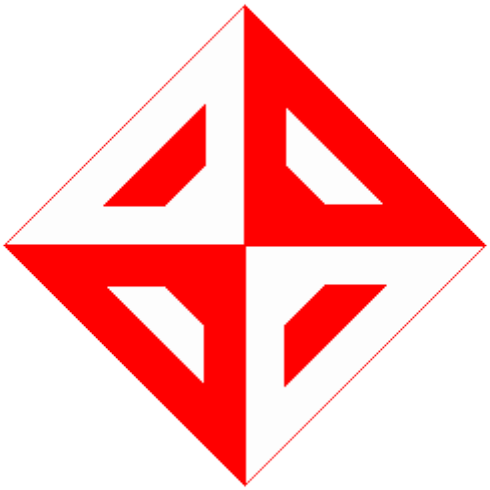

Barış ÇAVUŞ
Erenay DAYANIK
Memduh Çağrı DEMİR
Mesut BALCI

SYSTEM TEST DOCUMENTATION

The document is prepared by members of Vana Team; the document is about system test documentation about the project “Light My Way”.

03.05.2015

Table of Contents

1 Introduction

1.1 Document Identifier

This is the Software and System Test Document of the project named ‘Light My Way’ which is being developed by project group VANA. This document is based on IEEE 829-2008 Standard for Software and System Test Documentation. This document covers the all of the system tests with possible outcomes. These tests verify and validate the system’s features. This is the first version of the document which is written by the members of the team VANA.

1.2 Scope

The software product is a stand-alone mobile application developed in Unity3d game engine. It consists user interfaces for both single player and multiplayer game modes as well as game interfaces for multiple levels. The project’s features will be tested according to user interfaces’ functionalities as a result of the design. The test cases will include pre-conditions, test case descriptions, test case inputs as scenarios and expected outcomes.

1.3 References

[1] IEEE. (2008). IEEE Std. 829-2008 IEEE Standard for Software and System Test Documentation. IEEE Computer Society.

[2] Software Requirements Specification of the project Light My Way

[3] Software Design Descriptions of the project Light My Way

1.4 Level in the Overall Sequence

As mentioned in the section 1.2 Light My Way is a stand-alone project and thus it is not a part of the larger project. All of its functionalities are independent from each other. Combination of all of its individual functionalities creates the actual project. As a result, there will be only one level testing which is functional testing. Functional testing is a quality

assurance process that bases on its test cases. There are some inputs and expected outputs for each case independently.

1.5 Test Classes and Overall Test Conditions

All features of the product will be tested in this document and these tests will be covered on the next chapters of the document. All of the features are the functions which are described previously on Software Requirement Specifications and Software Design Descriptions documents by means of use cases. The product is tested to verify and to validate that it works as expected. Hence, all of the tests focus on meeting system's requirements.

1.6 Definitions, Acronyms & Abbreviations

User: A person who plays the game.

Character: The ball controlled by the player.

Unity3D: a rendering engine fully integrated with a complete set of intuitive tools and rapid workflows to create interactive 3D and 2D content.

Multiplayer: More than one players playing in the same game with separate devices.

Co-op / Co-operative: A multiplayer game type that allows players to play as a team to achieve the goal

Android: Android is the most popular mobile operating system (OS) based on the Linux kernel and currently developed by Google.

Ingame Screen: After invoking play in game menu, Ingame screen is the screen where users play with in a drawn interface.

GUI: Graphical user interface.

Maze: labyrinth in a level of map which player tries to solve.

SRS: Software Requirements Specification

SDD: Software Design Descriptions

2 Details for System Test Plan

The following sections describe test items and their identifiers, traceability matrix, features to be tested and not tested, approach that was followed through testing phase, pass/fail criteria for the items and test deliverables.

2.1 Test Items and Their Identifiers

The test items are the functionalities which were described in the Software Requirements Specifications document section 3.2 and were extended in the Software Design Document section 5.2 as a use cases. All of the test cases are listed below with their identifiers.

-Use Case 1: Play

-Use Case 2: Music

-Use Case 3: Rate

-Use Case 4: Sound

-Use Case 5: Single Player

-Use Case 6: Multiplayer

-Use Case 7: Back Home

-Use Case 8: Create Game

-Use Case 9: Refresh Games

-Use Case 10: Join Game

-Use Case 11: Move Up

-Use Case 12: Move Down

-Use Case 13: Move Left

-Use Case 14: Move Right

-Use Case 15: Interact Crystals

2.2 Test Traceability Matrix

The test traceability matrix is given below. First column shows the use cases which are denoted by UC and number e.g. UC10 represents Use Case 10: Join Game. First row shows the test cases related to the use case. Test cases are represented by TC and a number. There will be at least one test case for each use case.

	TC1	TC2	TC3	TC4	TC5	TC6	TC7	TC8	TC9	TC10	TC11	TC12	TC13	TC14	TC15	TC16	TC17
UC1	X																
UC2		X	X														
UC3				X													
UC4					X	X											
UC5							X										
UC6								X									
UC7									X								
UC8										X							
UC9											X						
UC10												X					
UC11													X				
UC12														X			
UC13															X		
UC14																X	
UC15																	X

2.3 Features to be tested

Features to be tested are listed in Section 1.2. They are the functionalities which were described in the previous documents. They are part of the user interface and ingame screen. Functionalities on the user interface (menu screens, game selections screens) are represented as buttons. Ingame screen functionalities are hidden on the mobile phone screen which handles the movements of the character and interactions. Both of these screens were implemented by the members of VANA team in C# language.

2.4 Features not to be tested

Since server uses the Unity3D's network server codes as a base, we will not test the core elements of this server implementation. We use the Unity master server source codes. This includes the connection tester, facilitator, master server and proxy server.

All source packages include the RakNet 3.732 networking library which handles the basic networking functions and provides plugins used by the networking servers.The packages include A Visual Studio 2008 solution for source codes.

The host information data is modified by the master server. The IP and port of the game which is registering, as seen by the master server, is injected into the host data. This way we can for sure detect the correct external IP and port in cases where the host has a private

address (NAT address). The IP and port in the host data sent by the game server is the private address and port and this is stored for later use. If the master server detects that a client is requesting the host data for a game server and the server has the *same* IP address then he uses the private address of the server instead of the external one. This is to handle cases where the client and server are on the same local network, using the same router with NAT addresses. Thus they will have the same external address and cannot connect to each other through it, they need to use the private addresses and those will work in this case.

2.5 Approach

All of the functionalities will be tested with knowing their internal structure and workings of the application. As a team VANA, we are the developers of these functionalities and we have all the knowledge about them. Since we know all of these information white-box testing technique will be used.

2.6 Item pass/fail Criteria

Expected outputs and the real outputs will be compared for each test case. The cases will be considered as passed if the outputs meet the expected outputs which were described in SRS and SDD. Otherwise, cases will be considered as failed.

2.7 Test Deliverables

Test deliverable for the project is this document. This document includes level test plans, level test cases and level test reports. Section 1, Section 2 and Section 3 cover test plans. Section 4 covers level test cases and Section 5 covers level test report.

3 Test Management

3.1 Planned Activities and tasks; test progression

In this section how the system can be tested and its performance in terms of stability, reliability and responsiveness will be provided.

First of all, testers who will test the system should be familiar with the design and implementations of the Light My Way project because white box testing method will be used

and it requires knowledge about the both functionalities and the test cases of the project. System boundaries and test goals should be decided to run these tests because testers want exact information about the test cases. Testers also design different scenarios for test cases. Purpose of these scenarios should be gathering maximum information about the test cases with minimal effort.

Once these requirements are met, testers can execute their test.

3.2 Environment/infrastructure

Light My Way project is based on mobile platforms, mainly Android platform. People who want to play the game and testers who want to run their tests must have a mobile device with Android platform. Single Player part of the project can be run with only this requirement. On the other hand, Multiplayer mode of the project requires an internet connection, Wi-Fi or 3G.

Our server uses the Unity3D's network server codes as a base, so we will not test the core elements of this server implementation as we mentioned in 2.4 section of the document. We have also deployed this server into Amazon EC2 instance that is located in Germany to decrease the delay of the game. Once we deploy the server we adapted it to our game. This server supports room creation and joining but not the matchmaking for users, so it is actually not too complex for to start.

4 Test Case Details

4.1 Use Case 1: Play Test Case

Pre-Conditions	The application shall be started
Description	To test Play button functionality
Input (Scenario)	User clicks the play button in the main menu
Expected Output	New interface should open with single player and multiplayer play option

4.2 Use Case 2: Music Test Case 1 – Open

Pre-Conditions	1. The application shall be started 2. The volume of the mobile device shall be open and more than zero value 3. The music of the game shall be closed means that music icon should show muted version. 4. The game should be on the main Screen.
Description	This test case is testing the user scenario for music opening.
Input (Scenario)	User touches the music button which is displayed as a speaker icon.
Expected Outputs	1. The main menu sound should be opened. 2. Music should start from the beginning and continue playing

4.3 Use Case 2: Music Test Case 2 – Close

Pre-Conditions	1. The application shall be started 2. The volume of the mobile device shall be open and more than zero value 3. The music of the game shall be opened means that music icon should show normal speaker version and sound can clearly be heard. 4. The game should be on the main Screen.
Description	This test case is testing the user scenario for music closing.

Input (Scenario)	User touches the music button which is displayed as a speaker icon.
Expected Output	The main menu sound should be closed.

4.4 Use Case 3: Rate Test Case

Pre-Conditions	1. The application shall be started. 2. The mobile device should have connected to internet with Wi-Fi or 3G 3. User must have Google Play account.
Description	This test case is testing the rate application on Google Play functionality is working or not.
Input (Scenario)	User touches the rate game button which is displayed as a star icon
Expected Outputs	1. Application rate page on Google Play should be opened. 2. User should be able rate application.

4.5 Use Case 4: Sound Test Case 1 – Open

Pre-Conditions	1. The application shall be started 2. The volume of the mobile device shall be open and more than zero value 3. The game should be on the main screen.
Description	This test case is testing the game play sounds.
Input (Scenario)	1. User touches the sound button in game main menu 2. User touches the single player button in play

	screen. 3. User touches the multiplayer button in play screen.
Expected Output	Music should start to play.

4.6 Use Case 4: Sound Test Case 2 – Close

Pre-Conditions	1. The application shall be started 2. The volume of the mobile device shall be open and more than zero value 3. The game should be on the main screen.
Description	This test case is testing the game play sounds.
Input (Scenario)	1. User touches the sound button in game main menu 2. User touches the single player button in play screen 3. User touches the multiplayer button in play screen.
Expected Outputs	1. Music should not start to play. 2. Sound effects should not be heard.

4.7 Use Case 5: Single Player Test Case

Pre-Conditions	1. The application shall be started 2. The Play button should be touched before to open play menu. 3. The game should be on the play menu where single player and multiplayer buttons are visible.
Description	This test case is testing the single player levels.

Input (Scenario)	User clicks the Single Player button.
Expected Output	1. The game should be opened. 2. Player should respawn.

4.8 Use Case 6: Multiplayer Test Case

Pre-Conditions	1. The application shall be started. 2. The mobile device should have connected to internet with Wi-Fi or 3G 3. The Play button should be touched before. 4. User should be on Play Menu rather than main menu
Description	This case is testing multiplayer game mode
Input (Scenario)	User clicks the Multiplayer button displayed with 3 people.
Expected Outputs	1. The new menu should be opened with Create 2. Game, Refresh Games and Join Game options

4.9 Use Case 7: Back Home Test Case

Pre-Conditions	1. The application shall be started 2. User should be on any other menu than a main menu
Description	Test case for return button on Android devices
Input (Scenario)	User touches the return button on their device
Expected Output	User should be able to return main menu which shows play, music, rate and sound buttons

4.10 Use Case 8: Create Game

Pre-Conditions	1. The application shall be started. 2. The mobile device should have connected to internet with Wi-Fi or 3G. 3. User should be on Multiplayer menu.
Description	This test case is testing creating the multiplayer game.
Input (Scenario)	User clicks create game button.
Expected Outputs	1. The multiplayer game should be opened. 2. Game should be hosted. 3. Player should respawn.

4.11 Use Case 9: Refresh Games

Pre-Conditions	1. The application shall be started. 2. The mobile device should have connected to internet with Wi-Fi or 3G. 3. User should be on Multiplayer menu.
Description	This test case is testing refreshing multiplayer games
Input (Scenario)	User clicks the refresh button.
Expected Output	User lists the online games.

4.12 Use Case 10: Join Game

Pre-Conditions	1. The application shall be started. 2. The mobile device should have connected to internet with Wi-Fi or 3G. 3. User should be on Multiplayer menu. 4. User should have listed multiplayer games
Description	Test case for joining a multiplayer game.
Input (Scenario)	User chooses the game that he/she wants to be joined from the multiplayer games menu
Expected Output	1. The multiplayer game should be opened. 2. Player should respawn.

4.13 Use Case 11: Move Up

Pre-Conditions	1. The application shall be started. 2. The game should be opened in single player mode. 3. Map should be visible and player should be spawned. 4. Player should stand still in the game screen.
Description	Test case for moving functionalities of player
Input (Scenario)	User sweeps the screen to upper direction by touching the screen.
Expected Output	Player should move to upper direction.

4.14 Use Case 12: Move Down

Pre-Conditions	1. The application shall be started. 2. The game should be opened in single player mode. 3. Map should be visible and player should be spawned. 4. Player should stand still in the game screen.
Description	Test case for moving functionalities of player.
Input (Scenario)	User sweeps the screen to down direction by touching the screen.
Expected Output	Player should move to down direction.

4.15 Use Case 13: Move Left

Pre-Conditions	1. The application shall be started. 2. The game should be opened in single player mode. 3. Map should be visible and player should be spawned. 4. Player should stand still in the game screen.
Description	Test case for moving functionalities of player.
Input (Scenario)	User sweeps the screen to left direction by touching the screen.
Expected Output	Player should move to left direction.

4.16 Use Case 14: Move Right

Pre-Conditions	1. The application shall be started. 2. The game should be opened in single player mode. 3. Map should be visible and player should be spawned. 4. Player should stand still in the game screen.
Description	Test case for moving functionalities of player.
Input (Scenario)	User sweeps the screen to right direction by touching the screen.
Expected Output	Player should move to right direction.

4.17 Use Case 15: Interact Crystals

Pre-Conditions	1. The application shall be started. 2. The game should be opened in single player mode. 3. Map should be visible and player should be spawned. 4. Player should stand still in the game screen. 5. A crystal should be visible and should be in lightened way(path should be reachable)
Description	Test case for crystal objects functionalities.
Input (Scenario)	User shall sweep the player towards a crystal that is on the visible path.
Expected Output	1. Crystal should explode. 2. Exploding effect should be visible. 3. Explode sound should be invoked.

5 System Test Report Details

5.1 Overview of Test Results

All items, specified in traceability matrix, were tested during the test procedure.

5.2 Detailed Test Results

5.2.1 TC1: Play

This functionality runs without any problem. The user can start the game and reach game mode selection menu.

Result: PASS

5.2.2 TC2: Music Open

This functionality runs without any problem. Music is successfully opened when the user touches the music open button.

Result: PASS

5.2.3 TC3: Music Close

This functionality runs without any problem. Music is successfully closed when the user touches the music close button.

Result: PASS

5.2.4 TC4: Rate

This functionality runs without any problem. User can open the application rate page on Google Play.

Result: PASS

5.2.5 TC5: Sound Open

This functionality runs without any problem. Sound effects and play music is clearly opened when user started the game after touched the sound button.

Result: PASS

5.2.6 TC6: Sound Close

This functionality runs without any problem. Sound effects and play music is not heard when user started the game after touched the sound close button.

Result: PASS

5.2.7 TC7: Single Player

This functionality runs without any problem. The user can start a new single player game.

Result: PASS

5.2.8 TC8: Multiplayer

This functionality runs without any problem. The user can open the multiplayer games menu which include create game, refresh games and join game.

Result: PASS

5.2.9 TC9: Back Home

This functionality runs without any problem. The user can move back to home menu.

Result: PASS

5.2.10 TC10: Create Game

This functionality runs without any problem. The user can create a new multiplayer game. This game will be listed on multiplayer games menu.

Result: PASS

5.2.11 TC11: Refresh Games

This functionality runs without any problem. The user can list the online multiplayer games.

Result: PASS

5.2.12 TC12: Join Game

This functionality runs without any problem. The user can join one of the online games from multiplayer games menu.

Result: PASS

5.2.13 TC13: Move Up

This functionality runs without any problem. The user can move up.

Result: PASS

5.2.14 TC14: Move Down

This functionality runs without any problem. The user can move down.

Result: PASS

5.2.15 TC15: Move Left

This functionality runs without any problem. The user can move to the left

Result: PASS

5.2.16 TC16: Move Right

This functionality runs without any problem. The user can move to the right.

Result: PASS

5.2.17 TC17: Interact Crystals

This functionality runs without any problem. The user can interact with the ingame objects and turn on the new lights.

Result: PASS

5.3 Rationale for Decisions

Rationale for decisions is that maximize the sequence of tests with the minimal effort. Although, test cases should be atomic and specific our test case decisions were inclusive without violating this principle. That was the decision behind separating each move test into different use cases when preparing this document and previously on the SRS and SDD documents.

5.4 Conclusions and Recommendations

In the testing phase 17 test cases have been tested by the members of the team VANA. All of the test cases PASSED and gave the expected outcomes. Each test case has been executed multiple times.