



PRACTICA 11: La envolvente convexa. Quickhull

Factor de ponderación [0-10]: 8

11.1. Objetivos

- Programación gráfica.
- Programación orientada a Eventos.
- Enfatizar la orientación a objetos en el desarrollo de aplicaciones.
- Desarrollo guiado por tests (TDD)
- Practicar el desarrollo de programas guiado por el comportamiento (BDD).

11.2. Contextualización

La envolvente convexa [1] de un conjunto de puntos en el plano es un polígono convexo cuyos vértices son algunos de los puntos del conjunto inicial de puntos.

En esta práctica se propone calcular y representar gráficamente la envolvente convexa de un conjunto de puntos del plano, generados aleatoriamente mediante el algoritmo concido como *Quickhull*.

11.3. Descripción

Desarrollar un programa `Quickhull.java` que calcule y represente gráficamente la envolvente convexa de un conjunto de puntos utilizando el algoritmo Quickhull [2].

Las siguientes deben tomarse como especificaciones de la aplicación a desarrollar:

- El programa ha de funcionar tanto como aplicación autónoma como en forma de applet. Compruebe su correcto funcionamiento tanto con `appletviewer` como con un navegador.
- El número de puntos que el programa genera de forma aleatoria para calcularles la envolvente, es un parámetro del programa que se introduce por línea de comando (o a través del fichero HTML del applet).

- También por línea de comando (o HTML) se introduce un segundo valor numérico que representa la velocidad de evolución del algoritmo: cuanto mayor sea este parámetro, más rápidamente veremos evolucionar el algoritmo en la interfaz gráfica.
- Si se ejecuta sin pasarle parámetros por línea de comandos, el programa deberá imprimir en la consola un mensaje explicativo de su modo de uso.
- La interfaz gráfica a desarrollar [3] consistirá en una ventana en la que se dibujan los puntos y se muestra la evolución del algoritmo hasta que finaliza su ejecución.
- La interfaz contendrá 4 botones: Inicializar, Ejecutar, Pausa, Paso. El botón Inicializar produce que se genere aleatoriamente un nuevo conjunto de puntos. El botón Ejecutar provoca la ejecución del programa. Pausa detiene momentáneamente la ejecución, y si se vuelve a pulsar, la reanuda. Paso a paso hace que la ejecución se ejecute paso a paso mostrando gráficamente la evolución del algoritmo.
- Trate de imitar (salvando las diferencias) la GUI [3] que se propone como ejemplo.
- Desarrolle un conjunto de tests unitarios de comportamiento que aseguren el correcto funcionamiento de la aplicación. Desarrolle estos tests ANTES del código de la aplicación propiamente dicha e intente que los tests evalúen situaciones no triviales.
- Evalúe el cubrimiento de código que realizan los tests desarrollados.
- Evite utilizar clases o métodos que no hayan sido explicados en las clases de teoría.

11.4. Referencias

[1] La envolvente convexa.

http://es.wikipedia.org/wiki/Envolvente_convexa

[2] QuickHull

<http://en.wikipedia.org/wiki/QuickHull>

[3] Una interfaz gráfica de usuario para el algoritmo Quickhull.

<http://www.cs.princeton.edu/courses/archive/spr09/cos226/demo/ah/QuickHull.html>