CPSC 3720 – Assignment 2

Overview

In this assignment, you will:

- Extend the functionality of another developer's software.
- Keep track of your progress using version control.
- Write passing unit tests for any additional Model classes.
- Use mocking to test your new Controller class.
- Determine how well your code is tested using code coverage.
- Use static analysis to detect bugs and avoid dangerous coding practices.
- Maintain a coding style with a style checker.
- Generate documentation for your code using doxygen.
- Use continuous integration to automate the running of software engineering tools.

Instructions

1. Fork the provided codebase at http://gitlab.cs.uleth.ca/course3720/assignments/old-maid
2. Extend the codebase to allow a player to also play "Rummy"
    a. The rules for Rummy can be found at https://bicyclecards.com/how-to-play/rummy-rum/
    b. Allow the user to choose between playing the two games.
    c. Implement the new game by modifying the existing code as little as possible.
        i. As the provided code uses MVC, two new classes should be needed: a Controller and a View.
        ii. You may need to make some changes to abstract classes as well for generalization.
        iii. Fix the problems your changes cause to the existing game.
3. Provide a retrospective report in a `report` directory that addresses the following questions:
    a. What was the easiest aspect of extending another developer's software?
    b. What was the most challenging aspect of extending another developer's software?
    c. How would you rate the quality of the code that was provided for extension? Provide a justification for your assessment.
    d. What could the previous developer have done to make new feature development easier?
    e. What could you have done with your software that would have made the job of the maintenance/new feature developer of your software easier?

Grading

Your assignment will be assessed by looking at:

- The status of most recent build in your repository's GitLab pipeline nearest the deadline.
- Playing the game Rummy and Go Fish.
- The number of passing unit tests and the code coverage for the unit tests of Rummy.
- The number and size of the memory leaks. This value should be 0 for the running the game, but not necessarily for the tests (i.e. the provided tests for Go Fish have known memory leaks).
- The number and type of errors identified by static analysis and style checking (there are some existing ones that should be easily fixed).
- Examining the generated documentation for Rummy.
- The quality and thoroughness of your retrospective report.

Submission

- There is nothing to submit, as the marker will follow the forks from the various repositories.
- Make certain that the marker (`mark3720`) has `Reporter` access to your fork. **You may receive 0 for the assignment** if we can't access it.