

Experiencia del proyecto

mruizm4

Junie 2020

Introducción

En nuestras clases de Programación estuvimos estudiando los fractales de Newton, Fractales de Julia y algunos sobre funciones iteradas, con algoritmos aleatorios y deterministas. Desarrollamos algoritmos para los dos primero y algunos para los dos últimos.

Como modo de proyecto final de la materia, se nos dio la tareas de crear una galería de fractales un repositorio en github y con su respectiva pagina tipo github.io.

En este documento se alojara mi experiencia de como desarrolle el repositorio y su pagina web.

Creación de la pagina

Para crear la pagina teníamos que crear un repositorio, y fue lo primera tarea que me puse hacer para empezar con el proyecto. Al crear repositorio le inserte las siguiente opciones licencia Apache 2.0, que se creara un archivo README, lo deje como publico además de ponerle nombre "Galeria-de-Fractales".

Con el repositorio hecho me puse a consultar como hacer la pagina web de tipo github.io. Para esto, dentro del repositorio me dirigí a configuraciones y estando hay en las opciones de mas abajo aparecía GitHub Pages desde hay elegí la fuente como "máster branch" y elegí un tema de colores de la pagina el cual puse minimalista. Al crear mi pagina web en la sección donde cree la pagina me apareció el link la pagina que acababa de crear.

Al crear la pagina en el archivo README me apareció el código de mi pagina en Markdown, que por defecto traía algunos consejos de como manejar el Markdown que utiliza GitHub Page. Y sobre ese archivo fue que trabaje el sitio web.

Creando los Fractales

Fractales de Newton

Para crear los fractales de newton utilice el código que vimos en clase el cual reflejaba la velocidad de convergencia de los puntos.

Para cambiar la función, colocaba la función que quería en sobre la función del código valga la redundancia en la parte de "return".

Para cambiar e color cambiaba los valores del numero correspondiente a las variables r,g,b. que sinceramente me toco usar el método de ensayo y error para que aparecieran los colores que se vieron finalmente en el sitio web

Fractales de Julia

Para los fractales de Julia también desarrollamos un código en clase, el cual para su implementación para crear los fractales era igual o muy similar al código de los de Newton. La diferencia es que Los Fractales de Julia recibe números complejos, entonces ponía la función en el mismo lugar que en el de Newton con el añadido de ponerlo numero complejos(para agregar números complejos utilizamos el comando "complex(a,b)" donde "a" es la parte real y "b" la imaginaria)

El cambio de colores también se hizo de manera idéntica a Newton es decir modificando los valores numéricos de las variables r,g,b

Fractales de funciones Iteradas

Algoritmos deterministas

De este tipo no me quedo muy claro el funcionamiento de los códigos así que honestamente opte por consultar en internet algunos códigos fractales que pudiesen funcionar con este tipo de algoritmos, en este caso los algoritmos que halle fueron los del triangulo de Sierpinski y el copo de nieve de Koch. Los links y articulo de donde saque estos códigos los deje en la pagina web y también aclare que no eran propios.

Curiosamente el del triangulo de Sierpinski funcionaba con el modulo "tk-inter" (los otros códigos los trabajamos con los módulos "matplotlib.pyplot" y "Pillow") que no lo había echado.Puse la imagen de 4 iteraciones del código, el cual en su función tenia una variable "level" que servía precisamente para cambiar el numero de iteraciones

El copo de Koch utilizaba otro modulo del cual tampoco habia trabajado, era el modulo "turtle" el cual también una variable "order" el la cual se le asignaba el numero de iteraciones para este caso se hicieron 5 iteraciones

Algoritmos Aleatorio

Para esta sección usamos un código visto en clase pero sacado de internet, este código consta del helecho de Barnsley, el cual con un algoritmo de que juega con

las probabilidades y la aleatoriedad de los puntos que se pintan, logra formar el fractal. Este algoritmo tiene un "rango de puntos" que se muestra en la salida el cual dependiendo del numero puesto se muestran mas o menos puntos, en la pagina se colocó 4 imágenes con este valor alterado. Como el nombre esta sección lo indica es un algoritmo aleatorio, el cual cada vez que ejecutas el código los puntos que se pintan son aleatorios lo que quiere decir que al menos la mayoría de los puntos aparecerán en una locación diferente si vuelves a ejecutar, lo que pasa es que el código esta hecho para aumentar las probabilidades de que los puntos que se pinten la figura de una manera bastante precisa.

Para el otro fractal si usamos un código desarrollado en clase el cual pinta un triangulo de Sierpinski, pero con un algoritmo aleatorio en vez de uno determinista. Lo implementamos para la pagina de manera idéntica al del párrafo interior con la diferencia que se tomaron 5 imágenes en vez de 4 con el valor de la densidad de puntos alterado, aunque su código es diferente.

Sección interactiva

Para la sección interactiva el profesor nos recomendó una pagina web ([Link de la pagina aquí](#)) para agregar un modulo interactivo a la pagina.

Para crear el modulo interactivo tome la función para crear los fractales de Newton y le asigne una función fija, pero la cual se puede cambiar el grado de la función puesta dentro del código, con ayuda del modulo "ipywidgets" cree una lista de valores que se le puede asignar a la función. También cree otro código con una función diferente para que haya mas opciones.

También a los valores de las variables r,g,b que son las que permite el cambio de color del fractal les di interactividad con 20 valores entre 0 y 100. Esto se hizo así ya que si se asignaba manualmente o con barra deslizante el programa calculaba cada valor asignado en el cuadro de texto, es decir el valor por defecto es 1 y si lo cambiabas a 20, calculaba el color con el valor de 2 y después de 20. lo cual resultaba algo incomodo. Si bien se pudo hacer un modulo interactivo mas completos, hacer esto resultaba mucho mas complejo y difícil de hacer y no logre desarrollar un modulo mas completo.

Después de hecho el código se abrió un "Anaconda prompt" (Anaconda es el navegador de herramientas de programación que utiliza) que es como una terminal de anaconda. hay instale el modulo "nbinteract" con los códigos de la pagina web que el profesor recomendó. Cuando instalo el modulo, dentro del Anaconda prompt puse la ruta de acceso del archivo donde tenia el código, para después insertar el una variante del código en la pagina para que me arrojase un archivo html del código de python, se utilizaron una variante y no el código directo debido a que la pagina esta hecha para los Gist no un repositorio de GitHub.

Ya con el archivo html del código interactivo lo cargue al repositorio en github. Después monte el código como un hipervínculo en la pagina. El único inconveniente es que el modulo tarda en inicializar o a veces no inicializa.