Checkers

Test Case Document

Version 1.2

March 04, 2018

Group Members	Chetan K Parakh, Cody Moser, Nicholas Nugent, Saugat Sthapit
Submitted To	Dr. Filippos Vokolos, Ph. D.

Table of Contents

1. Introduction

The purpose of this document is to describes the testing approaches used to validate and verify the outputs of Checkers game functionality as outlined in the requirements document v.0.1. Checkers is designed to be a web based multiplayer online game.

1.1. Definitions, Acronyms and Abbreviations

Please refer to Section 5 Appendix for the glossary for definitions, acronyms and abbreviations.

1.2. References

This documents contains terms and references that can be found in the preceding design document and the requirements document.

1.3. Types of Testing

Since this pertains more to GUI, we did web automation testing using Selenium Webdriver. For unit and integration testing, Chai was be used.

1.4 Code Coverage and Static Analysis Tools

IstanbulJs is the powerful tool built on top of Mocha which shows the code coverage and EsLint is used as the code analyzer.

2. Testing Environments

2.1. Windows based environment

Machine Name: Windows PC

Operating System: Windows 10

Interpreter Platform: Node.Js v6.9.5

Java JDK: 1.8.0_111

Browser:

- Google Chrome : Version 64.0.3282.186 (64 bit)

Number of clients: 2

Tester Name: Saugat Sthapit

State: PASS

2.2. Unix based environment

Machine Name: Apple MacBook Pro

Operating System: MacOS Sierra

Interpreter Platform: Node.Js v6.9.5

Java JDK: 1.8.0_111

Browser:

- Google Chrome : Version 64.0.3282.186 (64 bit)

Number of clients: 2

Tester Name: Saugat Sthapit

State: PASS

3. Setup Information and Prerequisites

Setup Information:

- Program and its associated test cases can be run by navigating to the correct address which is localhost:8080 in a web browser. Chrome and ChromeDriver binaries should be installed along with Gradle.

Following **conditions** should be met prior to running the application:

- Broadband internet connection for optimal performance.
- Hosting server must be up and be functioning properly.
- Updated version of server, dependencies and the packages installed.

4. Test Cases

4.1. Initialize Game

4.1.1. Description

This case states steps required to connect two players and start the game

4.1.2. Pre-Conditions for the Test Case

An internet connection, web app supported browser installed on the system, Node.js (server) is up and running

4.1.3. Scenario

ID	REQ	Description	Execution Steps	Expected	Actual
A1	R-17,18,31	Visit Homepage	Open a supported browser and visit localhost:8080	Home Screen displayed. Home screen contains buttons to start game and view game rules	Successful
A2	R-1,2,3,4,19, 32,33	First player connection	Connect to the http port and select start game button	Waiting Screen/info displayed	Successful
A3	R-1,2,3,4,5,6, 20,32,33	Second player connection	Connect to http port and select start game button after another player already has	Goes to game screen	Successful
A4	R-22,32,33	First player	Connect and	Goes to	Successful

		wait screen goes to game screen after second player connects	select start game. Wait for another player to connect and start game.	game screen	
A5	R-21,34	Third user must wait to play	Connect to server after two players already started	Player is informed they must wait	Not implemented
A6	R-23	First player cancel on waiting screen	Connect and start game as only player. Cancel on waiting screen	Removed from waiting	Successful

4.2. Game Play and Move Validation

4.2.1. Description

This case consists of testing steps required to play checkers by host and opponents. The test cases validate that the algorithm for the game play does not contradict with the rule of the checkers games as stated by The American Checkers Federation.

4.2.2. Pre-Conditions for the Test Case

An internet connection, web app supported browser installed on the system, Node.js (server) is up and running, two clients are connected to play the game.

4.2.3. Scenario

ID	REQ	Description	Execution Steps	Expected	Actual
B1	R-5,27	Pawn piece selected	1. Click on a pawn piece	Selected pawn piece is highlighted.	Successful
B2	R-5,27	Pawn piece deselected	1. Click on a pawn piece. 2. Click on the same pawn piece again	Highlight from the selected piece is removed.	Successful
B3	R-5,8, 27	Possible move location displayed	1. Click on a piece with one or more possible move location	The checker boxes where the selected piece can be move gets highlighted.	Successful
B4	R-5,27	Possible move location (checker box) clicked	1. Find a piece with one or more possible move location. 2. Click on that piece 3. Click on a highlighted checker box.	The selected piece move to the possible move location. (checker box)	Successful
B5	R-5,27	Pawn piece moving forward	1. Find a pawn piece with one or more possible forward move location. 2. Click on that piece 3. Click on one of the available highlighted checker box in forward direction	Checker piece moves to the highlighted checker box. The highlight is removed from the checker box	Successful
B6	R-5,27	Pawn piece capturing	1. Find a pawn piece that can move	-Pawn piece move to a	Successful

		opponent	diagonally forward over opponent piece. 2. Select the pawn piece 3. Move the pawn piece diagonally forward over opponent piece	new location. - Opponent’s piece is removed from the board. - Score is updated	
B7	R-5,27	Player tries to move his checker piece to an invalid checker box	1. Click on a pawn piece. 2. Click on any unhighlighted checker box.	No changes in game state.	Successful
B8	R-5,27	Pawn piece turning to king piece on reaching other end of the board	1. Play the game until one of the piece reached the other end of the board.	Pawn piece changes to the king piece.	Successful
B9	R-12	Updating score after capturing checker piece	1. Move a piece over opponents piece.	-Pawn piece move to the final location. - Opponent’s piece is removed from the board. - Score is updated	Successful
B10	R-5,27	King piece moving forward	1. Play the game until one of the piece reached the other end of the board. 2. Select the king piece. 3. Click on one of the possible move location box available in forward direction	- King piece disappears from its old checker box location - King piece appears in the selected checker box.	Successful
B11	R-5,27	King piece	1. Play the game until	- King piece	Successful

		moving backward	one of the piece reached the other end of the board. 2. Select the king piece with a possible move location in backward direction. 3. Click on the possible move location box available in backward direction of the king piece.	disappears from its old checker box location - King piece appears in the selected checker box.	
B12	R-5,27	King piece jumping forward	1. Play the game until a King piece can move forward diagonally over an opponent's piece. 2. Select the King piece. 3. Move the King piece diagonally forward over the enemy piece by clicking on that box.	-King piece is moved to the new location. - Opponent's piece is removed from the board. - Score is updated	Successful
B13	R-5,27	King piece jumping backward	1. Play the game until a King piece can move backward diagonally over an opponent's piece. 2. Select the King piece. 3. Move the King piece diagonally backward over the enemy piece by clicking on that box.	-King piece is moved to the new location. - Opponent's piece is removed from the board. - Score is updated	Successful
B14	R-5,27	Multiple jumps for pawn piece	1. Play the game until a pawn piece can make multiple forward diagonal moves over opponent pieces. 2. Select the pawn piece. 3. Move the pawn piece	-Pawn piece is moved to the new location. - Opponent's pieces are removed from the	Successful

			diagonally forward over the enemy pieces by clicking on that box.	board. - Score is updated	
B15	R-5,27	Multiple jumps for king piece	1. Play the game until a King piece can make multiple forward diagonal moves over opponent pieces. 2. Select the King piece. 3. Move the King piece diagonally forward over the enemy pieces by clicking on that box.	-King piece is moved to the new location. - Opponent's pieces are removed from the board. - Score is updated	Successful
B16	R-6,7,24,35,36,38	Player turn indicated	1. Start game. 2. Make a move. 3. Look to see that players have different colored pieces	Current players turn is shown and updated	Successful
B17	R-27,35,36,37	Player can make a move when it is their turn	1. Start game 2. Try to make a move on your turn	Able to make a move	Successful
B18	R-26,35	Player can not make a move when it is not their turn	1. Start game 2. Try to make a move with player when it is not their turn	Player can not make a move	Successful
B19	R-28,35,36,37,38	Player can see opponents move after they played it	1. Start game 2. Make a move	Both players can see the updated game board	Successful
B20	R-28,35,36,37,38	Player can make a move after other player's turn has ended	1. Start game 2. Make a move with one player. 3. Try to make a move with other player	Other player will see first players move and will be able to make a move	Successful

4.3 Ending the Game

4.3.1 Description

These cases describe the testing for when a game ends, whether by the normal gameplay ending, a player quitting, or a disconnection occuring.

4.3.2. Pre-Conditions for the Test Case

An internet connection, web app supported browser installed on the system, Node.js (server) is up and running, two clients are connected and playing the game.

4.3.3 Scenario

ID	REQ	Description	Execution Steps	Expected	Actual
C1	R-29,30	Game Ends in Win/Loss	1. Play the game until all pieces from one of the players have been taken.	- Both clients will be brought to the game over page - The winning client will have a message saying they won. The losing client will have a message saying they lost	Successful
C2	R-29,30	Game Ends in a Tie Because No Pieces have been removed in 40 turns	1. Play the game until both players are left with 1 king and pawn. 2. Move each player’s king piece so that no pieces are captured. 3. Move each player’s pawn forward so that	- Both clients will be brought to the game over page. - Both clients will have a message saying that the game ended in a draw.	Successful

			the other draw case doesn’t happen. 4. Repeat step two for the rest of the 38 moves.		
C3	R-29,30	Game Ends in a Tie Because neither player has advanced a non-crowned piece towards the king-row in 40 moves	1. Play the game until each player has solely 1 king on the board. 2. Move each player’s king so that neither one is captured for 40 turns.	- Both clients will be brought to the game over page. - Both clients will have a message saying that the game ended in a draw.	Successful
C4	R-25,30	Game ends by player resigning	1. Connect two clients to a game. 2. Have one client press the resign button.	- Both clients will be brought to the game over page - The winning client will have a message saying they won due to resignation. The losing client will have a message saying they lost due to resignation.	Successful
C5	R-30	Game ends from a disconnection	1. Connect two players to the game. 2. Disconnect one client from its internet connection.	- The client that stayed connected will be brought to the game over page. - The client that stayed connected will get the message that they won due to a disconnection	Successful

5. Appendix

5.1 Glossary

Backward - Direction a king piece can move

Forward - Direction that both a king and non-king piece can move

GUI - Graphical User Interface

Host: The player hosting the game. The player who launches chess application first and receives the unique game ID code from the application.