

Grobkonzept für die Maturitätsarbeit

Mikail Gedik

June 5, 2020

1 General

I plan to write my paper about fractals, like the Mandelbrot set. The goal is to implement a program which is able to calculate and display such fractals. As I don't have to specify the exact subject yet, I prefer to narrow it down later during my research.

2 Subject

My subject is the calculation of fractals with the help of modern computing power. There are two aspects I want to focus on: one is the mathematical part, the other is the computer science aspect. As i progress in my work, I will be able to look deeper into one of these two. The result of my work should be a program that can calculate, display and zoom into certain fractals.

2.1 Mathematics

I am less informed about the possibilities and methods of this field than the computer science aspect as I have only worked some time with the Mandelbrot set in the context of AM lessons.

Although I am inexperienced, fractals are still a core part of my thesis. This means that I will have to do a lot of research on this subject. In any way, fractals will be discussed in my paper.

2.2 Computer Science

Another part of my thesis is the coding of an application which is able to efficiently calculate and appealingly display fractals. To maximize performance, it is crucial to implement optimizations to the code.

Because my main coding experience is in Java, this will be the primarily used coding language. After the first working draft of my project is done, I can also imagine the following things (ranked from most important to least important) to improve the code or its speed:

2.2.1 Change Programming Language to C

Because C and C++ are much faster than Java, rewriting the code in one of these languages will result in a dramatic performance boost. I also consider leaving the frontend of the project in Java and only reprogram the backend in C.

2.2.2 Multithreading

Each point/pixel in the Mandelbrot set can be calculated independently from others. This makes it easy to compute it in parallel.

2.2.3 Graphics card

Instead of using only the CPU to calculate fractals, the GPU can also be used. But in order to do that, I'll have to learn a special programming language like OpenGL or Vulkan.

2.2.4 Graphical UI

Adding a nice and friendly user interface may be an important aspect, too. It is not essential, but being able to see and interact with the program in an intuitive way is important.

3 Title and Thesis

Because I have not yet selected a theme, the title is still open. As for the leading questions, I consider the following ones:

- Which fractals are interesting to take a look at?
- How can I program a tool that can visualize fractals?
- How can I save a sector of a fractal in a file (e.g. which file type)?
- How do I design a friendly UI?
- How do i optimize my program?
- How can I make use of my graphics card?
- Can I make my program run on different OS (like macOS, Windows, Linux)?

4 Tools

I am familiar with the two IDEs IntelliJ and Eclipse. Both offer compatibility with the VCS git.

5 Sources

Since the ETH-BIB and other institutes are closed due to the Corona-lockdown, I will be restricted to the internet.