TextVenturer
Test Plan

Version <1.0>

Revision History

Date	Version	Description	Author
<05/May/17>	<1.0>	<First Add>	Simon Vollmer, André Schmitt, Dominik Vogel

Table of Contents

<Iteration/ Master> Test Plan

1. Introduction

1.1 Purpose

The purpose of the Iteration 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 TextVenturer supports the following objectives:

- View
- Adventure Structure
- Game progress
    - Default Actions
    - Custom Actions

1.2 Scope

Unit tests with `CppUnitTestFramework` on Visual Studio. Testing all important things, that need testing.
We are testing the important game routines with a self-made input simulation tool. For some special cases, we have to test it manually.

1.3 Intended Audience

We are writing this Test Plan for ourselves, to prevent us from making any major mistakes and also everybody who is interested in our testing.

1.4 Document Terminology and Acronyms

- **tbd** → to be defined
- **n/a** → not available
- **txv** → TextVenturer

1.5 References

n/a

2. Evaluation Mission and Test Motivation

We are testing our project, so we don't get any funny bugs in it. It also eases the implementation of new features into the game, as it makes it less likely for bugs to occur. If we make big changes in fragile parts of our code it ensures, that everything will still be working correctly afterwards.

2.1 Background

We are testing our program, so that the user will have a bug-free experience of our game.

2.2 Evaluation Mission

As one of our team guidelines we want to make sure, that we deliver not only entertaining but also stable software.

2.3 Test Motivators

Going hand in hand with our team guidelines, we want to ensure, that our Beta-Testers, can experience the most of our software, without encountering bugs every two seconds, torturing them to death.

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.

- Adventure Objects
- File Finder
- File Stream
- Alias List

4. Outline of Planned Tests

4.1 Outline of Test Inclusions

tbd

*[Provide a high level outline of the major testing planned for the current iteration. Note what will be included in the plan and record what will explicitly **not** be included in the section titled Outline of Test Exclusions.]*

4.2 Outline of Other Candidates for Potential Inclusion

Penetration Testing.

4.3 Outline of Test Exclusions

tbd

*[Provide a high level outline of the potential tests that might have been conducted but that have been **explicitly excluded** from this plan. If a type of test will not be implemented and executed, indicate this in a sentence stating the test will not be implemented or executed and stating the justification, such as:*

- *"These tests do not help achieve the evaluation mission."*
- *"There are insufficient resources to conduct these tests."*
- *"These tests are unnecessary due to the testing conducted by xxxx."*

As a heuristic, if you think it would be reasonable for one of your audience members to expect a certain aspect of testing to be included that you will not or cannot address, you should note it's exclusion: If the team agrees the exclusion is obvious, you probably don't need to list it.]

5. Test Approach

- `CppUnitTestFramework`
- `InputSimulator`

5.1 Initial Test-Idea Catalogs and Other Reference Sources

Test-Scripts

5.2 Testing Techniques and Types

5.2.1 Data and Database Integrity Testing

n/a

5.2.2 Function Testing

Unit Testing

Technique Objective:	Ensure functions to work as intended.
Technique:	Extensive unit testing for each function.
Oracles:	tbd
Required Tools:	Visual Studio
Success Criteria:	No false assertions while the test is running.
Special Considerations:	n/a

5.2.3 User Interface Testing

Input Simulator

Technique Objective:	Ensure expected results for a given input.
Technique:	Send inputs to the program, that a user might enter.
Oracles:	tbd
Required Tools:	Input Simulator
Success Criteria:	Manually checking the expected result.
Special Considerations:	Try automating the currently manual checking of the result.

5.2.4 Installation Testing

tbd

Technique Objective:	tbd
Technique:	tbd
Oracles:	tbd
Required Tools:	tbd
Success Criteria:	tbd
Special Considerations:	tbd

6. Entry and Exit Criteria

n/a

7. Deliverables

- Test Logs
- Evidence Videos

7.1 Test Evaluation Summaries

The UnitTests are getting evaluated every time, the master-branch on our GitHub-Repository is pushed, which will happen every couple of days.

7.2 Reporting on Test Coverage

> 20%

7.3 Perceived Quality Reports

n/a

7.4 Incident Logs and Change Requests

n/a

7.5 Smoke Test Suite and Supporting Test Scripts

n/a

7.6 Additional Work Products

Input Simulator

7.6.1 Detailed Test Results

tbd

7.6.2 Additional Automated Functional Test Scripts

tbd

7.6.3 Test Guidelines

n/a

7.6.4 Traceability Matrices

Tbd

8. Testing Workflow

- Write code
- Write tests
- Manually test it (and fix the code until it works)
- Push onto the build-server
- Fix any possible future errors, found by the automatic testing

9. Environmental Needs

Visual Studio / Windows 7+

9.1 Base System Hardware

n/a

9.2 Base Software Elements in the Test Environment

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

Software Element Name	Version	Type and Other Notes
Visual Studio	2017	IDE
Windows	7/8/10	Operating System

9.3 Productivity and Support Tools

n/a

9.4 Test Environment Configurations

The following Test Environment Configurations needs to be provided and supported for this project.

Configuration Name	Description	Implemented in Physical Configuration
Build configuration	Special Configuration, to test our code	

10. Responsibilities, Staffing, and Training Needs

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

10.1 People and Roles

This table shows the staffing assumptions for the test effort.

*[**Note**: Add or delete items as appropriate.]*

Human Resources		
Role	**Minimum Resources Recommended** (number of full-time roles allocated)	**Specific Responsibilities or Comments**
Test Manager		Provides management oversight. Responsibilities include: • planning and logistics • agree mission • identify motivators • acquire appropriate resources • present management reporting • advocate the interests of test • evaluate effectiveness of test effort
Test Analyst		Identifies and defines the specific tests to be conducted. Responsibilities include: • identify test ideas • define test details • determine test results • document change requests • evaluate product quality

Human Resources		
Role	**Minimum Resources Recommended** **(number of full-time roles allocated)**	**Specific Responsibilities or Comments**
Test Designer		Defines the technical approach to the implementation of the test effort. Responsibilities include: • define test approach • define test automation architecture • verify test techniques • define testability elements • structure test implementation
Tester		Implements and executes the tests. Responsibilities include: • implement tests and test suites • execute test suites • log results • analyze and recover from test failures • document incidents
Test System Administrator		Ensures test environment and assets are managed and maintained. Responsibilities include: • administer test management system • install and support access to, and recovery of, test environment configurations and test labs
Database Administrator, Database Manager		Ensures test data (database) environment and assets are managed and maintained. Responsibilities include: • support the administration of test data and test beds (database).
Designer		Identifies and defines the operations, attributes, and associations of the test classes. Responsibilities include: • defines the test classes required to support testability requirements as defined by the test team

Human Resources		
Role	**Minimum Resources Recommended** **(number of full-time roles allocated)**	**Specific Responsibilities or Comments**
Implementer		Implements and unit tests the test classes and test packages. Responsibilities include: • creates the test components required to support testability requirements as defined by the designer

10.2 Staffing and Training Needs

n/a

11. Iteration Milestones

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

12. Risks, Dependencies, Assumptions, and Constraints

[List any risks that may affect the successful execution of this **Test Plan**, and identify mitigation and contingency strategies for each risk. Also indicate a relative ranking for both the likelihood of occurrence and the impact if the risk is realized.]

Risk	Mitigation Strategy	Contingency (Risk is realized)
Prerequisite entry criteria is not met.	<Tester> will define the prerequisites that must be met before Load Testing can start. <Customer> will endeavor to meet prerequisites indicated by <Tester>.	• Meet outstanding prerequisites • Consider Load Test Failure
Test data proves to be inadequate.	<Customer> will ensure a full set of suitable and protected test data is available. <Tester> will indicate what is required and will verify the suitability of test data.	• Redefine test data • Review Test Plan and modify • components (that is, scripts) • Consider Load Test Failure
Database requires refresh.	<System Admin> will endeavor to ensure the Database is regularly refreshed as required by <Tester>.	• Restore data and restart • Clear Database

[List any dependencies identified during the development of this **Test Plan** that may affect its successful execution if those dependencies are not honored. Typically these dependencies relate to activities on the critical path that are prerequisites or post-requisites to one or more preceding (or subsequent) activities You should consider responsibilities you are relying on other teams or staff members external to the test effort completing, timing and dependencies of other planned tasks, the reliance on certain work products being produced.]

Dependency between	Potential Impact of Dependency	Owners

[List any assumptions made during the development of this **Test Plan** that may affect its successful execution if those assumptions are proven incorrect. Assumptions might relate to work you assume other teams are doing, expectations that certain aspects of the product or environment are stable, and so forth].

Assumption to be proven	Impact of Assumption being incorrect	Owners

[List any constraints placed on the test effort that have had a negative effect on the way in which this **Test Plan** has been approached.]

Constraint on	Impact Constraint has on test effort	Owners

13. Management Process and Procedures

n/a