Checkers Game

Software Test Case Document

|  |  |
| --- | --- |
| **Group Members** | Anh Nguyen  Debdut Karmakar  Matthew Fuller  Zhuang Zijian |
| **Advisor** | Dr. Filippos Vokolos |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Change** | **Revision** |
| Anh Nguyen, Debdut Karmakar, Matthew Fuller, Zhuang Zijian | 08/20/2017 | Finalize the first version. | 1.0 |
| Anh Nguyen | 08/19/2017 | First Draft | 0.9 |

Contents

[1 Revision History 2](#_Toc491036055)

[2 Introduction 4](#_Toc491036056)

[2.1 Definition 4](#_Toc491036057)

[2.2 References 4](#_Toc491036058)

[3 Testing Environment 4](#_Toc491036059)

[3.1 Environment 1: Windows 10 4](#_Toc491036060)

[4 Setup Information and Prerequisites 5](#_Toc491036061)

[5 Test Cases 5](#_Toc491036062)

[5.1 Host Game 5](#_Toc491036063)

[5.1.1 Description 5](#_Toc491036064)

[5.1.2 Preconditions 5](#_Toc491036065)

[5.1.3 Scenario 5](#_Toc491036066)

[5.2 Join Game 5](#_Toc491036067)

[5.2.1 Description 5](#_Toc491036068)

[5.2.2 Preconditions 5](#_Toc491036069)

[5.2.3 Scenario 5](#_Toc491036070)

[5.3 Play Game 6](#_Toc491036071)

[5.3.1 Description 6](#_Toc491036072)

[5.3.2 Preconditions 6](#_Toc491036073)

[5.3.3 Scenario 6](#_Toc491036074)

[5.4 End Game 6](#_Toc491036075)

[5.4.1 Description 6](#_Toc491036076)

[5.4.2 Preconditions 7](#_Toc491036077)

[5.4.3 Scenario 7](#_Toc491036078)

[6 Appendix 8](#_Toc491036079)

[6.1 Glossary 8](#_Toc491036080)

# Introduction

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

## Definition

## References

This document may use terms that can be found in the requirement and design documents of the program.

# Testing Environment

## Environment 1: Windows 10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Machine Name** | Windows PC | **DB Directory** | N/A | | |
| **OS and Version** | Windows 10 Home;  16GB RAM | **Interpreter Platform** | Java 1.8.0 | **Client Server/ Back-end** | N/A |
| **Tester Name** | Anh Nguyen/Zijian Zhuang | | **Test Date** | 8/29/17 | |
| **New Log** | If necessary, list the new log after test has been run. | | | **State** | Passed |

3.2 Environment 2: macOS Sierra

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Machine Name** | MacBook Pro | **DB Directory** | N/A | | |
| **OS and Version** | macOS Sierra 10.12.6;  16GB RAM | **Interpreter Platform** | Java 1.8.0 | **Client Server/ Back-end** | N/A |
| **Tester Name** | Matt Fuller | | **Test Date** | N/A | |
| **New Log** | If necessary, list the new log after test has been run. | | | **State** | N/A |

3.3 Environment 2: Ubuntu 16.04

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Machine Name** | ThinkPad T440s | **DB Directory** | N/A | | |
| **OS and Version** | Ubuntu 16.04;  12GB RAM | **Interpreter Platform** | Java 1.8.0 | **Client Server/ Back-end** | N/A |
| **Tester Name** | Debdut Karmakar | | **Test Date** | 8/29/17 | |
| **New Log** | If necessary, list the new log after test has been run. | | | **State** | Passed |

# 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 GUI and is not meant for interaction through a console or terminal program.
* Program binaries specific to a platform can be launched on that platform if a working **Java Runtime Environment (JRE)** version 7 or higher installed on the system.
* An Internet/LAN connection is required.

# Test Cases

## Host Game

### Description

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

### Preconditions

Internet/LAN connection. **Java Runtime Environment (JRE)** installed on system.

### Scenario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | | | | | |
| **ID** | **Requirement** | **Description** | **Execution Steps** | **Expected Result** | **Actual Result** |
| A1 |  | Launch Application |  | GUI window showing main menu | Terminal ask to choose to host or join. |
| A2 | 4.1.1 | Host Game | Click “Host Game” Button | GUI show a dialog with IP address of the host and notice that it’s waiting for a player to connect. | Input 1 to terminal to host a game. IP address and port number is shown after. |
| A3 | R4.2.1 | A player connected |  | A new game starts. GUI displays the initial checker board. | A new game starts. GUI displays the initial checker board. |

## Join Game

### Description

The case consists of covering steps required to join a game.

### Preconditions

Internet/LAN Connection. **Java Runtime Environment (JRE)** installed on system.

### Scenario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | | | | | |
| **ID** | **Requirement** | **Description** | **Execution Steps** | **Expected Result** | **Actual Result** |
| B1 | R4.1.2 | Join Game | Click “Join Game” Button. Enter the IP address and press Enter. | A dialog shows up to ask for the IP address. | Terminal ask for IP address and port input. |
| B2 | R4.1.3 | Join game successfully |  | A new game starts. GUI displays the initial checker board. | A new game starts. GUI displays the initial checker board. |
| B3 | R4.1.4 | Join game failed – unable to connect |  | A dialog shows “Unable to connect to host.” GUI is back to main menu. | Error displays. Application exits. |
| B4 | R4.1.5 | Join game failed – host is busy |  | A dialog shows “Host is busy. Try again later.” GUI is back to main menu. | Error displays. Application exits. |

## Play Game

### Description

The case consists of covering (testing) the steps involved in mutual play of the game by both players. The test case essentially validates if the logic behind the moves has been correctly implemented and follows the rules of the game as dictated by World Checkers Draught Federation.

### Preconditions

Internet/LAN Connection. **Java Runtime Environment (JRE)** installed on system. A game already started.

### Scenario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | | | | | |
| **ID** | **Requirement** | **Description** | **Execution Steps** | **Expected Result** | **Actual Result** |
| C1 | R4.2.8 | Making a valid move | 1. Select a piece that is yours 2. Select a destination | The piece moves to the destination. | The piece moves to the destination. |
| C2 | R4.2.8 | Making an invalid move | 1. Select a piece that is yours 2. Select a destination | A prompt appears informing the user that the move is not valid and that they should select another space | The piece doesn’t move to new location. |

## End Game

### Description

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

### Preconditions

An Internet/LAN connection, **Java Runtime Environment (JRE)** installed on the system, a game is already setup and active.

### Scenario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | | | | | |
| **ID** | **Requirement** | **Description** | **Execution Steps** | **Expected Result** | **Actual Result** |
| D1 | 6.1 | Resign Game | 1. Host a session with host player and receive game ID. 2. Opponent joins session of client 1. 3. One of the Host or Opponent exits the application. | A “resigned” message is displayed to the Host if the Opponent resigns and vice-versa | Not implemented. |
| D2 | 4.2.12 | Win Game | 1. Host a session with host player and receive game ID. 2. Opponent joins session of client 1. 3. Players play the game until one of them wins and other loses by not having any moves left. 4. The Game results should be displayed properly to the winner and loser. | When a game ends, the application should prompt both parties if they want a rematch. If both parties want a rematch, the application should initialize a new game. | The winner is displayed. No rematch prompt is shown. |
| D3 | 4.2.12 | Lose Game | 1. Host a session with host player and receive game ID. 2. Opponent joins session of client 1. 3. Players play the game until one of them wins and other loses by not having any moves left. 4. The Game results should be displayed properly to the winner and loser. | When a game ends, the application should prompt both parties if they want a rematch. If both parties want a rematch, the application should initialize a new game. | The winner is displayed. No rematch prompt is shown. |

# Appendix

6.1 Glossary

**Checkers/Draughts** is a strategy board game for two players which involve diagonal moves of uniform game pieces and mandatory captures by jumping over opponent pieces

**Guest/Opponent** ­ The other player willing to join the game using the ID code provided by the Host player.

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

**World Checkers Draught Federation** A global organization of checkers that also holds the World Checkers Championship.