

SE 3XA3: Test Report

Big Two

Team 06, Aplus³
Jiaxin Tang and tangj63
Manyi Cheng and chengm33
Senni Tan and tans28

April 12, 2021

Contents

1	Functional Requirements Evaluation	1
2	Nonfunctional Requirements Evaluation	2
2.1	Look and Feel	2
2.2	Usability	3
2.3	Performance	4
3	Comparison to Existing Implementation	5
4	Unit Testing	5
4.1	Unit Testing for PlayerBot	5
4.1.1	BotPlayCards	5
4.1.2	BotStartingTurn	6
4.1.3	BotFreeTurn	6
4.1.4	BotSelectSingle	6
4.1.5	BotSelectPair	6
4.1.6	BotSelectFive	7
4.1.7	getAllFiveCards	7
4.1.8	getAllPairs	7
4.2	Unit Testing for Rules	7
4.2.1	newDeck	7
4.2.2	shuffle	7
4.2.3	isValidStartingPlay	8
4.2.4	isValidPlay	8
4.2.5	isValidSingle	8
4.2.6	isValidPair	8
4.2.7	isValidFiveCardPlay	8
4.2.8	isValidFlush	9
4.2.9	isValidFullHouse	9
4.2.10	isValidFourOfaKind	9
4.2.11	isStrongerPlay	9
4.2.12	isStrongerSingle	9
4.2.13	isStrongerPair	10
4.2.14	isStrongerFive	10
4.2.15	setFirstTurn	10
4.2.16	getCardsValue	10

5	Changes Due to Testing	10
6	Automated Testing	11
7	Trace to Requirements	11
8	Trace to Modules	13
9	Code Coverage Metrics	13

List of Tables

1	Revision History	ii
----------	-------------------------	-----------

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Apr 9	1.0	Initial draft
Apr 11	1.0	update

This document describes the test report generated from the Testing Plan used for BigTwo.

1 Functional Requirements Evaluation

Description of Tests: The purpose of these tests is to ensure that a user is able to use the product as specified in the Software Requirement Specification document. These tests include:

Test Name: FR-UI-1

Result: The game display a window of game of BigTwo.

Test Name: FR-UI-2

Result: The interface displays the game rule on the screen.

Test Name: FR-UI-3

Result: The interface displays a message to the game screen informing the result of the round, showing the scores of the player and dealers.

Test Name: FR-UI-4

Result: The game interface must display the interactive deck of cards the user possesses such that each card can be selected by the user upon clicking it.

Test Name: FR-UI-5

Result: The user interface display an image to simulate a real game of BigTwo.

Test Name: FR-UI-6

Result: Cards selected are removed from their deck, and displayed to the other players.

Test Name: FR-UI-7

Result: The turn goes to the next player to the right of user.

Test Name: FR-UI-8

Result: The Back of each player's card set is displayed (number cards in a card deck).

Test Name: FR-UI-9

Result: The user interface must indicate who is the current dealer by labelling the dealer's icon.

Test Name: FR-UI-10

Result: The interface should immediately return to the default window of a new game and clear any metadata from last game.

Test Name: FR-GR-11

Result: The user object will start the game with a random STARTING DECK OF CARDS, with 4 points and more, point counting rules: J=1, Q=2, K=3, A=4, 2=5, others=0.

Test Name: FR-GR-12

Result: Trick 1 starts, with the player with the Diamonds 3 being the first one to deal.

Test Name: FR-GR-13

Result: The next player in the counter-clockwise direction is the dealer.

Test Name: FR-GR-14

Result: The cards are dealt after verifying the combination.

Test Name: FR-GR-15

Result: The cards are dealt after verifying the combination, verifying the combination is higher than the one before, with the same number of cards.

Test Name: FR-GR-16

Result: The current trick is over, all cards are gathered up and a new trick will be started.

Test Name: FR-GR-17

Result: The game ends, and the scoring session begins.

2 Nonfunctional Requirements Evaluation

2.1 Look and Feel

Description of Tests: The purpose of the tests is to ensure the appearance of the game is clear. All tests were done by manual testing and observations

from testers and volunteers.

Test Name: TS-NF-L1

Result: A splash screen displayed the name of the game and the logo of the company(the group).

Test Name: TS-NF-L2

Result: After pressed the "Start" button, a new round of BigTwo started.

Test Name: TS-NF-L3

Result: The game displayed in a proper size fitting on a web.

Test Name: TS-NF-L4

Result: 100% of the volunteers said the Super Mario Themed appearance looked attractive to them.

Test Name: TS-NF-L5

Result: The game generated thirteen cards automatically for each of the four players. Each group of cards was placed on one of the four sides.

Test Name: TS-NF-L6

Result: The color of the cards are strongly different from the color of the background.

2.2 Usability

Description of Tests: The purpose of the tests is to ensure that the instructions of the game are easily accessed and realized. The tests are done manually to test all game features by using mouse clicking.

Test Name: TS-NF-U1

Result: The game successfully responded to the mouse clicking inputs that are functional.

Test Name: TS-NF-U2

Result: 100% of the volunteers completed the game easily.

Test Name: TS-NF-U3

Result: All texts in the game are shown in English.

Test Name: TS-NF-U4

Result: The screen displayed the instructions of how to play the game.

Test Name: TS-NF-A1

Result: The game is executable on the web browser.

2.3 Performance

Description of Tests: The purpose of the tests is to ensure that the game react to user's input in a reasonable time and the game is reliable.

Test Name: TS-NF-P1

Result: All responses from the game according to the input are performed in a fast manner within 1 second.

Test Name: TS-NF-P2

Result: The game displayed warning messages and refuse the invalid cards when selected invalid combination of cards.

Test Name: TS-NF-P3

Result: The game was executable on the web browser including Microsoft Edge, Firefox, Google Chrome, default built-in browsers on android phone.

Test Name: TS-NF-O1

Result: The new release of the game passed the test cases built for the previous version.

Test Name: TS-NF-C1

Result: No words or graphics were offensive to people with any culture.

Test Name: TS-NF-Le1

Result: The game did not violate any law.

Test Name: TS-NF-C2

Result: the game followed the MIT Open License.

Test Name: TS-NF-H1

Result: No gambling elements were involved in the game.

3 Comparison to Existing Implementation

Compare with the existing implementation of the web game on <http://www.onlinesologames.com/bigtwo>, our project will have similar functionalities with this web game except for the following differences:

- A timer component is integrated, the game ends when the timer reaches 0 as well.
- The user will be greeted with a rules window once the web page is loaded, instead of entering the game directly.
- The rules window will introduce player to the standard rules of big two and allows user to click "Start" button for the user to start the game.
- When the game is over, a game over window will be displayed, allowing player to see their score and play again by clicking the "Play Again" button.

Regarding testing, the existing implementation has no testing so a comparison to the the original is not possible.

4 Unit Testing

4.1 Unit Testing for PlayerBot

4.1.1 BotPlayCards

- Input: The last dealt card is a single card; the last dealt cards are a pair; the last dealt cards are a five-card combo
- Expected Output: A list of card that contains a single card stronger than the last dealt card; A list of cards that contains a pair stronger than the last dealt cards; A list of cards that contains a five-card combo stronger than the last dealt cards.
- Result: Pass

4.1.2 BotStartingTurn

- Input: A list of 13 cards contains a Diamond 3.
- Expected Output: A list that contains only a card of Diamond 3.
- Result: Pass

4.1.3 BotFreeTurn

- Input: A list of 13 cards contains only one five-card combo; a list of 13 cards contains only one pair; a list of 13 cards contains only one single card
- Expected Output: The five-card combo in the list; the pair in the list; the only single card in the list.
- Result: Pass

4.1.4 BotSelectSingle

- Input: A list of 13 cards that has at least one card greater than the last dealt card.
- Expected Output: The smallest single card in the list that is greater than the last dealt card.
- Result: Pass

4.1.5 BotSelectPair

- Input: A list of 13 cards that has at least one pair greater than the last dealt pair.
- Expected Output: The smallest pair in the list that is greater than the last dealt pair.
- Result: Pass

4.1.6 BotSelectFive

- Input: A list of 13 cards that has at least one five-card combo greater than the last dealt five-card combo.
- Expected Output: The smallest five-card combo in the list that is greater than the last dealt five-card combo.
- Result: Pass

4.1.7 getAllFiveCards

- Input: A list of 13 cards that has at least one five-card combo.
- Expected Output: All combinations of the five-card combo in the list.
- Result: Pass

4.1.8 getAllPairs

- Input: A list of 13 cards that has at least one pair.
- Expected Output: All combinations of the pair in the list.
- Result: Pass

4.2 Unit Testing for Rules

4.2.1 newDeck

- Input: None
- Expected Output: A deck of card with 52 cards.
- Result: Pass

4.2.2 shuffle

- Input: A list of cards
- Expected Output: A list of cards that the sequence is different than the original sequence.
- Result: Pass

4.2.3 isValidStartingPlay

- Input: A valid combination with a Diamond 3; a valid combination without a Diamond 3.
- Expected Output: true; false
- Result: Pass

4.2.4 isValidPlay

- Input: a valid/invalid single card; a valid/invalid pair; a valid/invalid five-card combo.
- Expected Output: true/false; true/false; true/false.
- Result: Pass

4.2.5 isValidSingle

- Input: A list contains a valid single card; a list contains a valid pair
- Expected Output: true; false
- Result: Pass

4.2.6 isValidPair

- Input: A list contains a valid pair; a list contains a valid single card
- Expected Output: true; false
- Result: Pass

4.2.7 isValidFiveCardPlay

- Input: A list contains a valid pair; a list contains a valid five-card combo
- Expected Output: false; true
- Result: Pass

4.2.8 isValidFlush

- Input: A list contains a valid pair; a list contains a valid flush
- Expected Output: false; true
- Result: Pass

4.2.9 isValidFullHouse

- Input: A list contains a valid pair; a list contains a valid full house
- Expected Output: false; true
- Result: Pass

4.2.10 isValidFourOfaKind

- Input: A list contains a valid pair; a list contains a valid Four of a Kind
- Expected Output: false, true
- Result: Pass

4.2.11 isStrongerPlay

- Input: Two single cards and one is stronger than another; Two pairs and one is stronger than another
- Expected Output: true; true
- Result: Pass

4.2.12 isStrongerSingle

- Input: Two single cards and one is stronger than another
- Expected Output: true
- Result: Pass

4.2.13 isStrongerPair

- Input: Two pairs and one is stronger than another;
- Expected Output: true
- Result: Pass

4.2.14 isStrongerFive

- Input: Two five-card combos and one is stronger than another;
- Expected Output: true
- Result: Pass

4.2.15 setFirstTurn

- Input: four card lists with only one lists contains Diamond 3
- Expected Output: the position of the list with the Diamond 3
- Result: Pass

4.2.16 getCardsValue

- Input: A single card
- Expected Output: The value of the single card
- Result: Pass

5 Changes Due to Testing

Through extensive testing on the game action buttons for the BigTwo game, it was discovered that the Timer would freeze when a user keeps performing action on the buttons, causing the parent component Game to re-render non-stop, which would lead to child component Timer to stop its useEffect due to React implementation. React will remember the useEffect, and call it later after performing the DOM updates. So useEffect runs both after the first render and after every re-render by default. The problem was resolved

by customizing the Timer useEffect hook to skip applying an effect if certain values haven't changed between re-renders. To do so, we pass the array [props.addSeconds, minutes, seconds] as an optional second argument to useEffect.

6 Automated Testing

While majority of the tests for this project were done manually, we have our unit testing done automatically with Jest and we also did the following test automatically.

Test Name: FR-GR-12

Initial State: Custom in-game state.

Input: The method startGame() is called.

Output: Pass/Fail Procedure: A random STARTING DECK OF CARDS of 52 unique cards is created, with 4 points and more, point counting rules: J=1, Q=2, K=3, A=4, 2=5, others=0.

Result: Pass

Test Name: FR-GR-16

Initial State: Custom in-game state, a trick has started.

Input: Player selects a valid combinations of cards to deal.

Output: Pass/Fail

Procedure: Confirm that the cards are only dealt after verifying the combination validity, verifying the combination is higher than the one before, with the same number of cards.

Result: Pass

7 Trace to Requirements

Test-ID	Requirements
Functional Requirements Testing	
FR-UI-1	FR1

FR-UI-2	FR2
FR-UI-3	FR3
FR-UI-4	FR4
FR-UI-5	FR5
FR-UI-6	FR6
FR-UI-7	FR7
FR-UI-8	FR8
FR-UI-9	FR9
FR-UI-10	FR10
FR-GR-11	FR11
FR-GR-12	FR12
FR-GR-13	FR13
FR-GR-14	FR14
FR-GR-15	FR15
FR-GR-16	FR15, FR16
FR-GR-17	FR15, FR17

Non-functional Requirements Testing

TS-NF-L1	3.1.1-1
TS-NF-L2	3.1.1-2
TS-NF-L3	3.1.1-3
TS-NF-L4	3.1.1-4
TS-NF-L5	3.1.2-1
TS-NF-L6	3.1.2-2
TS-NF-U1	3.2.1-1
TS-NF-U2	3.2.1-2
TS-NF-U3	3.2.2-1
TS-NF-U4	3.2.3-2
TS-NF-A1	3.3
TS-NF-P1	3.4.1
TS-NF-P2	3.4.3
TS-NF-P3	3.4.8
TS-NF-O1	3.5.4
TS-NF-C1	3.8
TS-NF-Le1	3.9.1
TS-NF-C2	3.9.3

TS-NF-H1	3.10.2
Automated Testing	
FR-GR-12	FR12
FR-GR-16	FR15, FR16

8 Trace to Modules

Modules	Tests
PlayerBot	4.1.1-4.1.8
Rules	4.2.1-4.2.16

9 Code Coverage Metrics

The BigTwo project has managed to produce code coverage of approximately 85-90% the entire project. This was determined by performing unit tests on our 9 internal modules, which had unit test cases written for the functionality of each module. The remaining modules, HTML files, and CSS files were covered in the manual testing suite, where the BigTwo game was played and simulated by multiple testers. Through integration and manual testing, the behavior and results of the game were observed for both the user/player and the dealer, and how they interacted with the button actions and game environment.

References