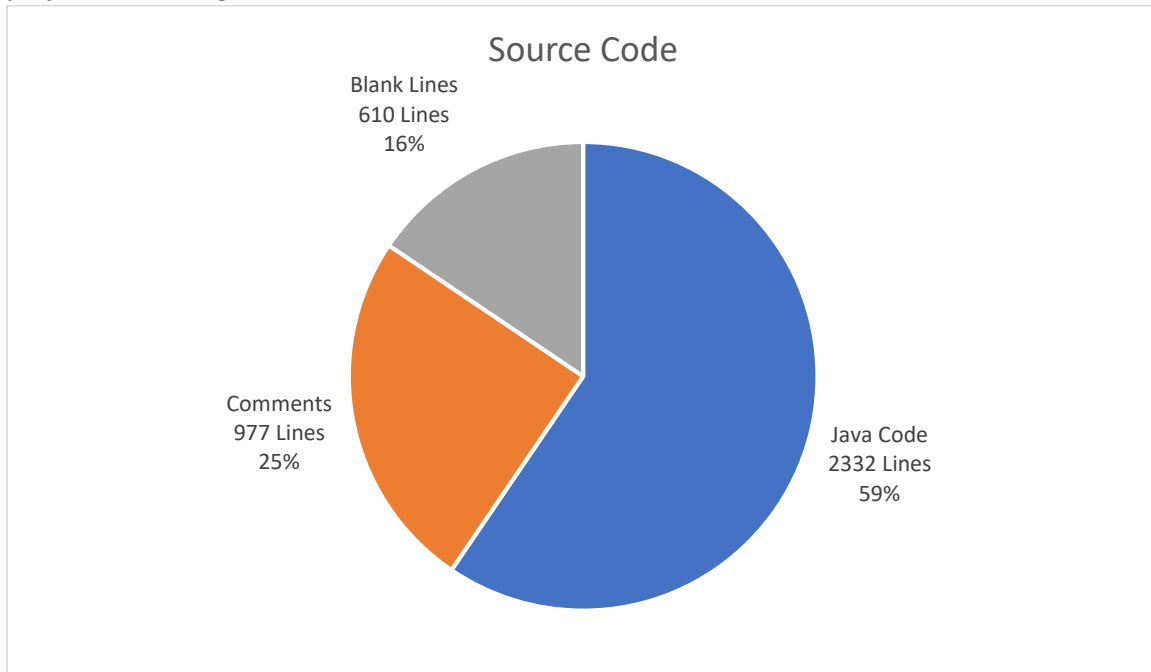


QA Report

In this document, a review of our Quality Assurance measures is made. Going through the measures one by one, the compliance to the measures defined in the QA concept is reviewed and evaluated.

Javadoc

The documenting of our source code by means of Javadoc-style comments was consistent and always more than enough. Source code documentation has held a high priority throughout the whole project. Thus, a significant share of the source code are comment lines.



Verdict: Very useful and well maintained

Unit Tests

This QA measure fell short in the project with the reason being the lack of resources. By splitting our initial group of four into groups of two, it was not feasible for us, to write sufficient tests for all major functions and classes. Instead, the need for core game development has held a higher priority over unit tests.

Fortunately, our development was not test-driven and thus, the lack of unit tests did not harm the development progress in a significant way.

In the end, the project contained 3 testing classes with a total of 7 tests, which does not match the expected results.

Verdict: Not well enough maintained, but lack thereof was not a major setback

Profiler

Even though the profiler provided interesting and fun insights about the program, we rarely felt that having a profiler in the development process was crucial.

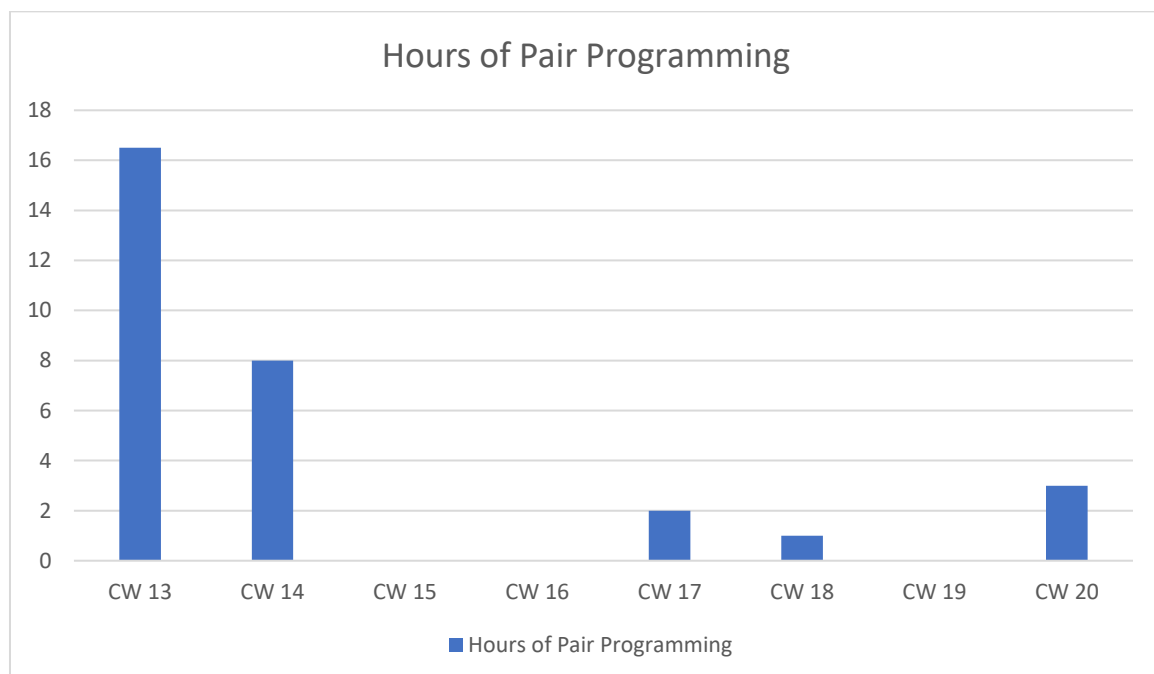
The crashes that occurred would be almost always traced back to networking issues and/or invalid arguments surrounding the source code itself, instead to a memory leak or bad resource management.

We think, that the scale of the project wasn't big enough for a profiler to play a big role in the development and thus, rendered the profiler near useless for us.

Verdict: Always maintained, but proved to be of little use

Pair Programming

In order to draw insights from the end results, we have been keeping a protocol on pair programming on the wiki of the repository. In the following a plot of the protocol:



As seen in the graph, the amount of pair programming has been declining ever since the group was in split in two after milestone 3 (April 10, calendar week 15). Following the restructure, the same workload had to be distributed between two members instead of four, which resulted in less resources available to allocate to pair programming.

Verdict: Not able to maintain due to lack of resources following the retirement of two members

Code Review

This measure was well maintained throughout the project, because the working members reliably updated each other after each progress. By adhering to the arrangement of reviewing big chunks of code upon pulling, we were able to constantly review each other's code and improve upon it as well.

Having clear and concise commit messages made maintaining this measure less challenging.

Verdict: Well maintained and improved code quality greatly