

# Group Report

## Unit Test Visualisation

**Team:** Group 4

**Course:** ELEN7046 - Software Technologies and Techniques

**Date Submitted:** 25 June 2012

**Source & Documentation:**

<https://github.com/KeaganPhillips/Wit-Group-4-project>

### Abstract

The design and implementation of a software visualisation tool for unit testing is presented. The visualisation tool is to be used to graphically display, enhance and ... unit tests. The development of both the unit tests and its visualisation are discussed. The tool was implemented using TDD and uses C# Reflection to obtain the necessary information which was then sent to the visualiser to display. The implementation suggests that the tool can play an important role in software visualisation education.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>License Agreement</b>	<b>4</b>
2.1	Overview . . . . .	4
2.2	Components used and their license agreements . . . . .	4
<b>3</b>	<b>Problem Description</b>	<b>4</b>
<b>4</b>	<b>Approach</b>	<b>4</b>
<b>5</b>	<b>Installation Procedure</b>	<b>4</b>

# 1 Introduction

This paper presents a report of an application that links software visualisation and software education. In the literature, it is stipulated that software visualization encompasses the development and evaluation of methods for graphically representing different aspects of software, including its structure, its execution, and its evolution.

The aspect of software that we present in this report is Unit Testing[1]. The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as expected. Each unit is tested separately before integrating them into modules to test the interfaces between modules. This testing mode is a component of Extreme Programming (XP)[2], a pragmatic method of software development that takes a precise approach to building a product by means of continual testing and revision.

Unit testing involves only those characteristics that are vital to the performance of the unit under test. This encourages developers to modify the source code without immediate concerns about how such changes might affect the functioning of other units or the program as a whole. Once all of the units in a program have been found to be working in the most efficient and error-free manner possible, larger components of the program can be evaluated by means of integration testing.

In this report, we primarily demonstrate an application that visualises all the unit tests in a given program. The application discussed in this report visualises not only the unit tests, but also the class diagrams together with their public methods and properties.

The remainder of this report is as follows. Section 2 discusses the licenses used in the development of the application. We provide the problem description in section 3, and discuss the approach we took to solve the problem in section 4. Installation process for the application is discussed in section 5. We provide a discussion on how the approach to the solution was implemented in section 6. In section 7, we discuss an analysis of our solution, and provide future works in section 8. We end the report with a conclusion

## 2 License Agreement

### 2.1 Overview

An analysis of software licenses was done considering the combination of both original work and third party components. The new BSD license also known as the 3-Clause BSD[3] license was chosen. The software and source code are made available under this license. Use, modification and redistribution is not restricted.

### 2.2 Components used and their license agreements

The following lists the components used within the system together with their respective license agreements.

- **Coffee Script:** MIT license[5]
- **Java script:** GNU GPL[6]
- **JQuery:** GNU GPL[6]
- **Kinetic.js:** GNU GPL[6]
- Boreli's report?

For the complete reference, please refer to the document 'ELEN7046 - Group4 - Licence Agreement.pdf'[4]

## 3 Problem Description

blah.. blah...

## 4 Approach

blah.. blah...

## 5 Installation Procedure

blah.. blah...

## References

- [1] <http://www.extremeprogramming.org/rules/unittests.html>
- [2] <http://www.extremeprogramming.org/>
- [3] <http://www.opensource.org/licenses/BSD-3-Clause>
- [4] License Agreement <https://github.com/KeaganPhillips/Wit-Group-4-project/tree/master/Documentation/>
- [5] <http://www.opensource.org/licenses/mit-license.html>
- [6] <http://www.gnu.org/copyleft/gpl.html>
- [7] b