SE 3XA3: Test Report
Fault in our Pong

Team 03, Pongthusiastics
Adwity Sharma - sharma78
Arfa Butt - buttaa3
Jie Luo - luoj3

December 8, 2016

Contents

Table 1: **Revision History**

Date	**Version**	**Notes**
7 December, 2016	1.0	Created test report

1 Functional Requirements Evaluation

Description of Tests: The purpose of these tests is to ensure that the functional requirements that were detailed previously. It also confirms the satisfaction of testing criteria set in test report. The test names are the same as the one used in the test report document. Our project passed all the functional requirements detailed below, hence our project is functionally adequate, in the fields that are tested below.

1.1 Mode Selection

1.1.1 Open new game

Test Name: FS-NG-1: New game
Test Method: Running the game
Result: When user chooses new game, a new page with the options for selecting a new game opens.

Test Name: FS-NG-2: Single Player Mode chosen
Test Method: Running the game
Result: In single player mode game against AI is started.

Test Name: FS-NG-3: Advanced Single Player Mode chosen
Test Method: Running the game
Result: An advanced single player mode is chosen with an obstacle added when advanced player mode is selected. There is a red ball that has to be avoided when in the obstacle mode.

Test Name: FS-NG-5: Back button to go back to the menu page
Test Method: Running the game
Result: There is a back button that takes us back to the menu page.

1.2 Load Game

Test Name: FS-LG-1: Load a saved game instead of starting a new one
Test Method: Running the game
Result: The last saved game opens instead of starting a new one when the load game is pressed.

Test Name: FS-LG-2: Game loaded is same as the game state that was saved the last time
Test Method: Running the game
Result: The game loaded is same as the game state that was saved the last time. The number of lives and the ball and paddle position is retrained from the previous game.

1.3 High Score

Test Name: FS-HS-1: Open high scores page when option is chosen from the menu page

Test Method: Running the game

Result: When high scores page is chosen from the options we are directed to a display page containing user name and score of twenty highest scoring players.

Test Name: FS-HS-2: Add a new high score and check if it is added at the right place in the high scores list

Test Method: Running the game

Result: When new high score is made there is a pop up window that allows the user to add in a user name and it is added at the right place in the high scores list.

1.4 Tutorial

Test Name: FS-TU-1: Open tutorial page when option is chosen from the menu page

Test Method: Running the game

Result: When the tutorial option is chosen from the main menu a tutorial page is opened.

Test Name: FS-TU-2: Back button to go back to the menu page

Test Method: Running the game

Result: when we press the back button we are taken back to the menu page.

1.5 Game State

Test Name: FS-GS-1: Paddle movement

Test Method: Running the game

Result: When the game is opened left key is pressed the paddle moves left on the console.

Test Name: FS-GS-2: Paddle movement

Test Method: Running the game

Result: When the game is opened right key is pressed the paddle moves right on the console.

Test Name: FS-GS-3: Increment scores

Test Method: Running the game

Result: The score is determined by the time the game is played. Playing the game for longer increases the score.

Test Name: FS-GS-4: Decrease lives (in single player mode)
Test Method: Running the game
Result: The program is run, in the single player mode game a turn is missed and the players lives decreased by 1.

2 Nonfunctional Requirements Evaluation

Our project passed all the nonfunctional requirements detailed below, hence our project is adequate, in the fields that are tested below.

2.1 Usability

Test Name: FN-1: This testing is done to ensure that the re-designing of the game has made the game better than it was originally.
Test Method: Survey
Result: A survey was conducted; the users were asked if the redesigned version of the game was better than the original version of the game. All the users that took part in the survey answered that the redesigned game is better than the original game. Hence, this test has been completed successfully.

Test Name: FN-2: Test to conrm that the game runs in all major operating systems such as Windows, Mac OX, Linux etc.
Test Method: Running the game in various operating systems
Result: The game was run in various operating systems and it compiled and run in each of the platform.

2.2 Spelling and Grammar

Test Name: FN-3: Making sure that the game does not have any grammar or spelling errors.
Test Method: Proofreading documents and the game display
Result: The folders within the game and the documents relating to it were checked for grammar and spelling errors during revision 1 to ensure the competence of this test.

2.3 Hardware requirement

Test Name: FN-4: Making sure that the speed of game doesnt change in dierent machines.
Test Method: Running the game in various computers and finding the speed of the ball in each machine.

Result: The game does not have tangible speed difference when it runs if various computers.

2.4 Entertainment

Test Name: FN-5: This testing is done to determine if the game is entertaining enough for the user.

Test Method: Survey

Result: Users rated the game as entertaining and more than 50 percent of the users were willing to play the game again.

2.5 Challenge

Test Name: FN-6: Test done to determine if the game is challenging enough for the user.

Test method: Survey

Result: The testing group played the game and around 60 percent of the users rated the game as easy and 40 percent of the users rated the game as medium. A medium means that the game isnt too hard but isnt so easy that it gets boring.

2.6 Controls

Test Name: FN-7: Test done to determine if the game controls are intuitive.

Test method: survey

Result: The testing group was asked if it is easy to play and run the game. All the participates of the survey replied this question with very easy.

2.7 Robustness and Performance

Test Name: FN-8: Test done to estimate Robustness and Performance.

Test Method: Manually browsing through the various states in the game

Result: The game was loaded and played 15 times. In this duration the game didnt crash a single time and the level of performance was never undermined.

3 Comparison to Existing Implementation

The nonfunctional requirement evaluation (FN-1) was done to ensure that the reconstruction of the game made it better among the user than it was originally. The users were asked if the redesigned version of the game was better than the original version of the game. All the users that took part in the survey answered that the redesigned game is better than the original game. Hence, this test ensured that the redesigned version of the game is better than the original.

4 Unit Testing

Junit was used to conduct the Unit testing. After rigorously testing the software, the various units that were tested yielded expected results. A full detail on the test cases that were performed can be viewed by accessing the j-unit test files that is compiled under the source folder with other java implementation code for this project.

5 Changes Due to Testing

During the testing phase of the project, the survey showed that due to lack of portability many of the users were not as willing to play or download our game. When we did the survey, we had not created a java jar file. This led us to problem as people without a java running environment did not want to download a java running environment just to play the game. Thus, after the survey data were evaluated our team decided to create a jar file to increase the portability of the project.

6 Automated Testing

Test Name: PC-1: Checking to see that the game sets score correctly.

Method: A new game state was created and the score at the beginning of the game is returned for both the user and the player. Then the results are evaluated to see that it matches the expected result. The unit test units: testSetTopScore() and testSetButtomScore() were used to perform this test.

Initial input: 10

Output: 10

Result: Pass.

Test name: PC-4: Checking to see that the position of the ball is set properly.

Method: A new game state is created and the position of the ball at the beginning of the game is returned and compared with the expected position of the ball.

Initial input: position of the ball from the game.

Output: (x,y) = (20,20)

Result: Pass.

Test name: PC-5: Checking to see that the potion of the paddle is set properly.

Method: A new game state is created and the position of the paddle at the beginning of the game is returned and compared with the expected position of the paddle.

Initial input: position of the paddle from the game.

Output: (x,y) = (20,20)

Result: Pass.

7 Trace to Requirements

Trace	**Requirements**
FS-NG-1	FR1
FS-NG-2	FR6
FS-NG-3	FR7
FS-NG-5	FR20
FS-NG-1	FR1,FR6
FS-LG-1	FR2,FR3
FS-NG-2	FR1,FR6
FS-HS-1	FR21
FN - 6	NFR2
FN-4	NFR4

Table 2: Trace Between Requirements and Testing

8 Code Coverage Metrics

Our group has managed to produce roughly 70 percent code coverage. Through j-unit test we have covered most of the modules in testing units. Also survey has helped us test most of the nonfunctional requirements.