

Test Report

November 28, 2015

SE 2XA3

Hui Chen

chenh43

Nareshkumar Maheshkumar

maheshn

Sam Hamel

hamels2

Group E

Contents

1	Revision History	2
2	Introductions	3
2.1	Testing Methods	3
2.2	Coverage Matrix	3
3	Nonfunctional Quality Tests	3
3.1	Usability Tests	3
3.2	Performance Tests	5
3.3	Robustness Tests	5
4	System Tests	6
5	Automated Unit Tests	6
6	Changes Due to Testing	6
7	Test Summary	6
7.1	Usability Tests	6
7.2	Performance Tests	7
7.3	Robustness Tests	7
8	Traceability to Requirements	7
9	Traceability to Modules	7

List of Tables

1	Revision History Table	2
2	Coverage Matrix Table	3
3	Usability Test Results Table	4
4	Performance Tests Table	5

1 Revision History

Table 1: Revision History Table

Revision #	Date	Team Member	Description of Change
0	November 26	Hui Chen	Started Test Report document

2 Introductions

2.1 Testing Methods

2.2 Coverage Matrix

Table 2: Coverage Matrix Table

Test 1: Game Start Speed		
Requirements	Results	Automation Coverage
3.4 1)	2.5: pass	None
3.4 2)	2.8: pass	None
3.4 3)	2.1-2.4: pass	Pass
3.4 4)	2.1, 2.7: pass	pass
3.4 5)	2.1-2.4: pass	pass
3.4 7)	2.1: pass	pass
3.4 8)	2.6: pass	pass

3 Nonfunctional Quality Tests

3.1 Usability Tests

Usability Tests are performed by recruiting 3 complete beginners with no knowledge of the game as well as 3 people who are already familiar with the game. The participants were given a series of tasks to perform. The amount of time required for them to complete the task as well as comments to the task are recorded. Participants will not be named to protect their identity. Further discussion of the results can be found in section 7.1.

Table 3: Usability Test Results Table

Task 1 - Play through the game until either the player or AI wins without help.		
Participant	Task Completion (sec)	Comments
Beginner 1	79	None
Beginner 2	20	I won without even knowing how to play
Beginner 3	53	I don't understand it.
Veteran 1	278	works
Veteran 2	38	None
Verteran 3	102	None
Task 2 - Help Page		
Beginner 1	28	None
Beginner 2	22	looks too plain
Beginner 3	30	None
Veteran 1	15	Needs more in depth explanations
Veteran 2	19	Simple and concise
Verteran 3	25	None
Task 3 - Rule Page		
Beginner 1	62	Decent explanation, wish it had some strategy guide as well.
Beginner 2	55	looks simple enough
Beginner 3	94	None
Veteran 1	13	It's missing something
Veteran 2	29	pretty basic
Verteran 3	10	None
Task 4 - Play through the game until either the player or AI wins after reading help and rules.		
Beginner 1	148	Makes a little more sense now
Beginner 2	54	I think it's a little easier now that I know what each button does and what the win conditions are, but that still doesn't change the fact that I don't know how to play this game
Beginner 3	495	None
Veteran 1	18	None
Veteran 2	91	None
Verteran 3	58	None

3.2 Performance Tests

Performance tests are to be performed using JUnit and a Stopwatch library to measure the time required to execute certain actions. Each JUnit test will run 10 times and if the average is within acceptable range, then the test passes. It is important to note that the performance may vary depending on the user's hardware, so for consistency, the tests will be conducted with the same machine. Further analysis can be found in section 7.2.

Table 4: Performance Tests Table

Test 1: Game Start Speed	
Test #	Results (ms)
1	777
2	62
3	76
4	77
5	49
6	57
7	48
8	67
9	51
10	56
avg:	132.0

3.3 Robustness Tests

To test for robustness, JUnit will be used to manually trigger events in which an error is produced which should be caught and displayed to the user. The results of such tests can be found under section 5.

4 System Tests

5 Automated Unit Tests

6 Changes Due to Testing

7 Test Summary

7.1 Usability Tests

Six participants were timed and observed while they perform tasks given by us within the program. We found that the results were well received, with minor complaints about the user interface.

For the game playing portion, the participants felt that the program was easy to navigate and the elements of the game were not hard to find on the user interface. The gameplay was smooth with minor issues that we have later resolved.

The help and rule page was met with little difficulty, the participants were easily able to pull up either pages and find the basic rules they needed to play the game. Some of the participants have found that the game or the pages are too simple. While we agree that it may be simple, but the idea is to not overwhelm the player with an absurd amount of information to digest, especially not with the beginners. It is true that Texas HoldEm is a complex game, we feel that if a player is armed with the basic knowledge and continuously practice by playing the game, they can understand the game and slowly but eventually come up with strategies on their own. Of course, for more advanced tactics, they would have to do their research but this game is not about having the best and smartest opponent to face but more of something that can be used to ease the transition between beginner and a seasoned player.

The participants were overall satisfied with the current state of the game and have made suggestions toward the improvement of the user interface. Their suggestion will be taken into consideration however there are no plans to implement any changes to the user interface at the present time.

7.2 Performance Tests

The performance tests were conducted in order to determine whether the user can the program in a reasonable time frame and also to see if the program can output the required actions in reasonable time. Our tests have shown that the software passes in these aspects and we have ideas on how to optimize the performance further however there is no plan to implement the ideas at the present time as the current system have exceeded the expected results.

7.3 Robustness Tests

The tests were conducted in JUnit where all possible and foreseeable errors were manually triggered and the results recorded. We found that we were able to catch and output all errors that can occur and no game actions were performed when errors occurred, therefore keeping the system error free. During manual testing, no errors were found that are not already included in our tests.

8 Traceability to Requirements

9 Traceability to Modules