Test Report for Group 15 - Flight Shooting Game

Yijun Chen (cheny161) Tianxing Li (lit20) Zefeng Wang (wangz217)

Contents

1 Revision History

Revision	Date	Change
3	12/05/18	Final Rev 1
0	11/05/18	Draft Complete

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 use the software according to the given requirements. These tests will include direction control testing, collision testing, ultimate testing, start, restart and quit testing.

Test Name: FS-PC-2

Results: The user is able to move around the flight by pressing "WASD" or direction keys.

Test Name: FS-R-3

Results: The user is able to collide with enemy flights and enemy bullets to lose health points.

Test Name: FS-PC-17

Results: The user is able to press "U" on keyboard to use ultimate when the ultimate condition is ready. After the use of ultimate, all enemies already on the will lose 3500 health points.

Test Name: FS-MC-1

Results: The user is able to press "SPACE" to enter the game at the start page.

Test Name: FS-MC-2

Results: The user is able to press "R" to restart the game at the end page. All game data will be reset and the game goes to the start page.

Test Name: FS-MC-3

Results: The user is able to press "Q" to quit the game at the end page. The game window will be closed right after pressing the key.

4 Non-functional Qualities Evaluation

4.1 GUI Testing

Description of Test: Usability of the Graphical User Interface (GUI) was tested by a small test group of ten McMaster students who are not in Software Engineering but are instead from different backgrounds (i.e. Social Sciences) to better reflect the technological experience of the potential users for this program. Participants were monitored to observe the time it took them to perform the requested task. After the participants were finished with all tasks, each participant gives feedback to the developers verbally.

4.1.1 Look and feel

Test Name: NF-L-1)
Results: All participants were able to successfully complete the task and tell the difficulty level of the enemies. However, the difficulty level between head boss and chramp boss are too close according to the feedback.

4.1.2 usability

Test Name: NF-U-1
Results: All the participants were able to successfully complete the task. According to the feedback, the game control keys are pretty common and popular from the existing games.

4.1.3 Performance

Test Name: NF-P-1
Results: According to the feedback, no latency or slow response occurred during game play.

4.1.4 Operational and Environmental

Test Name: NF-O-1
Results: No content that would violate culture, religion or politics was found.

5 Automated Testing

For this part, *unittest* library from Python is used. For each module in the system, a test code is written and conducted. Functional methods in such module are tested.

Description of tests for each module:

Given a module M and its set of functional methods S where $m_n \in$ S.

Create a unit test class T with a set of n test cases TC where $c_n \in$ TC.

Each test case refers to a module method.

$$c_n \Longleftrightarrow m_n$$

Each test case with a module method get a test result:

$$Conduct(c_n, m_n) \Rightarrow bool$$

True refers to pass and False refers to fail.

Here is the list of the functions for unit testing:

Item No.	Function
1	Player.reset(lifepoint)
2	SmartBullet.move()
3	SmartBullet.aim(player)
4	SmartBullet.reset(x,y)
5	Boss.move()
6	Boss.show_up()
7	Boss.check_death(lifepoint, score)
8	Boss.reset()
9	Bullet.move()
10	Bullet.reset()
11	Enemy.move()
12	Enemy.reset()
13	Enemy.check_death()
14	Enemy.not_collidable()
15	Gift.move()

Result: The end results of the testing were positive and all 15 test cases passed.

6 System Test

6.1 User Control Testing

Test Name	FS-PC-1
Initial State	Game is running and player plane stays still at the bottom of the window.
Input	âĂIJWâĂİ or âĂIJUPâĂİ button.
Expected Output	The player plane moves up for certain distance in range of 1/5 of the window vertically.

Table 2: Test for FS-PC-1

Test Name	FS-PC-2
Initial State	A graph of player plane stays still at the bottom of the window.
Input	âĂIJWâĂİ or âĂIJUPâĂİ button. (Hold)
Expected Output	The player plane keeps moving up until it reach the top of the screen

Table 3: Test for FS-PC-2

Test Name	FS-PC-3..4
Initial State	A graph of player plane stays still at the bottom of the window.
Input	âĂIJSâĂİ or âĂIJDownâĂİ button. (Hold)
Expected Output	The player plane keeps moving up until it reach the bottom of the screen

Table 4: Test for FS-PC-3..4

Test Name	FS-PC-5..6
Initial State	A graph of player plane stays still at the bottom of the window.
Input	âĂIJAâĂİ or âĂIJLeftâĂİ button. (Hold)
Expected Output	The player plane keeps moving up until it reach the left border of the screen

Table 5: Test for FS-PC-5..6

Test Name	FS-PC-7..8
Initial State	A graph of player plane stays still at the bottom of the window.
Input	âĂIJDâĂİ or âĂIJRightâĂİ button. (Hold)
Expected Output	The player plane keeps moving up until it reach the right border of the screen

Table 6: Test for FS-PC-7..8

Test Name	FS-PC-9..10
Initial State	A graph of player plane stays still at the bottom of the window.
Input	âĂIJW and AâĂİ or âĂIJUP and LEFTâĂİ button. (Hold)
Expected Output	The player plane keeps moving up until it reach the border of the screen

Table 7: Test for FS-PC-9..10

Test Name	FS-PC-11..12
Initial State	A graph of player plane stays still at the bottom of the window.
Input	âĂIJW and DâĂİ or âĂIJUP and RIGHTâĂİ button. (Hold)
Expected Output	The player plane keeps moving up until it reach the border of the screen

Table 8: Test for FS-PC-11..12

Test Name	FS-PC-13..14
Initial State	A graph of player plane stays still at the bottom of the window.
Input	âĂIJS and AâĂİ or âĂIJDOWN and LEFTâĂİ button. (Hold)
Expected Output	The player plane keeps moving up until it reach the border of the screen

Table 9: Test for FS-PC-13..14

Test Name	FS-PC-15..16
Initial State	A graph of player plane stays still at the bottom of the window.
Input	âĂIJS and DâĂİ or âĂIJDOWN and RIGHTâĂİ button. (Hold)
Expected Output	The player plane keeps moving up until it reach the border of the screen

Table 10: Test for FS-PC-15..16

Test Name	FS-PC-17
Initial State	Game running and play is alive with full âĂIJenergyâĂİ and certain enemy planes on the field.
Input	âĂIJU" button. (Press once)
Expected Output	All the enemy planes should be destroyed instantly. How test will be performed: When âĂIJenergyâĂİ is full, player press âĂIJUâĂİ once and all enemy planes not including the boss alive are destroyed instantly. If any stay alive, there is an error.

Table 11: Test for FS-PC-17

Test Name	FS-MC-1
Initial State	The game is on the start page.
Input	âĂIJSPACEâĂİ key. (Press once)
Expected Output	The game enters the game page and the game starts. How test will be performed: When it is on the start page, press âĂIJSPACEâĂİ once, the game starts immediately. If there is any other behaviours, there is an error.

Table 12: Test for FS-MC-1

Test Name	FS-MC-2
Initial State	The game is on the death page.
Input	âĂIJQâĂİ key. (Press once)
Expected Output	The game quits immediately. How test will be performed: When the game is on death page, press âĂIJQâĂİ once to quit the game. The windows should be closed.

Table 13: Test for FS-MC-2

Test Name	FS-MC-3
Initial State	The game is at the death page.
Input	âĂIJRâĂİ key. (Press once)
Expected Output	The game restarts immediately. How test will be performed: When the game is on death page, press âĂIJRâĂİ once to restart the game. A new game should start.

Table 14: Test for FS-MC-3

Test Name	FS-PC-2
Initial State	A graph of player plane stays still at the bottom of the window.
Input	âĂIJWâĂİ or âĂIJUPâĂİ button. (Hold)
Expected Output	The player plane keeps moving up until it reach the top of the screen

Table 15: Test for FS-PC-2

Test Name	FS-PC-2
Initial State	A graph of player plane stays still at the bottom of the window.
Input	âĂIJWâĂİ or âĂIJUPâĂİ button. (Hold)
Expected Output	The player plane keeps moving up until it reach the top of the screen

Table 16: Test for FS-PC-2

6.2 System Control Testing

Test Name	FSR1
Initial State	The game on the start page.
Input	Play one round
Expected Output	Modification on high mark

Table 17: Test for FSR1

Test Name	FSR2
Initial State	The game is running and player is alive
Input	None
Expected Output	The player plane is shooting automatically

Table 18: Test for FSR2

Test Name	FSR3
Initial State	The game is running and player is alive
Input	Some operations to make the player's plane collide other enemy objects
Expected Output	Loss of life point

Table 19: Test for FSR3

Test Name	FSR4
Initial State	The game is running and player is alive
Input	Some operations making the player plane collide with gift object
Expected Output	add life point with "+" sign gift power up fire with "R" sigh gift

Table 20: Test for FSR4

7 Changes Due to Testing

7.1 GUI Testing

After conducting look and feel on GUI, the following changed are applied to differentiate the difficulty level between head boss and chramp boss:

The image of chramp boss has been modified to look more aggressive

7.2 Unit testing

After conducting Unit test, no modification need to be applied.

8 Trace to Requirements

Test	Requirements
	Functional Requirements Testing
FS-PC-1-16	FR1,FR2,FR3,FR4,FR5,FR6,FR7,FR8,FR9,FR10
FS-PC-(2...16)	FR7
FS-PC-17	FR8
FS-MC-(1...3)	FR4
	Non-functional Requirements Testing
NF-L-1	NF1,NF2,NF3,
NF-U-1	NF4, NF5, NF6,NF7,NF8,NF9,
NF-P-1	NF10, NF11, NF13,NF14,NF15,
NP-O-1	NF20, NF24, NF25,NF26,NF27
	Automated Testing
UF-1	FR9
UF-2	FR9
UF-3	FR10
UF-4	FR10
UF-5	FR8,FR10
UF-6	FR10
UF-7	FR10
UF-8	FR8
UF-9	FR5
UF-10	FR5
UF-11	FR8
UF-12	FR12
UF-13	FR9
UF-14	FR6

Table 21: Trace Between Tests and Requirements

9 Trace to Modules

Test	Modules
Functional Requirements Testing	
FS-PC-1-16	M1...M7
FS-PC-(2...16)	M1...M7
FS-PC-17	M1...M7
FS-MC-(1...3)	M1...M7
Non-functional Requirements Testing	
NF-L-1	M1...M7,
NF-U-1	M1...M7
NF-P-1	M5
NF-O-1	M1...M7
Automated Testing	
UF-1	M5
UF-2	M3
UF-3	M3
UF-4	M3
UF-5	M1
UF-6	M1
UF-7	M1
UF-8	M1
UF-9	M2
UF-10	M2
UF-11	M4
UF-12	M4
UF-13	M4
UF-14	M6

Table 22: Trace Between Tests and Modules

10 Code Coverage Metrics

The Flight Shooting Game group has managed to produce roughly 90 percent code coverage through our tests. This number is based off the fact that all of the modules have been covered in testing along with the output. Please refer to the trace to modules section. There it is clearly shown that we have covered every module multiple times, once again confirming the 90 percent coverage rate.