**Test plan**

**Test plan identifier**

version 1.06 unit test plan, this plan's’ purpose is to identify the requirements and analyse the features that will be tested.

Authors: Jason Pearson, John Young

**Introduction**

The objective of the test plan is to outline how and what will be tested for the program that is being designed. First an interpretation of the requirements will be done then test plans will be done for the project. After this the coding will be started, then error handling will be done. Afterwards testing methods will be put in the code. Finally the testing methods will be documented and the project will then be signed off. A resource constraint is that we have a limited amount of allocated time.

**Test items**

**Driver 1.0**

**Map 1.0**

**Location 1.0**

**Street 1.0**

**Random Passer 1.0**

**Features to be tested**

* Insuring the driver’s position changes H
* Command line only takes one integer value M
* Driver leaves at correct exit H
* One driver runs at a time H
* If drivers exits make sure location changes to outside. M
* If chosen to stay driver doesn’t exits the city H
* Driver is able to go forward or back to its previous location. H
* Insure random seed range is within range. H
* Make sure five dashes are printed after each iteration. L

The reason why some features were given High level is because if they are not running as expected it will cause the whole program to not run as intended and likely crash. The reason why some features were given medium level risk is because they will likely not lead to a crash at first but will probably later while the program runs. The reason why some features were given low risk level because they will likely not cause any major impact on how the program runs or on other features that are running.

**Features not to be tested**

* Random Number generator does provide the same number with the same seed.

**Non functional**

* It may be easier to understand the map of the city visually
* A driver may start in any of the four locations listed in FUN-City-LOCS.

**Outliers**

* Spelling error there

* Drivers may not start outside of the city
* While random number isn’t tested it will be hard to test methods surrounding it
* The simulation shall end if a driver encounters the outside city location.(but five drivers have to go)
* Five drivers, numbered 1 through 5, shall traverse the city one after the other.(doesn’t say if five drivers are going at the same time or waiting until finished.)
* No other arguments shall be accepted. If not arguments are provided, more than one argument is provided, or the single argument is not an integer, the system shall inform the user and exit.(the system is accepting more than one requirement.)

**Approach**

* The testing framework Mockito is being used because of that it allows developers to verify the behaviour of a system under test without establishing expectations beforehand. Another reason is Mockito has good void method handling since the product might have a lot of void methods this framework Mockito is going to the better choice of testing framework.
* The testing framework will require some special training due to us developers have not used this testing framework before.
* The test approach that will be taken is proactive technique where the test design process is initiated early to find and fix defects before a build is created.

**Item Pass/Fail Criteria**

* For an item to pass it needs to run as expected and to be repeatable with the same input values.
* For an item to fail either it doesn't run as expected or crashes the program.

**Suspension Criteria and Resumption Requirements**

* hardware or software are not available at the time indicated in the project schedule.
* the build contains many serious defects which seriously prevent or limit testing progress.
* assigned test resources are not available when needed by the test team.

**Test Deliverables**

* Test Plan
* Test Result Report
* Test summary report
* Defect Report
* Testing Strategy
* Test Case Documents

**Testing Tasks**

* Planning what test scenarios are needs
* Creating test cases
* Creating testing methods
* Executing testing methods
* Reporting buds
* Documenting Results

**Environment needs**

* Hardware requirements that are needed are to run NetBeans are only a processor 800 MHz and memory 512Mb and around 750 MB of free disk space. No single version of java SE Development Kit is needed but JDK 8u60 is recommended for NetBeans 8.1 JDK 6.0 doesn’t work with NetBeans 8.1

**Staffing and training needs**

Even though both members are familiar with java, both members are not familiar with JUnit so this will require some training to learn how to properly use the testing framework.

**Responsibilities**

Both members are in charge of determining risks that are involved with the project.

What features will be tested and what features that are not being tested will be determined before testing begins.

The training for JUnit will be provided by internet sources like websites and videos.

Time management will have to be track off, to avoid being behind schedule.

**Schedule**

**interpretation of requirements 15/04/17 - 29/04/17**

**Test plan 30/04/17 - 26/04/17**

**Program coding 27/04/17 - 18/05/17**

**Error handling 28/04/17 - 22/05/17**

**Testing Methods 23/05/17 - 29/05/17**

**Documentation of Testing 24/05/17 - 6/06/17**

**Signed Off 7/06/17 - 8/06/17**

**Risks and contingencies**

Some risks that could happen are having incompatible hardware or software that is required to avoid this make sure the right version of NetBeans and java JDK are installed.

Another risk is falling behind schedule to help minimise the risks a todolist should be created, to make it easy to determine what needs to be completed.

Another risk is that training for JUnit might take longer than expected and cause delays in testing methods to minimize the risk is to do check up with team members early to determine if they understand the testing framework.

One other risk is that either the requirements or design was misunderstood or was changed throughout the project timeline. To minimise this risk interpretation of the risks and communication between the customers can be done.

**Approval**

**Stakeholder:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    Signed:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**