Group Project Reflection

Statement

For the group project, I was delegated to the Persistence module which involved creating a parser for the game levels written in JSON files. I was also responsible for the definition of the levels themselves, which helped with developing the parser because I knew what I was working with.

The following table below contains my most significant commits to the module on GitLab and their corresponding description:

URL Link to Most Significant GitLab Commits	Description of Commits
https://gitlab.ecs.vuw.ac.nz/course-work/swen225/2020/groupproject/team11/chaps_challenge/-/commit/2552171733cac35f9efe5fe7f7960cd914a6b650	This commit involved adding the very first iteration of the JSON parser I planned on implementing. I had very little idea of the Jackson JSON library initially so at this point in the development I was experimenting with how it worked.
https://gitlab.ecs.vuw.ac.nz/course-work/swen225/2020/groupproject/team11/chaps_challenge/-/commit/39ee54b68969e6b8c997c2cf19b74899d12c5fd9	This commit was for the final version of the JSON parser I planned on implementing from the beginning. I decided to use BufferedReader for the final edition because it was a more reliable method of reading the file and transferring the data using ObjectMapper into the Maze class.
https://gitlab.ecs.vuw.ac.nz/course-work/swen225/2020/groupproject/team11/chaps_challenge/-/commit/08ebdd2d6342d3abaa4376035400239f11246143	This commit is for the JSONLoaderTest class which handles JUnit testing for the Persistence package. Tests included loading of file, performance and failure tests etc.

Group Member Rankings By Contribution

Name	ECS Email	Ranking
Matt Romanes	romanematt@ecs.vuw.ac.nz	3
Frances Julaton	julatofran@ecs.vuw.ac.nz	4

Scott Jacobs	jacobsscot@ecs.vuw.ac.nz	5
River Rivera	riverarive@ecs.vuw.ac.nz	4
Stuart Hood	hoodstua@ecs.vuw.ac.nz	1

What Knowledge And/Or Experience Have I gained?

Being involved with the group project helped me gain a lot of experience in developing full-fledged programs like games. Prior to SWEN 225, I only knew how to add to a program (since the programs given to me were pre-defined by the lecturers). So to create a program entirely from scratch was definitely something I had to acquaint myself with. This group project (as well as the previous assignments) also gave me a glimpse of the real world in terms of collaboration with my classmates. Although I knew my group members really well, we would clash over certain things like implementation, coding style etc. So in this regard I did gain experience working with other people, which is an essential skill in the real world.

What were the major challenges, and how did we solve them?

One of the biggest challenges we faced during the group project was managing the group itself. We often didn't communicate, and we often fought over little things like coding style, implementations of methods etc. More of this will be explained in the last section where I go into further detail as to what happened in the group during the duration of the project. But for this section, I will say that managing the group was difficult for each and every one of us, despite the fact we knew each other well (except for one of our members who came to fill in the 5th slot). This was down to the fact that we didn't really establish a reliable system of ensuring adequate project management, including relying on software to keep the group in check. We could only apply a temporary fix to this issue by ensuring communication between each member, asking them on their progress etc.

Another challenge we had to face during the group project was actually understanding how the game worked. Having never played Chip's Challenge ever, my group struggled to get started because we lacked understanding of the rules of the game as well as the actions the player and objects performed. Throughout the duration of the project, we only relied on the handout provided to us by the lecturers. Looking back in hindsight, this problem could've been avoided. Only relying on the handout severely hindered our understanding of the game, resulting in a fairly incomplete game. We never really overcame this problem because we didn't think about it; it's as simple as that.

Which technologies and methods worked for me and the team, and which didn't, and why?

The technology that didn't work for me was GitLab, the place where our team would upload files and directories concerning the group (as per the instructions for the project). The first problem involved mostly me; my lack of understanding for GitLab resulted in most of GitLab's features not either not working for me or I didn't understand the features well enough to figure out how they worked. The second problem with GitLab was the fact that when it came to submission, our team's attempts to merge our different branches together simply didn't work for any of us. And when they did work, some of our programs were overwritten, and some of the merges conflicted, so our submission program was botched.

Discuss how you used one particular design pattern or code contract in your module. What were the pros and cons of using the design pattern or contract in the context of this project ?

The design pattern I used for my module was Mediator; this design pattern defines simplified communication between classes. At the time I was working on the Persistence module, I realized that my classes would need to communicate with other classes for it to work, which is why I went for this design pattern. My module relied on the Maze module, and the person responsible for that module ensured that his classes could easily communicate with my classes. In the end, this was a good call from me; I didn't think any other way could've worked.

As I was delegated into the Persistence module, I was also given the freedom by the course to use Java external libraries. In our project, I used the Jackson library, perceived by many as the best library for JSON handling due to its performance and simplicity. Because of its simplicity, the design of my parser doesn't really involve much apart from using the BufferedReader class. Both BufferedReader and Jackson provided classes and executed the parsing of the files themselves; I determined how it would be done. My design involved making a BufferedReader object read the file provided then used Jackson's ObjectMapper class to read the data from the BufferedReader object into the Maze class. I found no problems using this method of file reading.

What would I do differently if I had to do this project again?

If I was given the opportunity to work on this group project, I would still work on the Persistence module, but perhaps consider using another JSON library. Throughout the development process, I found it difficult to acquaint with the Jackson library. Despite the praise it gets for simplicity and performance, there was a learning curve involved with using Jackson. I know this is the case for the other libraries, but having looked at the likes of Google GSON, which I understood more of, I would probably use this if I had the chance.

The other thing I could do differently is my implementation of the parser itself and its integration with other modules. I created my parser without consulting the group, and as a result when it came to integration, it didn't work out. Basically, my parser didn't integrate with anyone else's modules. I'll explain more of this in a later paragraph, but I decided to talk about it for a bit in this section because the lack of integration was partly my responsibility too. If I had the chance, I would make sure that my parser could be easily integrated into other people's modules.

What should the team do differently if we had to do this project again?

As mentioned in an earlier section, there were moments of animosity in the group. One of our group members (codenamed St) had absolutely no sense of diligence; they made the least contribution to the project, and actually attempted to pull out not only our group, but the course itself. They also sabotaged the efforts of another group member (codenamed Sc), which resulted in Sc's Lab 5 marks being deducted. St often pulled health issues out of nowhere to justify not doing any work. This made the development process more difficult for others and myself because we were reliant on their module. These problems went as far as reporting the member to the tutors and lecturers, because it was getting out of hand. To sum it up, while our communication was adequate, our project management skills were unprofessional and lacked discipline, resulting in a botched game. So if we were to do this project again, we would remedy these issues by enforcing discipline upon every group member and utilize a project management system which will help keep the group and the project in check.

Another thing we would do differently is allow our modules to easily integrate with one another, like connecting two jigsaw puzzle pieces together. As mentioned in an earlier paragraph, we worked on our modules independently, as in we wouldn't communicate with each other on how our modules were going to integrate with one another. So when it came to playing our game for the first time, nothing worked. The reason; our modules wouldn't integrate with each other. For example, my Persistence module didn't integrate well with the Application module, which in turn didn't integrate with the Renderer module and so on and so forth. If we had to do this project again, we would ensure that our modules could integrate very well and very easily, meaning we would have to create entirely different implementations for each individual module.