Name: Daniel McGuckin

User Name: dam44

Group Project: N06

Role: Programmer

I found the project ran very smoothly. Most people turned up to each meeting and we understood what we each had to do.

My Tasks

My jobs were spike testing of the database, UML and implementation of many parts of the system, specifically handling persistent data and writing the shop, monster worth, fighting, breeding, monster aging, logging in/out, registering and monster/user design.

The spike testing for the database connection had a few problems. I set it up in a regular java project and it worked perfectly. I thought, mistakenly, that this would therefore work for the entire project however when applying it to the web-based java project we found connection problems which continued throughout the week. Looking back I realize that I should have tried connecting it up with a web java project, but in my mind at the time I thought that the difference between the web java project and the regular one was just on the front end. i.e. the output using the server with JSP's rather than using a main method. I was evidently mistaken for problems occurred and possibly I should have been slightly more thorough at this angle.

My writing of the persistence class to connect the database to the java code was very successful, largely thanks to the spike testing. Although connection through the persistence.xml had problems, the persistence manager class worked well and we could easily persist, update, remove and search for monsters in the database from the beginning of the implementation and testing week assuming the database connection was working.

I was also given the task of creating the UML for the project. This included the use-case, class and sequence diagrams. I felt this went well; we used the use-case diagram to aid us in deciding and implementing our user requirements. The class diagram helped us in putting the system together and the sequence diagram aided in our understanding of how the system was meant to run. I had never created a sequence diagram prior to the project so this was a nice learning curve.

In terms of coding I was tasked with writing a large amount of the core system. I prepared a lot of it before coding week and some I wrote in the first two days. This included the fighting algorithm that determined a winning monster when passed two as parameters. It calculated the winning monster by comparing stats and using a random wheel. I also wrote the breeding algorithm which used much of the same techniques but generated a new monster by taking the parent stats and using these with a random wheel.

I also wrote the shop (buy and sell) algorithms. This involved changing/updating monster and user details in the database. I also wrote the algorithms for determining a monster worth, this was used by the shop and looked at the monsters stats and decided a worth based on those and it's age. I assessed the monsters age with another algorithm which outputted a number between 0-10 depending how far the monster was from its peak age. This number would then change the monster's effectiveness in battle and alter its worth.

I wrote the logging in/out and registering algorithms. These accessed the database to check if user names and passwords were correct and created new entities in the database when registering. I initially added some regular expressions for sanitizing input but we commented this out for ease of access and I did not have time to implement it at the end.

I also created the monster aging algorithm. I did this by using a separate time thread when the user was logged in. The theory of this was that when the user logged off the monster would stop aging and would only age so long as the user was active. The thread worked but was not perfect and did occasionally create some memory issues. I created the algorithm to check if users were logged in, this was for if a user had closed their window without pressing the log-out button or if the user had gone away from their computer for a long time. This was another thread that activated (like the aging thread) when the user logged in and died when the user was deemed as logged out. When users did things like breeding monsters it would change their state to active, if after so many minutes they did not do anything to change their state to active they were deemed as logged out.

If I could do the project again I would make sure the database was 100% working on a Java Web application before coding week, I made the mistake of not testing it on this platform and it cost us time. I would also make our time management better, especially in terms of having more code completed by coding week (Although most of it was).

## The Team

I teamed up with Ben during coding week to aid in integrating the back code with the jsp's and to attempt to get our friends list working. I found this pair coding style fun, productive and I thoroughly enjoyed it.

I found the team dynamic flowed with no personality clash or in-house fighting. We were directed well by our project leader and we all got on very well to get the job done. Members worked very well together with a united persistence and determination to get the job done to the best standard we could. I found that by the end, we had all handled the stress well and I found that I was proud to be a member of this team. I would work with them all again and enjoyed almost every second of it. I wish I could write more on this subject but there is little drama to write about, we worked efficiently and I enjoyed it.

## Leadership

Our team leader (James Slater) was very organized and led the team well. We, as a group worked well under his easy, relaxed yet determined leadership. He motivated and inspired the group to work efficiently even under a lot of pressure. Although almost all the members consistently performed, when one or two slacked for certain periods of time he dealt with them fairly and ultimately I believe the reason almost everybody in the team worked so consistently was influenced a lot by him.

I always knew my task throughout the entire project, and if I ever had a question James was ready to answer it. We had group meetings every week outside of our compulsory meeting where tasks would be set and questions could be answered by a member of the team. There was not a dictator-like policy in our team, everybody could speak their mind and James would listen.