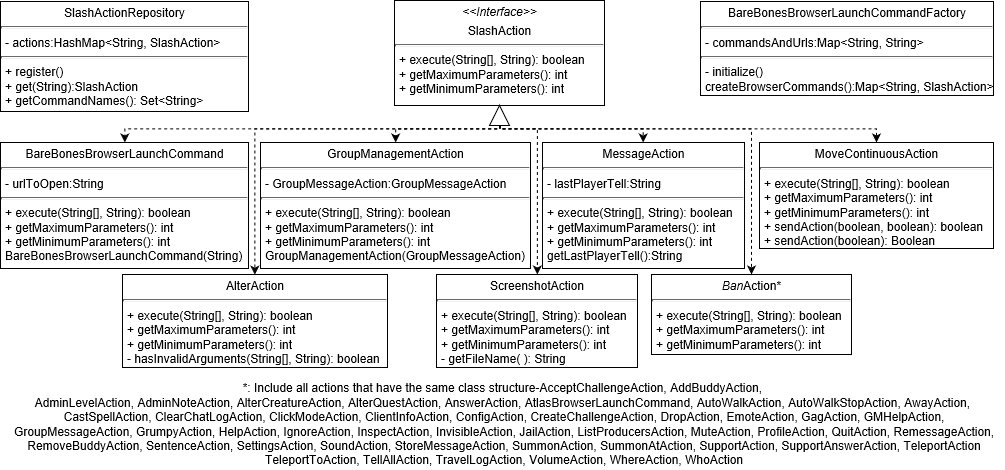
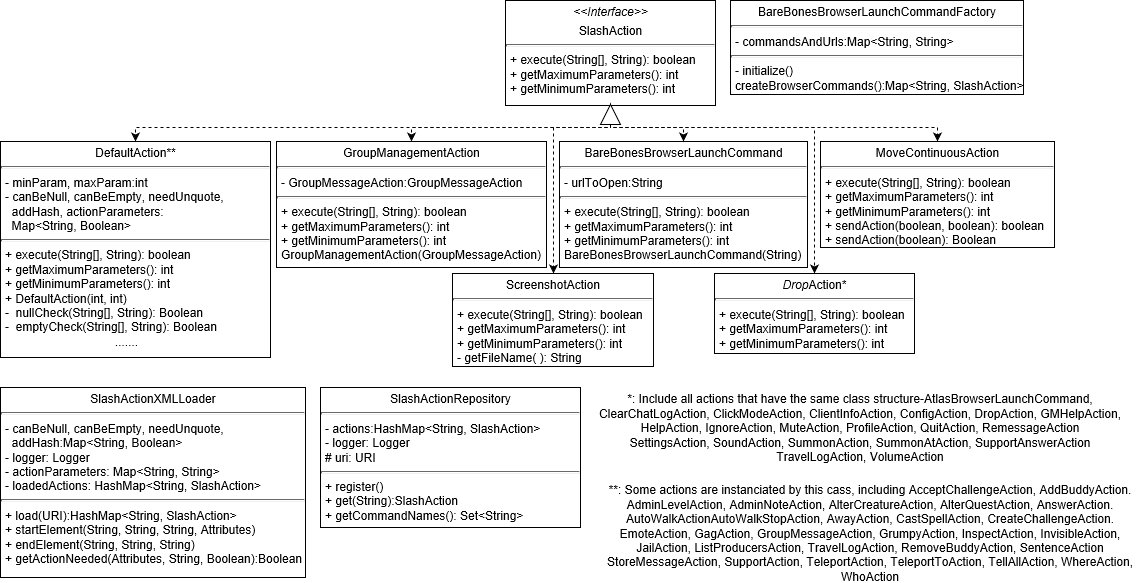
1.



UML graph of structure in package games.stendhal.client.actions before migration



UML graph of structure in package games.stendhal.client.actions after migration

Changes: a new DefaultAction is created for those migrated actions, including BareBonesBrowserLaunchCommand, AlterAction and some classes previously represented by BanAction. A new SlashActionXMLLoader is added into the package; SlashActionRepository is updated.

2.

Total size of class files for 32 classes that has been migrated: 56.6kb; after migration, these classes are represented by 3 class files and an XML file, altogether 19.8kb, other files are left unchanged. Code coverage: before migration: instruction: 37%; branch: 27%; complexity: 37%; line:37%; method: 42%; class: 97%; after migration: instruction: 70%; branch: 54%; complexity: 65%; line:68%; method: 77%; class: 90%.

Migration of actions to XML file effectively reduces the line of codes for them by standardizing parameters, so as to achieve same function by a smaller database. This is helpful for code maintenance, as parameters in XML files are easier to read and update. With a codebase before migration, we probably need to add nearly an entire class file for a new action, but we only need to add less than 5 lines after it. Since all migrated, disappeared codes are almost simple repetition of the same part, such a migration improves the efficiency of code; also, less code also means less likeliness that bug appears.

3a.

To complete an issue successfully, one need to do WBS carefully and have a clear idea of what kind of techniques shall be used. If one has no idea of, say, what kind of data structure should be used or what steps to be taken, then definitely (s)he cannot even make an estimation. Apart from that, if one’s issue depends on that of others, then chances are that (s)he’s procedure is blocked by incompleteness (for example, lack of robustness, unsatisfactory code organization or even bugs) of their code, and the time spent on waiting their reparation is not easy to estimate. In this group coursework, I worked on test coverage and gave a time estimation of 3 days, however some test gives an error but they are supposed to pass. The reason of those errors was that handling of null pointers was not given in source code. My groupmate responded to me quickly and got that problem solved at once and eventually such an interruption did not take me much time, though our actual time spent was still half an hour longer than estimation, but it did prove that such a circumstance may appear and the problem cannot be expected to be solved that fast.

3b

Writing test code surely takes some time, but they will save more in a longer term. Running tests exposes unexpected code behaviors explicitly, given a reasonable coverage, without the need (though sometimes still need) to go under single step debug, which can be even more time-consuming. Also, testing help us handle cases that is less likely to appear in our software. For example, when unexpected input is given, we want to get our application still behaves properly, or at least goes into fail-safe state, instead of a total collapse. The higher test coverage is, the more likely that lines that cause errors can be discovered at once, and corner cases handled properly. Apart from that, test codes can help monitor possible code changes in the future. Some changes are made that are improper, then corresponding tests will fail or even gives an error, making (s)he know that such a change on codebase shall be reconsidered; without test, an error cannot be so easily inspected, and it will probably appears in release version, which is definitely what we do not want. In this group coursework, out tests did not work as expected, thus did what test should do perfectly. Some of our test gave errors at first where they are supposed to pass, and the cause was that null pointer was missing in source code at that time. We handled corner cases later, and those tests passed smoothly. Without these tests we would have forgotten to do those handling and codebase should be more likely to give out bugs then.