Test Plan

for

Game of Checkers

Version 1.0 approved

Prepared by Natalie Ownby, Anna Cowsar, Cassandra Coyle, Niraj Regmee, and Alexander Rodriguez

Texas State University, CS 3398

October 4, 2018

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
0.1	Natalie Ownby	Initial requirements to be tested	10/10/18
0.2	Natalie Ownby	Formatted test cases and added test cases for additional functionality	10/26/18

1 Introduction

The “Game of Checkers” project includes designing and implementing software that allows a user to play a game of checkers against an AI. The test plan document outlines all types of testing that will be completed including specific requirements that will be tested.

1.1 Test Plan Objectives

This document outlines the tests that will be used to verify and validate the code for "Game of Checkers". This will include some unit testing as well as testing the functional and non-functional requirements outlined in the SRS.

2.1 Test Strategy

2.1.1 System Test

System tests will ensure that the software allows the user to play a game of checkers following all rules outlined in the SRS document. Detailed tests are outlined in section 4. Success in all section 4 tests will indicate a successful system.

2.1.2 Stress/Performance Test

No stress/performance testing is required.

2.1.3 Security Test

No security testing is required

2.1.4 Automated Test

No automated testing is required.

2.1.5 Recovery Test

No recovery testing is required.

2.1.6 Documentation Test

Documentation testing will ensure all requirements and design aspects outlined in the SRS and SDD are met.

2.1.7 Beta Test

No Beta testing is required.

2.1.8 User Acceptance Test

User acceptance tests will be performed by the developers to ensure the code meets all requirements outlined in the documentation. This stage of testing will include playing multiple games of checkers and noting any defects in the software.

3.1 Environment Requirements

3.1 Environment

Testing will ensure the program operates in a web browser. This will be done by cloning the github file and opening the index.html file in a Internet Explorer, Google Chrome and Firefox web browser.

4 Functions To Be Tested

4.1 Game Play

4.1	**REQ 1**
Condition	The user has started the game and it is the users turn
Action	User clicks on one of their pieces
Expected Result	Alert shows location of selected piece
Actual Result	Pass

4.1	**REQ 2**
Condition	User has started the game. It is the users turn.
Action	The user selects their piece and the tile they want to move it to (a valid tile for a valid move)
Expected Result	Piece is moved to the selected tile. The user interface is updated to show this move.
Actual Result	Pass

4.1	**REQ 3**
Condition	The user has started the game. It is the users turn.

Action	The user will click on a piece and click on an invalid tile the piece cannot be moved to.
Expected Result	The piece is not moved to the invalid location. The user interface does not change.
Actual Result	Pass

4.2 AI Implementation

4.2	**REQ 1**
Condition	It is the AI's turn to move a piece.
Action	User takes no action
Expected Result	AI selects and moves piece to a valid spot on the board. The user interface is updated.
Actual Result	N/A

4.2	**REQ 2**
Condition	It is the users turn.
Action	User takes no action
Expected Result	The AI does not move a piece
Actual Result	N/A

4.2	**REQ 3**
Condition	Through the duration of the game
Action	User plays checkers
Expected Result	The AI follows the rules of checkers
Actual Result	N/A

4.3 Rule Enforcement

4.3	**REQ 1**
Condition	It is the users turn. The piece is not a king and not able to jump.
Action	The tester clicks on a piece and clicks on an invalid space to move it to. This is repeated for all non-king pieces and all invalid spaces they can move to.

Expected Result	The piece is not moved to the selected location.
Actual Result	Pass

4.3	**REQ 2**
Condition	It is the user’s turn and the user can jump a piece
Action	The user jumps a piece
Expected Result	The user’s piece is moved to the correct location and the jumped piece is removed from the board.
Actual Result	Pass

4.3	**REQ 3**
Condition	User can move their piece to a space the far side of the board.
Action	User moves their piece to the spot on the opposite side of the board.
Expected Result	Piece is “kinged” and the user interface is updated to show the king symbol on the piece (white piece turns yellow, green piece lighter green).
Actual Result	Pass

4.3	**REQ 4**
Condition	The user has a piece that has been kinged.
Action	User selects the king piece and selects a tile that is backwards diagonal
Expected Result	The king piece is moved to the backwards diagonal space.
Actual Result	Pass

4.3	**REQ 5**
Condition	The user has one move left to remove the final opponent piece from the board.
Action	User jumps the final piece.
Expected Result	The program displays that the user won the game.
Actual Result	Pass

4.4 Night Mode

4.4	**REQ 1**
Condition	Any point throughout the game. The game is not currently in night mode
Action	The user clicks the “Night Mode” button
Expected Result	The UI updates to show the black and blue night mode board.
Actual Result	Pass

4.4	**REQ 2**
Condition	At any point in the game, the game is already in night mode.
Action	The user selects night mode
Expected Result	The UI returns to normal mode with the red and black board.
Actual Result	Pass

Defect/s: 4.4 - If User decides to start a new game mode and the board is in night mode the board will not revert to the original black and red. It will stay in the night mode at the start of a new game.

4.5 Player Name Entry

4.5	**REQ 1**
Condition	The user started the game
Action	The user clicks on the name entry text box
Expected Result	The user is able to type in the box
Actual Result	Pass

4.5	**REQ 2**
Condition	The user typed their name in the name entry box
Action	The user clicks “Submit Name”
Expected Result	The users name appears at the top of the game board.
Actual Result	Pass

4.5	**REQ 3**
Condition	The user wins the game

Action	No further action
Expected Result	The system displays an alert that says, “USER NAME won!”
Actual Result	Pass

Defect/s: 4.5 REQ 2 - If User decides to re-enter different name the previous name will not disappear. The new name will overlap the previous one above the board.

4.6 Music

4.6	**REQ 1**
Condition	User is playing the game and wants to hear music
Action	User clicks play button
Expected Result	Music starts playing while in game
Actual Result	Pass

4.6	**REQ 2**
Condition	User wants to stop hearing music
Action	User clicks pause button
Expected Result	Music stops playing
Actual Result	Pass

4.7 Timer

4.7	**REQ 1**
Condition	User starts the game
Action	User clicks the start timer button
Expected Result	The timer starts counting up.
Actual Result	Pass