SE 3XA3: Test Report
Dragon Age

Group 8: Team Eight
Stanley Liu (MacID: liuz23)
Toni Miharja (MacID: miharjat)
Zhi Zhang (MacID: zhangz1)

December 6 2017

Contents

1 Revision History

Revision	Author	Date	Change
1	Stanley Liu	12/06/17	Final Copy Rev 1
0	Stanley Liu	12/05/17	Final Copy Rev 0

Table 1: Revision History for Test Report Document

2 List of Tables and Figures

3 Functional Qualities Evaluation

Description of Tests: The purpose of these tests is to ensure that the user is able to play the game according to the given requirements. These tests will include testing for enemy, dragon tower, bullet and path.

Test Name: FRE-1

Results: Enemy enters the path when game starts

Test Name: FRE-2

Results: Speed and number of enemies increases as the wave increases

Test Name: FRE-3

Results: Tower launches bullet and hit enemy, does damage to enemy

Test Name: FRE-4

Results: Bullet is able to kill enemy is bullet damage is greater than the rest of enemy health

Test Name: FRD-1

Results: Dragon tower is upgraded correctly

Test Name: FRD-2

Results: The *isInRange* equation is working correctly

Test Name: FRT-1

Results: Bullets are all removed when all enemy exit

Test Name: FRT-2

Results: Damage is set on enemy correctly

Test Name: FRT-3

Results: All enemies are removed and game over when life equals 0

Test Name: FRP-1

Results: Game always starts with enemy wave coming in

Test Name: FRP-2

Results: *onboard* function is working correctly

Test Name: FRP-3

Results: *upgradeBound* function is working correctly

Test Name: FRP-4

Results: *onRoute* function is working correctl

4 Non-Functional Qualities Evaluation

Description of Tests: The purpose of these tests is to ensure the usability and performance of the game. These tests will include testing for usability and performance.

Test Name: NF-U-1

Results: Game runs successfully on every operating system

Test Name: NF-U-2

Results: All buttons and button outcome generated correctly

Test Name: NF-U-3

Results: Average rating of team members is 4.3 (greater than 3)

Test Name: NF-U-4

Results: Average rating of team members is 4.0 (greater than 3)

Test Name: NF-P-1

Results: The game is loaded into the introduction under 2 seconds

Test Name: NF-P-2

Results: The reaction time of all button clicks to achieve function is under 0.5 seconds

5 Changes Due to Testing

Interfacing-wise, the Dragon Age team has upgraded the introduction page and changed the map to have a more middle earth feel. Coding-wise, the team has been test running the game at the development process, therefore, very few changes has been made due to testing.

6 Automated Testing

We ran the *pygame unit test suite* in command line to ran 680 test cases for our game. The test cases include image testing, pixel testing, syntax testing and so on. All the test cases passed in 15.754 seconds. In this document, we will represent automated testing by "ATT".

7 System Tests

7.1 Enemy Testing

These are the tests to test enemy of the game.

Test Name	FRE-1
Initial State	No enemy enters the board
Input	User press start button
Expected Output	One wave of enemy enters the path

Table 2: Test for FRE-1

7.2 Dragon Tower Testing

These are the tests to test dragon tower of the game.

Test Name	FRE-2
Initial State	A small enemy wave at level 1 moves on board
Input	User defeats wave 1
Expected Output	At next wave, number of enemy increases and enemy speed increases

Table 3: Test for FRE-2

Test Name	FRE-3
Initial State	Tower is static
Input	Enemy walks into launch range of tower
Expected Output	Tower launches bullet and hits enemy, does damage to enemy

Table 4: Test for FRE-3

Test Name	FRE-4
Initial State	Enemy moves and tower launches bullets
Input	Bullet hits enemy and the damage is greater than the rest of the enemy health
Expected Output	Enemy is killed and removed from board

Table 5: Test for FRE-4

Test Name	FRD-1
Initial State	Dragon tower is not upgraded
Input	User select update button with sufficient gold
Expected Output	Dragon tower is upgraded 1 level

Table 6: Test for FRD-1

Test Name	FRD-2
Initial State	Dragon tower is build and enemy wave moves in
Input	User waits until the enemy comes into tower range
Expected Output	Output if the enemy is in range or not therefore checks *isInRange* equation

Table 7: Test for FRD-2

7.3 TimerFired Testing

These are the tests to test timerFired functions of the game.

Test Name	FRT-1
Initial State	Dragon tower placed and enemy came in
Input	Tower shots at enemy
Expected Output	Bullets are removed when all enemy exit

Table 8: Test for FRT-1

Test Name	FRT-2
Initial State	Bullet shots at enemy
Input	Enemy is hit by bullet
Expected Output	Damage is set on enemy

Table 9: Test for FRT-2

Test Name	FRT-3
Initial State	Game started
Input	Game life equals to 0
Expected Output	All enemies are removed and game over

Table 10: Test for FRT-3

7.4 Path Testing

These are the tests to test path of the game.

Test Name	FRP-1
Initial State	Game loaded into introduction page
Input	User clicks on start game
Expected Output	Game start with enemy wave coming in

Table 11: Test for FRP-1

Test Name	FRP-2
Initial State	Game started
Input	Check if any tower is on board
Expected Output	True if on board, otherwise False

Table 12: Test for FRP-2

Test Name	FRP-3
Initial State	There is enough coins to upgrade a dragon
Input	User clicks on dragon on board and try to upgrade
Expected Output	Check if the mouse is in upgrade button bound

Table 13: Test for FRP-3

Test Name	FRP-4
Initial State	Dragon tower placed on board
Input	User clicks on the dragon tower on board
Expected Output	Output if the dragon tower is on game route

Table 14: Test for FRP-4

7.5 Non-Functional Test

7.5.1 Usability

These are the tests to test usability of the game.

Test Name	NF-U-1
Initial State	Game file downloaded onto personal computers
Input	Launch the game on personal computers running Windows, Mac OS and Linux
Expected Output	Game runs successfully on every operating system

Table 15: Test for NF-U-1

Test Name	NF-U-2
Initial State	Game is opened on personal computers
Input	User plays the game by mouse clicks
Expected Output	All buttons work correctly, outcome generated correctly

Table 16: Test for NF-U-2

Test Name	NF-U-3
Initial State	Testing team has tested previous two tests
Input	Member in test group are asked to rate the overall satisfaction of our game
Expected Output	The average rating is greater than 3

Table 17: Test for NF-U-3

Test Name	NF-U-4
Initial State	Testing team has tested NF-U-1 and NF-U-2
Input	Member in test group are asked to rate the overall satisfaction of Pokemon Tower Defense
Expected Output	The average rating is greater than 3

Table 18: Test for NF-U-4

7.5.2 Performance

These are the tests to test performance of the game.

Test Name	NF-P-1
Initial State	Game is launched
Input	User opens the game
Expected Output	The game is loaded into the introduction under 2 seconds

Table 19: Test for NF-P-1

Test Name	NF-P-2
Initial State	Game is loaded into the game screen where user start to play
Input	User plays the game by mouse clicking
Expected Output	The reaction time of button click to achieve function should be under 0.5 seconds

Table 20: Test for NF-P-2

8 Trace to Requirements

Test	Requirements
	Functional Requirements Testing
FRE-1	FREW-1
FRE-2	FREWU-1
FRE-3	FR-TDE-1
FRE-4	FR-TDE-2
FRD-1	FR-UDT-2
FRD-2	DT-RNG-1
FRT-1	DT-BR-2
FRT-2	DT-BR-1
FRT-3	FR-GOD-2
FRP-1	FR-SPSS-2
FRP-2	FR-BNDT-3
FRP-3	FR-UDT2
FRP-4	FR-PDT-2
	Non-functional Requirements Testing
NF-U-1	NFR-UB-1
NF-U-2	NFR-UB-2
NF-U-3	NFR-UB-3
NF-U-4	NFR-UB-4
NF-P-1	NFR-P-1
NF-P-2	NFR-P-2
	Automated Testing
ATT	NFR-UB-1, NFR-UB-2, NFR-P-1, NFR-P-2

Table 21: Trace Between Tests and Requirements

9 Trace to Modules

Test	Modules
Functional Requirements Testing	
FRE-1	M3, M11
FRE-2	M3, M11
FRE-3	M3, M11
FRE-4	M3, M11
FRD-1	M1, M9
FRD-2	M1, M9
FRT-1	M4, M5, M2
FRT-2	M5, M2
FRT-3	M3, M4, M5, M11
FRP-1	M12
FRP-2	M12
FRP-3	M1, M12
FRP-4	M1, M12
Non-functional Requirements Testing	
NF-U-1	M8
NF-U-2	M8
NF-U-3	M8
NF-U-4	M8
NF-P-1	M8
NF-P-2	M8
Automated Testing	
ATT	M1 - M13

Table 22: Trace Between Tests and Modules

10 Code Coverage Metrics

The Dragon Age team has managed to produce at least 85 percent of the code coverage through testing. This is based on the number of test cases the team wrote for every module, requirement and specific function. Please refer to the traceability in the upper section.