



Presentación

Algoritmo Lepp-Delaunay

Gabriel Sanhueza Sanhueza

Arquitectura y librerías

- Arquitectura ModelView
- Lenguaje: C++
- Interfaz: Qt 5.8



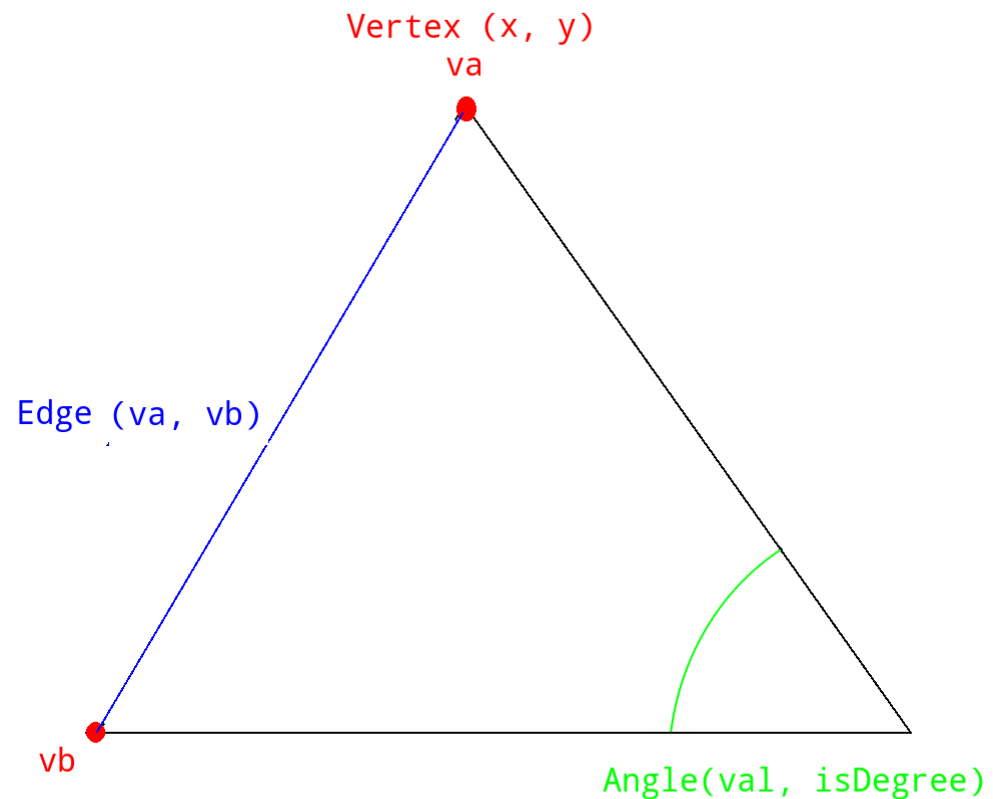
Code less.
Create more.
Deploy everywhere.

Archivos relevantes (.h,.cpp)

- Angle : Abstracción de ángulos
- Canvas : Pantalla de dibujo
- Constants : Alto y ancho de pantalla
- Edge : Detección de arista más larga
- Model : Algoritmos de refinamiento
- Triangle : Triángulos de *Vertexs*
- Vertex : Vértices (Puntos parseados)
- View : Ventana principal
- Main : Archivo para correr programa

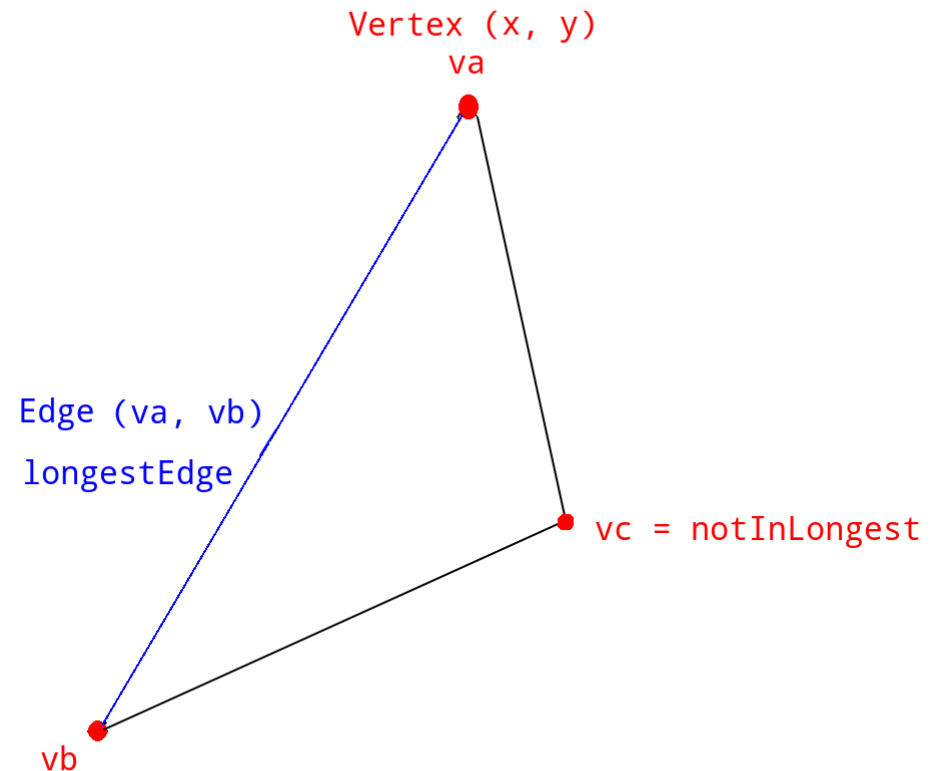
Estructuras de datos

- Vertex
 - int x
 - int y
- Edge
 - Vertex va
 - Vertex vb
- Angle
 - double val
 - Bool isDegree



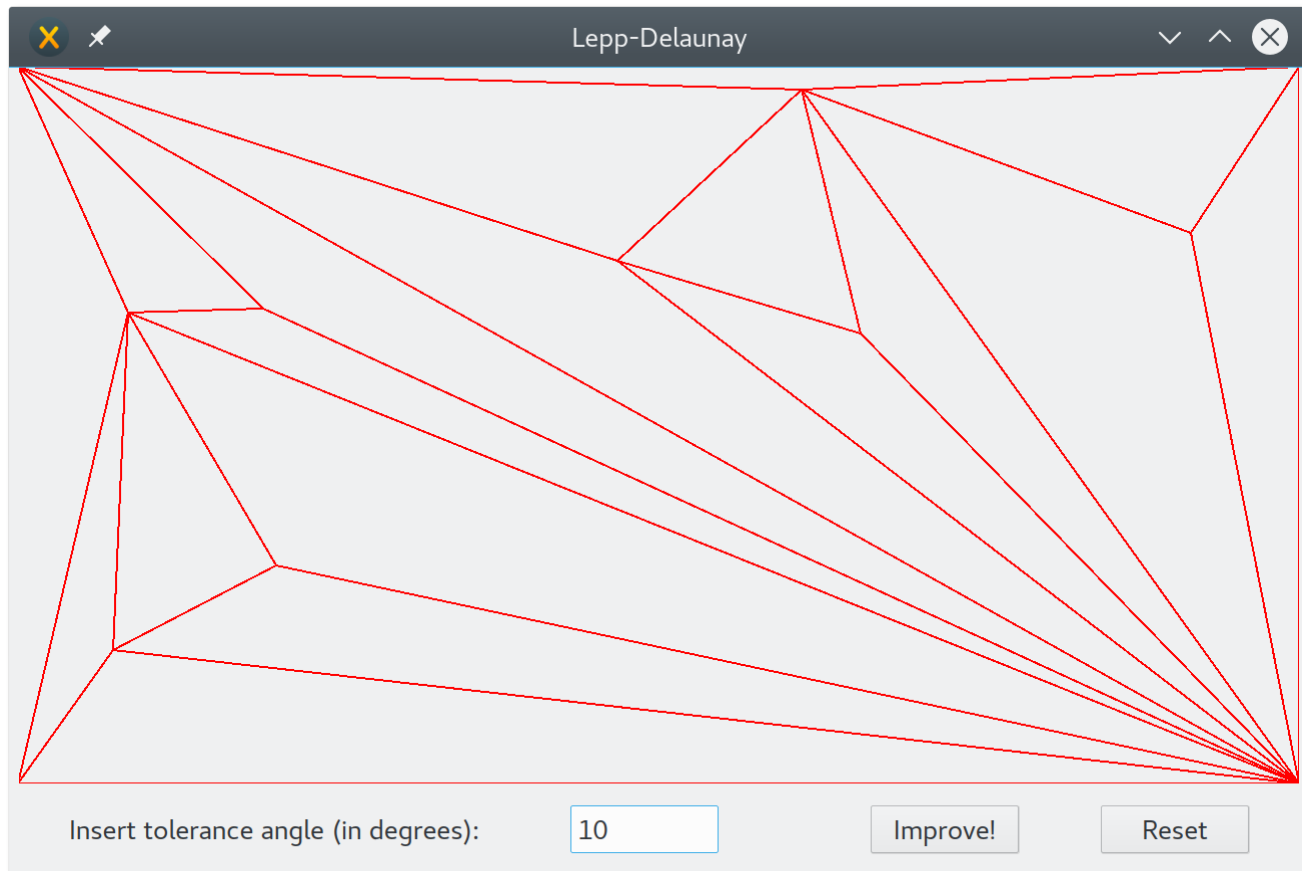
Estructuras de datos

- Triangle
 - Vertex va
 - Vertex vb
 - Vertex vc
 - Vertex notInLongest
 - Edge longestEdge



Estado inicial

- Triangulación parseada desde archivo
 - Input de tolerancia por parte del usuario



Algoritmo base

- Encontrar triángulos malos ($0 < O_{tol}$)
- Mientras hayan triángulos malos:
 - Tomar uno
 - Crear lista Lepp
 - Insertar en:
 - Centro de Edge más largo si en borde
 - Centroide si hay 2 triángulos terminales
 - Actualizar triángulos malos

Encontrar triángulos malos

- Para cada triángulo de la triangulación:
 - Si ángulo mínimo menor que tolerancia:
 - Agregar a lista de triángulos malos
- Retornar lista

Lista Lepp

- Por ahora está con fuerza bruta
- Para cada triángulo de la triangulación:
 - Si ya estoy viendo este triángulo, omitir
 - Si no,
 - Si triángulo lleva a borde
 - Agregar triángulo y marcar borde
 - Si triángulo A lleva a B y el vecino más largo vuelve a A
 - Agregar triángulo y marcar terminal
 - Actualizar edge más largo visto
- Retornar lista



Inserción en borde

- Tomar borde más largo del triángulo
- Dividirlo
 - Crear 2 triángulos respecto al dividido
 - Borrar el dividido

Inserción de centroide

- Tomar los 2 triángulos terminales
- Calcular centroide como el promedio de sus coordenadas
- Crear 4 triángulos con 2 de los Vertex anteriores y el centroide para cada uno
- Eliminar los 2 triángulos terminales de la triangulación
- Agregar los 4 nuevos triángulos



Actualizar triángulos

- Volver a escanear la triangulación
- Detectar los ángulos mínimos nuevamente

Resultados

