**Государственное бюджетное общеобразовательное учреждение "Президентский физико-математический лицей № 239"**

**«Пересечения прямых и треугольника»**

Годовой проект по информатике

Автор: Марюшко Виктор, 10-7 класс

Санкт-Петербург

2021

**Постановка задачи**

На плоскости задано множество точек, и "параллельный" прямоугольник. Множество точек образует все возможные прямые, которые могут быть построены парами точек множества. Найти такую прямую (и такие две точки, через которые она проходит), что эта прямая пересекает указанный прямоугольник, и при этом длина отрезка прямой, находящейся внутри прямоугольника, максимальна. В качестве ответа: выделить найденные две точки, нарисовать прямую, которая через них проходит, а также выделить на этой прямой отрезок между двумя найденными точками пересечения.

**Уточнение исходных и выходных данных**

**Входные данные**

В качестве входных данных выступает параллельный прямоугольник и список точек. Прямоугольник задается двумя точками-вершинами. Точки попарно не совпадают. Точки задаются координатами типа double, однако отрисовка точек с помощью OpenGL ограничивает эти координаты до отрезка [-1; 1].

**Выходные данные**

В качестве выходных данных выступают две точки, дающие максимальный отрезок, прямая, проходящая через них, и отрезок. Их нужно выделить.  
  
**Математическая модель**

1. **Пересечение прямых**

Треугольник состоит из 3-х сторон. Эти стороны принадлежат некоторым прямым. Рассмотрим, как узнать точку пересечения двух прямых. Пусть у нас есть две прямые:

Ax + By + C = 0

ax + by + c = 0

Преобразуем:

y = -A/B \* x - C/B

y = -a/b \* x - c/b

**x = (b \* C / B - c) / (a - A \* b / B)**

**y = -(A \* X + C) / B**

Кроме того, нужно проверить, принадлежит ли эта точка треугольнику. Что бы сделать это необходимо найти расстояние между точками, по которым строится отрезок(d) и расстояние от полученной точки до каждой из этих точек(d1 и d2) и если она принадлежит этому отрезку, то d = d1+d2.

1. **Расстояние между точками**

Найдя точки пересечения, которые задают отрезок, чтобы найти его длину, нужно воспользоваться формулой:

**D = √((x1 - x2)2 + (y1 - y2)2)**

**Анализ используемой структуры данных**

Нужно хранить треугольник, список точек(входные данные) для того, чтобы их нарисовать, а так же концы отрезка (ответ). Они хранятся в отдельном списке.

**Выбор метода решения**

Для решения задачи я перебираю все пары точек, строю по ним прямую и ищу ее пересечение с треугольником. Если есть пересечение, то проверяю его длину, если оно длиннее того, что находил до этого, то сохраняю эти две точки.

**Комментированный листинг**

ArrayList<Point> points;

//массив точек(входные данные)

Rectangle treug;

//треугольник

Point[] TruePoint = new Point[2];

//ответ

boolean ans;

//флаг, решена ли задача

public void solve(){

ans = false;

//ставим флаг на false

double maxDist = 0;

// максимальная дистанция

for(int i = 0; i < points.size(); i++){

for(int j = i+1; j < points.size(); j++){

//бежим по всем точкам

ArrayList<Point> t = treug.intersection(new Line(points.get(i), points.get(j)));

// строим линию по точкам и пересекаем

if(t.size() > 0){

//если есть пересечения

double dist = t.get(0).distanceTo(t.get(1));

//считаем длину

if(dist > maxDist){

//если она больше предыдущей

maxDist = dist;

//обновляем длину

ans = true;

TruePoint[0] = t.get(0);

TruePoint[1] = t.get(1);

//обновляем ответ, и ставим флаг, что задача имеет решение

}

}

}

}

}

**Пример работы программы**

triangle

(0,15; 0,74), (0,09; -0,51), (-0,57; 0,67)

points

(0,98; 0,24)

(0,38; -0,85)

(0,75; -0,12)

(-0,45; -0,59)

(-0,09; -0,63)

(-0,12; -0,49)

(0,38; 0,75)

(0,72; -0,94)

(0,61; 0,75)

(0,92; -0,06)

(0,91; -0,32)

(-0,85; -0,87)

(0,88; 0,27)

(-0,73; 0,92)

(0,90; -0,89)

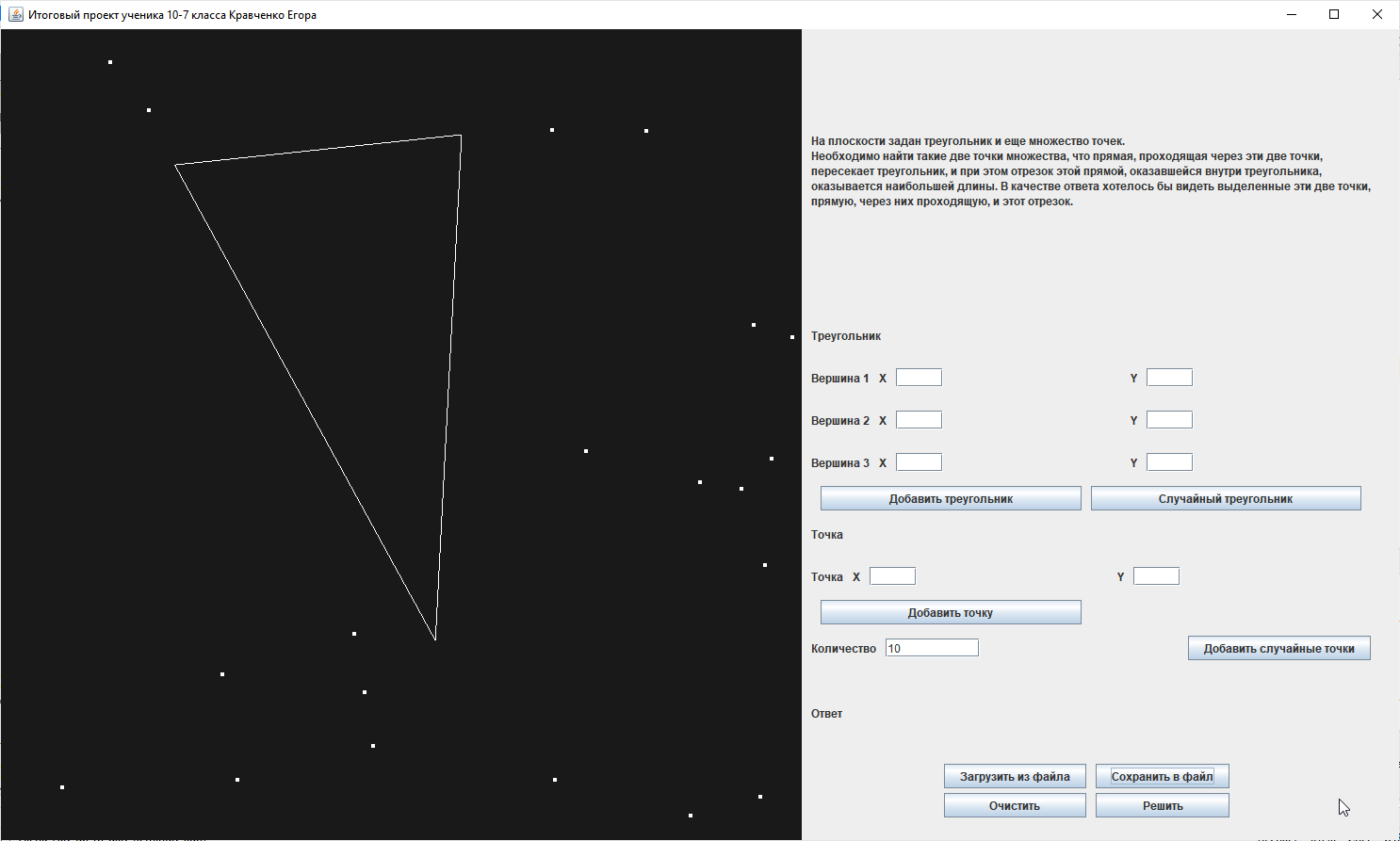
(0,85; -0,13)

(0,46; -0,04)

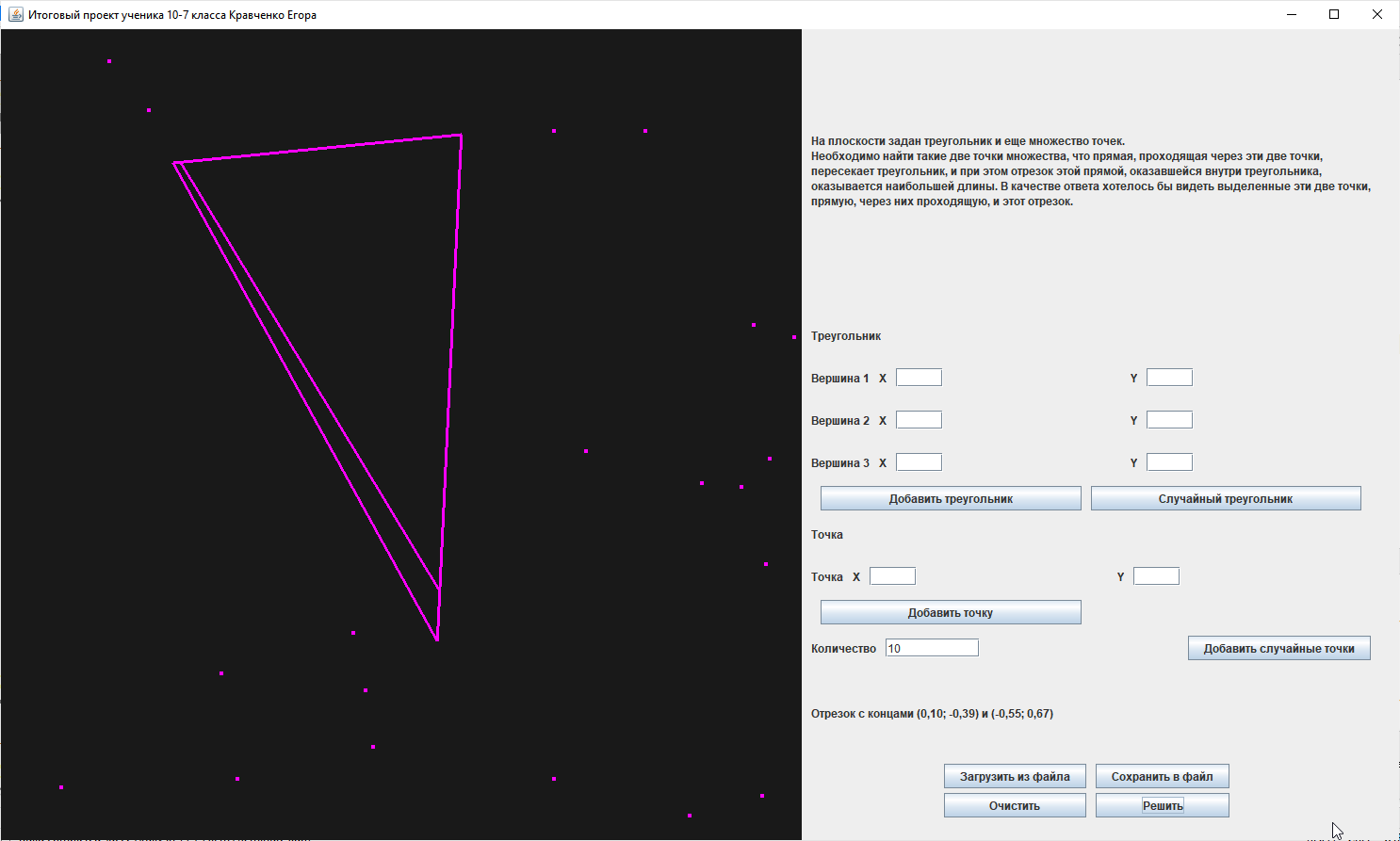
(-0,41; -0,85)

(-0,63; 0,80)

(-0,07; -0,77)



Отрезок с концами (0,09; -0,38) и (-0,55; 0,67)



**Анализ правильности решения**

В основе решения задачи используются точные и простейшие математические модели: пересечение прямых и расстояние между точками. Программа точно вычисляет точки пересечения и округляет их координаты до сотых из соображений эстетики. Приведу несколько примеров задач с маленьким количеством прямоугольников, решение которых очевидно, и программа его же и выдаёт.

