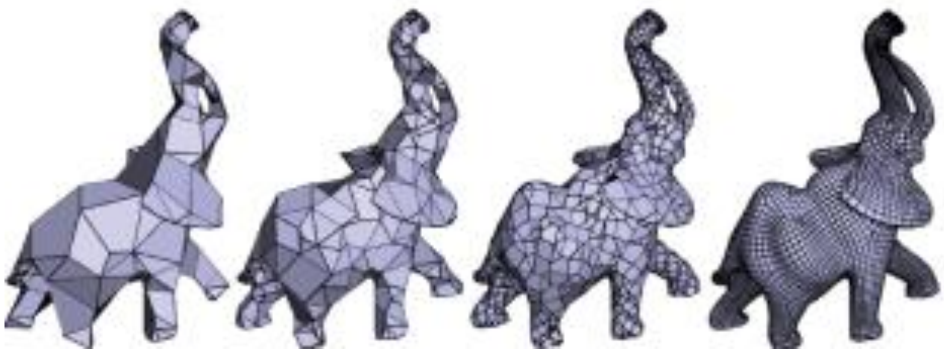
Instituto Tecnológico de Costa Rica

Escuela de Computación

IC3002 Análisis de Algoritmos – Grupo 1

Segundo Semestre de 2016

Proyecto (válido por examen parcial 2)



Motivación: En Graficación por Computadora (CG, o Computer Graphics) se utilizan segmentos de recta para construir polígonos para crear imágenes en dos o tres dimensiones. Por ello, la Geometría Computacional tiene gran importancia en esta disciplina. Por supuesto, la técnica básica para representar en la computadora los espacios geométricos es establecer un sistema de coordenadas, y representar los puntos como pares o tripletes ordenados, según sea que lo que se represente esté en un espacio 2D o 3D. Las formas básicas, como rectas y círculos quedan determinados mediante parámetros, dependiendo del esquema de representación utilizado. Como siempre, hay varias formas de resolver los problemas en esta área, y una de ellas es simplemente utilizar las construcciones clásicas de la geometría euclídea, solo que con el lenguaje cartesiano, pues las construcciones euclídeas ya de por sí son algoritmos, justificadas por los teoremas. He aquí un par de ejemplos sencillos:

a) Dados dos puntos en posición, B y C, encontrar un tercer punto A tal que ΔABC es un triángulo equilátero.

b) Dados tres puntos en posición: B, C y G, asumir que los dos primeros son los vértices de un triángulo, y G es el centroide (punto de intersección de las medianas). Encontrar el punto A del triángulo ΔABC.

Como se ve en estos ejemplos, el camino geométrico puro brinda algoritmos que pueden involucrar menos operaciones, con una consecuente menor pérdida de precisión, y un mejor tiempo de ejecución.

Entonces, esta es una técnica diferente de programar y hacer algoritmos, y el objetivo de este curso es precisamente mostrar al estudiante las diferentes técnicas y posibilidades que hay para diseñar la solución a problemas computacionales.

El problema específico planteado: Dados tres puntos en posición, encontrar un cuarto punto tal que la suma de las distancias del punto a los puntos dados es mínima. Ese punto existe y es interior al triángulo definido por los puntos, y único si para todos los ángulos definidos por los puntos originales, la medida de tales ángulos es menor a ${}^{2\pi}/_{3}$. Ese punto es llamado el *punto de Fermat*.

El programa a desarrollar: el programa que los estudiantes deberán escribir, en lenguaje Python, deberá usar la construcción geométrica de la solución para encontrar las coordenadas del punto.

Elementos geométricos computacionales a considerar:

a. El programa usará un sistema de coordenadas para designar los puntos en un espacio de dos dimensiones, por lo que todo punto será representado por un par ordenado de números de punto flotante. Entonces para todo punto P del espacio se tendrá: P = (x,y), donde x, y son las coordenadas correspondientes a un sistema de referencia.
b. Para que este sea un espacio geométrico hay que declarar una función de distancia geométrica. Entonces si $P_1 = (x_1, y_1)$ y $P_2 = (x_2, y_2)$ son puntos, definiremos la longitud del segmento P_1P_2 como $|P_1P_2| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.
c. El punto medio de P_1P_2 tiene coordenadas $\left(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}\right)$.
d. Cualquier recta en el plano euclídeo tiene una ecuación de la forma
$$ax + by + c = 0$$
donde a, b, c son constantes con $a^2 + b^2 \neq 0$.

Observación 1. Si $b \neq 0$, sea $m = {}^{-a}/_{b}$, $d = {}^{-c}/_{b}$; el número *m* es la pendiente de la recta. La ecuación toma entonces la forma
$$y = mx + d$$

Observación 2. Si (x_0, y_0) es un punto de la recta, su ecuación es entonces
$$y - y_0 = m(x - x_0)$$

Y si (x_1, y_1) es otro punto de la recta, vale $y_1 - y_0 = m(x_1 - x_0)$; y la ecuación de la recta se convierte en

$$(y_1 - y_0)(x - x_0) - (x_1 - x_0)(y - y_0) = 0$$

Con el empleo de estas fórmulas es posible hallar la ecuación de una recta, dependiendo de los datos que se tengan de ella.

e. Un círculo tiene una ecuación de la forma

$$x^2 + y^2 + 2Gx + 2Fy + C = 0$$

donde (-G,-F) es su centro y vale $C = G^2 + F^2 - r^2$ donde r es su radio.
Claramente, tenemos que

$$\begin{aligned} x^2 + y^2 + 2Gx + 2Fy + C = 0 &\Rightarrow x^2 + 2Gx + y^2 + 2Fy + C = 0 \\ &\Rightarrow x^2 + 2Gx + y^2 + 2Fy + G^2 + F^2 - r^2 = 0 \\ &\Rightarrow x^2 + 2Gx + G^2 + y^2 + 2Fy + F^2 - r^2 = 0 \\ &\Rightarrow x^2 - 2(-G)x + G^2 + y^2 - 2(-F)y + F^2 = r^2 \\ &\Rightarrow (x^2 - 2(-G)x + G^2) + (y^2 - 2(-F)y + F^2) = r^2 \\ &\Rightarrow (x - (-G))^2 + (y - (-F))^2 = r^2 \end{aligned}$$

Donde esto último es una fórmula alternativa para un círculo de radio r y centro en (-G,-F).

f. Dos rectas $a_1x + b_1y + c_1 = 0$ y $a_2x + b_2y + c_2 = 0$ son paralelas o iguales si y solo si $a_1 : b_1 = a_2 : b_2$

g. Si dos rectas $a_1x + b_1y + c_1 = 0$ y $a_2x + b_2y + c_2 = 0$ no son paralelas, entonces su intersección es un punto (x_0, y_0) tal que satisface simultáneamente ambas ecuaciones. Ese punto es único.

h. Dos círculos pueden intersecarse en 0, 1 o 2 puntos. Nótese que encontrar el o los puntos de intersección equivale a resolver simultáneamente dos sistemas de ecuaciones cuadráticas.

Fundamento teórico:

Teorema 1: Sea ΔABC un triángulo. Construya sobre los lados AC y AB triángulos equiláteros ΔABD y ΔACE. (a) Demuestre que $\Delta ADC \cong \Delta ABE$. (b) Sea F el punto de intersección de DC y EB, demuestre que $\angle \mathrm{DFB} = {}^{\pi}/_{3}$. F es el punto de Fermat. (c) Demuestre que si se construye un tercer triángulo equilátero ΔBCG, la recta BA es concurrente con DC y BE en el punto F.

Teorema 2: Demuestre que en un triángulo tal que ningún ángulo es mayor a ${}^{2\pi}/_{3}$ el punto F del teorema anterior es tal que para todo punto P del plano diferente de F se cumple

$$|AF| + |BF| + |CF| < |AP| + |BP| + |CP|$$

(Indicación: utilice el teorema anterior)

Interacción con el usuario: El programa deberá ser capaz de recibir del usuario los tres puntos que se le propongan, deberá verificar que no hay entre ninguno de ellos un ángulo mayor a ${}^{2\pi}/_{3}$, y que los tres puntos son diferentes; y deberá luego encontrar el punto de Fermat **usando la construcción del teorema 1**. En caso de que entre los puntos se encuentre un ángulo “incorrecto” el programa lo dirá al usuario, indicando el ángulo “culpable” y se pondrá disponible para recibir un nuevo conjunto de puntos. Asimismo, un conjunto de puntos tales que dos o más de ellos son iguales se considerará inapropiado. Si los puntos forman un conjunto apropiado se procederá al cálculo de las coordenadas del punto de Fermat. El programa deberá dar al usuario las coordenadas de los puntos D y E, adicionalmente. No es obligatoria una interfase gráfica que muestre el plano y los segmentos y puntos involucrados, sin embargo la distribución Anaconda de Python viene provista de librerías para graficación sumamente atractivas, pero no es obligatorio usar tales recursos.

Sugerencia para las pruebas: el programa Geogebra, que es de distribución gratuita y de fácil instalación, permite que uno defina puntos y rectas con facilidad a partir de sus coordenadas y parámetros, y además puede encontrar puntos de intersección de rectas y círculos. Estas características pueden hacer de él un aliado valioso a la hora de verificar los resultados de su programa.

Detalles de evaluación:

a. Fecha de revisión: Jueves 24 de Noviembre de 2016. Las revisiones se efectuarán durante todo el día, de 10:00 am a 5:00 pm
b. Se podrá efectuar en grupos de a lo sumo dos personas.
c. Rubros de la calificación
    a. Interacción con el usuario (20%)
        i. Captura mediante interfase gráfica el conjunto de puntos (5%)
        ii. Detecta los conjuntos de puntos inapropiados. (15%)
    b. Ejecución del programa (65%)
        i. Encuentra y muestra correctamente el punto D (20%)
        ii. Encuentra y muestra correctamente el punto E (20%)
        iii. Encuentra y muestra correctamente el punto F (20%)
        iv. Calcula y muestra correctamente la suma de las distancias a los vértices A, B y C desde F. (5%)
    c. Elaboración teórica (15%)
        i. Demostración del Teorema 1. (5%)
        ii. Demostración del Teorema 2. (5%)
        iii. Cálculo del O() del programa. (5%)