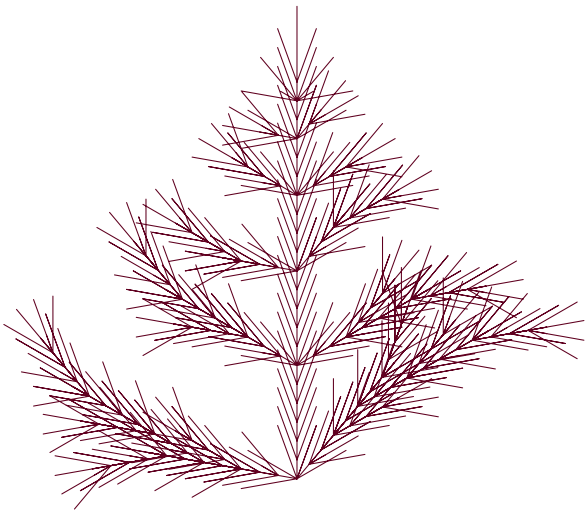
Lindenmayer's Garden

Scuola d'Arti e Mestieri di Trevano (SAMT)
Diaries

Paolo Bettelini



Contents

1 Working Sessions

1.1 2023-05-02

Work hours:
08:30 - 10:30: Initial analysis
10:30 - 11:25: Repository setup

Today I read time the requirements for the project for the first. I thoroughly analyzed each requirements with my advisor, with whom, after a long discussion, we decided to drastically change.
The initial premise and functionality of the project does remain the same, but it will be executed with a wider approach, which both augments its functionality and simplifies its workings.

In the second half of the working session I setup the repository with the initial files, folder and documents.

The plan for the next working session is to create the Gantt chart.

1.2 2023-05-03

Work hours:
09:05 - 10:00: Requirements
10:00 - 10:15: Documentation
10:15 - 10:35: Initial Gantt Chart
10:50 - 12:10: Use Cases
13:20 - 13:50: UI Design
13:50 - 14:10: Uses Cases
14:10 - 14:45: `lsystems-engine`
15:05 - 15:40: `lsystems-renderer`
15:40 - 16:20: `lsystems-gui`

Today I started adding content to the documentation. I wrote the requirements of the project and some other minor section.

I created the following sections:

- `Introduction.Information`
- `Analysis.Requirements`
- `Analysis.Use Cases`
- `Analysis.GUI Design`

The `Technologies.Rust` section was recycled from a previous project.

In the second half of the working session I drew the UI design sketch and started coding.

I plan to structure the project with the following crates:

- `lsystems-engine`: Basic L-System string expansion
- `lsystems-renderer`: Rendering logic with abstract operations
- `lsystems-parser`: Config parser for `lsystems-renderer`
- `lsystems-cairo`: Rendering implementation for `cairo`
- `lsystems-gui`: Graphical User Interface

I created the `lsystems-engine` lib and implemented the basic L-systems string expansion. I also added a unit test to make sure the string is expanded correctly.

I created the `lsystems-renderer` lib and started sketching the library structure and

I created `lsystems-gui` and setup a simple GTK4 application to make sure everything is ready.

I am ahead of the initial planning.

Git status:

1. Created `gui` branch
2. Created `renderer` branch

1.3 2023-05-04

Work hours:
08:20 - 08:40: Renderer abstraction
08:40 - 09:50: Renderer implementation
10:05 - 11:25: Fractal drawing

Today I created the rendering abstraction and wrote a basic implementation of the fractal drawing.

To achieve this I needed to implement the drawing abstraction for `cairo-rs`.

I hardcoded a fractal renderer into the code and tried to draw it. Something resembling the fractal is renderer onto the canvas, but it is still broken.

The plan for the next working session is to render the fractal correctly and continue the documentation.

I am still ahead planning.

Git status:

1. Merged `gui` into `master`
2. Merged `renderer` into `master`
3. Created `parser` branch
4. Created `fractal-drawing` branch

1.4 2023-05-05

Work hours:
08:20 - 08:50: Fixed fractal drawing
08:50 - 10:40: Parser implementation
10:40 - 11:20: Fractal from file
13:35 - 14:35: Documentation

Today I implemented the parser for the fractal configuration. The parser is not ready but it is stable enough and supports the most part of the basic grammar.

The program is now able to read a file containing the fractal specifications and then render it.

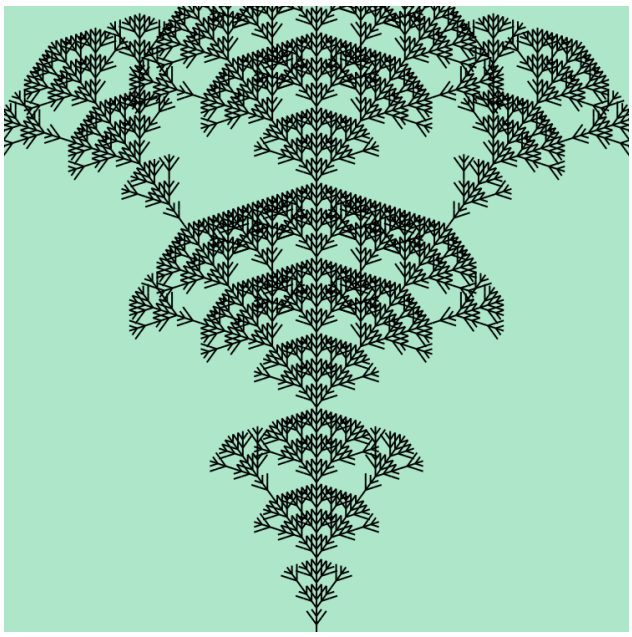
The file looks like the following:

```
LINE = 10
ANGLE = 0.62832

F -> F[+FF][-FF]F[-F][+F]F

axiom F
iter 5
initial_pos 375,750

F: forward LINE
+: rotate ANGLE
-: rotate -ANGLE
[: push
]: pop
```



Git status:

1. Merged `fractal-drawing` into `master`
2. Merged `parser` into `master`

In the second half of the working session I continued the documentation.

I modified or created the following sections:

- `Technologies.GTK4`
- `L-Systems.Definition`
- `L-Systems.Examples`

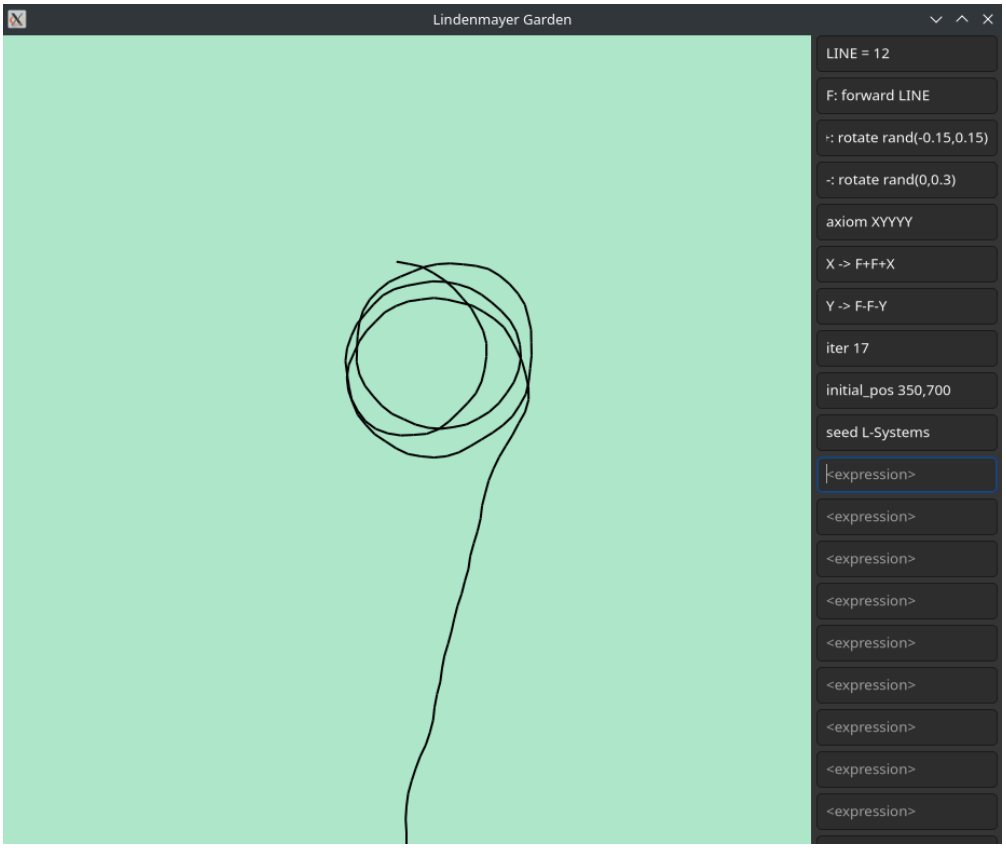
1.5 2023-05-08

Work hours:
09:05 - 12:30: GUI editor
13:15 - 14:45: Stochastic behavior
15:00 - 16:20: Generic improvements

Today I implemented the first version of the GUI editor. When the application starts, some textboxs are loaded with the default lines of a fractal configuration. By changing the content of the textboxs the fractal is updated in real-time.

In the second half of the working session I implemented the stochastic behavior, namely the `rand(lower, upper)` function and the `seed` command.

The following image illustrates such behavior.



Git status:

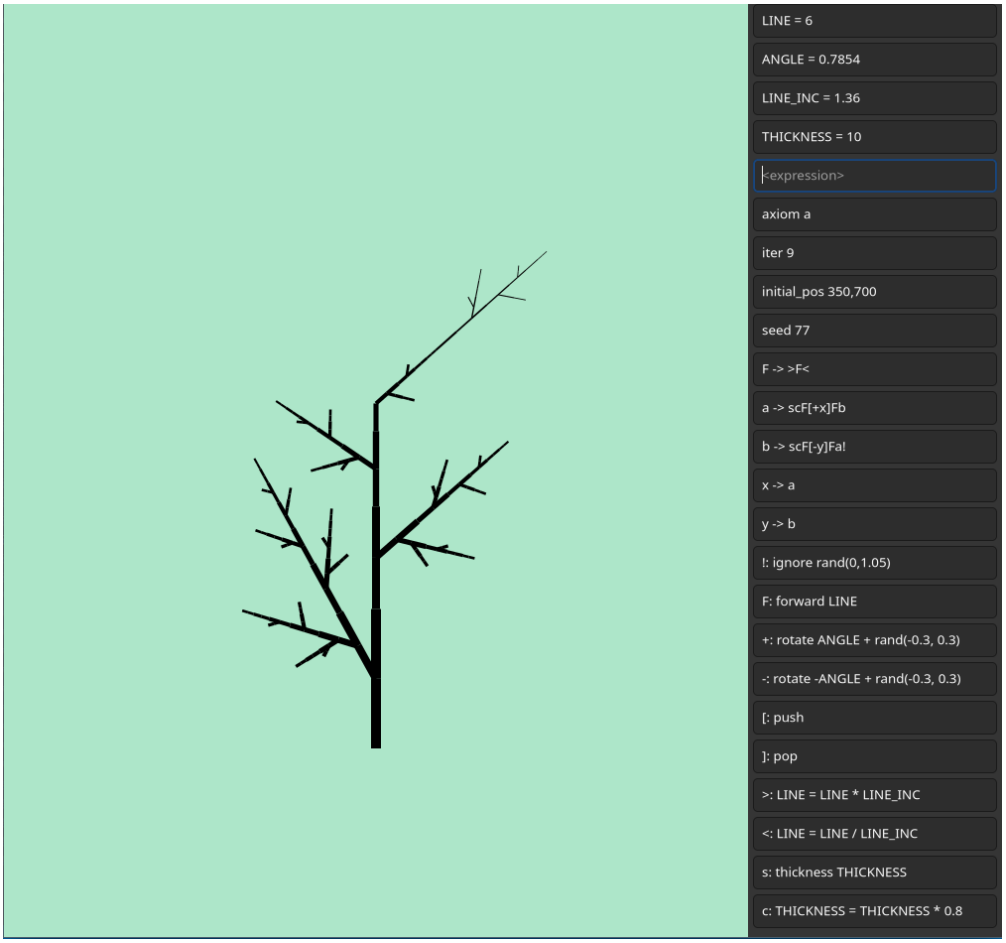
1. Created `gui-editing` branch
2. Merged `gui-editing` into `main`
3. Created `stochastic` branch
4. Merged `stochastic` into `main`
5. Created `development` branch (generic improvements and fixes)

1.6 2023-05-09

Work hours:
08:30 - 09:00: Ignore feature
09:00 - 09:50: Thickness feature
10:20 - 11:15: Generic improvements and features
11:15 - 11:30: Logging

Today I implemented the **ignore** and **thickness** features.

The ignore feature implements a command to ignore the next N symbols in the string.
I found the symbol defined as `!: ignore rand(0,1.05)` very useful to give some trees a more natural growth. Here's an example



I also fixed some bugs, implemented the `canvas` command to se the canvas size, the `initial_thickness` and the `initial_pos` command.

Git status:

1. Merged `development` into `main`
2. Created `ignore-feature` branch

3. Merged `ignore-feature` into `main`
4. Created `thickness` branch
5. Merged `thickness` into `main`
6. Created `development` branch (generic improvements and fixes)
7. Merged `Development` into `main`
8. Created `logging` branch

The plan for the next working session is to continue the documentation and maybe some GUI editor improvements.

1.7 2023-05-10

Work hours:
09:10 - 12:30: GUI Editor
13:10 - 14:15: GUI Editor
14:15 - 14:45: Status label
14:45 - 16:30: Documentation

Today I continued developing the GUI editor.
The various configuration textboxes are now separated into their own retractable section. If a textbox contains an incorrect line it will be colored red. Lines put in a different section than the one they should be put in (e.g. a variable in the rules section) will also produce an error.
The config section is hardcoded; there is a specific textbox for the `axiom`, `iter` and so on.

I also added a status label which tells the user if an error has occured (e.g. a variable or functioned used does not exist).

The editor does not yet create/remove empty texboxes when needed.

In the remaining time I continued the documentation. I only modified the `Implementation` section.

Git status:

1. Created `better-ui` branch
2. Merged `better-ui` into `main`
3. Merged `logging` into `main`

I am in line with the planning.

1.8 2023-05-11

Work hours:
08:40 - 09:50: Color feature
10:10 - 11:20: Refactor

Today I implemented the color feature.
A symbol may change the color of the line by using the `color <color>` operation. Any valid CSS color is a valid color.
I also added the `initial_color <color>` command.

In the second half of the working session I refactored some code. I had some problems with the memory management and GTK4.

In the next working session I will implement the dynamic editor behavior, namely creating or removing textboxes as needed.
I will also start implementing the animation system.

Git status:

1. Created `color-feature` branch
2. Merged `color-feature` into `main`

I am in line with the planning, expect for the dynamic editing feature which should have been implemented within today.

1.9 2023-05-12

Work hours:
08:20 - 13:40: Animation features

Today I started implementing the animations feature.
I had to refactor some code in order to be able to host this feature.
I was able to make a basic draw loop and animate the fractal, while keeping the same random seed to something changes between frames.
When the configuration of the fractal changes the animation is reset.
I have yet to add a play/resume button and the `frame` variable to the mathematical expressions.

This feature is very difficult to implement and I had some problems with the memory management.

Git status:

1. Created `animations` branch

1.10 2023-05-15

Work hours:
09:10 - 11:10: UI Playback and fixes
11:10 - 13:45: Refactor
13:45 - 14:00: Depth feature
14:00 - 14:50: Optimization
14:50 - 16:20: Injections

Today I implemented the UI playback. In the GUI I added:

- A toggle button to play/stop the animation
- A button to reset the animation (frame=0)

- A counter for the current frame
- The elapsed time between each render

I also added the following hardcoded variables

- **FRAME**: The current frame count
- **DEPTH**: The current stack depth
- **INDEX**: The current index in the fractal string
- **LENGTH**: The length of the fractal string

I did a lot of refactor and separated a lot of logic into its own file. I fixed some bugs and optimized the program overall.

I am also almost done with the `injections` features, but the parser is still broken.

1.11 2023-05-16

Work hours:
08:20 - 09:15: Injections
09:20 - 09:50: Logging
10:05 - 11:20: Dynamic entries

Today I finished implementing the injections features.
Injections can be made using the command `inject`.
E.g. `inject 0,FF 100,! 1000,+F-F`

I implemented a simple terminal logger and added a label under the canvas containing the length of the fractal string.

I also added some logic to add new empty textboxes if needed. The removal of them has yet to be implemented.

Git status:

1. Merged `injections` into `main`
2. Created `logging` branch
3. Merged `logging` into `main`
4. Created `dynamic-inputs` branch

The only feature left is the `import/export` of l-system files.
The plan is to finish the code this week, and only document in the next one.

1.12 2023-05-16

09:10 - 16:10: Import/Export

Today I implemented the import/export buttons.
The export works but the import doesn't, although the logic is there.

Everything is basically done and I will finish the code in the next working section. There are still a couple of small improvements that could be made.
I will then spend the rest of the last week to complete the documentation.

Git status:

1. Merged `dynamic-inpzts` into `main`
2. Created `import-export` branch

1.13 2023-05-22

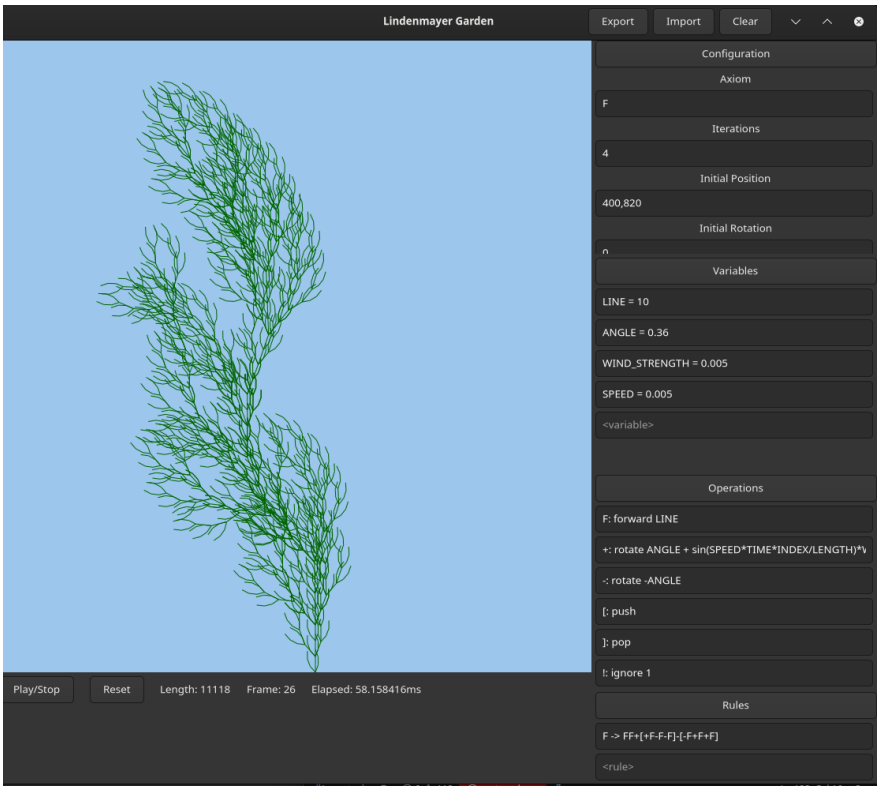
09:10 - 16:10: Minor features and improvements

Today I finished implementing the application. I implemented many small features and improvements, namely

1. Added a variable `TIME`.
2. Bug fixes.
3. Better error label message.
4. Added a **Clear** button.
5. Finished the import feature.
6. Added `jump` operation.
7. Added `dot` operation.
8. Added `background` command.

Everything was easy to implement and I fortunately did not have any problem in doing so.

Here's a picture of the final product



In the remaining time I continued the `Semantics` subsection of the documentation.

Git status:

1. Merged `import-export` into `main`
2. Created `development` branch
3. Merged `development` into `main`

I am still ahead of planning and the documentation is the only thing left to do.

1.14 2023-05-23

08:20 - 11:10: Documentation

Today I continued the `Implementation` section of the documentation.

I found two minor bugs in the code and easily fixed them. I also added a `Time` label to the UI.

Half of the documentation is done, I should be able to finish it in time. The major remaining part of the documentation is the testing cases and the implementation.

1.15 2023-05-24

09:05 - 10:05: Final Gantt Chart
10:05 - 16:20: Documentation

Today I continued the documentation. I added the final gantt chart and wrote my considerations about it.

I continued the implementation, added references, testing section and some generic changes.

I also created the abstract file.

The documentation is almost done and I should be able to finish it tomorrow. I mainly need to finish the implementation section and the introductory section about L-Systems.

1.16 2023-05-25

08:25 - 11:15: Documentation

Today I almost completed the documentation. I completly rewrote the `L-Systems` sections. I fixed some typos and added some content throughout the documentation. There are still a couple of things missing from the `Implementation`, which I will add tomorrow.

I also wrote the abstract page and everything is almost done.

1.17 2023-05-26

08:20 - 11:35: Documentation

Today I finished the documentation. I fixed some typos and added some content in various sections. I added the glossary and submitted my project.