



Proyecto de CGI

fract'ol

Resumen:

Este proyecto busca crear fractales gráficamente bonitos.

Versión: 3

Índice general

| | | |
|-------------|----------------------------------|-----------|
| I. | Avance | 2 |
| II. | Introducción | 3 |
| III. | Objetivos | 4 |
| IV. | Instrucciones generales | 5 |
| V. | Parte obligatoria | 7 |
| V.1. | Renderizado | 8 |
| V.2. | Representación gráfica | 8 |
| VI. | Parte extra | 9 |
| VII. | Entrega y evaluación | 11 |

Capítulo I

Avance

Esto es lo que dice Wikipedia del aceite de oliva:

El aceite de oliva es un aceite vegetal de uso principalmente culinario. Se obtiene del fruto del olivo (*Olea europaea*), denominado oliva o aceituna. Casi la tercera parte de la pulpa de la aceituna es aceite. Por esta razón, desde la Antigüedad se ha extraído fácilmente con una simple presión ejercida por un molino. En España, las instalaciones donde se obtiene el aceite reciben el nombre de almazara. También se ha empleado para usos cosméticos, medicinales, religiosos y para las lámparas de aceite.

La oliva o aceituna no se suele comer cruda debido a la amargura de su sabor (principalmente debida a la presencia de compuestos fenólicos). Este sabor se reduce mucho aplicando diversos procesos de macerado. No obstante el 90 % de la producción mundial de olivas se emplea en producir aceite.

El aceite se extrae de aceitunas maduras de entre seis y ocho meses, cuando contienen su máxima cantidad de aceite, lo que suele ocurrir a finales de otoño. Las aceitunas se someten a una primera presión con el objeto de extraer su zumo. La calidad del aceite depende en gran medida del procesado posterior. Por esta razón, los productores vigilan estos pasos con sumo cuidado. La calidad del aceite de oliva se juzga por sus propiedades organolépticas y por su contenido de ácidos grasos libres. Existen regulaciones en la Unión Europea sobre las clasificaciones del aceite en seis categorías, en función de la concentración de ácidos grasos.

Este proyecto no se llama *fract'oliva*, por lo que no tiene relación con la producción de aceite.

Capítulo II

Introducción

El término fractal fue usado por primera vez por el matemático Benoit Mandelbrot en 1974, se basó en la palabra latina *fractus*, cuyo significado es “roto” o “fracturado”. Un fractal es un objeto de abstracción matemática, como una curva o superficie, que conserva un patrón similar independientemente de la escala.

Varios fenómenos naturales — como el romanesco — tienen algunas de las características de los fractales.



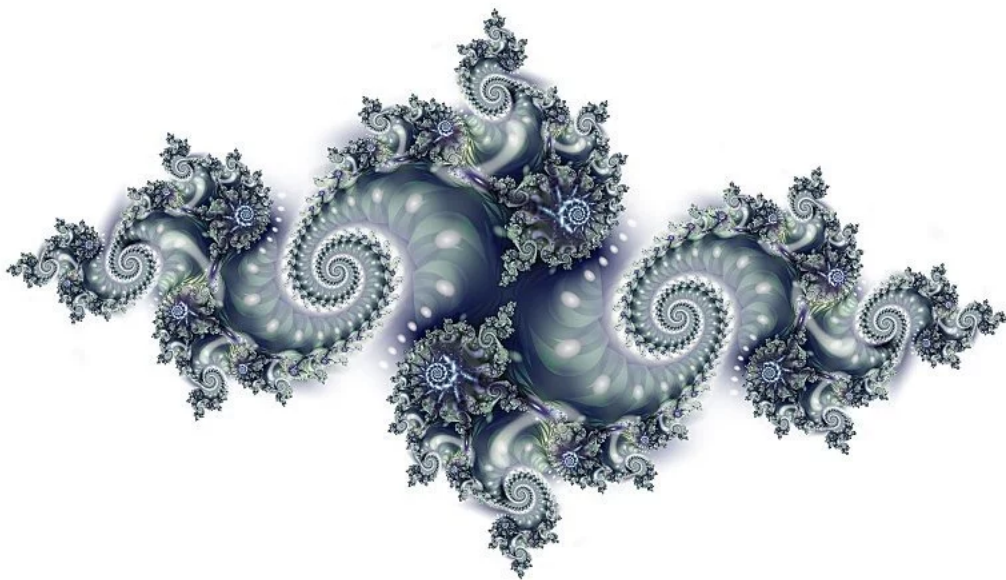
Ahora, te toca a ti generar algunos fractales majestuosos.

Capítulo III

Objetivos

Es el momento de que crees un proyecto básico de gráficos por ordenador. Vas a utilizar la librería gráfica del campus: la `miniLibX`. Esta librería ha sido creada de manera interna e incluye herramientas básicas necesarias para abrir una ventana, crear imágenes y hacer frente a los eventos del teclado y del ratón.

Este nuevo proyecto será tu oportunidad para familiarizarte con la `miniLibX`, para que descubras o utilices la noción matemática de **números complejos** y eches un vistazo al concepto de **optimización** en CGI y practiques el manejo de acontecimientos.



No olvides ver vídeos de fractales.

Capítulo IV

Instrucciones generales

- Tu proyecto deberá estar escrito en C.
- Tu proyecto debe estar escrito siguiendo la Norma. Si tienes archivos o funciones adicionales, estas están incluidas en la verificación de la Norma y tendrás un 0 si hay algún error de norma en cualquiera de ellos.
- Tus funciones no deben terminar de forma inesperada (segfault, bus error, double free, etc) ni tener comportamientos indefinidos. Si esto pasa tu proyecto será considerado no funcional y recibirás un 0 durante la evaluación.
- Toda la memoria asignada en el heap deberá liberarse adecuadamente cuando sea necesario. No se permitirán leaks de memoria.
- Si el enunciado lo requiere, deberás entregar un **Makefile** que compilará tus archivos fuente al output requerido con las flags `-Wall`, `-Werror` y `-Wextra` y por supuesto tu **Makefile** no debe hacer relink.
- Tu **Makefile** debe contener al menos las normas `$(NAME)`, `all`, `clean`, `fclean` y `re`.
- Para entregar los bonus de tu proyecto deberás incluir una regla `bonus` en tu **Makefile**, en la que añadirás todos los headers, librerías o funciones que estén prohibidas en la parte principal del proyecto. Los bonus deben estar en archivos distintos `_bonus.{c/h}`. La parte obligatoria y los bonus se evalúan por separado.
- Si tu proyecto permite el uso de la `libft`, deberás copiar su fuente y sus **Makefile** asociados en un directorio `libft` con su correspondiente **Makefile**. El **Makefile** de tu proyecto debe compilar primero la librería utilizando su **Makefile**, y después compilar el proyecto.
- Te recomendamos crear programas de prueba para tu proyecto, aunque este trabajo **no será entregado ni evaluado**. Te dará la oportunidad de verificar que tu programa funciona correctamente durante tu evaluación y la de otros compañeros. Y sí, tienes permitido utilizar estas pruebas durante tu evaluación o la de otros compañeros.
- Entrega tu trabajo en tu repositorio `Git` asignado. Solo el trabajo de tu repositorio `Git` será evaluado. Si Deepthought evalúa tu trabajo, lo hará después de tus com-

pañeros. Si se encuentra un error durante la evaluación de Deepthought, esta habrá terminado.

Capítulo V

Parte obligatoria

| | |
|-----------------------|--|
| Nombre de programa | fractol |
| Archivos a entregar | Makefile, *.h, *.c |
| Makefile | all, clean, fclean, re, bonus |
| Argumentos | El tipo de fractal a mostrar y cualquier otra opción disponible |
| Funciones autorizadas | <ul style="list-style-type: none">• open, close, read, write, malloc, free, perror, strerror, exit• Todas las funciones de la librería de matemáticas (-lm man man 3 math)• Todas las funciones de la miniLibX• ft_printf y cualquier función equivalente que hayas escrito |
| Se permite usar libft | Sí |
| Descripción | El objetivo de este proyecto es crear un pequeño programa de exploración de fractales. |

Tu proyecto tendrá que seguir las siguientes reglas:

- Tienes que utilizar la miniLibX, ya sea la versión disponible en los ordenadores del campus o la que instales usando sus fuentes originales.
- Tendrás que entregar un Makefile que compilará tus archivos fuente. No debe hacer relink.
- Las variables globales están prohibidas.

V.1. Renderizado

- Tu programa debe ofrecer los conjuntos de **Julia** y **Mandelbrot**.
- La rueda del ratón hace zoom in y zoom out y lo hace casi infinitamente (dentro de los límites del ordenador). Ese es el concepto de un fractal.
- Debes poder crear un conjunto de Julia diferente mediante los parámetros del programa.
- Un parámetro se pasará a través de la CLI para definir el tipo de fractal visualizado.
 - Puedes usar más parámetros como opciones de renderizado.
 - Si no se introduce un parámetro, o si el parámetro es inválido, el programa mostrará una lista de parámetros disponibles y terminará correctamente.
- Debes usar al menos unos pocos **colores** para mostrar la profundidad de cada fractal. Es incluso mejor si te adentras en los efectos psicodélicos.

V.2. Representación gráfica

- Tu programa tiene que mostrar la imagen en una ventana.
- El manejo de tu ventana debe ser fluido (cambio a otra ventana, minimización y demás)
- ESC debe cerrar la ventana y salir del programa de manera limpia.
- Hacer click en la cruz del marco de la ventana debe cerrarla y cerrar el programa de manera limpia.
- El uso de **images** de la MiniLibX es obligatorio.

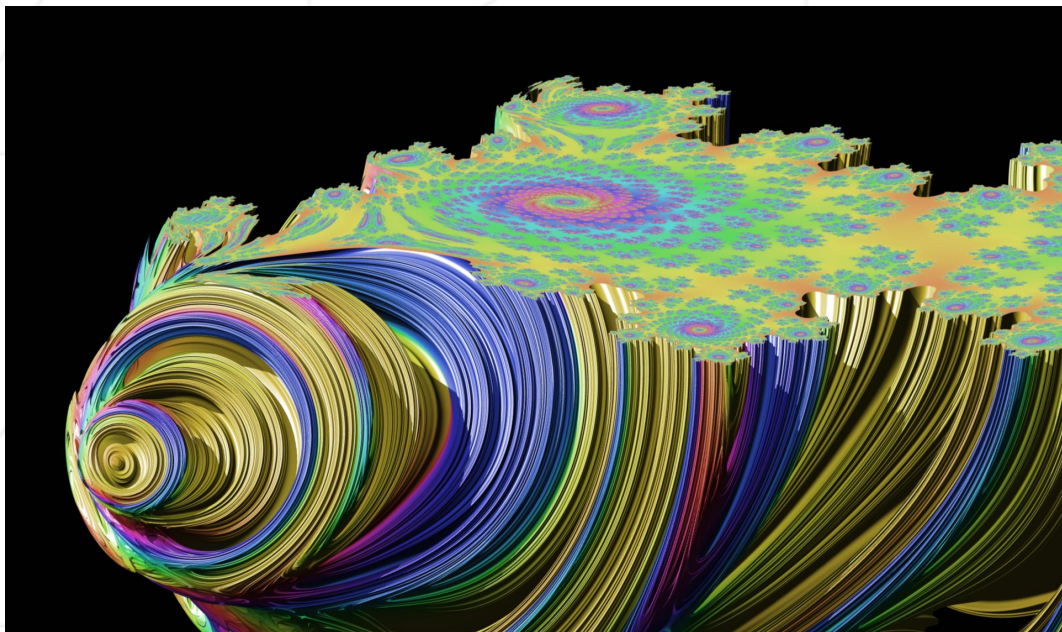
Capítulo VI

Parte extra

Habitualmente, te animaríamos a que desarrollaras nuevas características extra. Sin embargo, habrá proyectos gráficos mucho más interesantes más adelante. ¡Te están esperando! ¡No pierdas demasiado tiempo en este ejercicio!

Conseguirás algunos puntos extra con lo siguiente:

- Un fractal diferente más (hay más de cien referencias a distintos tipos de fractales online).
- El zoom sigue la posición actual del ratón.
- Adicionalmente al zoom: moverse con flechas.
- Haz que el rango de color cambie.





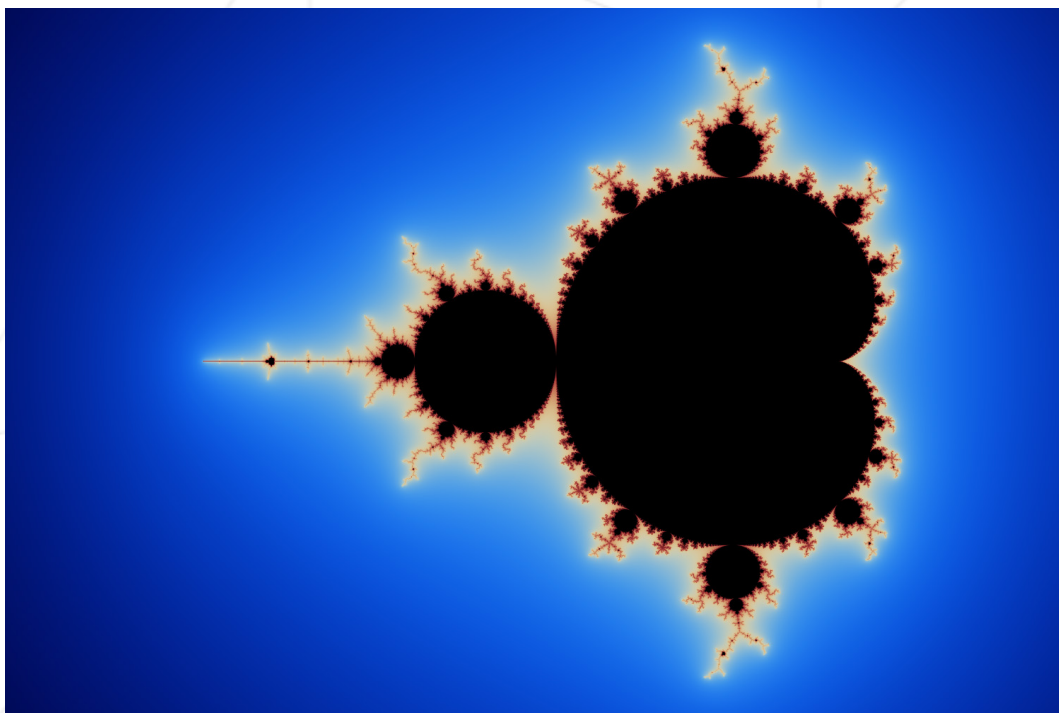
La parte bonus será exclusivamente evaluada si la parte obligatoria está PERFECTA. Perfecta significa que todos los requisitos de la parte obligatoria deben estar correctamente completados y funcionan perfectamente. De otro modo, tus bonus no serán evaluados de ninguna manera.

Capítulo VII

Entrega y evaluación

Entrega tu trabajo en tu repositorio `Git` como de costumbre. Solo el trabajo de tu repositorio será evaluado. Comprueba dos veces si es necesario que los nombres de tus archivos sean los correctos.

Como estos ejercicios no se verifican con un programa, puedes organizar tus archivos como quieras, siempre que entregues los archivos obligatorios y cumplas los requisitos.



```
file.bfe:VAD2s02qgbqPEXR0eASmsgnY0o0sDMJev7zFHhw  
QS8mvM8V5xQQpLc6cDCFXDWTiFzZ2H9skYkiJ/DpQtnM/uZ0
```