Personal Report – David Haenze

Role in group

At our first group meeting I was assigned the role of deputy QA manager, with Phil being the QA leader. However, it became apparent over the successive group meetings that I would be more valuable to the group as one of the lead programmers, hence I spent very little time assuring the quality of our project, with Phil having done most of that work. I still kept the title, however.

After we realized the skill sets of the various group members, I in essence was the deputy head programmer, under Thomas Hull. Together we figured out much of the program. I wrote the algorithms for breeding and fighting, as well as improved the initial design of our JSP files. Additionally, I worked on some of the client-side validation, for selling prices and the like. Finally, I spent a lot of time on just general code tidying up and Javadoc writing.

I was also assigned the role of "GitHub guru", and I was tasked with figuring out how git works and how to interface it with our IDE of choice, Netbeans. Unfortunately, I was less than succesful in this venture, and Phil had to step up and take over for me, especially once I became more preoccupied with the coding side of our project. However, I still continued to research the functioning of Git, especially as we encountered various problems whenever we tried to sync commits.

Troubles encountered and resolution

Initially, there was a lot of trouble with figuring out how the code worked – Thomas had written a lot of it all on his own, so the general outline, i.e., most of the database files (Monster.java, MonsterDOA.java, etc), the servlets and the JSPs had (at least) rudimentary code, so getting my head around it took a little time, but with some research, both online and with Thom, I managed to understand our code and work with it.

Our version control system, GitHub caused us a lot of trouble, with deletion of code happening more than once. Part of this was due to failures on my part, as I did not wholly understand the functioning of git, and part due to improper training in the use of git, leading to us coders overwriting the wrong files at times and the like. However, after many hours spent perusing the various online manuals for Git, I came to understand it and these troubles disappeared.

Overall success

Overall, I feel our group was successful, at least in terms of providing a functional program that fit the specification and met most of the testing requirements. Personally I would never play such a game, and there were many other things that could have been done better, especially our interface – due to time constraints, with our interface having to be effectively re-written the morning of the submission. I myself had more of a vision for it, including images for our monsters, play-by-play description of a fight, such as "You attacked, but missed! Enemy monster attacked and did 10 damage!" etc, but in the end we were only able to implement a simple "You won! Here are you monster stats".

To sum up, we did a good overall job fulfilling requirements but could have delivered a much better and more polished product with better time management.