Реализация алгоритма Сазерленда-Ходжмана на языке C++

Чернов Ярослав, группа 206

13 апреля 2025 г.

1 Введение

Алгоритм реализует отсечение произвольного многоугольника другим многоугольником. Основная идея — на каждом шаге модифицировать исходный многоугольник, сохраняя только часть, находящуюся в правой полуплоскости от текущего ребра секущего многоугольника.

2 Используемые математические методы

2.1 Классификация точки относительно ребра

Для ребра $e = (org, dest)$ и точки p вычисляется **ориентированная площадь**, чтобы узнать ориентацию данной точки относительно данного ребра:

$$\text{sign} = (dest.x - org.x)(p.y - org.y) - (p.x - org.x)(dest.y - org.y)$$

- sign > 0: точка слева от ребра (LEFT)
- sign < 0: точка справа от ребра (RIGHT)
- sign $= 0$: коллинеарные точки (BEHIND/BEYOND/...)

```
36 PointClass Point::classify(const Edge& e) const {
37     Point a = e.dest - e.org;
38     Point b = *this - e.org;
39     double sa = a.x * b.y - b.x * a.y;
40     if (sa > 0.0) return LEFT;
41     if (sa < 0.0) return RIGHT;
42     if ((a.x * b.x < 0.0) || (a.y * b.y < 0.0)) return BEHIND;
43     if (a.length() < b.length()) return BEYOND;
44     if (e.org == *this) return ORIGIN;
45     if (e.dest == *this) return DESTINATION;
46     return BETWEEN;
47 }
```

Рис. 1: Реализация классификации точек относительно ребра в программе

2.2 Поиск пересечения отрезков

Для двух отрезков определяется точка пересечения (если она существует) путем решения системы линейных уравнений. Проверяются условия:

- Отрезки должны пересекаться в общих пределах (параметр t между 0 и 1)
- Вычисляются координаты точки пересечения через параметрическое представление
- Обрабатываются особые случаи параллельных и совпадающих прямых

3 Работа алгоритма

Алгоритм реализуется двумя ключевыми методами:

clipPolygonToEdge

- Принимает исходный многоугольник и ребро отсечения
- Строит новый многоугольник, сохраняя:
  - Вершины, лежащие внутри полуплоскости
  - Точки пересечения с ребром отсечения
- Возвращает отсеченный многоугольник

clipPolygon

- Последовательно применяет clipPolygonToEdge() для всех ребер секущего многоугольника
- На каждом шаге передает результат предыдущего отсечения
- Выполняет проверку на валидность промежуточных результатов
- Возвращает финальный отсеченный многоугольник

4 Клиент-серверная часть

4.1 Файл client.cpp

В файле client.cpp присутствует только функция main, которая подключает пользователя к серверу и позволяет передать на него данные о двух полигонах, после чего получает ответ и выводит результат на экран.

4.2 Файл server.cpp

В файле server.cpp реализованы все классы и алгоритмы, в том числе вышеуказанные clipPolygon и clipPolygonToEdge. Функция main запускает сервер, принимает данные от клиентов, выполняет для них алгоритм отсечения и возвращает результат работы алгоритма обратно.

5 Заключение

Вообще без понятия, что тут нужно писать, просто у других заключение было. Ну, алгоритм работает и на том спасибо.