TDDC88 Lab 5

Software Metrics and Quality Factors

Introduction

In this lab you will work with the open source java project FreeCol. FreeCol is a turn-based strategy game based on the old game *Colonization*, and similar to *Civilization*. During this lab you will gain insight into the topic of software metrics and how they can be used to estimate software quality factors of arbitrary code. You will have to work with a fairly large code base, but don't let this intimidate you. The purpose of this lab is not to understand the code in detail, even if a broad understanding of the code structure will be helpful. The purpose is to find out to what extent we can evaluate code which might not necessarily be known beforehand.

For the first part of the lab you will look at specific parts of the code. The aim here is to get acquainted with how software measurements are taken and how they can be used to get software metrics.

For the second part of the lab you will apply what you learned from the first part of the lab and perform a study on the whole code base. You will choose one software quality factor to study and find a software metric relating to the chosen quality factor. In order to perform measurements, so that you can calculate your metrics, you will use an Eclipse plug-in called Metrics.

Start-up and installation

To download FreeCol you first open a terminal and change directory into your Eclipse workspace. You shall now clone a Gitlab project containing the game using this command:

```
git clone http://gitlab.ida.liu.se/tddb84/freecol.git
```
Then create a new Java project in Eclipse and enter the name of the folder you just cloned and press finish, if you have not changed the folder's name it should be called freecol.

You should now be able to see the project in the package explorer. Installation of the Metrics plug-in will be explained later on. The plug-in is not required for the first part of the lab.
Part A – *Introduction to Software Metrics*

The purpose of this first part is to get acquainted with metrics and you will thus be performing the measurements and calculating the metrics by hand. The relation between the software metrics and software quality factors are of interest for part B, so keep this in mind while working.

**Tasks**

You are to take measurements of a certain part of the code base and calculate some different software metrics.

- Make sure that you have followed the earlier steps of how to download the FreeCol code base and make it available as a project in Eclipse.
- Find and open the following classes
  - `Tile` in `src → net.sf.freecol.common.model → Tile`
  - `AIPlayer` in `src → net.freecol.server.ai → AIPlayer`
  - `RandomChoice` in `src → net.sf.freecol.common.util → RandomChoice`
- Choose at least one method that has at least 10 lines of code from each class and calculate the different metrics *cyclomatic complexity*, *number of lines of code* and *hierarchical depth*. The hierarchical depth is the number of superclasses that the class has.

**Tips**

- If you are unsure of how to calculate the cyclomatic complexity look up the lecture on software metrics and chapter 11.4 in the course book\(^1\) where it is called cyclomatic number.
- The hierarchical depth can be found by pressing f4 when having focus in target class in the eclipse code window and observing the tree structure that shows up.

**Examination**

When you have completed all the steps you present your results to the lab assistant. Be prepared to answer questions about how you calculated the metrics.

---

\(^1\) Pfleeger, S. L. and Atlee, J. M. *Software Engineering - Theory and Practice*, 4th edition
Part B – Software metrics and FreeCol

In this part of the lab you will perform and analyze measurements on the whole FreeCol code base, instead of just looking at a few classes and methods. To be able to perform the measurements you will use the Metrics Eclipse plug-in. The goal of this lab is to introduce how metrics can be used to get an estimation of the software quality factors of a software product.

Eclipse metrics plug-in

In order to install Metrics first go to Help → Install New Updates... Then press add to add the update site http://metrics.sourceforge.net/update with a name of your choice. If should look like the picture below.

Choose the Metrics plug-in and press next and go through the installation process. After installation has been completed you will need to restart Eclipse. If the installation was successful you should be
able to launch the plug-in by going through Window → Show View → Other... and in the pop up window choose Metrics → Metrics View.

In the bottom part of the screen you will see the metrics view together with instructions for calculating the metrics. Right click on the FreeCol project in the package explorer and then go to Properties. Then you to Metrics and check Enable Metrics.

After this the project will perform a full rebuild. When this is completed you should be able to see a list of metrics. In order to just get the metrics for the source and not the tests, select the src folder in the package explorer.
If you need more information about installation and usage of the Metrics Eclipse plug-in it can be found at the following link: [http://metrics.sourceforge.net/](http://metrics.sourceforge.net/).

**Task**

Your task is to perform a study of software quality factors of the open source project FreeCol, and from that write a short report (about 2 pages). The study should be done as follows:

- Select **ONE** software quality factor of your choice from the following list as a basis for your study.
  - Changeability
  - Testability
  - Understandability

- Decide and describe **ONE** software metric you will use to evaluate the quality factor and which measurements you will need to perform on the code base to get the metric. You need to describe why your chosen metric will give you a good estimation of the quality factor, and also how the metric is calculated.

- Perform measurements on FreeCol using the Metrics plug-in and find locations in the code where your metric has especially high or low values.

Using your study and based on the extreme points you found, you will then write a discussion where you discuss the following questions:

1. What is causing the metric to have a particularly high/low value in the code locations indicated by your extreme points? Are the areas of very high or very low software quality or are the values caused by something else?

2. Are you able to draw any conclusions regarding software quality factors in FreeCol? Why/why not?
3. Do you know of any other metrics handling the same software quality factor as your chosen metric? How are they different from the metric you used? Do you think other metrics are required in order to get a good estimation of the software quality factor of the product?

4. Do you have any suggestions for improvement in the locations indicated by your study? This could be either on a method or class level or a larger structural change.

5. Do you think your study gave you a better understanding of the code base? Why/why not?

**Examination**

Write a short report (about 2 pages) containing a description of the study you have performed, as well as a discussion answering the questions. The description of the lab shall be in sufficient detail for a fellow student to replicate the study, and the discussion shall be supported by references if needed. Send in your report as a PDF-file or plain text to your lab assistant using his/her urkund-address which is stated on the lab page.