

# Individual Project Reflection

Azer Hojlas, CINTe-19

May 29, 2022

## 1 Introduction

This assignment will briefly cover the general experiences gained during the project duration (mostly what went well), as well as what can be extracted and used in future projects down the road. Furthermore, improvements to the finished project are discussed in hindsight. To recap and refresh the memory of the reader, the project group I was involved with created an online musical synthesizer that could play waveforms inputted as mathematical functions. This instrument could be played through a piano-like graphical display, directly on the keyboard.

## 2 General Reflection

I was very lucky to land a group where the majority of the members were already experienced web developers. This meant that the project work could be started instantly, resulting in a functional albeit sub-par prototype during the first sprint. I personally however, am not experienced in web-development. Thus, learning proper Javascript was the top priority during the first half of the first sprint.

I was in charge of implementing the visual representations of the waveforms (which were inputted as functions), either as a single function or multiple. Due to the reasons mentioned above, progress was slow and the process confusing. Every few hours, I would hit a dead end, upon which someone would take a seat next to me and resolve the issue within a few minutes. More often than not, this was the role of the scrum master, who did not have an as fixed position as the rest of the team members, acting more like a "libero", assisting team members wherever help was needed. This process of failure and assistance would repeat constantly over the first two sprints. The point is, help was always readily available, in no small part thanks to a relaxed and friendly group-dynamic that did not punish sub-par performance when someone was stuck and asked for help. Initially, I was slightly hesitant to call for assistance due to the problems being miniscule in comparison to the level of experience of the other members. This would however delay the work process and waste additional time before I was ready to admit defeat and call for help. Suitable advice for my future self would be to call for assistance immediately. It sounds like an obvious platitude, but it is comparatively more advantageous to be slightly humbled (through asking for help) instead of wasting precious man-hours.

As the project work proceeded, the code became more and more complex, inevitably making it harder and harder for me to actively contribute on the same level as my colleagues. Thus I tried to make myself useful in any way that I could. A rough timelapse of my work-process looked something like this: First half of the first sprint was dedicated to learning Javascript, HMTL and so forth. The second half of the first sprint up until the end of the second sprint consisted of coding and implementing functionality. Due to aforementioned reasons (more on that later), the third sprint was dedicated to refactoring code and bug-fixes. The fourth sprint saw me more or less switch to miscellaneous and administrative work. This included preparing an informative expo-site and more banal assignments such as cleaning a midi-piano, fixing power-point slides etc.

## 3 Two Good Ways Of Working

Due to a lack of understanding the instructions in the assignment page, I am unsure whether the instructions allude to the academically accepted practices described in the "Practices and Tools" page (as well as the ones listed in the book), or if I am allowed to freely discuss specific ways of working within the group. Therefore, one course-related and one team-specific point will be discussed.

### 3.1 Version Control with Git

The project work was very much modular, meaning that the web-app functionality and subsequent labour were divided into mini-projects, where small teams of one to two members would take on said mini projects or module and implement it according to some predetermined specification. This division of labour brought about a set of advantages, namely that each and every group/module did not have to wait for another group to finish their work in order to develop their own module. This enabled a work environment where everybody could work independently in parallel, thus allowing for a fast and efficient work flow. This however does not come entirely without disadvantages. One of which is that the project could not be completely modularized, thus, some modules were not completely independent from others, i.e a worst case scenario could ensue where one module could impede one or several others.

The above stated problem was solved quite easily with git branching, that is, each and every branch contained a single functional module that could be improved and changed at the discretion of whoever was in charge of said module. These branches were later merged, and if any conflicts or issues were discovered, these were resolved before merging. This way of working proved to be a big asset, greatly contributing to cutting down on time-consuming bug-fixes.

### 3.2 Live-share

When engaging in group-work, members would connect to each others computers with a tool called "live-share". This tool enabled programming on one machine simultaneously by two or several developers, while allowing the participants to view each others work in real time. This was used as a substitute for or in lieu of pair-programming. What usually transpired was that two developers would work on different sections of the code while sitting next to each other. This enabled further division of labour within each module, thus having the same positive aggregate effect as the previously mentioned use of modularization. Whenever problems arose, one developer would drop whatever they were doing and either observe and review the others work or take over and let their counterpart observe. In practice, this resulted in pair-programming with additional.

## 4 Two Improvements In The Work Process

Even though I was generally very pleased with the project work, there are some improvements that could have been made, which will be discussed below.

### 4.1 Weekly Internal Code Review

As previously mentioned, the division of labour made it harder to keep tabs on modules that other team members were responsible for. As time progresses, increasingly complex functionality required increasingly complex code. If one was not directly involved in developing said module, the code would seem alien and it would become impossible to improve upon somebody elses work. This however only applied to those of us that were not seasoned developers. If I would have had a better understanding of the code belonging to my team members, I could have contributed to modules where I was not present at their inception.

The above mentioned problem could have been remedied by weekly code reviews. As a matter of fact, this is something that we did have, the focus however was not on code but on functionality and overall design. I personally believe that code reviewing was a victim of our own success. As the work progressed without issue, we became complacent and disregarded presenting every section of code that we had written. Instead we opted for continuing working on whatever we were doing. I do feel that

this was more of an individual problem, as it would have been easier to develop someone else's code if I were a better developer myself. All in all it was not all too troublesome as I found other ways of making myself useful.

## 4.2 Quarantine Protocol

Numerous members of the group became sick during the last sprint (including yours truly). This impeded our ability to work and was a drag on overall performance. Due to the strict rules regarding absence, we did not allow ourselves the luxury of adequate sick-leave. Thus, even though we would stay at home circa one or two days, fear of extra work ensured that in the affected person coming to school on the third day, healthy or not. Inevitably this resulted in the infection spreading to the rest of the group. This is something that in hindsight perhaps should have been expected. Enforcing face-masks, hygienic routines and proper distancing (classroom areas designated as quarantined zones) could have played a role in hindering the spread of the disease or at least slowed it down. This kind of issue will no doubt surface in future projects, regardless of whether it is school- or work-related. It sounds self-evident, however it is of vital importance that each member contributes to preventing spread of diseases at the work place. Whoever disregards this does so at their peril, i.e. risking the future success of the project.

## 5 Conclusions

Even though I am not very interested in web-development and music in general, it has been very rewarding to work with competent team members, not only because I consider them smart, but also because it was socially enjoyable to work with them.

On a different note, it was unfortunate that progress was halted in the last sprint due to disease. One suggestion for future course iterations is to change the conditions under which one is required to do additional work. I am aware of the fact that the course responsible already has given much consideration to this, and that more lenient conditions probably would lead to abuse of the system (healthy students calling in sick instead of going to school etc). I do however believe that additional rules regarding sick-leave would benefit the course. How this would be organized I leave to someone that is more administratively capable to figure out.

With that said, I am pleased to say that the course has been appreciated and instructive (both in direct work related experience and soft skills regarding group work and cooperation), and I wish both the course-responsible and my team members an enjoyable and happy summer break.