The sketch extra homework

Kristof Cseh

University of Bristol
Department of Computer Science,

January 15, 2018

Introduction

In this program we can introduce ourselves into the world of fractals in an interactive way. Four different fractals (or sets of fractals) are implemented. We can plot and zoom the Mandelbrot set with three different colouring methods, the Julia set, the Buddhabrot fractal and the basic Lindenmayer system. We can use the software by clicking the binary icon in the file browser and without using command line/terminal window. When we zoom in the Mandelbrot set or Julia set it is useful to see the metadata printed in the terminal window but otherwise the detailed graphical interface helps to navigate. A guide option is also added but the documentation is put here instead of there because the logic part is done there in the code and I find it is more the task of graphics designers (or I need more time to finish).
My aim in this homework was to practise graphics, memory management (avoid memory leaks) and practise using modules.

Libraries

Some third party libraries are used. I tried to use cross-platform libraries. The libraries I used are enumerated below:

- SDL2 (Simple DirectMedia Layer) [1]
- SDL2 ttf (SDL2 extension which allows to render true type fonts) [2]
- GNU MPFR (multi-precision floating point library) [3]

We can install them with the following commands (Linux):

```
sudo apt-get install libmpfr-dev
```

```
sudo apt-get install libsdl2-dev
```

```
sudo apt-get install libsdl2-ttf-dev
```

In case MPFR depends on this we may also need

```
sudo apt-get install libgmp3-dev
```

Modules

A lot of modules are used.
complex.c
This module is a basic complex number library which uses MPFR. We do do math part of the Mandelbrot and Julia sets with this module.

l-system.c
This module is basically picked from a former imperative programming assignment (string assignment). There are some changes to improve memory management and there is a serious bug fix. This module is responsible for the string handling part of the Lindenmayer system part of the program.
font.c
With this we can render text in SDL in a brief comfortable way. We have 3 different functions to render text. One to render one line text in white. One to render wrapped text (multi-line text) and the last one is to render one line coloured text. The font we use is Courier and it is downloaded from the internet (`COURIER.TTF`).
buddha.c
We can draw the Buddhabrot [6] with this module. Some random number routines are implemented as well.
guide.c
In this unit we made a scrollable text.
triangle_further.c
This unit is from my triangle assignment. It is used to detect mouse over the text.
menu.c
It is responsible for the user interface in the menu.
stack.c
This is the exact copy of the singly linked list module from the lecture notes. The only change is the some new structures are added in it and the `int` is rewritten to `state`.
sdl_lsystem.c
This unit converts the l-system string into drawing. User interface is added.

The logic of the program

The heart of the program is the `main.c`. Every module is included here. To compile the program type `make main` in the terminal. The program is a huge while loop which ends if the boolean `running` is set to be false.

Initially the `currentMenuItem` is set to be zero, which means we selected the Mandelbrot menu point (the chosen one is denoted by '>' before it). If we select another one then eg. Julia then the value is 1 and so on. Once we select something then the `currentMenuItem` is set to be `currentMenuItem = -currentMenuItem - 1`, with this value we exit the menu and draw the corresponding task. After finishing the task we can go back to the menu.

We have to options to navigate in the menu: first we can simply move the mouse over the text or use arrow keys (left, down, right, up). If the chosen menu point is highlighted then we can run it by pressing <ENTER> or clicking if the mouse is over the text.

We detect the mouse over the text by cutting the text box into two triangles and use the triangle function from `triangle_further.c`.

In Mandelbrot set we can see a menu with some sort of user interface. It contains a radio button and a checkbox. With the radio button we can select the colouring method. With the checkbox we can decide if we want to plot the bifurcation diagram [4] corresponds to the Mandelbrot set in the first frame. Once we selected our options by pressing <ENTER> we can draw the Mandelbrot set.
In the Mandelbrot set if we click with the left button then the program redraws the Mandelbrot fractal with the centre where we clicked and zoomed in (right button zooms out). If we opened the program in the terminal then we can see the current zoom level, the zooming factor and the number of the maximum iteration. If we press the left and right arrows then we can change the zoom factor. Initially it double the image every time, but with this we can achieve deeper zooms without waiting so long. With the up-down or '+'-'-' keys we can change the number of the maximum iteration. If we press the key 'j' we can see the corresponding Julia set [5]. The colouring type of it is the same as the chosen colouring for Mandelbrot set. To exit from this Julia set simply click exit window button. If we want to exit from the Mandelbrot zoom click exit button.
The Julia set option in the menu is very similar to the Mandelbrot set without user interface.
The Buddhabrot fractal [6] option draws The Buddhabrot fractal, which is a different rendering type of the Mandelbrot set. The disadvantage of this is that it is not zoomable. If we very want to zoom then we face with extremely long rendering time. In the rendering method we need some kind of random numbers. I made an experiment with the random distributions and found that the most efficient is the uniform(-1,1). I tried normal distribution as well, using the central limit theorem, but it is slow.
When the drawing is finished we can see a white line appearing in the top left corner. Exit with the exit button.
The L-system needed the most advanced user interface. It contains several input boxes. If the input box is selected then the cursor starts to flashing in that box. We can write and delete with <BACKSPACE>. Once we typed all the rules we need by clicking the draw box or by pressing <ENTER> we can start the rendering of the L-system. The output is centred and resized to fit the box. Initial values are set in the boxes to test the program quicker. We can stop drawing by click exit and can exit L-system menu by clicking exit button. There are many types of error handling in the program.
The rules:

- A, B, C, D, E, F: draw
- G, H, I, J, K, L: move
- +: turn left
- -: turn right
- |: turn 180 degrees
- [: save state

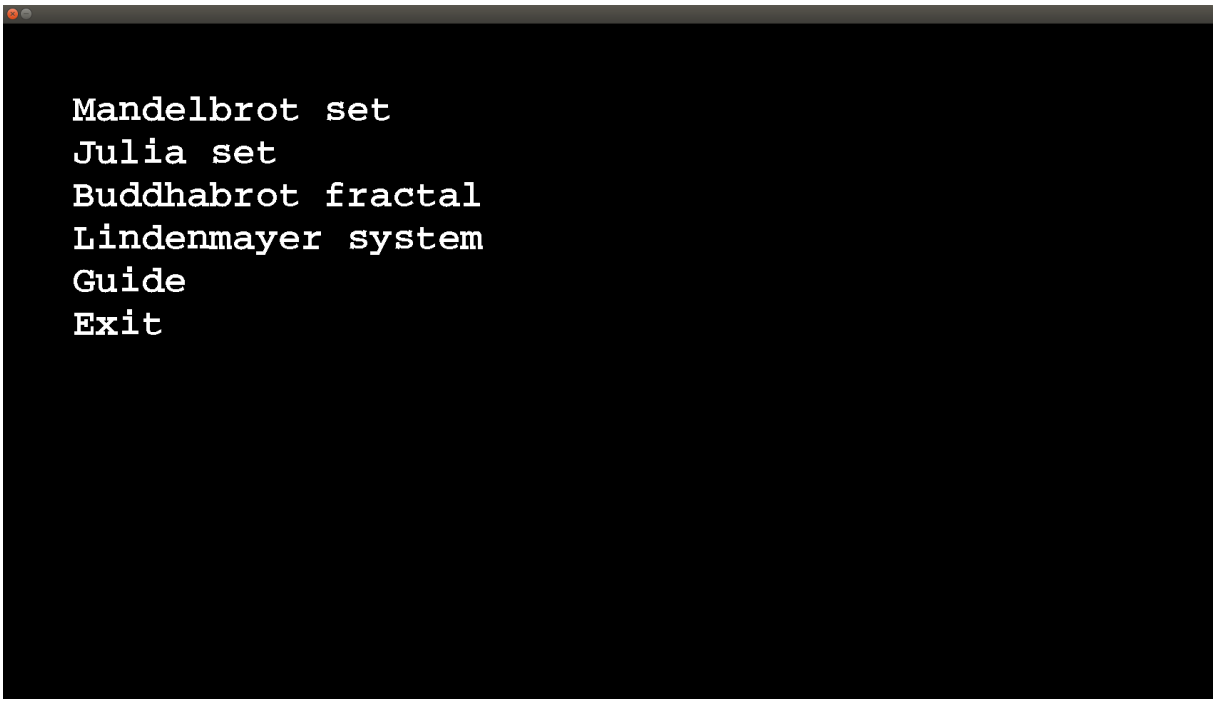
-]: load the previously saved state

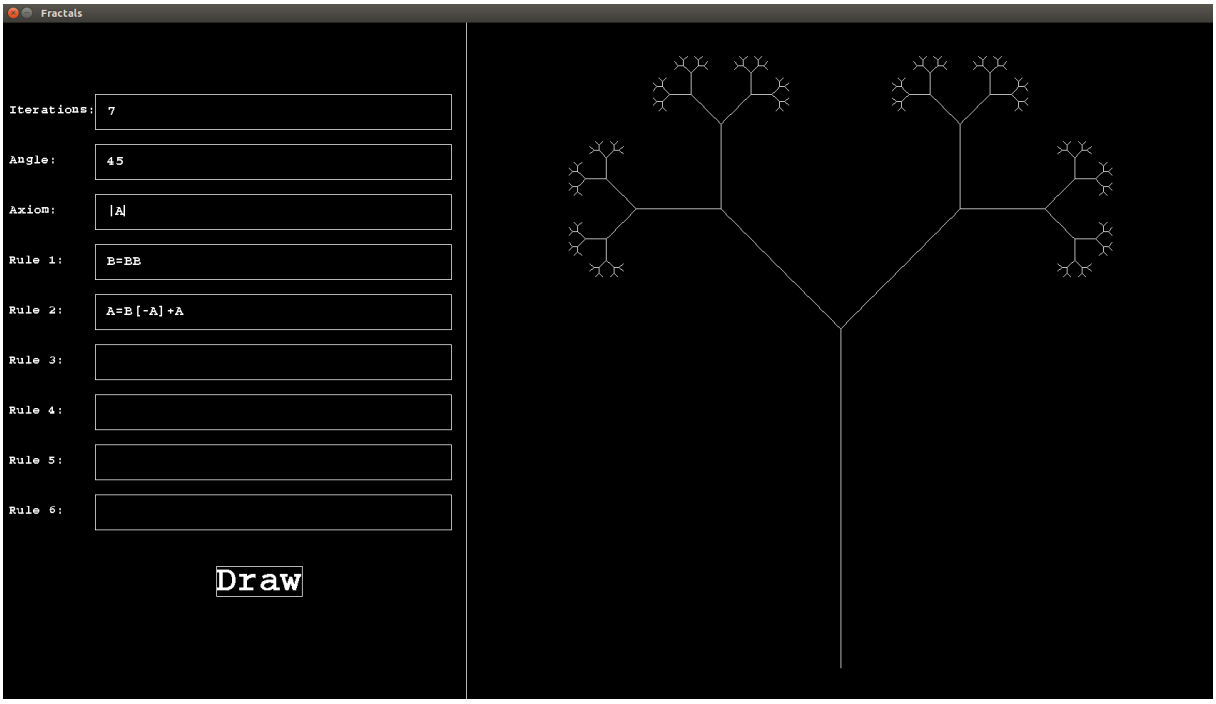
In the guide we can put some text about the program, development team, other information, and the page is scrollable by mouse drag and move, arrow keys or mouse wheel. Exit with <BACKSPACE> or <ENTER>

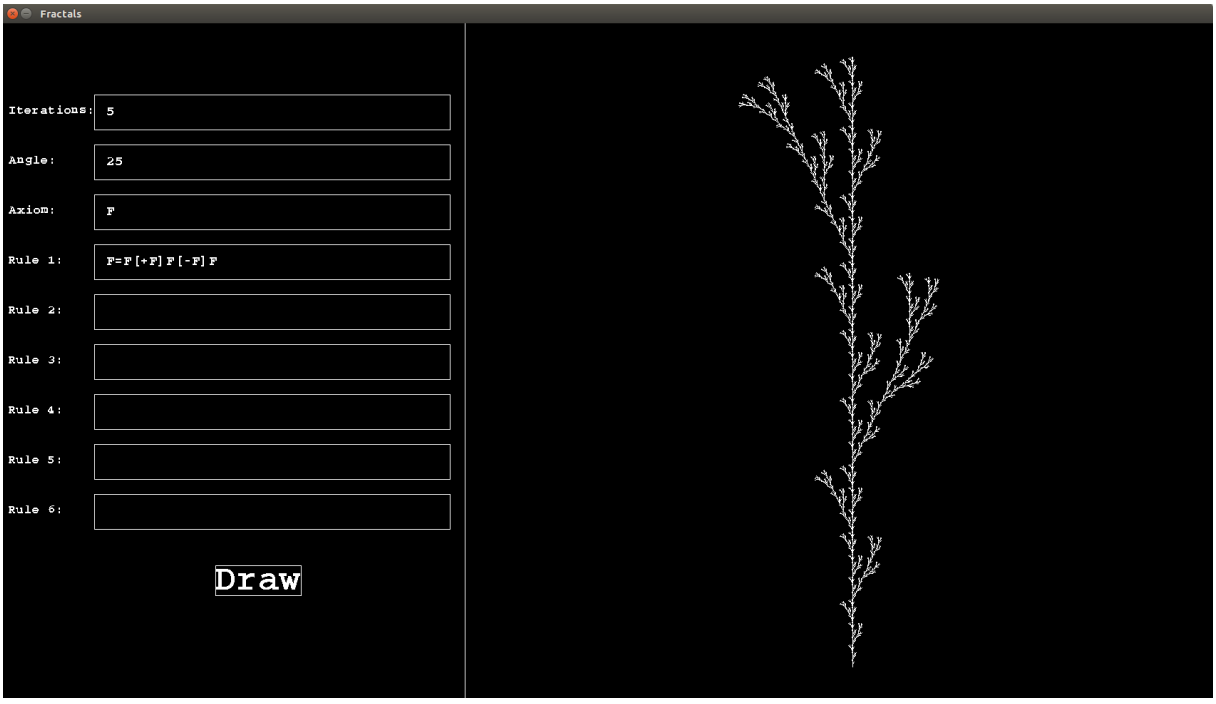
Version 2

1. bug-fixes in `l-system.c`
2. interface added to Mandelbrot
3. colouring method added to Mandelbrot
4. bifurcation diagram added to Mandelbrot
5. corresponding Julia set added to Mandelbrot
6. because of the previous the dependency between `mandel.c` and `julia.c` is reversed.
7. zoom factor and maximum iteration are changeable
8. error handling added in L-system

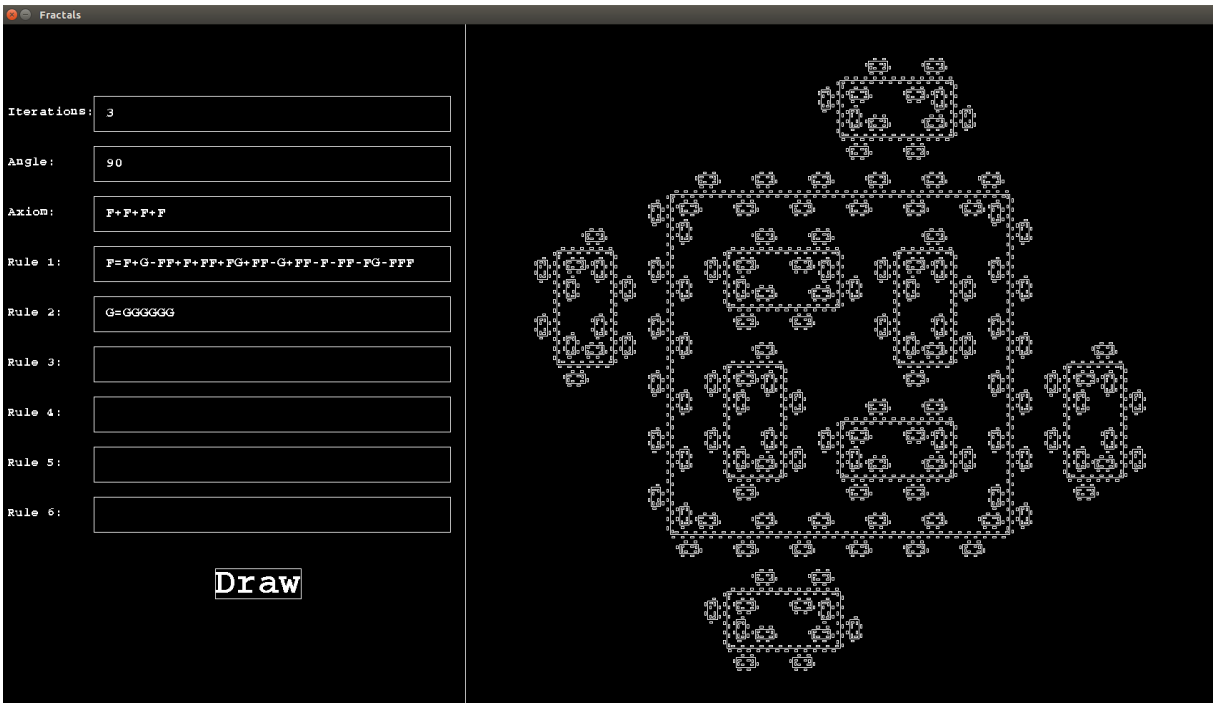
Some screenshots from the program:

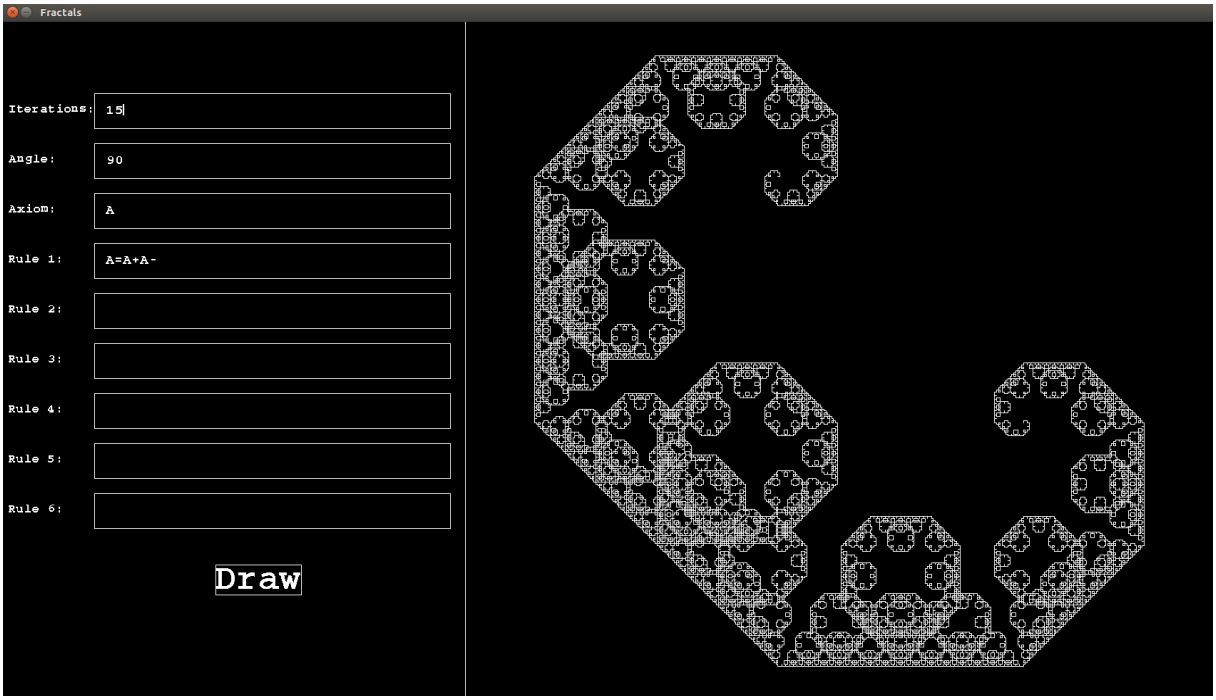


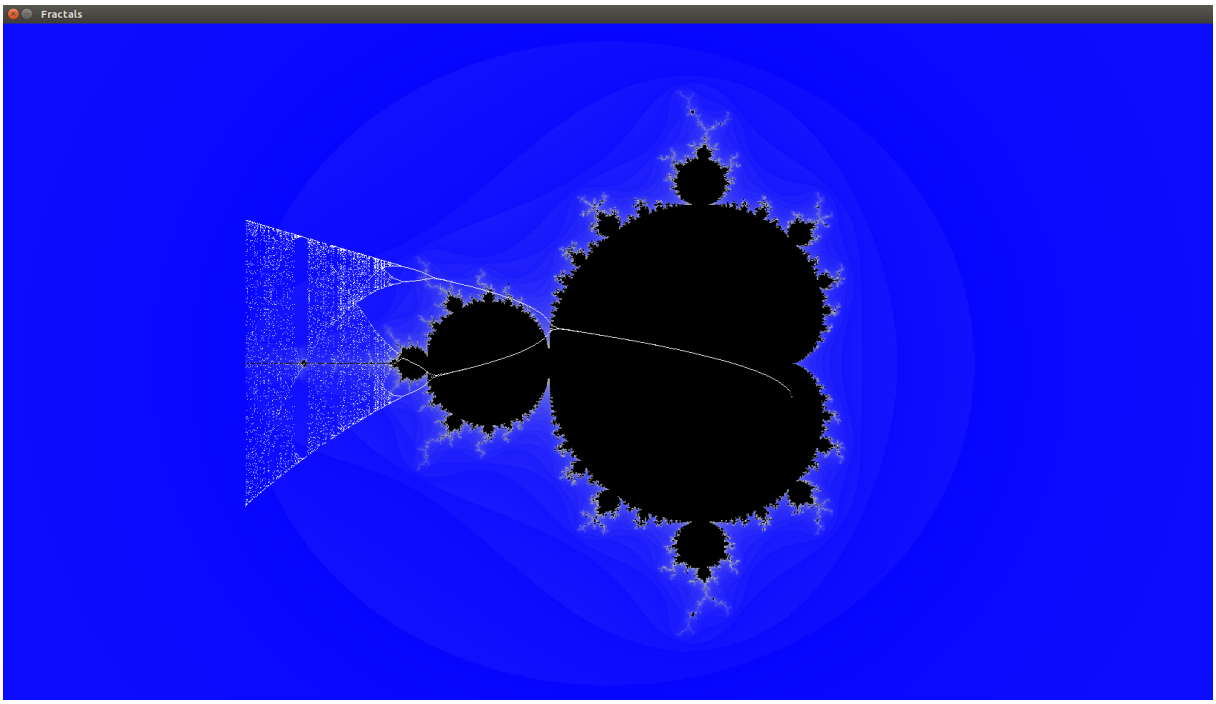

Fractals
Iterations: 7
Angle: 45
Axiom: |A
Rule 1: B=BB
Rule 2: A=B[-A]+A
Rule 3:
Rule 4:
Rule 5:
Rule 6:
Draw

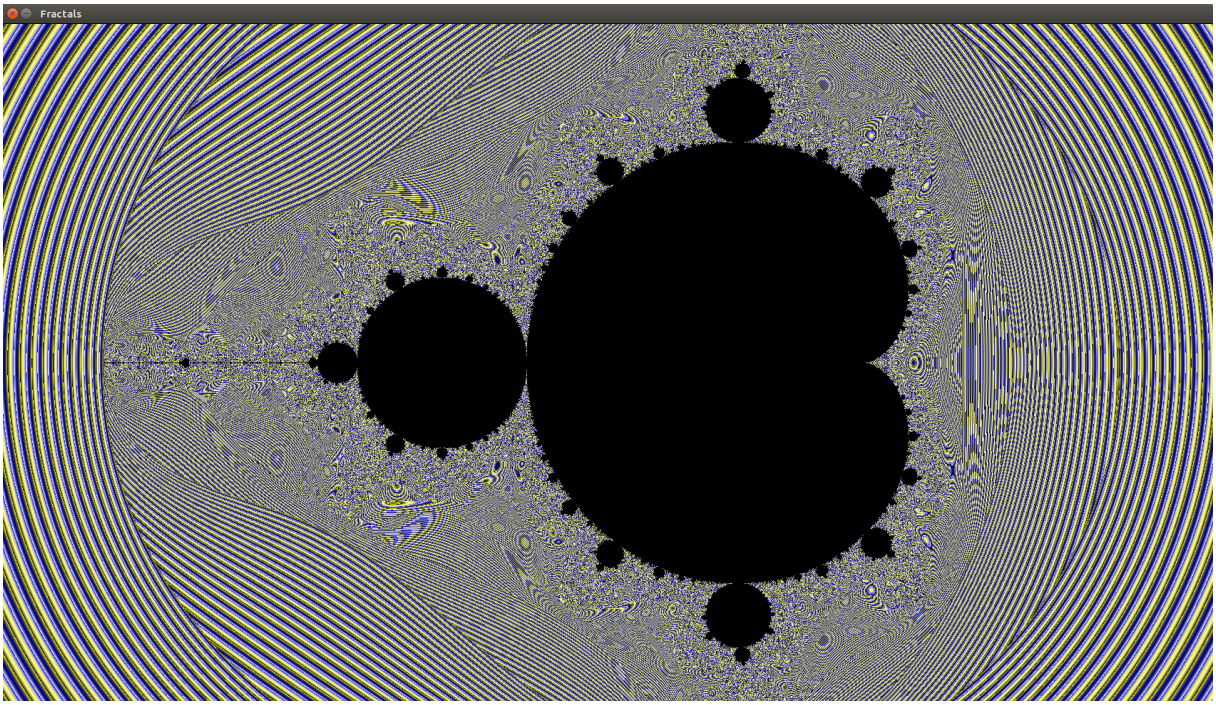

Fractals
Iterations: 5
Angle: 25
Axiom: F
Rule 1: F=F[+F]F[-F]F
Rule 2:
Rule 3:
Rule 4:
Rule 5:
Rule 6:
Draw

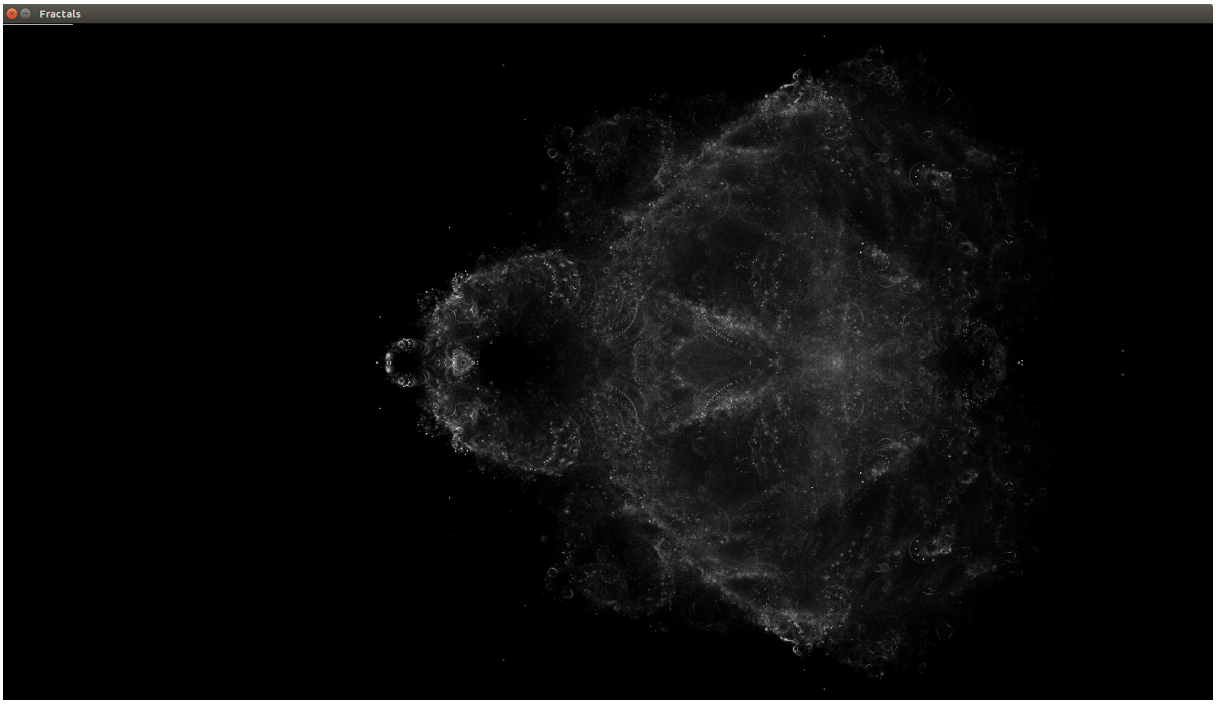

Fractals
Iterations:
5
Angle:
60
Axiom:
A
Rule 1:
A=ALA
Rule 2:
L=LLL
Rule 3:
Rule 4:
Rule 5:
Rule 6:
Draw


Fractals
Iterations:
3
Angle:
90
Axiom:
F+F+F+F
Rule 1:
F=F+G-FF+F+FF+FG+FF-G+FF-F-FF-FG-FFF
Rule 2:
G=GGGGGG
Rule 3:
Rule 4:
Rule 5:
Rule 6:
Draw


Fractals
Iterations: 15
Angle: 90
Axiom: A
Rule 1: A=A+A-
Rule 2:
Rule 3:
Rule 4:
Rule 5:
Rule 6:
Draw


Fractals


Fractals


Fractals

References

[1] Documentation of GNU MPFR:
http://www.mpfr.org/mpfr-current/mpfr.html

[2] SDL2 wiki:
https://wiki.libsdl.org/

[3] SDL2-ttf
https://www.libsdl.org/projects/SDL_ttf/docs/SDL_ttf.html

[4] Bifurcation diagram:
https://en.wikipedia.org/wiki/Bifurcation_diagram
https://en.wikipedia.org/wiki/Mandelbrot_set#/media/File:
Verhulst-Mandelbrot-Bifurcation.jpg

[5] Julia set:
https://en.wikipedia.org/wiki/Julia_set
http://mathworld.wolfram.com/JuliaSet.html
https://www.youtube.com/watch?v=mg4bp7G0D3s

[6] Buddhabrot fractal:
https://en.wikipedia.org/wiki/Buddhabrot#Rendering_method
http://superliminal.com/fractals/bbrot/bbrot.htm

[7] Smooth colouring:
https://stackoverflow.com/questions/369438/smooth-spectrum-for-mandelbrot-set-rende