Inf2C: Software Engineering
Coursework 3

Creating an abstract implementation of a auction house system

Project log

Michael Andrejczuk s1703773

November 27, 2018

1 Introduction and overview

This document acts as a log of my work on this project, completed as a joint project with Dylan Thinnes.

In total I estimate about 26 hours of work on this project. This is estimated from my Git commit logs, and split as follows:

- 2 hours preparation: planning, preparing skeleton files (for code, report, project logs), reading recommended documentation;
- 2 hours on unit tests and JavaDoc for the Money class;
- 4 hours on unit tests for our own code;
- 10 hours for implementation;
- 8 hours for our report and project logs.

2 Saturday, November 17

This was our first chance to look at the coursework specification as a team. Our aim was to:

- Solidify our understanding of the task, ensuring we both understood the requirements;
- Plan our approach, deciding on key milestones;
- Begin implementation.

We achieved:

- We decided we would try to get the bulk of the coding side done by Saturday 24th, and everything else done by Monday 26th.
- We read over the recommended documentation (on coding style, logging, JavaDoc, JUnit) together and discussed how it would fit into our approach.
- We decided what development environments we would use, choosing to stick to our usuals: Emacs/Vim with Git and Bash scripts. We considered using Eclipse but decided against it, as we were more familiar and productive with our usual tools.
  On reflection this was a good idea, as it allowed us to work quickly with familiar tools. For example, rather than needing to learn how to run tests in Eclipse, I simply wrote a small Bash script to compile and run our tests. Having a similar development environment also allowed us to work together (relatively) seamlessly.
- I also pushed the coursework contents to our shared (private) Git repository. We had been using this previously to write our report, and decided to continue using it to ensure our development work was also tracked.
  On reflection it was an excellent choice to set this up, as it allowed us to work in a distributed manner, resolving merge conflicts as needed. The commit logs also acted as a very good project log, for tracking both the work I did and the time I spent on it.

- I began adding and completed the required unit tests for MoneyTest.java. I also wrote the JavaDoc annotation for Money.java.
  We tried pair programming for this and neither of us found the practice beneficial, so we decided not to use it going forward.

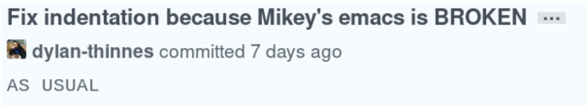
In total I spent around 6 hours working today, and we achieved everything we intended.

3 Sunday, November 18

Work today was mostly focussed around implementation:

- We discussed how closely we would follow our CW2 spec, choosing to only make small changes when necessary.

- An issue popped up with my Emacs settings indenting Java code incorrectly, meaning Dylan had to re-indent it when merging my changes. Thanks to Git we noticed this quickly and I was able to change my settings, allowing us to continue to write code in a consistent and coherent style.
  On reflection this did cause some conflict, so I'm glad we sorted it quickly. One heated commit message sums up the mood:



Fix indentation because Mikey's emacs is BROKEN ···

dylan-thinnes committed 7 days ago

AS USUAL

- I focused on writing tests, leaving most of the implementation to Dylan for the day. The tests written today were primarily unit tests, rather than system-level tests. We considered swapping off writing code and tests across the day but decided against it.
  On reflection, this was a good choice. It meant that each of us could focus on what we were working on, without needing to context switch. Instead we planned to swap off who would be doing code and tests on alternate days.

- We tried following test-driven development but decided against it, choosing instead to focus on coding out a rough implementation and then looking to pass tests once the implementation was more concrete.
  On reflection this did allow us to focus more on code, but meant that we had to spend some time patching up edge cases.

In total I spent around 7 hours working today, and we achieved everything we intended.

4 Thursday, November 22

I spent about 1 hour working today, patching up an edge case that we hadn't noticed. In hindsight if we had followed a more test-driven approach, we would have caught this sooner. Luckily, our implementation is modular and easy-to-understand enough that this was an easy fix to make.

5 Friday, November 23

Today was again focussed on implementation:

- I implemented two new features, working around the code Dylan had added already. Since we had agreed a coding style and approach previously, it was easy to implement these and understand the code they were a part of.
- With the due date quickly approaching we discussed altering our plans so far, and I began tracking TODOs across the code in my log.
  On reflection this was beneficial as I could easily refer back to this when looking for things to do.

In total I spent around 3 hours working today, and we achieved everything we intended. The most valuable part of today was that I spent the bulk of the time on implementation, becoming very familiar with the codebase Dylan had written so far. This meant that in following days I was very comfortable working with it.

6 Saturday, November 24

Today was in part implementation, and in part documentation.

- We began really working on getting tests passing today — indeed, we succeeded in getting tests passing by the end of the day. I added a great deal of logging code.
  On reflection this proved very useful when running tests later as we could more easily track the execution path of failing tests. On more than one occassion it allowed me to identify a bug very quickly.
- I added our report skeleton and began work on our class diagram. As we had already finalised our design at this point, and due to its similarity to our CW2, it was relatively quick work. On reflection, it helped that we were confident in this design at the beginning of the coursework, as it made writing the code and the report relatively painless.

In total I spent around 4 hours working today.

7 Sunday, November 25

- I fixed a few more edge cases.
- I finished writing tests for our code. On reflection this was beneficial as it allowed us to test and find edge cases we hadn't considered before; it also meant that we could be confident in the correctness of parts of our implementation.
- I started writing up my project log. This was useful as it allowed me to reflect on the work done so far.

In total I spent around 5 hours working today.

8 Monday, November 26

- I checked our code compiled on DICE.
  On reflection, both of us choosing to develop on Linux was beneficial as it allowed us to work in a Linux environment from the beginning, making the transition to DICE seamless. Additionally, choosing to develop using Git meant that all we needed to do in order to transfer our code to DICE for submission was a simple `git pull`.

- We jointly read over the report and discussed our post-mortem, reflecting on what went well and what didn't. These discussions are included in the bulletpoints above in my reflection.

In total I spent around 1 hour working today. With the coursework being finished and submitted, we met our expected target date for submission.