

# Software Test Case Document

## **Checkers**

Version 1.0

Created by:

Dustin Nguon  
Mariah Sheehan  
Mariana Quinde Garcia  
Stevie Parris

# Revision History

Name	Date	Reason for Change	Version	Editor
Dustin Nguon Mariah Sheehan Mariana Quinde Garcia Stevie Parris	August 10, 2016	Initial Version	1.0	Dustin Nguon, Mariah Sheehan, Mariana Quinde Garcia, Stevie Parris

# Table of Contents

## 1. [Introduction](#)

[1.1 Definitions, Acronyms and Abbreviations](#)

[1.2 References](#)

## 2. [Testing Environments](#)

[2.1 Environment 1: Windows](#)

[2.2 Environment 2 : Mac OSX](#)

## 3. [Setup Information and Prerequisites](#)

## 4. [Test Cases](#)

[4.1 Test Cases 1: Initialize Game](#)

[4.1.1 Description](#)

[4.1.2 Precondition](#)

[4.1.3 Scenario](#)

[4.2 Test Cases 2: Web Navigation](#)

[4.2.1 Description](#)

[4.2.2 Preconditions](#)

[4.2.3 Scenario](#)

[4.3 Test Cases 3 Play Game](#)

[4.3.1 Description](#)

[4.3.2 Preconditions](#)

[4.3.3 Scenario](#)

[4.4 Test Cases 4: Ending Game](#)

[4.4.1 Description](#)

[4.4.2 Preconditions](#)

[4.4.3 Scenario](#)

## **5. [Appendix](#)**

[5.1 Glossary](#)

# 1. Introduction

The purpose of this document is to describe the testing approaches used while evaluating the functionality and performance of **Checkers** program as to meet the requirements outlined in the requirements document. **Checkers** is a node/javascript application that allows two users to play a game of checkers from remote locations via connecting to a web server.

## 1.1 Definitions, Acronyms and Abbreviations

Please refer to the Appendix and Glossary sections for any definitions and abbreviations.

## 1.2 References

The document may feature terms and references which can be found in the preceding requirements and design documents related to **Checkers** program.

## 2. Testing Environments

The program and associated test cases have been run within the following test environments.

### 2.1 Environment 1: Windows

Machine Name	Windows PC	DB Directory		N/A		
OS and Version	Windows 8.1; 8 GB RAM; 1 TB HDD	Interpreter Platform	Google Chrome	Client	Google Chrome	
				Server	Node JS	
Tester Name	Dustin Nguon	Test Date		07/13/2016		
New Log	If necessary, list the new log after tests have been run.	State		Complete		

### 2.2 Environment 2: Mac OSX

Machine Name	Mac OSX	DB Directory		N/A	
OS and Version	Max OSX El Capitan 10.11.2 16 GB RAM; 500 GB SSD	Interpreter Platform	Google Chrome	Client	Google Chrome
				Server	Node JS
Tester Name	Stevie Parris	Test Date		07/13/2016	
New Log	If necessary, list the new log after tests have been run.	State		Complete	

### 3. Setup Information and Prerequisites

Prior to running the program the following prerequisites must be met.

- The program and associated test cases can be run by launching the program. The program features a web GUI and is not meant for interaction through a console or terminal program.
- Program url are not particularly specific to a platform and can be launched on most platforms as long as a working version of Chrome is installed on the system.
- An internet connection (broadband recommended) is required for the program.
- Tux server along with the associated checkers server program must be up and functioning properly, without any glitches.

## 4. Test Cases

The purpose of this document is to describe the testing approaches used while evaluating the functionality and performance of **Checkers** program as to meet the requirements outlined in the requirements document. **Checkers** is a web application that allows two users to play a game of checkers from remote locations via connecting to a web server.

### 4.1 Test Cases 1: Initialize Game

#### 4.1.1 Description

The case consists of covering the steps required to host a game on a computer.

#### 4.1.2 Preconditions for this test case

An internet connection, Java runtime environment installed on the system, Tux (Node server) is up and running.

#### 4.1.3 Scenario

Test Case					
ID	Req	Description	Execution Steps	Expected Result	Actual Result
A1.	1.4.1	Launch App	1. Launch application by navigating to url	Start window	As expected
A2.	1.2, 1.3	Start Game	1. Click 'Start'	Launches or joins a game	As expected



## 4.2 Test Case 2: Web Navigation

### 4.2.1 Description

The case consists of covering (testing) the steps involved in navigating the web application and make sure all elements display correctly.

### 4.2.2 Preconditions for this test case

An internet connection, Chrome on the system, Tux (server) is up and running, a the home page is on.

### 4.2.3 Scenario

Test Case					
ID	Req	Description	Execution Steps	Expected Result	Actual Result
D1.	5.1.1	Navigation Header	1. Navigate through all the following pages and make sure the navigation header displays correctly and links works: 1.1. About Page 1.2. Instructions Page 1.3. Game Page 1.4. Results Page	The navigation header appears on the top of all pages with the Home, New Game, Instructions, Player's Turn (only on game pages) links appear and work.	As expected
D2.	5.1.2	Page Footer	1. Navigate through all the following pages and make sure the footer displays correctly and links works: 1.1. Home Page 1.2. About Page 1.3. Instructions Page 1.4. Game Page 1.5. Results Page	The footer appears on the bottom of all pages with the About and Instructions links appear and work.	As expected
D3.	5.1.3	Home Page	1. Start application (links and buttons tested in other cases)	When game is up the Home Page displays and looks like the prototype	As expected
D4.	5.1.4	About Page	1. Open About page through footer link	The About page displays correctly (as shown in prototype)	As expected
D5.	5.1.5	Instructions Page	1. Open Instructions page through footer link or header link.	The Instructions page displays correctly (as shown in prototype)	As expected
D6	5.1.6	Game Page	1. Open Game page	The Game page displays	As expected

			though start button on Home page or header New Game link	correctly (as shown in prototype)	
D7	5.1.7	Results Page	1. Finish a game and get to the New Page Game	The New Page game displays correctly (as shown in prototype)	As expected

## 4.3 Test Cases 3: Play the Game

### 4.3.1 Description

The case consists of covering (testing) the steps involved in mutual play of the game by the host and the opponent. The test case essentially validates if the logic behind the moves has been correctly implemented and follows the rules of the game as set forth by **USA Checkers**.

### 4.3.2 Preconditions for this test case

An internet connection, Chrome installed on the system, Tux (Node server) is up and running, a game is already setup and active between the host and the opponent.

### 4.3.3 Scenario

Test Case					
ID	Req	Description	Execution Steps	Expected Result	Actual Result
B1.	3.1	Can only click on owned pieces and when turn player.	1. Click on anywhere on the gameboard except for an owned piece.	Nothing happens	As expected
B2.	3.2	Man Forward Movement	1. Click on Man with a free space diagonally adjacent and forward left or right 2. Click on a free space that is diagonally adjacent and forward left or right to the Man	Man is erased from initial space and placed into new chosen space.	As expected
B3.	3.2	King Forward Movement	1. Click on King with a free space diagonally adjacent	King is erased from initial space and placed	As expected

			and forward left or right to it 2. Click on a free space that is diagonally adjacent and forward left or right to the King	into new chosen space.	
B4.	3.2	King Backward Movement	1. Click on King with a free space diagonally adjacent and backward left or right 2. Click on a free space that is diagonally adjacent and backward left or right the King	King is erased from the initial space and placed into the new chosen space.	As expected
B5.	3.2	Capturing a piece using a Man.	1. Click on a Man with an enemy piece that is diagonally adjacent and forward left or right and an empty space adjacent to the enemy piece on the same diagonal. 2. Click on the free space that is adjacent to the enemy piece and along the same diagonal.	The enemy piece is erased. The player piece is erased and moved to the new chosen space.	As expected
B6.	3.2	Capturing a piece with a King.	1. Click on a king with an enemy piece that is diagonally adjacent along with an empty space adjacent to the enemy piece on the same diagonal. 2. Click on the free space that is adjacent to the enemy piece and along the same diagonal.	The enemy piece is erased. The player piece is erased and moved to the new chosen space.	As expected
B7.	3.3	Ending turn on Man movement	1. Perform steps in B2 and observe B2's expected result.	The UI updates to show that it is now the other player's turn.	As expected
B8.	3.3	Ending turn on King forward movement	1. Perform steps in B3 and observe B3's expected result.	The UI updates to show that it is now the other player's turn.	As expected
B9.	3.3	Ending turn on King backward movement	1. Perform steps in B4 and observe B4's expected result.	The UI updates to show that it is now the other player's turn.	As expected
B10.	3.3	Ending turn on capture with a Man	1. Perform steps in B5 without landing in a space to capture another piece and observe B5's expected result.	The UI updates to show that it is now the other player's turn.	As expected

B11.	3.3	Ending turn on capture with a King	1. Perform steps in B6 without landing in a space to capture another piece and observe B6's expected result.	The UI updates to show that it is now the other player's turn.	As expected
B12.	5.6.2.2	Chain capture	1. Perform a capture that places the piece in position to capture another enemy piece.	The player's turn does not end and they are required to legally capture all pieces using the piece that was moved initially.	As expected
B3.	6.2.3	Crowning	1. Legally move a Man to the opposite end of the board.	B2's expectation except a king will appear on the chosen space.	As expected
B14.	5.6.2.1	Invalid moves of a Man	1. Click on a Man piece. 2. Click on any space that is adjacent and diagonal and occupied.	Nothing	As expected
B15.	5.6.6.2	Invalid moves of a king	1. Click on a King 2. Click on any space that is not diagonally adjacent	Nothing	As expected

## 4.4 Test Case 4: Ending the game

### 4.4.1 Description

The case consists of covering (testing) the steps involved in resigning, drawing, or winning a game from an active gaming session.

### 4.4.2 Preconditions for this test case

An internet connection, Chrome on the system, Tux (Node server) is up and running, a game is already setup and active between the host and the opponent.

### 4.4.3 Scenario

Test Case					
ID	Req	Description	Execution Steps	Expected Result	Actual Result
C1.	3.4	Resign Game	1. Either player exits the browser/closes tab.	A Results Page displays saying "Game Lost"  **A "resigned" message is displayed to the user and game ends. Returns to home screen.	As expected
C2.	3.4	Player Wins	1. Player makes a move that removes opponent's final piece	A Results Page displays saying "Game Won"  **A "winner" message is displayed to the player and the game ends. Returns to home screen.	As expected
C3.	3.4	Players Loses	1. Opponent makes a move that removes player's final piece	A Results Page displays saying "Game Lost"  **A "loser" message is displayed to the player and the game ends. Returns to home screen.	As expected
C4.	3.4	Three Repeated Move Draw	1. Players move pieces such that the board's configuration is the same three times in 6 moves	A Results Page displays saying "Game Tied"  **A "draw" message is displayed to the player and the game ends. Returns to home screen.	As expected
C5.	3.4	Uncrowned	1. Neither player has	A Results Page displays	As expected

		Movement, no piece removal draw	advanced an uncrowned piece in their previous 40 moves and no piece has been removed the board in their previous 40 moves.	saying “Game Tied”  **A “draw” message is displayed to the player and the game ends. Returns to home screen.	
--	--	------------------------------------	--	--	--

## 5. Appendix

### 5.1 Glossary

**Guest/Opponent** : The *other* player willing to join the game.

**Man**: A checkers piece that is not a king.

**Tux** : Servers available to computer science students at Drexel University. Tux servers will be used for running game sessions.

**USACheckers** : Organization that sets the rules of checkers for international play. Refer to their website (<http://www.usacheckers.com/>) for the complete set of rules which are the same rules this application should follow.