

**Escape The Island**

**Master Test Plan**

**for Escape The Island  
2D point-and-click-adventure**

**Version 1.3**

Escape The Island	Version: 1.3
Master Test Plan	Date: 19/05/15

## Revision History

Date	Version	Description	Author
01/05/15	1.0	Described first chapters	Alisa Koksjuk
13/05/15	1.1	Described next chapters and updated first chapters	Alisa Koksjuk
14/05/15	1.2	Finished all as far as possible, filled in 'tbd' in some cases	Alisa Koksjuk
19/05/15	1.3	Formated and convert to PDF	Alisa Koksjuk

Escape The Island	Version: 1.3
Master Test Plan	Date: 19/05/15

# Table of Contents

## [Introduction](#)

### [Purpose](#)

### [Scope](#)

### [Intended Audience](#)

## [Evaluation Mission and Test Motivation](#)

### [Evaluation Mission](#)

### [Test Motivators](#)

## [Target Test Items](#)

## [Outline of Planned Tests](#)

### [Outline of Test Inclusions](#)

### [Outline of Test Exclusions](#)

## [Test Approach](#)

### [Testing Techniques and Types](#)

#### [Function Testing](#)

#### [User Interface Testing](#)

#### [Configuration Testing](#)

## [Deliverables](#)

### [Test Evaluation Summaries](#)

### [Reporting on Test Coverage](#)

### [Incident Logs and Change Requests](#)

## [Testing Workflow](#)

## [Environmental Needs](#)

### [Base System Hardware](#)

### [Base Software Elements in the Test Environment](#)

### [Productivity and Support Tools](#)

## [Responsibilities, Staffing, and Training Needs](#)

### [People and Roles](#)

### [Staffing and Training Needs](#)

## [Iteration Milestones](#)

## [Risks, Dependencies, Assumptions, and Constraints](#)

## [Management Process and Procedures](#)

### [Measuring and Assessing the Extent of Testing](#)

### [Assessing the Deliverables of this Test Plan](#)

Escape The Island	Version: 1.3
Master Test Plan	Date: 19/05/15

# Iteration Testplan

## 1. Introduction

### 1.1 Purpose

The purpose of the Master Test Plan is to gather all of the information necessary to plan and control the test effort for a given iteration. It describes the approach to testing the software, and is the top-level plan generated and used by managers to direct the test effort.

This Test Plan for the project Escape-The-Island supports the following objectives:

- The items that should be targeted by the tests.
- The motivation for and ideas behind the test areas to be covered.
- Outlines of the testing approach that will be used.
- The required resources and provides an estimate of the test efforts.
- The deliverable elements of the test project.

### 1.2 Scope

We defined three different levels of testing. The first one is Testing on the surface with which we test Functionality in the form of Scenarios which user is able to create.

The second Testing is a Unit testing and concerns different types of testing like Functionality of single Code Classes, Reliability and Performance.

The third Testing include Parts with different weighting of importance. These parts are View, Functionality, Usability, Reliability, Performance and Support in some cases.

The size of our project and the amount of time restrict our testing potential. For that reasons we are not testing explicitly Performance, Database and Configuration.

### 1.3 Intended Audience

Our Audience for whom we are writing the Test Plan are the people who are interested in our work and our product.

## 2. Evaluation Mission and Test Motivation

Our Mission is to provide a high quality click adventure and therefore we are very motivated to make everything work and make our costumes and game users happy.

### 2.1 Evaluation Mission

Mission contains several points which are imported for us. The following description gives the nail on the head.

- find as many bugs as possible
- find important problems, assess perceived quality risks
- advise about perceived project risks
- verify a specification (requirements, design or claims)
- to provide a funny game which is also children welcome
-

Escape The Island	Version: 1.3
Master Test Plan	Date: 19/05/15

## 2.2 Test Motivators

According to our Mission we are very motivated to test our work, because we want to provide a high quality software and a funny game to make our customer and our users happy.

To be in time with our project we want to avoid every risks which can influence our working hours and lead to the delay of our final state.

Our motivators are:

- provide a functionality with less bugs as possible
- provide a funny and welcome children game

## 3. Target Test Items

The listing below identifies those test items \_software, hardware, and supporting product elements \_that have been identified as targets for testing. This list represents what items will be tested.

*tbd*

## 4. Outline of Planned Tests

This section provides an outline of the testing that will be performed. The outline in this section represents an overview of both the tests that will be performed and those that will not.

### 4.1 Outline of Test Inclusions

The tests that will be performed are Cucumber testing, JUnit testing and Human factors testing. The last one includes in some cases more or less following tests:

- Quality testing
- Configuration testing
- Compatibility test (Internet Browser needed)
- Timing testing

### 4.2 Outline of Test Exclusions

This tests will not be performed:

- Security testing
- Environmental test
- Stress Testing
- Volume testing
- Recovery testing

## 5. Test Approach

We defined three different levels of testing as seen in chapter 1.2. The first one is Testing on the surface with which we test Functionality in the form of Scenarios which user is able to create. Everything that concerns the View except the game is tested with Cucumber.

The second Testing is a Unit testing and concerns different types of testing like Functionality of single Code Classes, Reliability and Performance. This Testing is realized with JUnit.

The third Testing include Parts with different weighting of importance. These parts are View, Functionality, Usability, Reliability, Performance and Support in some cases. This is an Useability test and will be executed by several testers on the Escape-The-Island game.

Escape The Island	Version: 1.3
Master Test Plan	Date: 19/05/15

The size of our project and the amount of time restrict our testing potential. For that reasons we are not testing explicitly Performance, Database and Configuration.

## 5.1 Testing Techniques and Types

### 5.1.1 Function Testing

Function testing of the target-of-test are focused on any requirements for test that can be traced directly to use cases or business functions and business rules. The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. This type of testing is based upon black box techniques; that is verifying the application and its internal processes by interacting with the application via the Graphical User Interface (GUI) and analyzing the output or results.

The following table identifies an outline of the testing recommended for each application. These Testing is realized with Cucumber, therefore it needs feature-files and step definition -files.

Technique Objective:	From Use Cases defined Narratives will be implemented in Gherkin as feature-files and related step-definitions execute the feature-files via running cucumber.
Technique:	<p>Cucumber execute each use-case scenario's individual use-case flows or, functions and features, using valid and invalid data, to verify that:</p> <ul style="list-style-type: none"> <li>the expected results occur when valid data is used</li> <li>the appropriate error or warning messages are displayed when invalid data is used</li> <li>each business rule is properly applied</li> </ul>
Required Tools:	<p>The technique requires the following tools:</p> <ul style="list-style-type: none"> <li>ruby</li> <li>cucumber</li> <li>Java IDE (IntelliJ IDEA 13.1.6)</li> <li>GITHUB</li> </ul>
Success Criteria:	<p>The technique supports the testing of:</p> <ul style="list-style-type: none"> <li>collect Items</li> <li>combine Items</li> <li>get statistics</li> <li>interact with objects</li> <li>interact with NPCs</li> <li>save the game</li> <li>register, log in</li> <li>edit profile</li> <li>using items on objects</li> <li>walk around the island</li> </ul>

### 5.1.2 User Interface Testing

User Interface (UI) testing verifies a user's interaction with the software. The goal of UI testing is to ensure that the UI provides the user with the appropriate access and navigation through the functions of the target-of-test. In addition, UI testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.

Escape The Island	Version: 1.3
Master Test Plan	Date: 19/05/15

Our User Interface Testing is a part of the Usability Testing. Therefore there are no especial deep testing in several parts of the UI.

Technique Objective:	Several people act as user and test the game in their whole functionality until the end of the game and showing the statistics web page.
Technique:	The evaluation of tests are made by a survey 'How did you enjoyed the Escape-The-Island Click Adventure'.
Required Tools:	The technique requires an Internet Browser and a Google survey.
Success Criteria:	All issues are in the area from 'excellent' until 'good'

### 5.1.3 Configuration Testing

We decided not to make explicit configuration test, because our project needs no explicit hardware. Nevertheless we had to test our web-based click adventure on several browsers. This is described in form of a table in chapter 9.2.

## 6. Deliverables

In this section are the various artifacts that will be created by the test effort that are useful deliverables to the various stakeholders of the test effort. There are listed only those that give direct, tangible benefit to a stakeholder and those by which we want the success of the test effort to be measured.

### 6.1 Test Evaluation Summaries

The test evaluation summaries are realized in form of an internet survey. The content of the tests evaluation summaries is divided in three parts:

- JUnit Testing
  - Percentage of errors
- Cucumber Testing
  - Percentage of errors
- Usability Testing
  - Was the game interesting?
  - Is the game children welcome?
  - Is the game easy to play?
  - Does the game has a good performance?
  - tbd

JUnit tests and Cucumber tests will be tested every day. Usability will be tested in a survey once a week. Then all JUnit and Cucumber tests were successful there will be a big survey which contains the summary of all Tests.

### 6.2 Reporting on Test Coverage

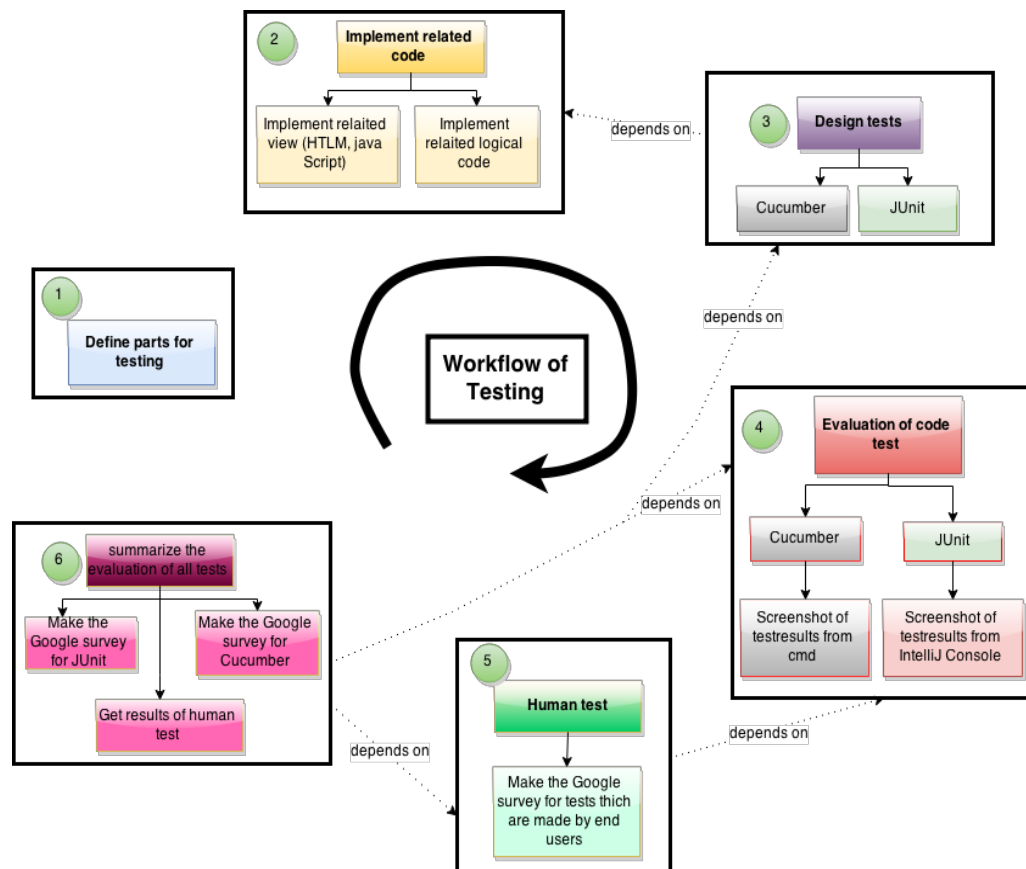
The following Screenshot from Jenkins (where JUnit and Cucumber tests are running by building) shows the test Coverage.

### 6.3 Incident Logs and Change Requests

Test	Test Area	Screen shot
Cucumber	tbd	tbd
JUnit	tbd	tbd

## 7. Testing Workflow

The following graphic shows the brief outline of our testing Workflow. The several single test which are not mentioned here in that necessary detail are explained in related chapters above.





Escape The Island	Version: 1.3
Master Test Plan	Date: 19/05/15

## 8. Environmental Needs

### 8.1 Base System Hardware

The following table sets forth the system resources for the test effort presented in this Test Plan.

System Resources		
Resource	Quantity	Name and Type
Database Server		
Server Name	1	TBD
Database Name	1	TBD
Client Test PCs	5	Lenovo, Win 7; Samsung Win 7, Fujitsu Win 7; ALIENWARE, Win 8.1;
Test Repository	1	Branch of Test Master
Network or Subnet		TBD
Server Name		TBD
Test Development PCs		ALIENWARE, Win 8.1

### 8.2 Base Software Elements in the Test Environment

The following base software elements are required in the test environment for this Test Plan.

Tested Browser	Version	Type and Other Notes
Firefox	tbd	Internet Browser
Chrome	tbd	Internet Browser
Internet Explorer	tbd	Internet Browser
Firefox Developer Edition	tbd	Internet Browser
Opera	tbd	Internet Browser

### 8.3 Productivity and Support Tools

The following tools will be employed to support the test process for this Test Plan.

Tool Category or Type	Tool Brand Name	Vendor or In-house	Version
Build server	Jenkins	tbd	tbd
Project Management	Open Project Jira	OpenProject Foundatin e.V. (OPF)	tbd
DBMS tools	tbd	tbd	tbd

Escape The Island	Version: 1.3
Master Test Plan	Date: 19/05/15

## 9. Responsibilities, Staffing, and Training Needs

This section presents the required resources to address the test effort outlined in the Test Plan including the main responsibilities and the knowledge or skill sets required of those resources.

### 9.1 People and Roles

The followed table shows the staffing assumptions for the test effort. Because our Team has only three members, there is only one person who is responsible for testing. This person has the roles of Test Manager, Test Analyst and Test Designer and Implementer (green mark). The role of Tester is divided more or less among the team members (marked purple). Everything that belongs to the database is the work of another team member (marked blue). We decided not to test our database explicitly.

Human Resources	
Role	Specific Responsibilities or Comments
Test Manager	Provides management oversight. Responsibilities include: <ul style="list-style-type: none"> <li>• planning and logistics</li> <li>• agree mission</li> <li>• identify motivators</li> <li>• acquire appropriate resources</li> <li>• present management reporting</li> <li>• advocate the interests of test</li> <li>• evaluate effectiveness of test effort</li> </ul>
Test Analyst	Identifies and defines the specific tests to be conducted. Responsibilities include: <ul style="list-style-type: none"> <li>• identify test ideas</li> <li>• define test details</li> <li>• determine test results</li> <li>• document change requests</li> <li>• evaluate product quality</li> </ul>
Test Designer	Defines the technical approach to the implementation of the test effort. Responsibilities include: <ul style="list-style-type: none"> <li>• define test approach</li> <li>• define test automation architecture</li> <li>• verify test techniques</li> <li>• define testability elements</li> <li>• structure test implementation</li> </ul>
Tester	Implements and executes the tests. Responsibilities include: <ul style="list-style-type: none"> <li>• implement tests and test suites</li> <li>• execute test suites</li> <li>• log results</li> <li>• analyze and recover from test failures</li> <li>• document incidents</li> </ul>

Escape The Island	Version: 1.3
Master Test Plan	Date: 19/05/15

Test System Administrator	<p>Ensures test environment and assets are managed and maintained.</p> <p>Responsibilities include:</p> <ul style="list-style-type: none"> <li>• administer test management system</li> <li>• install and support access to, and recovery of, test environment configurations and test labs</li> </ul>
Database Administrator, Database Manager	<p>Ensures test data (database) environment and assets are managed and maintained.</p> <p>Responsibilities include:</p> <ul style="list-style-type: none"> <li>• support the administration of test data and test beds (database).</li> </ul>
Designer	<p>Identifies and defines the operations, attributes, and associations of the test classes.</p> <p>Responsibilities include:</p> <ul style="list-style-type: none"> <li>• defines the test classes required to support testability requirements as defined by the test team</li> </ul>
Implementer	<p>Implements and unit tests the test classes and test packages.</p> <p>Responsibilities include:</p> <ul style="list-style-type: none"> <li>• creates the test components required to support testability requirements as defined by the designer</li> </ul>

## 9.2 Staffing and Training Needs

The person who is responsible for Cucumber and JUnit tests and the person who is responsible for builds tests are well trained. The tester for the Usability tests do not need a special training of qualification.

## 10. Iteration Milestones

The following table shows the milestones of Testing effort on build server Jenkins.

Milestone	Planned Start Date	Actual Start Date	Planned End Date	Actual End Date
Iteration Plan agreed	tbd	tbd	tbd	tbd
Iteration starts	tbd	tbd	tbd	tbd
Requirements baselined	tbd	tbd	tbd	tbd
Architecture baselined	tbd	tbd	tbd	tbd
User Interface baselined	tbd	tbd	tbd	tbd
First Build delivered to test	tbd	tbd	tbd	tbd
First Build accepted into test	tbd	tbd	tbd	tbd
First Build test cycle finishes	tbd	tbd	tbd	tbd
[Build Two will not be tested]	tbd	tbd	tbd	tbd
Third Build delivered to test	tbd	tbd	tbd	tbd
Third Build accepted into test	tbd	tbd	tbd	tbd
Third Build test cycle finishes	tbd	tbd	tbd	tbd

Escape The Island	Version: 1.3
Master Test Plan	Date: 19/05/15

Fourth Build delivered to test	tbd	tbd	tbd	tbd
Fourth Build accepted into test	tbd	tbd	tbd	tbd
Iteration Assessment review	tbd	tbd	tbd	tbd
Iteration ends	tbd	tbd	tbd	tbd

## 11. Risks, Dependencies, Assumptions, and Constraints

Risk	Mitigation Strategy	Contingency (Risk is realized)
Prerequisite entry criteria is not met.	<p>&lt;Tester&gt; will define the prerequisites that must be met before Load Testing can start.</p> <p>&lt;Customer&gt; will endeavor to meet prerequisites indicated by &lt;Tester&gt;.</p>	<ul style="list-style-type: none"> <li>Meet outstanding prerequisites</li> <li>Consider Load Test Failure</li> </ul>
Test data proves to be inadequate.	<p>&lt;Customer&gt; will ensure a full set of suitable and protected test data is available.</p> <p>&lt;Tester&gt; will indicate what is required and will verify the suitability of test data.</p>	<ul style="list-style-type: none"> <li>Redefine test data</li> <li>Review Test Plan and modify components (that is, scripts)</li> <li>Consider Load Test Failure</li> </ul>
Database requires refresh.	<System Admin> will endeavor to ensure the Database is regularly refreshed as required by <Tester>.	<ul style="list-style-type: none"> <li>Restore data and restart</li> <li>Clear Database</li> </ul>
Failed Tests which are retained directly to the code can hinder the developing workload.	<Tester> should test only on his branch.	<ul style="list-style-type: none"> <li>Last commits should be able to be reconstructable</li> </ul>

Dependency between	Potential Impact of Dependency	Owners
Logical Code ↔ JUnit Tests	Deadlines for tests	Maik Wandrai ↔ Alisa Koksjuk
View ↔ Cucumber Tests	Deadlines for tests	Norman Naujokat ↔ Alisa Koksjuk

Our important assumptions (related to testing) which should be proven are tightly connected with dependencies listed above.

Assumption to be proven	Impact of Assumption being incorrect	Owners
Logical Code (Classes should be well structured, e.g. Clean Code Development)	Test Master a give a specified plan to owners which include how to improve their work	Mak Wandrai
View (should be well structured e.g. HTML with unique ids...)	Test Master a give a specified plan to owners which include how to improve their work	Norman Naujokat

## 12. Management Process and Procedures

### 12.1 Measuring and Assessing the Extent of Testing

The summary of all test evaluations will be shown in a special graphic generated by a survey.

Escape The Island	Version: 1.3
Master Test Plan	Date: 19/05/15

## 12.2 **Assessing the Deliverables of this Test Plan**

As already mentioned The size of our project and the amount of time restrict our testing potential. Therefore this Test Plan has the important and necessary tests and sufficient completely the defined requirements.