Tarea 1

Bea Graboloza

Enunciado:  
  
Dado un conjunto de puntos en 2D contenidos en el interior un rectángulo, construya una triangulación de calidad 'aceptable', en base a las dos operaciones 1 y 2 para construir una triangulación válida, y a la operación 3 para mejorar la calidad de dos triángulos vecinos  
  
 1. Inserción simple de un punto contenido en un triángulo, uniendo el punto con los vértices del triángulo.  
2. Inserción simple de un punto que está sobre una arista, uniendo el punto con los vértices del cuadrilátero que forman los triángulos que comparten la arista.  
3. Intercambio de la arista que comparten dos triángulos vecinos (que forman un cuadrilátero convexo) por la diagonal alternativa   
  
Se pide que:  
A. Diseñe un algoritmo INCREMENTAL (los puntos se agregan de uno en uno) que resuelva el problema en base a las ideas de las NOTAS y lo que discutiremos en clases. Escriba el algoritmo en seudo lenguaje.  
B.  Diseñe y discuta una estructura de datos para su implementación.  
C.  Implemente el algoritmo y úselo para construir triangulaciones de conjuntos      
      aleatorios de puntos, y para puntos ubicados sobre una grilla rectangular. Todos en el   
       interior de un rectángulo.

Pseudocódigo algoritmo inicial:

Dado el rectángulo inicial, dividido en 2 genera la triangulación inicial

For each punto en el conjunto de puntos:

Añadir el punto en su lugar

Encontrar un triángulo t de la triangulación actual que contenga el punto P  
if P está en el interior de t:

Realizar operación 1

else (si P esta sobre una arista):

Realizar operación 2

Revisar los ángulos de los 4 triángulos formados con esa operación

For each triangulo

Mejorar = false

For each angulo

If angulo < limit\_inf or angulo > limit\_sup:

Mejorar = True

If Mejorar:

Realizar operación 3 en ese triangulo

La estructura de datos escogida fue Half Edge, ya que con esta se pueden obtener todos los datos necesarios para las operaciones a partir de obtener 1 solo half Edge de la estructura, ya que con el half Edge puedes obtener el opuesto, el siguiente, el siguiente del siguiente, el anterior, su cara (triangulo) y así ir navegando toda la actual triangulación, concentrando toda la información en los half edges liberando a las caras y aristas de este peso, haciéndolas mucho más livianas en información haciendo en general la triangulación general menos pesada de lo que podría ser.

Se adjuntarán imágenes de referencias que acompañan al código adjunto

Imagen referencial para el método create\_triangle:

Gráfico, Gráfico radial

Descripción generada automáticamente

Imagen referencial para el método add\_point\_in\_face:

Gráfico, Gráfico radial

Descripción generada automáticamente

Imagen referencial para el método add\_point\_in\_edge:

Diagrama

Descripción generada automáticamente

Imagen referencial para el método change diagonal:

Diagrama

Descripción generada automáticamente