

Chapter III

Objectives

In this project you will discover the basics of graphic programming, and in particular how to place points in space, how to join them with segments and most importantly how to observe the scene from a particular viewpoint.

You will also discover your first graphic library: `miniLibX`. This library was developed internally and includes the minimum necessary to open a window, light a pixel and deal with events linked to this window: keyboard, mouse and “expose”. This project introduces you to “events” programming. Don’t forget to watch the e-learning videos!

Chapter IV

General Instructions

- The executable file must be named `fdf`.
- You must submit a `Makefile`.
- Your `Makefile` must compile the project and must contain the usual rules. It must recompile and re-link the program only if necessary.
- If you are clever, you will use your library for your `fdf`. Submit also your folder `libft` including its own `Makefile` at the root of your repository. Your `Makefile` will have to compile the library, and then compile your project.
- You cannot use global variables.
- Your project must be written in accordance with the Norm.
- You have to handle errors carefully. In no way can your program quit in an unexpected manner (Segmentation fault, bus error, double free, etc).
- You'll have to submit a file called `author` containing your username followed by a `'\n'` at the root of your repository.

```
$>cat -e author
xlogin$
```

- You must use the `miniLibX`. Either in the version that is available on the system, or from its sources. If you choose to work from the sources, you will need to apply the same rules for your `libft` as those written above.
- Within your mandatory part, you are allowed to use the following functions:
 - `open`
 - `read`
 - `write`
 - `close`

- `malloc`
 - `free`
 - `perror`
 - `strerror`
 - `exit`
 - All the functions defined in the library `math` (`-lm` et `man 3 math`)
 - All the functions defined in the library `miniLibX`.
- You are allowed to use other functions to complete the bonus part as long as their use is justified during your defence. Be smart!
 - You can ask your questions on the forum.

Chapter V

Mandatory part

This project is about creating a simplified graphic “wireframe” (“fils de fer” in french, hence the name of the project) representation of a relief landscape linking various points (x, y, z) via segments. The coordinates of this landscape are stored in a file passed as a parameter to your program. Here is an example:

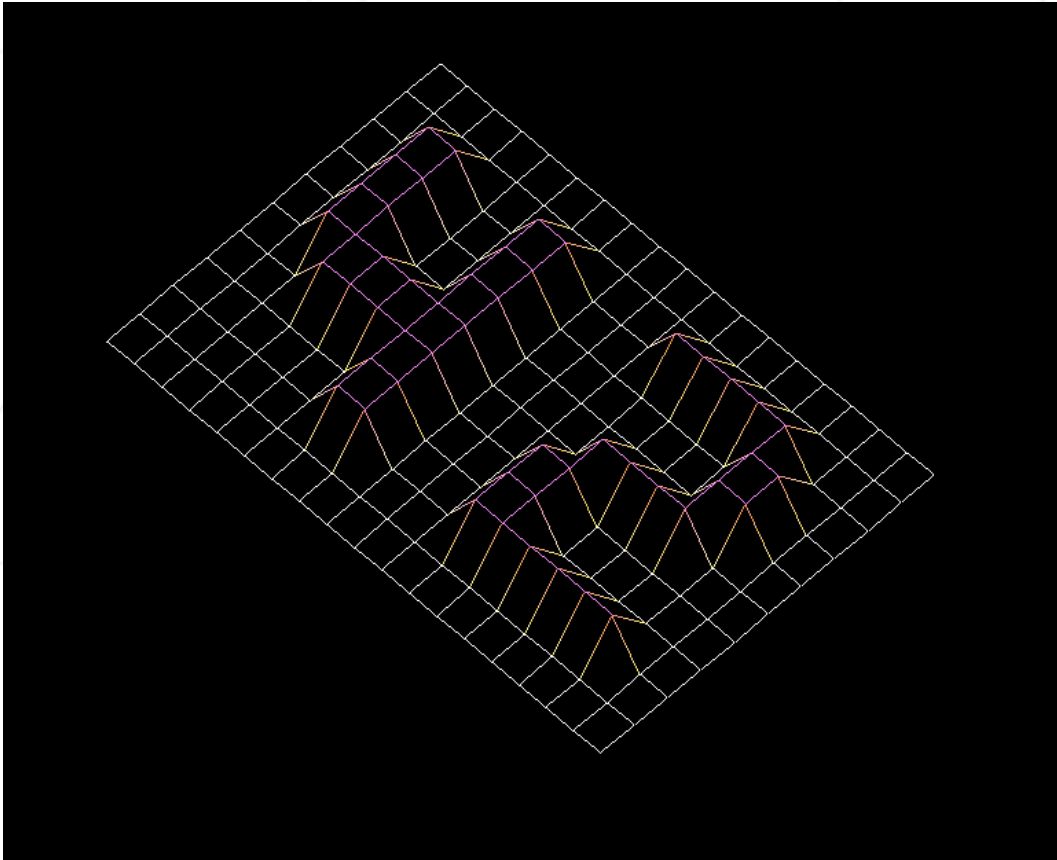
```
$>cat 42.fdf
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 10 10 0 0 10 10 0 0 0 10 10 10 10 10 0 0 0
0 0 10 10 0 0 10 10 0 0 0 0 0 0 0 10 10 0 0
0 0 10 10 0 0 10 10 0 0 0 0 0 0 0 10 10 0 0
0 0 10 10 10 10 10 10 0 0 0 0 10 10 10 10 0 0 0
0 0 0 10 10 10 10 10 0 0 0 10 10 0 0 0 0 0 0
0 0 0 0 0 0 10 10 0 0 0 10 10 0 0 0 0 0 0
0 0 0 0 0 0 10 10 0 0 0 10 10 10 10 10 10 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$>
```

Each number corresponds to a point in space:

- The horizontal position corresponds to its axis.
- The vertical position corresponds to its ordinate.
- The value corresponds to its altitude.

If you execute your program `fdf` on this file, we should see something like this:

```
$>./fdf 42.fdf
$>
```



Remember to make optimum use of your `libft`! The use of `get_next_line`, `ft_split` and `ft_getnbr` will allow you to quickly and simply read data from the file.

Regarding the graphic representation:

- You can choose the type of projection: parallel, iso, conic.
- You must manage `expose` correctly.
- You must be able to quit the program by pressing `'esc'`.
- The use of `images` from `minilibX` is strongly encouraged.
- Find attached a binary called `fdf` as well as the example `42.fdf` inside `fdf.zip`).



`man mlx`

Chapter VI

Bonus part

Here are a few interesting ideas of bonuses to create, or even to use. You can of course add your own original bonuses that will then be evaluated directly by your correctors. You will gain points if you are able to:

- Fill the boxes with a color according to altitude (green at the bottom, then brown, then white on top for ex.)
- Specify the color spectrum in parameter.
- Manage correctly the hidden surfaces.
- Change types of projection.