SE 3XA3: Test Report
Mini-Arcade

Andrew Hum	William Lei	Arshan Khan	Jame Tran
huma3	leim5	khana172	tranj52
400138826	400125240	400145605	400144141

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	**Version**	**Notes**
4/2/2020	1.0	Andrew - Formatted Document & Section 1

1 Functional Qualities Evaluation

Description of Tests: These tests are used to demonstrate that the requirements of the product are fulfilled based upon the functional requirements from the Software Requirements Specification. These tests will demonstrate basic functionality of the game and its correctness.

Test Name: FRCG
Results: The user is able to choose a game to play

Test Name: FRSS
Results: The user is able to access and see the scoreboard

Test Name: FRPM
Results: The user is able to play the Maze game

Test Name: FRPP
Results: The user is able to play the Pong Game

Test Name: FRPF
Results: The user is able to play the Flappy Game

Test Name: FRRL
Results: The user is able to return to the launcher

Test Name: FRME
Result: The user is able access all of the menu elements for each game

Test Name: FRKM
Result: The user is able to move the player using the keyboard or mouse (varies per game)

Test Name: FRSG
Result: The game produces a final score and records it for the scoreboard

2 Nonfunctional Qualities Evaluation

2.1 Usability

Description of Tests:

Test Name:

Results:

2.1.1 Example Test Scenario

Description of Tests:

Test Name:
Results:

2.2 Performance and Robustness

2.2.1 Example Test Scenario

Description of Tests:

Test Name:
Results:

3 Changes Due to Testing

3.1 Launcher Testing

3.2 Scoreboard Testing

3.3 Maze Testing

3.4 Pong Testing

3.5 Flappy Testing

4 Automated Testing

4.1 Launcher Testing

4.2 Scoreboard Testing

4.3 Maze Testing

Description of tests:

Results:

4.4 Pong Testing

Description of tests:

Results:

4.5 Flappy Testing

Description of tests:

Results:

5 System Tests

5.1 Launcher Testing

Test Name -
Initial State -
Input -
Expected Output -

5.2 Scoreboard Testing

Test Name -
Initial State -
Input -
Expected Output -

5.3 Maze Testing

Test Name -
Initial State -
Input -
Expected Output -

5.4 Pong Testing

Test Name -
Initial State -
Input -
Expected Output -

5.5 Flappy Testing

Test Name -
Initial State -
Input -
Expected Output -

6 Trace to Requirements

6.1 Functional Requirements Traceability Matrix

Test	**Req.**
FT-1	-
FT-2	-
FT-3	-
FT-4	-
FT-5	-
FT-6	-
FT-7	-
FT-8	-
FT-9	-

Table 2: Trace Between Tests and Functional Requirements

6.2 Non-Functional Requirements Traceability Matrix

Test	**Req.**
NFT-1	-
NFT-2	-
NFT-3	-
NFT-4	-
NFT-5	-
NFT-6	-
NFT-7	-
NFT-8	-
NFT-9	-

Table 3: Trace Between Tests and Non-Functional Requirements

6.3 Automated Testing Traceability Matrix

Test	**Req.**
AT-1	-
AT-2	-
AT-3	-
AT-4	-
AT-5	-
AT-6	-
AT-7	-
AT-8	-
AT-9	-

Table 4: Trace Between Automated Tests and Requirements

7 Trace to Modules

7.1 Functional Requirements Traceability Matrix

Test	Modules
FT-1	-
FT-2	-
FT-3	-
FT-4	-
FT-5	-
FT-6	-
FT-7	-
FT-8	-
FT-9	-

Table 5: Trace Between Functional Tests and Modules

7.2 Non-Functional Requirements Traceability Matrix

Test	Modules
NFT-1	-
NFT-2	-
NFT-3	-
NFT-4	-
NFT-5	-
NFT-6	-
NFT-7	-
NFT-8	-
NFT-9	-

Table 6: Trace Between Non-Functional Tests and Modules

7.3 Automated Testing Traceability Matrix

Test	Modules
AT-1	-
AT-2	-
AT-3	-
AT-4	-
AT-5	-
AT-6	-
AT-7	-
AT-8	-
AT-9	-

Table 7: Trace Between Automated Tests and Modules

8 Code Coverage Metrics

With the tests mentioned throughout this document, our group has produced an approximate 85% code coverage. This is evident through the traceability matrices above, as they show that every module has been covered, with some being covered multiple times.