UNIVERSITY OF YORK

DEPARTMENT OF COMPUTER SCIENCE

ENG1 Group Assessment 2
Team 1

Auber

Test2.pdf - Testing

Group Members:

Jonathan Davies

Jamie Hewison

Harry Smith

Zee Thompson

Mark Varnaliy

Testing methods and approaches

We decided to use blackbox testing and perform tests for each of our functional requirements. One of the types of blackbox testing that we decided to implement is regression testing. It is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features. This type of testing is appropriate for our project because it is necessary to add and change code without breaking and adding bugs into the previous team's work.

Another type that we implemented is functional testing. It is defined as a type of software testing that validates a system against the functional requirements. This involved some unit tests, but mainly consisted of manual testing. Therefore, because test cases were not implemented in code, we decided to replace a traceability matrix with a functional requirements test table. It contains the unique ID of the test (TCID), whether it is automated or manual (Automated/Manual), the functional requirement related to this test (Requirement tested), aim of the test (Test performed), specified condition (Input test data), required output (Expected output), actual output (Output data) and result of the test (Pass/Fail).

The last two tables contain more precise descriptions of implementations of tests. The manual tests table has step-by-step instructions of required actions. Another table contains names of files for automated tests and their explanations. This format will allow future developers to conduct tests quicker and easier. All tests performed were successful which means that our implementation of another team's game satisfies the most crucial functional requirements.

The evidence of testing can be found at the end of this document.

Tests

TCID	**Automated/ Manual?**	**Requirement tested**	**Test performed**	**Input test data**	**Expected Output**	**Output data**	**Pass/Fail**
001	Manual	FR_GAME_WIN	Is the game won when all operatives are arrested?	All operatives arrested	Game won	Game won	Pass
002	Manual	FR_GAME_LOSE	Is the game lost when all 15 systems are destroyed?	All 15 systems destroyed	Game lost	Game lost	Pass
003	Automated	FR_OPER_NUM	Are there 8 operatives in the game?	JSON 'mapdata' file	8 operatives in file	8 operatives in file	Pass
004	Manual	FR_GAME_MUTE	Is the game muted when the button is pressed?	Mute button is pressed and game is played	All sounds disappear, including music and SFX	All sounds disappear, including music and SFX	Pass
005	Manual	FR_UI_SCALABLE	Are all elements of the game scalable?	Game is run with different resolutions	All elements must adjust to user's	All elements adjust to User's resolution	Pass

					resolution		
006	Manual	FR_PLAYER_HEAL	Can the player heal their health?	Player enters medbay or pick up Heal power up	Player health increases	Player health increases	Pass
007	Manual	FR_PLAYER_TELEPORT	Can the player teleport to different rooms around the map?	Player presses ‘E’ on teleporter, then clicks on room	Player teleports to room	Player teleports to room	Pass
008	Manual	FR_PLAYER_NOTIFIED	Is the player notified when a system is under attack?	Player waits for the system to be under attack.	Notification of ‘system under attack’ in bottom right	Notification appears	Pass
009	Manual	FR_PLAYER_SYSTEM_LOCATION	Is the location of the system that is under attack displayed?	Location of system under attack displayed in message in the bottom right	Location displayed	Location displayed	Pass
010	Automated	FR_MAP_SYSTEMS	Are there at least 15 systems in the game?	Number of systems in the game	Systems greater than or equal to 15	Systems greater than or equal to 15	Pass
011	Automated	FR_MAP_ROOMS	Are there at least 4 room types in the game?	Number of unique rooms in game	Unique rooms greater or equal to 4	Unique rooms greater or equal to 4	Pass
012	Manual	FR_MAP_LAYOUT	All systems on the map must be reachable by operatives (without the use of teleport pads), and thus by the player	Player moves to every room	Player can reach every room	Player can reach every room	Pass
013	Automated	FR_MAP_TELEPORTERS	Does the map must contain at least two teleporters?	Number of teleporters	Teleporters greater or equal to 2	Teleporters greater or equal to 2	Pass
014	Automated	FR_MAP_INFIRM	Does the map contain an infirmary room?	Number of infirmary rooms	Infirmary room greater or equal to 1	Infirmary room greater or equal to 1	Pass
015	Automated	FR_MAP_INFIRM_TELEPORTER	Does the infirmary include a teleport pad?	Medbay’s teleporter coordinates	Medbay’s teleporter coordinates not equal to null	Medbay’s teleporter coordinates not equal to null	Pass
016	Manual	FR_MAP_INFIRM_HEAL	Does the infirmary contain some way to heal the player?	Player enters infirmary with less than max health	Player’s health regenerates	Player’s health regenerates	Pass

017	Manual	FR_GAME_DEMO	Does the game have a demo mode, not requiring user input?	Player clicks demo mode button, without further input	Demo mode displays and runs	Demo mode displays and runs	Pass
018	Manual	FR_GAME_DIFFICULTY	Are there different difficulty modes?	Player clicks easy, medium, and hard game	Games are of different difficulty	Games are of different difficulty	Pass
019	Manual	FR_PLAYER_POWERUP	Can the player gain powerups during the game?	Player presses 'SPACE' on powerups	Notification appears, along with effect	Notification appears, along with effect	Pass
020	Manual	FR_GAME_SAVE	Can the player save the game, and then reload it?	Player presses 'save game' in ESC menu, and then exits the game. Player then loads game.	Game loads with same save state	Game loads with same save state	Pass
021	Automated	Assets exist?	Do the relevant assets exist within the game files?	Game asset files	All assets exist	All assets exist	Pass

All of the above tests managed to pass with our current implementation. We believe these tests are sufficient to show that the game meets all the requirements necessary for this assessment. More tests would be necessary if we were to produce a game that would be released to the mass market, as we'd then need to be 100% certain that the game is flawless and could not be exploited in any way.

Implementation of manual tests:

TCID	Steps required
001	Tester must arrest all operatives before they destroy all systems and 'Game won' must appear.
002	Tester must let operatives destroy all systems and 'Game lost' screen must appear.
004	Tester must click Mute button and then start the game and make sure that all in-game sounds (teleportation, walking ,etc.) are gone as well. Also, mute button can be pressed again and same test can be implemented in order to test efficiency of button and sounds of the game.
005	Tester should change the resolution of the computer and run the game. Then, check 'HOW TO PLAY' menu and play the game.
006	Tester should take damage from the operative. Then, enter the medbay system and check if the health of the player is increasing gradually. After that, take damage again and pick up heal or regen power up and the health should increase instantly or gradually respectively. (Medbay and power ups can be tested in any order).
007	Tester should stand on every teleport pad implemented in the game and teleport to every possible location by pressing 'E' and choosing destination. (Tester should not press 'E' while teleporting as It can cause image overlap and is not expected to be done by user. If this happens tester should move out of the teleport pad range).
008/009	Tester should start the game and wait until systems will be under attack and check whether notification about what system and when appeared in the notification log at the bottom right of the screen.
012	Tester should move to every location shown in minimap (Available by holding 'M' key).
016	Tester should take some damage from an infiltrator, and then move to the infirmary (shown on minimap). They should then check that after some time, their health increases.

017	Tester should launch game, click play, and then click the ‘DEMO’ button, and give no further input
018	Tester should try and win game on each difficulty mode, and report whether they found one was more difficult than the other.
019	Tester should find powerups (in the form of syringes), press ‘SPACE’ on them and notice a notification as well as the shown affect.
020	Tester should pause mid-way through a game, press ‘save game’ button, then exit the game and press play again. Upon being asked whether they would like to load from a saved game, they should press ‘yes’.

Implementation of automated tests:

TCID	Test Method(s) Name	Explanation
003	operativeEqualsEight()	Checks whether there are 8 operatives listed in the ‘mapdata.json’ file
010	systemsEqualsFifteen()	Checks if there are at least 15 rooms, and each room has a system so num_rooms = num_systems
011	fourUniqueRoomsExist()	Checks at least 4 rooms with unique names. Uses a helper function to loop through list comparing each item and increments rooms by 1, and checks whether >= 4.
013	twoTeleportersExist()	Checks if there are at least two rooms with a teleporter in them.
014	oneInfirmaryRoomExists()	Loops through the rooms until it finds one with the name ‘Medbay’
015	medbayTeleporterExists()	Looks at the JSON entry for the medbay and checks if an integer list exists at the ‘teleporterCoords’ entry
021	mapAssetsExist(); skinAssetsExist(); audioAssetsExist(); imgAssetsExist(); imgMenuAssetsExist()	All methods check whether the relevant assets exist in the game files

Proof of Design & Testing:

Material showing our testing material, consisting of test design and evidence of Testing, can be found on our website, at https://eng1-eclipse.github.io/website2/software-testing/. This page features a list of the tests carried out which you can click on to learn more about each specific test. Videos are available for each manual test, and test reports that were generated are also available, that display the relevant tests passing. The videos strictly follow the guidelines set out in our ‘Implementation of manual tests’ table.

You can also directly access the page for a test by clicking their TCID number in any table within this document. This will take you directly to the relevant page containing test information, and evidence of the testing taking place.