Contributions Report

Group 36

Friday 6th Decemeber 2019

Contribution Report

For the duration of Assignment 2, the production of our game Jungle Hunt, our group all made significant steps individually towards the common goal, however it was only possible to complete the feat with full teamwork and cooperation. Although some minor issues were raised between the team mid-way through the assignment, these were insignificant enough to avoid having any major effect on our final game. Because of this reason, we managed to complete the game relatively early and to a high standard.

As we had already produced a Design Document for our final game, heading into the first meeting we all had a strong idea for what we wanted to produce, and the general direction we would head in order to complete this. This meant that people could immediately start to work on their tasks individually. In the initial meeting it was decided that rather than stick to the classes which we first laid out in our Design Document, we would flex around which people could work on when.

Although this method would allow for everybody to work at their own pace and help each other on the classes, it would provide difficulties in the start to create a functioning base on which we could expand on. To combat this, George took on a heavy load of work at the beginning of the project to get the JavaFX working on a basic level, producing a screen which could load a level and objects into the level. This allowed the rest of the members of the group to implement their features in the classes, and thus creating a game.

In the first meeting it was also discussed that we would use a single GitHub repository, with branches for every member to develop. When being merged, a separate member of the group would merge a branch. This was so that 2 people had checked the code before it was pushed to master, and any glaring mistakes could be stopped and edited out before they reached distribution. Early on, player movement was established, which allowed for us to test interactions with other classes. George also developed the Lumberjack class (named such as it handles the logs) early on into the development. This class was put in place to easily manage log messages and help with debugging code.

Throughout the development of Jungle Hunt, and because of the modular nature of the project features could be gradually added and tested. This meant that each feature could be either fully or partly implemented before or during other the development of other features. We used Scrum to quickly work on different parts of the system, where everyone continually knew what they wanted to implement, and would implement it on their own branch before pushing to the master.

Towards the mid part of the project and after the majority of the code had been completed George also set up an 'Experimental' branch where either non-mandatory features or code improvement features that may affect the testing of new features could be developed and tested. This branch allowed us to investigate the idea of several new classes that might affect how the game would run and how better to access information from classes within the program. Eventually however, and due to the modular code base, everything developed on this branch, position class, flipping rather than rotating sprites and a more efficient level builder ended up in the final project.

As agreed in the first meeting, and as repeated from Assignment 1, we decided to hold two weekly meetings where everyone could both discuss the progress of the project before breaking up into smaller groups where the code could be analysed and worked upon in a casual environment. In these meetings, we discussed what had been implemented so far, what was planned implementation and how far we needed to go in order to reach the next stage of our implementation. A key feature of our meetings was team development, and more specifically pair programming. This meant that anyone who was struggling to implement a feature, or just wanted to implement a feature more quickly would get a group member to help. This method of working together also allowed us to easily bug fix in person, rather than having to fix them remotely, although try as we might this still occurred, usually at 3am.

We managed our time well as a team, and evenly distributed work. This meant that from an early stage we felt that we were ahead of where we should in theory be. Because our implementation was flexible, implementation of features was often easy, meaning that work could often be very productive in short stints of time. This efficiency in our work and meetings meant that, although we continued working late into the project, we had completed the specification in adequate time.

Issues

Throughout the development process of creating Jungle hunt, every member of the group had some issues, which had to be resolved in order to complete the development efficiently. As we worked in a team to complete the tasks, these issues often meant working together to achieve the outcome.

GitHub

Early on into Assignment 1 we had decided to use GitHub as our version control platform, using GitHub's services, we had to learn how to use it clearly and effectively to all work on separate parts of the project, which could then be merged. Throughout the course of development, a few merge conflicts arose which had to be resolved, usually caused by someone being a derp and not pulling master. These often were fixed quickly and efficiently, as we tried to never work on the same part of the program at the same time with everyone implementing their own features.

Speed

During the development of the program, especially towards the end when multiple features were running each tick we noticed that the game was extremely RAM hungry, more so than Google Chrome, some work was put in by both George and Maz to reduce this as at times the game would simply take minutes to respond or simply just crash.

This would have been a big problem, especially when creating the video and if the examiner tried to run our code. In order to combat this, we looked into the code and spotted where we were making multiple JavaFX elements and not closing them. This quickly chewed up RAM, and once fixed because a fast running game. Earlier on in the project we had also had issued with loading multiple sprites ad-hoc at build time and as such decided to make all of the images rather than related to their objects but their classes which in turn halved memory consumption in the early stages of the project.

Significant Changes

As a group we did not have to make any huge changes to the hierarchy of the classes however one class 'Impassable' was removed due to it being totally under-utilised throughout the project and several new classes were created to handle specific methods. Some examples of these would be the BFSVertex class, used solely for calculating the trajectory of the Smart Enemy, the Save class, used solely to write data to a file and the Position class, which is used almost as a tuple or co-ordinate in an almost useful way because Java doesn't have support for these built in.

Whilst it is true we did not have to make many changes to the class heirarchy, we definitely did have to make changes to the methods within them, most noticeably here is how our Player and Enemy move in the game. Initially we had planned to just return an integer value, from 0 to 3, to represent a direction but early on into the project it became easier to handle Player movement internally and simply return the new grid with the Player object already re-located, later on in the project this same logic allowed George to implement Enemy movement in mere minutes as it was a simple case of running a for-each loop over the Enemy array which was built at build time.

Members

Chuks Ajeh

Chuks worked predominantly on the Smart Enemy class with Angelo, and thus had a large contribution to the most complex algorithms within the Game. Working in tandem, and using Pair programming they created a working Smart Enemy. Chuks also helped with bug fixing for the rest of the group. At the end of the project Chuks also contributed as a voice over for the team video.

> ”For this project I oversaw the Smart Enemy variant, Scene Transition structure and video voice over. I am not the strongest programmer either so along with Angelo, this was a chance to strengthen my programming skills and practice working in a team. Getting up to speed with project meant using Git which I was very unfamiliar with but alongside learn it online other member within the team helped me enabling to contribute my work to the team. The Smart Enemy proved to be a very challenging task for me.
>
> I, along with Angelo was tasked with implementing the wave front algorithm; A Breadth First Search followed by a path finding algorithm. Initial conceptualisation was not an issue and implementing a generic version was fairly straightforward, but modifying the code to meet our specifications proved very difficult. Working alongside Angelo provided much needed support and a valuable learning experience. Consistent communication of ideas and progress was key, and the result was hugely rewarding; a fully function smart enemy.
>
> Towards the the of the project I pivoted towards scene transition and voice over which were largely straightforward. Scene transitions involved thinking through how the user would interact with the system in the best way possible in terms of easier and user experience. The voice over was merely me talking about the game our project functionality and extra features.
>
> Overall, I believe my contribution was effective, in the future I would like to play a more active role in the programming sections of the task should my skills suffice, apart from that the team worked extremely well playing to each person’s individual strengths.”

Angelo Balistoy

Angelo worked with Chuks on the Smart Enemy class. Along with Chuks they created a working Smart enemy which searched for, and moved towards the player. Angelo was also the lead when it came to Javadoc, and ensured that most classes at the start received proper JavaDoc comments so that the JavaDoc could be generated. Along with this, Angelo helped bug fixing for the rest of the team.

> ”For this project, my main job was to Javadoc and comment lines of code and code clean-up. To a lesser extent, I also took any job that needed to be done such as Class implementation and debugging. I am one of the weakest programmers on the course so I was glad to be assigned like this; where I could learn what coding as a team is like without jeopardising the entire project. I had trouble with learning Git but eventually learned how to work with it to contribute to the team.
>
> The JavaDoc and commenting was not difficult, but needed for not only the specification but for future code revisions. It allowed us to know what each method should do and made sure we did not deviate too much from the original design. My biggest problem was implementing the SmartEnemy class.

> I had encountered an algorithm for the Smart Enemy known as Wave-front planning which is a BFS search followed by a path-finding algorithm. Implementing the algorithm separately was not a problem but modifying it such that it melded well with other methods was quite difficult. To help with this, I worked alongside Chuks Ajeh in a pseudo-pair-programming approach and it proved very effective.
>
> Despite this difficulty, I really enjoyed this task as I learned a new and interesting algorithm, how to implement it, and practised adapting tried-and-tested algorithms to work for a project. Around the end of the project, I was still working on the SmartEnemy, as well as cleaning up code for efficiency and readability.
>
> Overall, I believe I contributed effectively in this group due to my efforts in debugging, JavaDoc, commenting and the implementation of classes such as SmartEnemy. As a team, I believed we worked extremely well. Luckily, had no problems with working together which allowed us to finish on time with no rush at the end."

George Carpenter

George took on a leading role early into the development once again, and created a base for the game to be built upon. They helped with the implementation of the Message of the day, game builder, enemy movement, player interactions and teleporters. Throughout development George also worked to clean code and refactor it in order to keep conventions.

> "For this project I once again took on the mantle of "the boss baby" in that I guided the implementation right from the start and throughout, beginning the implementation of many features and leaving it up to others to finalise them and instead only guiding them on how it could be done. Throughout the project I also took on the role of "cleaner" in the sense that I would periodically check over any already implemented features and try to refactor, extract and clean up their implementations to improve the code whilst trying to maintain the same logic behind the scenes.
>
> Early on in the project I found myself banging my head on a desk, a lot, so to ease the pain I decided I should get high.. Wait no, that's bad for you don't do it. To aid myself at the time, and with the version of helping the other members of the group in the then future, I set up a class specifically designed to help with debugging and error hunting which was affectionately called Lumberjack, because it logs.
>
> Over the course of the project it has proved invaluable as it has helped many members of the group to track down errors in nearly every stage of implementation. Towards the end of the project I took a more observational role in ensuring everything we had implemented was up to a good standard and improved elements where obvious improvements could be made.
>
> Overall I feel as though this project went extremely smoothly for our group as once again we all knew where we fitted into the team, who was doing what, where they were strong and where they may need assistance. If I had to pick my favourite points of the project from features I implemented or helped to finalise they would have to be the moment I implemented Enemy movement and the game suddenly just started to appear in front of me as from that point on the project really started to feel as though we were in the endgame and everything we had been working on since the middle of October had finally come into being."

Blake Davies

Blake took on the lead role in the design of the aesthetics for the game. Working excessively on sprites and game art, designing all of the sprites from scratch, and creating a unique and captivating feel to the game. Following this Blake also helped with the implementation of the User and Save classes.

> "For this project I was mainly working on the aesthetic side of the project, like the sprites and the video for the project, so that all of our sprites used for our game were unique and fitted the style of the game exactly how we wanted it to. Though this was

> my main focus I also helped with the implementation of some of the classes like the User class and the Save class with Sam and Maz. While implementing this class we did have few issues with creating a user for a while as their save file would never actually save as their name it would just be a null, but thanks to a bit of help from George's Lumberjack class the error was resolved, even if it did take a while.
>
> I think as a team we worked extremely well together, I think that everyone in the team knew their strengths and we capitalised on that and gave people roles that fitted the well for the project. My favourite part from the entire assignment was when I finally finished all of the sprites for the game, and replaced all of the placeholder sprites and the game looked like how I envisioned it from the moment that we chose our jungle theme."

Ioan Mazurca

Ioan, more commonly known as Maz, took lead on implementing the GUI and Menus, and their interactions with the game. Maz had a massive impact on the project implementing all the menus and how they create and load files. Maz also helped work on Wall Enemy movement, and making a level complete and restart at the right times. Towards the enw of the project Maz also developed a minimap, an original idea Maz had had since the start of the project.

> "For this project, most of my effort has been spent in linking the actual game with the menus as well as some of the extra features that we implemented, such as the inventory display and the mini map. Although in the beginning of the assignment I did not have too many tasks to do, I managed to help George in implementing the Wall Enemy and the Teleporters, which required extra attention when it came to the special cases that these classes involved. Considering that I had some previous experience working with CSS and that the GUI takes a lot of time to implement, I decided to step in in order to take care of the front end part of the project.
>
> While implementing the GUI, I had some difficulties in understanding how to efficiently make the use of the stage, scenes and panels, but after practicing with the application structure elements I succeeded in thoroughly learn JavaFX. The team worked well together, as everyone managed to make a contribution to the project. The feature that I'm most proud of is the implementation of the minimap, as I believe is a really nice one to have, as it allows the player to see where the goal is, but also see places where they could find the keys and the tokens."

Samuel Roach

Samuel took lead of the Save function, however helped implement many classes. Working on High scores, saving, dumb enemies, line enemies, player item management and door interactions. Predominantly Sam worked on the save function, getting it to save level files and user files after levels are complete. Sam also helped bug fix and worked in between members of the team to solve problems. Towards the end they also implemented music, created solely by themselves.

> "For the duration of this project I worked excessively on the back-end of the system, implementing some core functions required in the spec. Early on I created line enemy movement, and dumb enemy movement, so that we had some enemies to avoid when testing. I then moved onto player inventory work. This was a lot of finicky things, ensuring that Keys and Tokens are stored properly, so that multiple tokens are not stored. Once implemented I then got Keys and Tokens to interact with their corresponding doors, so that the player could begin puzzle solving.
>
> Following this I then worked on the Save file, and did most of the File reading for the project. I created many functions to write certain parts of the file, which could then be handpicked depending on what you needed. Finally I helped implement high scores and users, which could then be used by Maz to show the leader board at the end of each level. Throughout the process I continuously bug fixed and worked to help other members as much as I could."