



Universidad del Istmo de Guatemala
Facultad de Ingenieria
Ing. en Sistemas
Informatica 1
Prof. Ernesto Rodriguez - erodriguez@unis.edu.gt

Proyecto Final: Fractales

Fecha de entrega: 28 de Octubre, 2019 - 11:59pm

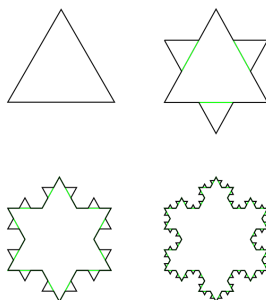
Instrucciones: Resolver cada uno de los ejercicios siguiendo sus respectivas instrucciones. El trabajo debe ser entregado a traves de Github, en su repositorio del curso, colocado en una carpeta llamada "Proyecto Final". Al menos que la pregunta indique diferente, todas las respuestas a preguntas escritas deben presentarse en un documento formato pdf, el cual haya sido generado mediante LaTeX.

A continuación se presenta el proyecto final. El objetivo de este proyecto es familiarizar al estudiante con la utilización de la programación funcional para crear programas interactivos. El objetivo es que el estudiante construya un programa que dibuje fractales.

El proyecto se puede llevar a en grupos de un maximo de 3 integrantes. Este proyecto tiene como valor 15 puntos de los 40 que conforman el examen final. Adicionalmente, existe la oportunidad de 15 puntos extra de los cuales 5 seran para el examen final y 10 para recuperar nota en los parciales anteriores.

Aparte de la entrega final, el dia 29 de Octubre, tendra que presentarle el codigo al profesor durante la clase. El profesor tiene derecho de hacerle preguntas sobre cualquier parte del codigo a cualquiera de los integrantes del grupo. La nota final del proyecto es individual (no grupal) y dependera de la habilidad de cada integrante en responder las preguntas que el profesor le haga sobre el codigo.

Copo de Nieve de Koch (5pts)



El copo de nieve de Koch (https://en.m.wikipedia.org/wiki/Koch_snowflake) es un poligono definido recursivamente. Dado un poligono (un triangulo por ejemplo), se puede generar el siguiente poligono asi:

1. Dividir cada linea del poligono en 3 partes exactas.
2. Crear un triangulo nuevo en el espacio abarcado por la linea del medio
 - La longitud de cada linea del triangulo nuevo es la misma que la linea del medio.

Su tarea consiste en definir una función llamada `snowflake` : $\mathbb{N} \rightarrow \text{List}(\mathbb{R}, \mathbb{R})$. Esta función debe aceptar un numero natural indicando la cantidad de veces que se repetira la recursion. El valor 0 representa el copo más simple de Koch (un tirangulo) y numeros mayores a 0 representan la cantidad de repeticiones que se llevaran a cabo en la producción del copo de nieve. El resultado de esta función es una lista de parejas de numeros reales (**Float**) que representan los vertices del copo de nieve en un plano cartesiano de coordenadas.

Triangulo de Sierpinski (5pts)

El triangulo de Sierpinski (https://en.wikipedia.org/wiki/Sierpinski_triangle) es otro fractal generado recursivamente. El caso base consiste en un triangulo regular. Luego, cada paso recursivo divide cada uno de los triangulos que conforman el triangulo mediante otro triangulo que cada lado del triangulo esta ubicado en la mitad de cada una de las lineas del triangulo anterior.

Su tarea consiste en definir la función `sierpinski` : $\mathbb{N} \rightarrow \text{List}(\text{List}(\mathbb{R}, \mathbb{R}))$ la cual debe aceptar un numero natural como parametro y generar una lista de listas de parejas de numeros reales. La interpretación de este resultado es que cada lista de numeros corresponde a uno de los muchos triangulos que conforman el triangulo de Sierpinsky.

Visualización (5pts)

Utilizar el modulo Html y Canvas de Elm para crear una interfaz grafica que permita visualizar el triangulo de Sierpinski y el copo de Koch. Esta interfaz debe tener lo siguiente:

- Un campo para seleccionar el fractal que se desea dibujar
- Un campo para seleccionar el numero de repeticiones que se utilizar para crear el fractal.
- Un area donde el poligono seleccionado es dibujado

Esta interfaz debe ser interactiva, lo que significa que se pueden modificar los valores y area de visualización dibuja el fractal con los parametros actualizados.

Extras (15pts)

Se otorgaran hasta 15 puntos extras por articulos adicionales en la entrega que esten relacionados al trabajo. Algunos ejemplos podrian ser:

- Utilizar css para mejorar la apariencia visual del programa
- Inclusion de otros fractales
- Fractales de 3 o más dimensiones
- Utilizar color diferente para cada recursion dibujada para mejor visualizacion del fractal.
- Opcion de dibujar el fractal de forma probabilistica. Ie. a los puntos que conforman las esquinas del fractal se les modifica la posición ligeramente mediante numeros aleatorios.

- Interactividad con el mouse
- Cualquier cosa extra que usted se imagine, discutir con migo si desea saber exactametine el puntaje que se dara por su extra.

Proyectos Alternativos

Se aceptan propuestas para proyectos alternativos. Se requiere que los proyectos involucren los temas aprendidos en clase, por lo cual se debe utilizar un *lenguaje puramente funcional* como Elm, Haskell, OCaml, SML, Prolog o Agda.

Es posible negociar aun más puntos extras (adicionales al 15% extra) si su proyecto tiene como resultado un aporte a la comunidad de software libre (open-source). Si lo desea, puede buscar en Google Summer of Code `summerofcode.withgoogle.com` ideas de proyectos.