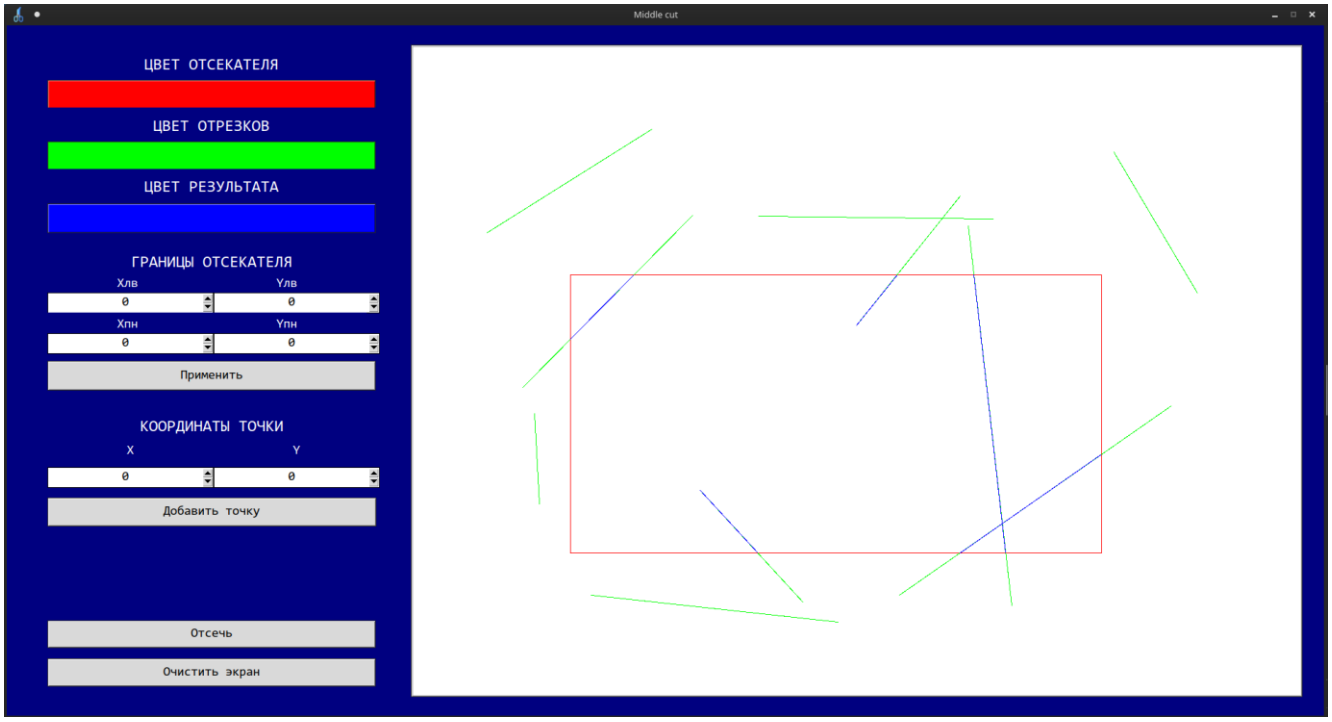


Пересечения как в одной, так и в двух точках, а также отсутствие пересечений.

До:

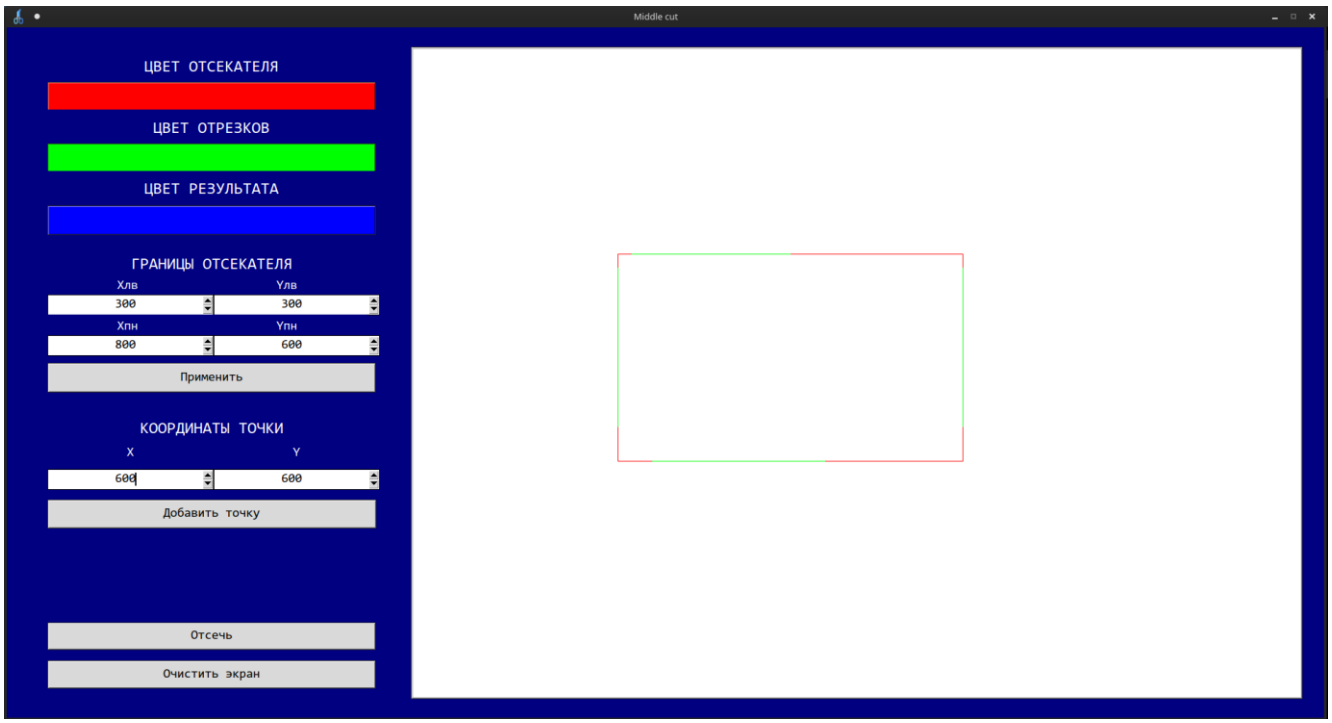


После:



Отрезки располагаются на границах отсекателя.

До:



После: