EIF206 Programación 3

Proyecto de programación

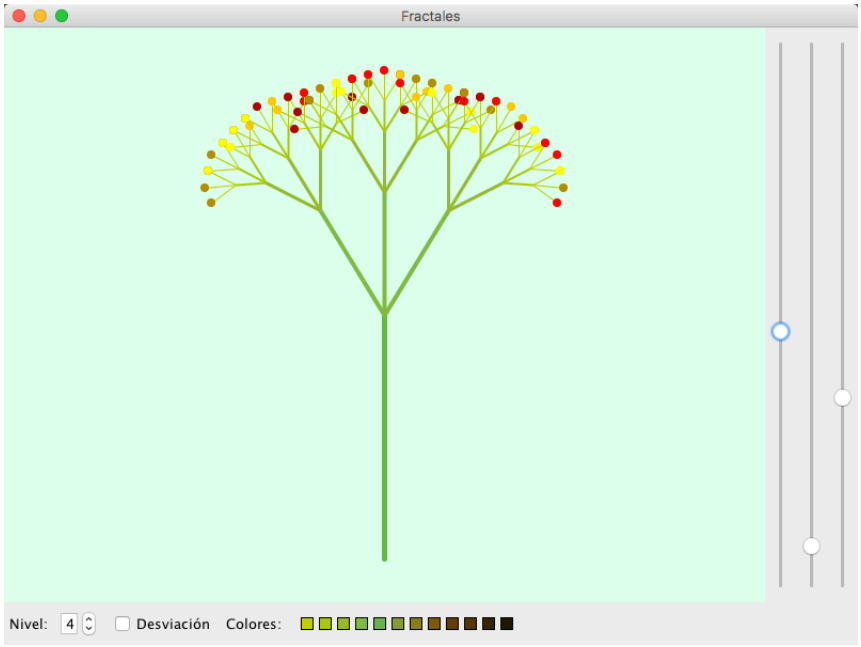
Profesor: Georges Alfaro Salazar

Descripción del proyecto

Un **fractal** es un objeto geométrico cuya estructura básica, fragmentada o aparentemente irregular, se repite a diferentes escalas. El término fue propuesto por el matemático Benoît Mandelbrot en 1975 y deriva del latín *fractus*, que significa quebrado o fracturado. Muchas estructuras naturales son de tipo fractal. La propiedad matemática clave de un objeto genuinamente fractal es que su dimensión métrica fractal es un número no entero.

Si bien el término "fractal" es reciente, los objetos hoy denominados fractales eran bien conocidos en matemáticas desde principios del siglo XX. Las maneras más comunes de determinar lo que hoy denominamos dimensión fractal fueron establecidas a principios del siglo XX en el seno de la teoría de la medida.[1]

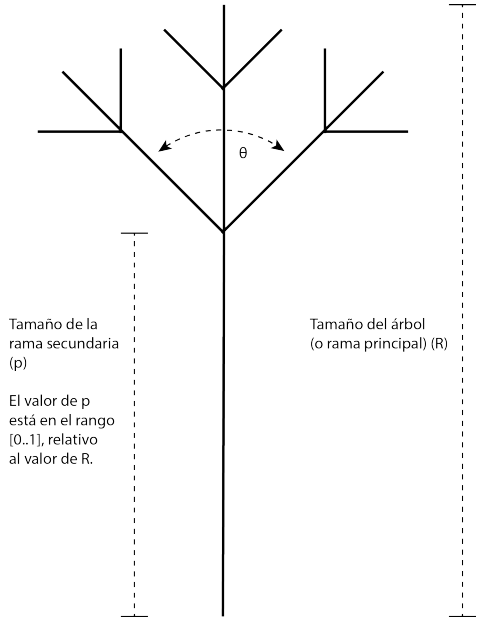
El proyecto a completar dibujará un árbol con estructura fractal. El árbol de la imagen que se muestra abajo tiene una sola raíz con tres ramas conectadas en su extremo. Cada una de esas ramas tiene otras tres ramas en su extremo. Esas ramas tienen tres ramas y así sucesivamente. Cada sección del árbol se parece al árbol mismo. Esto se conoce como *auto-similitud*. Mandelbrot se refería a esto diciendo que cada parte es una "copia de tamaño reducido del todo".[2]



[1] https://es.wikipedia.org/wiki/Fractal
[2] http://natureofcode.com/book/chapter-8-fractals/

El árbol del ejemplo es perfectamente simétrico y las partes son, de hecho, réplicas exactas del conjunto. Sin embargo, los fractales no tienen que ser perfectamente auto-similares.

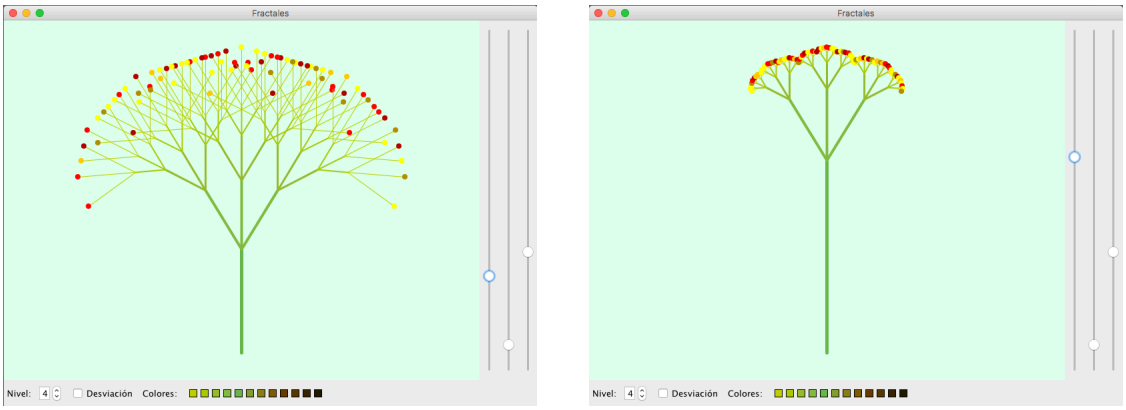
En el programa a desarrollar, cada árbol se dibujará utilizando cuatro parámetros, que pueden ser ajustados por medio de controles en la ventana, como se muestra. Además, se incluirá un control para poder cambiar el conjunto de colores utilizado para mostrar el árbol. Por último, se agregará un control de verificación (*checkbox*) para considerar un factor de aleatoriedad en el dibujo. Cuando se seleccione este control, el programa introducirá un factor aleatorio (que será determinado en el código) para modificar el dibujo del árbol. Para dibujar el árbol, se especifica un punto de inicio y otro final, de manera que el árbol ajuste su tamaño de acuerdo con la ventana.



Los controles a mostrar en la ventana serán entonces:

- Control de tamaño
- Control de factor de ramificación
- Control del ángulo de las ramas
- Control de color
- Control del nivel de recursión
- Control de desviación

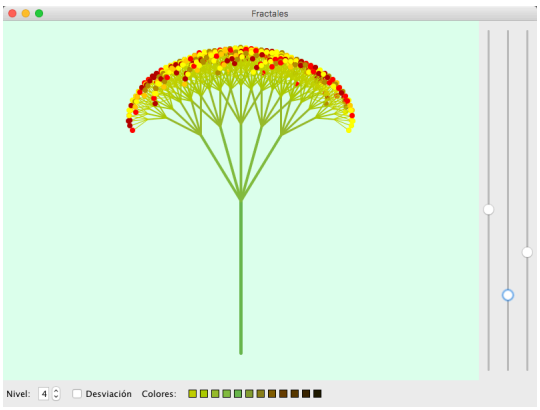
Control de tamaño.



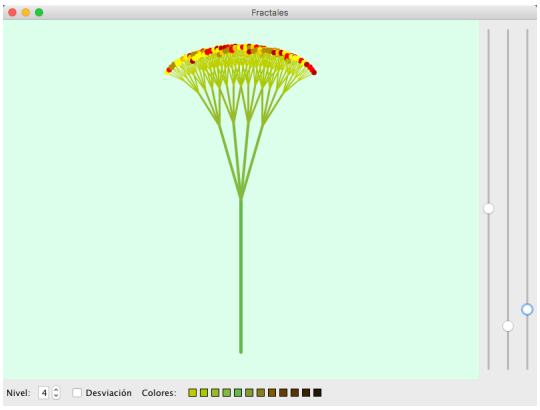
Se utilizará un control de tipo *JSlider*, con un valor porcentual (de 0 a 100) para controlar la posición, según el tamaño de cada rama, donde deben aparecer las ramas secundarias ($0 \leq p \leq 1$)

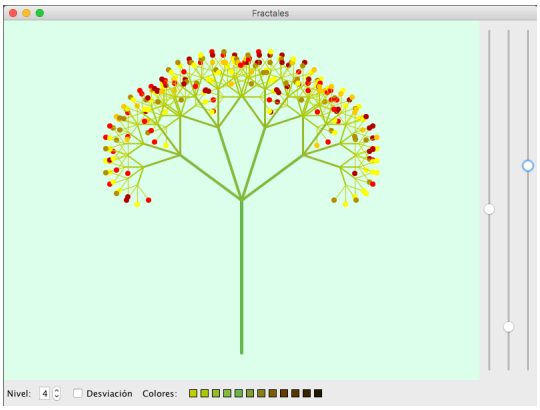
Control del factor de ramificación

Un control de tipo *JSlider* se empleará para indicar cuántas ramas deben colocarse **en cada nodo**. Habrá dos ramas como mínimo. El máximo de ramas a colocar será de 15.



Control del ángulo de las ramas





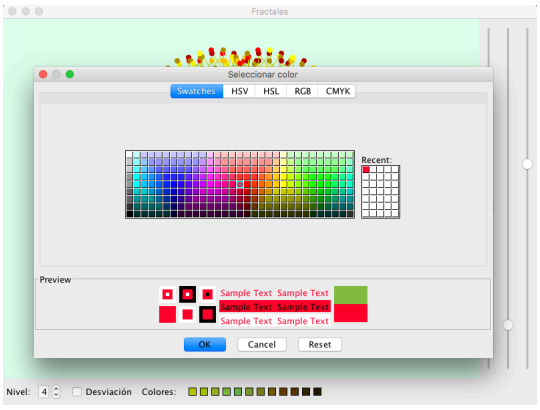
El ángulo de las ramas (θ) también se controlará por modelo de un control *JSlider*. El ángulo se calcula relativo a la línea de la rama que las contiene. Observe que el ángulo es la apertura total entre las ramas de cada extremo. La separación entre cada rama depende del total de ramas. Si el número de ramas aumenta, disminuye el espacio entre cada una, pero no cambia el ancho total del árbol.

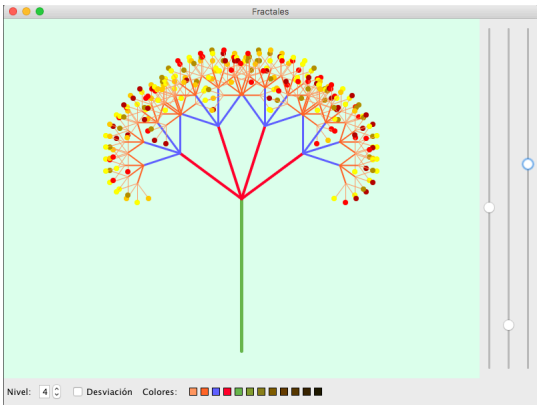
Para el ángulo de las ramas:

$$\theta \in [0 \cdots \pi]$$

Control de color

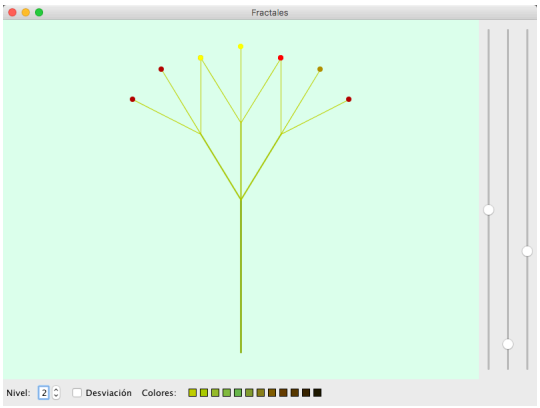
Para la selección de colores, se mostrará una paleta de 12 o 16 colores (de acuerdo con el nivel máximo de recursión permitido) que permita seleccionar el color a utilizar para las ramas de cada nivel. Al seleccionar un color de la paleta, se mostrará un control *JColorChooser* estándar.

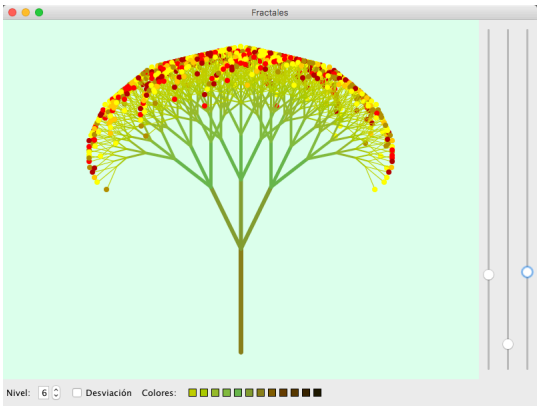




Nivel de recursión

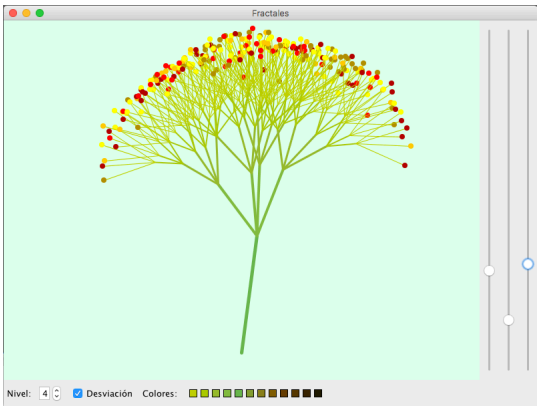
El nivel de recursión es un valor numérico que determina el nivel máximo de recursión a utilizar para dibujar el árbol. Puede usar un menú, campo de texto o un control de tipo *spinner*. Como el dibujo del árbol es más complejo a medida que aumenta el nivel de recursión, determine el máximo nivel práctico que pueda utilizar.





Control de desviación

Incluya un *checkbox* para considerar un factor de desviación al calcular el tamaño de las ramas y su ángulo. Determine el valor máximo de desviación a utilizar.



Objetivos

El objetivo principal del proyecto es adquirir habilidades en el manejo de interfaces de usuario, incluyendo el maneo de controles y el uso de primitivas gráficas. También se evalúa el uso apropiado de la recursividad.

Referencias

https://es.wikipedia.org/wiki/Fractal

http://natureofcode.com/book/chapter-8-fractals/

http://fractalfoundation.org/resources/fractivities/fractal-trees/

http://robertfathauer.com/FractalTreesArt.html

Consideraciones de implementación

El proyecto deberá completarse en el lenguaje de programación Java (versión 7 como mínimo). Se empleará *NetBeans* como plataforma de desarrollo (IDE). Es posible entregar las carpetas con el código fuente sin utilizar el IDE, pero en este caso, habrá que incluir los archivos de comandos necesarios para poder compilar y empaquetar la aplicación completa. Todas las pruebas de revisión se harán en el sistema operativo Linux o Windows indiferentemente.

Observaciones

Junto con los archivos de proyecto, habrá un documento que comprenda:

- Una descripción breve de la solución implementada
- Los diagramas de clase y cualquier otro diagrama UML que sea conveniente (diagramas de secuencia, colaboración, estado, etc.)
- Una explicación breve del uso del programa. Aquí no deben venir aclaraciones sobre el funcionamiento incorrecto del programa.
- En caso de que el programa no funcione adecuadamente, un análisis de los resultados obtenidos, posibles causas de los problemas presentados y sugerencias de solución

Se deberán incluir todos los diagramas de clase correspondientes, y deben coincidir con la implementación. Los diagramas deberán ajustarse a la sintaxis establecida en UML versión 2.

En la definición de cada clase, en el código fuente, se incluirán al inicio comentarios indicando:

- Descripción de la clase
- Fecha de creación e historial de modificaciones
- Nombre del autor o autores.
- Nombre del revisor o revisores.

En cada método se hará una descripción del contrato (objetivo o función implementada por el método) y los parámetros correspondientes.

Evaluación

El proyecto será evaluado de acuerdo con el detalle de la rúbrica indicada. Se puede completar el proyecto de manera individual o en pareja. Si se realiza en pareja, deberán enviar un mensaje de correo al profesor indicándolo y solamente uno de los participantes subirá los archivos del proyecto al aula virtual.

La entrega del proyecto será por medio del aula virtual, en las fechas provistas. El límite de entrega será el día **sábado 17 de junio**. Es importante recalcar que la documentación del proyecto no tiene puntaje asignado, pero se rebajarán puntos de calificación si esta no se presenta o se presenta de manera incorrecta o incompleta.

Criterio	Valor
Despliegue de la ventana y los controles de usuario	15%
Dibujo del árbol (ajuste de tamaño respecto a la ventana)	10%
Asociación de controles (MVC)	15%
Dibujo del árbol (recursividad)	25%
Dibujo del árbol (definición de parámetros)	20%
Control de aleatoriedad	15%
Puntaje total	**100%**