COMP710 Group Game Project Post-Mortem Nikkolas Diehl- 16945724

Overview:

For my COMP710 group project with Jordan Carter, we chose to create a 2D racing game. The process for creation of the product was set out early on via trello as project design, development and reflection. During the very short design phase, Jordan and I brainstormed and wrote up some ideas for a game we wanted to create. Afterwards, we worked to quickly write up a technical design document and game design document.

We went over quite a few ideas during the design phase but ended up settling on some specific requirements that we knew we wanted. These initial design ideas where to have the game be: Top down, multiplayer, cars, racing, city racing, 2D game obstacles, and maze like. Once we had these primary ideas, we decided on creating a city racer game with a crucial design element. We wanted the core of the game to be heavily data driven so we decided early on that the “map” that the player would race on would be bitmap driven.

We also decided to use the Box2D physics engine to speed up our development process and allow for much easier, more controlled and accurate game play.

Development phase was quite busy but straight forward. I had a lot of other responsibilities at the time of developing the product so my work flow was sporadic and broken up across the 4-5 weeks we worked on the product. I used SVN to iteratively build the game alongside Jordan until the final gold build was finished.

What Went Right:

Compared to the struggle, and inefficiencies I felt in the first solo project, this team project had a lot more going for it and a lot more things went really well.

This mortem will only go over what went right for me and my tasks/development on the project.

Firstly, my proudest creation in this project had to be the level loading and bitmap reader. I wanted to be able to create a dynamic, easy to use and easy to setup level loading system that only needed a map number to load a map. Initially there was some issues upskilling with SDL’s surface reading technology, but I quickly progressed and once I got it working, I was pleasantly surprised with the visuals of the map. I found it quite satisfying effectively getting it working first time because once I finished coding it, and ran it, it immediately worked and displayed the map with the correct sprites in the correct positions. I ended up driving around for a while searching for issues and genuinely happy and excited that I got it working so well.

Secondly, the GUI system. A lot of the initial design was pulled from Jordan’s code, but after improving it a lot and working to abstract the code and objects through many layers, I was extremely pleased with myself and the GUI system over all. I wanted to make sure that creating buttons and HUD elements alike was easy and efficient so when I finally completed it, all you needed was text, a position and a function call back. The hidden customisability behind it though makes me exceptionally happy and I wish to use my own system in future.

Lastly, the particle system. Last time, during my personal project both the particle system, and basically every other system within my code base was very fixed and hack-y. I didn't work very hard to abstract and compartmentalise the code so a lot of it was hard coded and fixed within the game. This time around, I made sure to abstract everything and make sure there was still a lot of customisability and great looking graphics so I worked add heaps of options for how particles are emitted and what you can do with them. When it was finally finished, I could easily create a new emitter, feed it some settings and run it and it would just work. It was extremely pleasing to drive around with a tone of different particle effects working in the background.

What Went Wrong:

The biggest problem I had in my previous project was time management. The exact same could be said about this project. Even though we where able to combine a lot of our work from the previous game and also work together, due to a lot of external responsibilities, we did not have time to complete what we wanted originally.

In our original ideas; although not expressed in the game design document, we wanted to make a multiplayer game to be able to verse another player and race around. This turned out to not be feasible within the time frame and we had to cut it. We were also not really able to create any obstacles besides the main walls and buildings within the scene which disappoints me.

In terms of coding, we where never able to complete the initial plan and add obstacles and sounds. Whilst there are basic menu sounds, that was the limit of what we could complete during the time frame we where given. I am quite sad that there was no time to add a fully-fledged sound system, but I am happy with what we did manage to complete. And at least we have working menu sounds.

Lessons Learnt:

I learnt quite a few lessons during the development of the project.

Time management has always been an issue with almost any university project I've ever worked on. I needed to try and split my time up more and manage what I was to work on each week. Unfortunately, due to extreme time pressure for research and development, I did not have enough time to even manage my time. This inevitably let me down.

Design phase: Whilst we did manage to create a product that somewhat resembles the original game design document and technical design document, there are many things lacking. I learnt both in my solo project and in this team project, to more carefully plan out what was actually feasible. I have learnt more and more about my true working speed and have found that I usually over estimate what I am able to complete.

I also learnt a lot more this time around about the SDL framework and was able to work more in depth with the rendering kit provided. In my previous project, I did not go into depth with SDL development and this led to a lot of performance issues and problems. This time around, I went more into depth with rendering and the SDL framework and it is how I learnt how to create the bit map reader and was able to massively increase the performance in the game.

I also learnt a lot more about particle systems and how to optimise them best to run in a way that allows me to create many particle systems all running at the same time whilst barely dropping frames. I am quite impressed with how the particle system turned out and I have learnt a lot about creating them in future. Next time I would like to learn more about creating particle systems that run fully on the GPU to increase performance even more and add even more functionality to my particle systems.

Thirdly, I learnt quite a lot about function call-backs and compartmentalizing my code. In my solo project, most, if not all of my GUI was hardcoded and worked only within the singular menus and situations they where created in. During this team project, Jordan showed me the usefulness of function call-backs that allowed me to create buttons and functionality dynamically and anywhere within the code without any issue and with relative ease. I am extremely happy to have programmed in C++ now and feel that my overall knowledge and skill level in C++ has increased 10 fold.

Lastly, whilst I did not work on it as much as Jordan did, I learnt a lot about the creation and use of Box2D within a 2D game. Previously, in my solo project I used my own physics engine which, whilst working, became very laggy and inefficient when in more extreme situations. This time around, with the use of Box2D, we where able to hugely optimise the process and game and allow for some really cool physics simulations. The experience of creating a large game/project in C++ with a team mate will remain helpful for the rest of my development life.

Conclusion:

In conclusion, I enjoyed working alongside Jordan to create the 2D racing game and felt I have learnt a substantial number of new things within this project. From the GUI system with function call-backs, to bitmap reading and level loading; I thoroughly relished my experience. Whilst there was a lot of time management issues, and a lack of content that we where able to create, I am happy with what we where able to finish. And whilst it wasn’t smooth sailing, I definitely want to explore C++ game development more in future and expand my game development repertoire.