Introduction

Over the course of this project we have set aims and deadlines and learned numerous new skills, this report aims to outline some of the tasks that went well and what could have been improved. Much of the success of our game came down to the planning phase and setting out a plan for both the game and the development of the game. Most importantly, the development process went generally smoothly resulting in an overall game we are happy with.

Planning

From the start of the project it's clear that we all had a general vision of the elements of our game, however one problem we found is that not all of our visions were aligned to integrate into the final result. Our initial project report did a good job of outlining the individual features and functionality of the game, from the core aspects such as the user's bars to the way we were going to incorporate our data sets, yet the exact way these would work and interact was not sufficiently documented. Aspects such as the way the user would interact with the core game loop did not get fully ironed out and discussed until later into the project when this should have been a key discussion point in both the report and our group.

There was some better parts of the planning, for example our ability to split our group into two subgroups helped speed up the development. Doing this successfully allowed us to work concurrently on the project without getting in each others way. As the UI was able to be setup without the immediate integration of the core game, we were able to develop a more refined product without rushing at the end to fit the gameplay into the UI. Furthermore, this allowed us to make use of our own talents such as Adam's art experience. This built confidence in our group and overall resulted in a better project.

Finally, our ability to plan out the deadlines and tasks of our project was a little hit or miss. The Gantt chart we initially developed did not really come into much use and was not used throughout. Instead we worked better just setting our deadlines based on the assignment and handling out individual tasks. This is where Trello was more helpful. Using Trello meant that we could keep track of what was done, needed doing, and was less important. This split up the workflow much better and kept us both on task and on time.

Development

Overall development of the game went well after initial difficulties with members getting to grasp with C++. At the start of the project, the code is noticeably less well implemented, there is a mix of library aspects such as some functions using the Qt string while others use the C++ std string. This sort of stuff began to get ironed out as members became more and more familiar as the code base grew. C++ best practices began to get implemented such as using constant references to pass structs through to functions without slowly copying and overall code proficiency resulted in a much better final implementation of our game's engine.

We also were confident in the functionality of our game as we started to test aspects using unit tests. The unit tests were useful for checking that our code worked but we found that they were much more useful at ensuring that our code *kept* working. Having the unit tests meant that if we ever changed areas of code, knowingly or unknowingly, we could be assured that the final product would be working. CI/CD also helped by automating this process and letting the group know whether the tests were working, this was further amplified by the use of the discord integration which made sure we knew the results.

Conclusion

If we were to do this again it's clear that the process would be much more refined and improved. Some of this is just inherently, our overall coding proficiency, both in general and in C++, greatly improved along with our ability to work in a team. Other aspects would be improved by our new found understanding of the importance of planning and communication. Overall, we are happy with the result of our project and the process that led us there.