**Software Test Plan (STP) Template**

Items that are intended to stay in as part of your document are in **bold**; explanatory comments are in italic text. Plain text is used where you might insert wording about your project.

This document is an annotated outline for a Software Test Plan, adapted from the IEEE Standard for Software Test Documentation (Std 829-1998).

Tailor as appropriate. Where you decide to omit a section, you might keep the header, but insert a comment saying why you omit the element.

**Nicholas Visalli**

[**nvisal1@students.towson.edu**](mailto:nvisal1@students.towson.edu)

**Amro Al-Suwaida**

[**aalsuw2@students.towson.edu**](mailto:aalsuw2@students.towson.edu)

****

**FreeCol 0.11.6**

**Software Quality Assurance Plan**

**Version: (0.11.7) Date: (05/16/2018)**

**Document History and Distribution**

1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision # | Revision Date | **Description of Change** | **Author** |
| 1 | 04/17/2018 | Generating a report of needed packages | N.V and A.A |
| 2 | 04/24/2018 | Refactoring and improving the quality of those packages | N.V and A.A |
| 3 | 05/08/2018 | Adding features to the software | N.V and A.A |
| 4 | 05/15/2018 | Testing the features | N.V and A.A |
| 5 | 05/16/2018 | Completing the STP document | N.V and A.A |

**Table of Contents**

1. Introduction...................................................................................................................... 1

2.Test Items........................................................................................................................ 2

3. Features To Be Tested........................................................................................................... 3

4. Features Not To Be Tested.................................................................................................. 3

5. Approach................................................................................................................................ 3

6. Pass / Fail Criteria................................................................................................................. 5

7. Testing Process...................................................................................................................... 5

8. Environmental Requirements............................................................................................. 6

9. Change Management Procedures...................................................................................... 7

# 

# **1. Introduction**

**1.1 Objectives**

Our goal for this project is to improve the quality of various packages in the software. We created a report that indicates the priorities of packages.

**1.2 Testing Strategy**

Specific Testing Strategy:

* Use CodePro to detect cyclomatic complexity for each package.
* Determine packages that have a cyclomatic complexity above 50. This number is the highest maximum from each package, not the average.
  + We decided to improve the following packages
    - Net.sf.freecol.server.model
    - Net.sf.freecol.server.ai.mission
    - Net.sf.freecol.server.ai
    - Net.sf.freecol.common.networking
  + Not only did these packages have a cyclomatic complexity above 50, they also contained more classes than other packages that had high complexities.
* Refactor the chosen packages to reduce the average cyclomatic complexity as much as possible without changing or defecting any previously implemented game logic.
  + In order to ensure that our refactorings do not deter the game from running as intended, we run the “ant testall” command every time we finish refactoring an individual class. If all 430 tests still pass, then we move on to the next class. If not, we remove our refactorings and try again.
* Generate JavaDocs for the specified packages.
* Introduce a new change to the game
  + We decided to change the font of the panel headers to BeachSocietyLight. This change can be immediately noticed when starting a new game. The panel header that reads "Setup new game" will be changed.
* Perform a regression test after changing the font.
  + We did not need to introduce any new tests for this simple quality of life improvement. Rather, we ensured that all existing tests still passed after changing the font. To do this, we simply run “ant testall”. If all 430 tests still pass, then we determine that our quality of life update was successfully integrated within the game. If not, we examine the cause of failure and reimplement our change.

**1.3 Scope**

* This update covers the following packages (refactoring and generation of JavaDocs)
  + Net.sf.freecol.server.model
  + Net.sf.freecol.server.ai.mission
  + Net.sf.freecol.server.ai
  + Net.sf.freecol.common.networking
* This update also changes the following files (quality of life update -- font change)
  + Resources.properties
    - Path: FreeCol/data/base/Resources.properties

**1.5 Definitions and Acronyms**

* This update does not introduce any new acronyms that require interpretation.

# **2. Test Items**

(Specify the test items included in the plan. Supply references to the following item documentation:

* The following packages and files were modified in this update
  + Net.sf.freecol.server.model
  + Net.sf.freecol.server.ai.mission
  + Net.sf.freecol.server.ai
  + Net.sf.freecol.common.networking
  + Resources.properties
* Link to the official documentation for the specified files
  + [**http://www.freecol.org/docs/FreeCol.html**](http://www.freecol.org/docs/FreeCol.html)
    - Specially, refer to section 3.6 (Customization)

**2.1 Program Modules**

* New modules were not added in this update.

**2.2 User Procedures**

* Users will immediately notice the font change when starting a new game. The panel header that reads "Setup new game" will be changed. All other panel headers are changed as well, this is simply the first noticeable instance that a user will notice.

# 

# **3. Features To Be Tested**

* Introduce a new change to the game
  + We decided to change the font of the panel headers to BeachSocietyLight. This change can be immediately noticed when starting a new game. The panel header that reads "Setup new game" will be changed.

# **4. Features Not To Be Tested**

* All previously implemented tests were tested both while refactoring and while implementing the quality of life update. We did this in order to ensure that our changes did not negatively impact the already existing system in any way.

# **5. Approach**

* This is a reiteration of the above “Testing Strategy” Section. In this section, however, we specify the results (final average cyclomatic complexity) for each of the four packages.
* Use CodePro to detect cyclomatic complexity for each package.
* Determine packages that have a cyclomatic complexity above 50. This number is the highest maximum from each package, not the average.
  + We decided to improve the following packages
    - Net.sf.freecol.server.model
      * Final average complexity: 4.37
    - Net.sf.freecol.server.ai.mission
      * Final average complexity: 2.64
    - Net.sf.freecol.server.ai
      * Final average complexity: 3.23
    - Net.sf.freecol.common.networking
      * Final average complexity: 1.85
  + Not only did these packages have a cyclomatic complexity above 50, they also contained more classes than other packages that had high complexities.
* Refactor the chosen packages to reduce the average cyclomatic complexity as much as possible without changing or defecting any previously implemented game logic.
  + In order to ensure that our refactorings do not deter the game from running as intended, we run the “ant testall” command every time we finish refactoring an individual class. If all 430 tests still pass, then we move on to the next class. If not, we remove our refactorings and try again.
* Generate JavaDocs for the specified packages.
* Introduce a new change to the game
  + We decided to change the font of the panel headers to BeachSocietyLight. This change can be immediately noticed when starting a new game. The panel header that reads "Setup new game" will be changed.
* Perform a regression test after changing the font.
  + We did not need to introduce any new tests for this simple quality of life improvement. Rather, we ensured that all existing tests still passed after changing the font. To do this, we simply run “ant testall”. If all 430 tests still pass, then we determine that our quality of life update was successfully integrated within the game. If not, we examine the cause of failure and reimplement our change.

**5.1 Component Testing**

* Because this update does not introduce new program modules, individual component testing was not conducted in this update

**5.2 Integration Testing**

* + In order to ensure that our refactorings do not deter the game from running as intended, we run the “ant testall” command every time we finish refactoring an individual class. If all 430 tests still pass, then we move on to the next class. If not, we remove our refactorings and try again.
  + We ensured that all existing tests still passed after changing the font. To do this, we simply run “ant testall”. If all 430 tests still pass, then we determine that our quality of life update was successfully integrated within the game. If not, we examine the cause of failure and reimplement our change.

**5.3 Interface Testing**

* Interface Testing is not covered by our update.

**5.4 Security Testing**

* Security Testing is not covered by our update.

**5.5 Performance Testing**

* Security Testing is not covered by our update.

**5.6 Regression Testing**

* We changed the font of the panel headers within the game. In order to ensure that our changes did not adversely affect previously tested functionality, we ran all 430 tests that were provided with the software. All of these tests still pass after changing the panel header font.

**5.7 Acceptance Testing**

* Acceptance testing is not covered by our update.

**5.8 Beta Testing**

* Beta testing is not covered by our update.

# **6. Pass / Fail Criteria**

* In order to run all 430 tests, run “ant testall” from the terminal within the appropriate directory.
* Refactoring Pass / Fail Criteria
  + If all 430 tests still pass, then we move on to the next class. If not, we remove our refactorings and try again.
* Quality of Life Update (Font Change) Pass / Fail Criteria
  + If all 430 tests still pass, then we determine that our quality of life update was successfully integrated within the game. If not, we examine the cause of failure and reimplement our change.

**6.1 Suspension Criteria**

* None of the provided tests needed to be seperated or suspended during the implementation of this update.

**6.2 Resumption Criteria**

* Resumption Criteria is not applicable to this update.

**6.3 Approval Criteria**

* New tests were not added with this update.

# 

# 7. **Testing Process**

(Identify the methods and criteria used in performing test activities. Define the specific methods and procedures for each type of test. Define the detailed criteria for evaluating test results.)

**7.1 Test Deliverables**

* All metric reports are located in a directory within the newest commit on Github
* Every test check (running all 430 provided tests to ensure green bar) is not provided in the form of a report. We simply ran these in the terminal. The tests are already provided in the FreeCol project and do not require any extra download.
* Test input: In order to run all 430 tests, run “ant testall” from the terminal within the appropriate directory.
* Test output: The number of passed and failed tests.

**7.2 Testing Tasks**

* All tests can be run immediately after downloading(cloning) the directory.
  + The only requirement to run the tests is to have Ant installed on your device.
    - Ant setup: <https://stackoverflow.com/questions/3222804/how-can-i-install-apache-ant-on-mac-os-x?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa>
  + To run tests, run “ant testall” from the terminal within the appropriate directory.

**7.3 Responsibilities**

* **Nicholas Visalli** 
  + Refactor specified packages and generate JavaDocs (Package names listed above)
  + Manage issue logs on Github
  + Run tests, using Ant
* **Amro Al-Suwaida**
  + Run metric reports (before and after refactoring of packages)
  + Determine packages to refactor
  + Change the font of panel headers with in the game
    - New font: BeachSocietyLight

**7.4 Resources**

* The only requirement to run the tests is to have Ant installed on your device.
  + Ant setup: <https://stackoverflow.com/questions/3222804/how-can-i-install-apache-ant-on-mac-os-x?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa>
  + To run tests, run “ant testall” from the terminal within the appropriate directory.

**7.5 Schedule**

* Generating a metrics report detailing packages that need improvement.
* Refactoring and improving the average cyclomatic complexity of those packages.
  + Ensuring that all provided tests still pass after implementation.
* Generate JavaDocs for specified packages
* Implementing quality of life change.
  + Changing the font of the panel headers in the game to BeachSocietyLight.
  + Ensuring that all provided tests still pass after implementation.
* Completing the STP document.

# 

# **8. Environmental Requirements**

**8.1 Hardware**

* We implemented this update using Macbook Pro laptops

**8.2 Software**

* We implemented this update with Eclipse Oxygen.

**8.3 Security**

(Identify the testing environment security and asset protection requirements.)

* We used GitHub as our remote repository.

**8.4 Tools**

We are going to use multiple eclipse plug-in such as:

* Junit.
* Spots Bugs.
* JavaDoc.
* CheckStyle.
* Code Coverage.
* PIT Mutation test. \*\*
* Bad Smells.
* CodePro:
  + Audit Code.
  + Compute Metrics.
  + Generate Test Cases.
* AutoRefactor.

**8.5 Risks and Assumptions**

* Approach reevaluation
  + We initially wanted to add more unit tests, providing better coverage for the specified packages.
    - Due to time constraints, we decided it was practical to change our plan to improve the software.

# 

# **9. Change Management Procedures**

* Items from our original test plan that did not make the final implementation of the update
  + Addition of new unit tests
  + Changing the icon of the boat that carries a user’s units

# 