Физико-математический лицей № 239

**Нахождение максимальной площади пересечения треугольников**

Отчет о домашней работе по информатике

Работу выполнил

Ученик 10-2 класса

Колпачков Всеволод

1. **Постановка задачи**

На плоскости задано множество треугольников. Найти такие два треугольника, что площадь фигуры, находящейся внутри обоих треугольников, будет максимальна. Выделить найденные два треугольника, выделить контур фигуры, находящейся внутри обоих треугольников, желательно "залить цветом" внутреннее пространство этой фигуры.

1. **Уточнение исходных и выходных данных и ограничений на них**

**2.1 Исходные данные**

Во входном файле **input.txt** содержатся координаты треугольников. Координаты должны иметь вид (x1,y1, x2,y2, x3,y3) и каждая координата не должна быть больше 1 и меньше -1.

* 1. **Выходные данные**

Необходимо вывести на экран два треугольника, площадь пересечения которых максимальна, и саму эту площадь

1. **Математическая модель**

В задаче требуется найти площадь пересечения треугольников. Находится пересечение двух треугольников за счет пересечения прямых, которыми создается треуголник.

**4. Описание алгоритма**

Для того, чтобы решить поставленную задачу я буду делать следующее: создается треугольник, координаты его вершин хранятся в массиве, затем создается еще один треугольник, координаты вершин которого тоже хранятся в массиве. Каждому треугольнику присваивается тип “не используется ” в начале. Так как треугольники состоят из прямых, то зная уравнение каждой из прямой( находится по двум точкам), нахожу пересечения прямых, составляющих каждый треугольник. В массив записывается координаты точки пересечения каждой прямой. Получаю фигуру из с вершинами в точках пересечения прямых. Нахожу площадь полученной фигуры: найдя центр получившейся фигуры, разбиваю на треугольники, нахожу площадь каждого маленького треугольника, а затем складываю их. В массив записывается полученная площадь. Так делаю с каждой парой треугольников. Сравниваю полученные площади, нахожу максимум. Выделяю пару треугольников, площадь пересечения которых максимальна.

**5. Листинг программы**

|  |
| --- |
| public void addTriangle(double x1, double y1, double x2, double y2, double x3, double y3) { |
|  | triangles.add(new Triangle(new Vec2(x1, y1), new Vec2(x2, y2), new Vec2(x3, y3))); |
|  | } // Создание треугольника |
|  |  |
|  | public void solve() { |
|  | Triangle tri1 = new Triangle(), tri2 = new Triangle(); |
|  |  |
|  | for (Triangle tri : triangles) |
|  | tri.setType(UNUSED); // Устанавливаем всем треугольникам тип “не используется” |
|  |  |
|  | for (Triangle first : triangles) { |
|  | for (Triangle second : triangles) |
|  | { |
|  | if (first == second) |
|  | continue; |
|  |  |
|  | Polygon p = first.intersect(second); |
|  | if (p.area > polygon.area) { |
|  | polygon = p; |
|  | tri1 = first; |
|  | tri2 = second; |
|  | } |
|  | } |
|  | } // Сравниваем площади пересечения и выделяем треугольники, если площадь оказывается больше |
|  |  |
|  | for (Triangle tri : triangles) |
|  | if (tri == tri1 || tri == tri2) |
|  | tri.setType(CHOOSEN); |
|  | } // Устанавливаем на найденные треугольники тип “используется” |
|  |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |  |
|  | public void addRandomTriangles(int n) { |
|  | polygon = new Polygon(); |
|  | for (Triangle tri : triangles) |
|  | tri.setType(UNKNOWN); // Создаем рандомный треугольник и присваиваем ему тип “неизвестный” |
|  |  |
|  |  |
|  | for (int i = 0; i < n; i++) { |
|  | Triangle tri = new Triangle(); |
|  | tri.randomize(); |
|  |  |
|  | triangles.add(tri); |
|  | } |
|  | } // Сравниваем каждый треугольник с каждым и присваиваем тип “используется/не используется” |
|  |  |
|  |
|  |
|  |
|  |
|  |  |
|  | public void render(GL2 gl) { |
|  | for (Triangle tri : triangles) |
|  | if (tri.type != CHOOSEN) |
|  | tri.render(gl); |
|  |  |
|  | for (Triangle tri : triangles) |
|  | if (tri.type == CHOOSEN) |
|  | tri.render(gl); |
|  |  |
|  | polygon.render(gl); |
|  | } |
|  | } //Закрашиваем треугольники соответственно их типу : “используется” закрашиваем в яркий цвет и площадь их пересечения тоже, а “ не используется” в темный цвет |

**5.2 Пример программы**

**Входные данные: Выходные данные:**

* (0.2 ; 0.7) ; ( 0.6; -0.3) ; (-0.4 ; -0.5) (0.2 ; 0.7) ; ( 0.6; -0.3) ; (-0.4 ; -0.5)
* (0.0 ; 0.0) ; ( 0.8; -0.1) ; (-0.9 ; 0.9) (-0.7 ; -0.8) ; ( 0.3; -0.2) ; (-0.1 ; 0.5)
* (-0.7 ; -0.8) ; ( 0.3; -0.2) ; (-0.1 ; 0.5)

**6. Анализ правильности решения**

Поскольку пересечения двух треугольников может не быть, то и выделяться ничего не должно, что и делается. Также пересечением может быть прямая, то есть выделяться тогда должна именно эта прямая. Если ввести два одинаковых треугольника, то программа и выдаст эти два одинаковых треугольника