SIMON FRASER UNIVERSITY
Department of Computing Science
CMPT 276: Project

Project, Phase 2: Implementation

Deliverables

You will be working on your group repository on SFU's GitHub. Each team member should contribute continually to all phases of the project.

Create a "Documents" directory in your team's repository for maintaining the report file of this phase. The TAs will mark the report along with the history and final version of your code on SFU GitHub on the specified deadlines. In addition to the phase deadline. See Canvas for deadlines.

Implementing Your Game

The second phase of the project revolves around the implementation of the game. You will be creating the classes you have designed in your UML diagrams, and implementing the functionality you have specified in your use cases. You may adjust your design where necessary, and create additional classes and methods as needed. You will need to make many design decisions as well as decisions about what external libraries and frameworks to use, e.g., for the construction of the user-interface. You are allowed to use external libraries for specific tasks (e.g., creating a user-interface or parsing a file) but you must implement the core logic of your game's algorithm yourself, and this code should be written in Java.

Build Automation

You must use Apache Maven for managing and building your project. Based on Maven's documentations, "Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information." Build automation tools such as Maven facilitate the process of understanding, building, executing, testing, and in general managing software projects by providing a uniform build system.

One goal of Maven is to provide an infrastructure for better development practices. As such, each Maven project includes a POM file, a source tree for your application's sources and a source tree for your test sources. This is the standard layout for Maven projects (the application sources reside in `${basedir}/src/main/java` and test sources reside in `${basedir}/src/test/java`, where `${basedir}` represents the directory containing pom.xml) The specification, execution, and reporting of unit tests are part of the normal build cycle using Maven. The test source code, which you will write in Phase 3, will be kept in a separate source tree that is parallel to the production code source tree.

Anyone should be able to automatically build, run, and later test your program using Maven. Please refer to Maven's description and features for more information.

Code Quality

While implementing the game, you need to make multiple decisions. Always have the quality of your code in mind during the decision makings. If you notice any flaws in your design from Phase 1 that negatively impact the quality of your implementation, revise your design and justify your actions in the report of this phase.

Code Style and Documentation

You should aim at making your code as understandable and maintainable as possible. In addition to paying attention to the design of the program and the quality of the code, you should also provide appropriate comments and documentation along with your code. While implementing the game in Phase 2, you must also write descriptive comments and appropriate JavaDocs comments.

Deliverables for Phase 2

- **The code**: you should follow the standard layout for Maven projects, where the source code is maintained in the `${basedir}/src/main/java` directory, where `${basedir}` represents your project's home directory (the directory containing pom.xml). You will create all your "production" packages and classes within this directory.
  (You will create your tests in `${basedir}/src/test/java` in Phase 3.)
- **The report**, which should:
  - describe your overall approach to implementing the game,
  - state and justify the adjustments and modifications to the initial design of the project (shown in class diagrams and use cases from Phase 1),
  - explain the management process of this phase and the division of roles and responsibilities,
  - list external libraries you used, for instance for the GUI and briefly justify the reason(s) for choosing the libraries,
  - and discuss the biggest challenges you faced during this phase.

Create a directory in your repository named "**Documents.**" Create a report The report should be named "**Phase2Report.pdf**," and should be at most 4-5 pages.