Acceptance Test Plan

Networked Chess

Tomer Shemesh

Kelly Shiptoski

Matt D'Amore

Alex Marion

March 9th 2016
Revision 2

Revision History

Name	Date	Reason For Change	Version
Tomer, Kelly, Matt, Alex	02/27/16	Initial Version	1.0
Tomer, Kelly, Matt, Alex	03/09/16	Revised Version	2.0

Table of Contents:

1. Introduction

1.1. Background

Our goal with this document is to list out all of the test cases for both the front and back end of our competitive chess program. This document was influenced by the Software Requirement Specifications (see §1.2) which governed the design and behavior of the program. The document will go into detail about the tests that will be performed on the front end, backend, as well as integration tests which test the interaction between the front and back end.

1.2. References

The following provides a URL to our most recent SRS document -- https://docs.google.com/document/d/1MU82lTAWkP8FAi9lfXHgDKAeDbM1zyf2bjd3jH-t490/edit

This document is also available as a submission on Blackboard Learn.

1.3. Definitions and Acronyms

1.3.1. SRS: Software Requirement Specification

1.3.2. Pawn Two-Move Head Start: If a pawn has not moved, it can move two spaces forward. If it has already moved it may not do this.

1.3.3. Pawn Promotion: If a pawn has reached the end of the board (on the opponent's side) it is replaced by a queen piece.

1.3.4. Castling: If the following conditions are satisfied, the king may move two squares toward the rook and the rook may move one square over the king.

1. The king and one of the target rook must not have moved

2. There are no pieces between them
3. The king is not in check
4. The king does not end up in check
5. The king does not pass through any square that is attacked by an enemy piece

2. Test Approach and Constraints

This section describes the the objectives, structure, and constraints of the test plan for our chess system.

2.1. Test Objectives

The main objective of the Acceptance Test Plan is to verify that the chess software functions as specified in the SRS document. The Acceptance Test Plan will provide criteria to define if the project succeeds or fails.

2.2. Test Structure

Each scenario will consist of of a precondition, an action, a postcondition, and the specifications that are covered by the scenario.

1. The precondition will describe the state of the software before the tested action occurs.
2. The action is what is being tested.
3. The postcondition is the desired state of the software after the action has executed.
4. The specifications are a list of the requirements that the scenario uses.

2.3. Test Constraints

The Acceptance Test Plan will be limited to testing the functionality of the software as defined in the SRS document. It will

not test the design and implementation of the chess game source code. It will not fully test the GUI or the backend server.

3. Test Assumptions and Exclusions

This section provides greater details about which functions and features of the chess program will be covered by the Acceptance Test Plan process and which functions of the chess software will not be covered.

3.1. Test Assumptions

It is assumed that all issues covered by the Acceptance Test Plan were also previously addressed by unit test, integration test and system tests of the chess system. This Acceptance test plan will cover:

- Functional requirements of the system listed in the Software Requirements Specification
- Consistency of user related system documentation.

3.2. Test Exclusions

It is assumed that all issues not covered by the Acceptance Test Plan were not previously tests, integration tests, and system tests. The Acceptance Test Plan will not cover:

- Systems outside the scope of the Software Constraints listed in the SRS.
- Structural integrity of the source code.

4. Entry and Exit Criteria

This section lists the criteria which must be satisfied in order for the Acceptance Test Plan to begin, as well as the criteria which must be satisfied in order for the Acceptance Test Plan to stop.

4.1. Entry Criteria

The Acceptance Test can begin after the following conditions have been met:

1. All other unit tests, system tests, and integration tests are complete.

2. A proper environment that meets Software **r3.2.6** outlined in the SRS is available.
3. A copy of the latest version of the SRS has been received.
4. The latest version of Chess has been compiled and installed
5. Consent from the Team Leader
6. Consent from the Client Representative

4.2. Exit Criteria

The Acceptance Test Plan should be halted after either of the following

- All Priority 1 requirements were tested without deviation from the expected behavior (Success)
- At least one Priority 1 requirement deviated from the documented specification (Failure)

5. Testing Participants

This section describes the roles and responsibilities of the parties involved in the Acceptance Test Plan, as well as the procedure for reporting the test results.

5.1. Roles and Responsibilities

- Test Team Leader: Tomer Semesh
- Client Representative: Alex Marion
- Testers: Kelly Shiptoski, Matthew Damore

5.2. Training Requirements

All parties involved in the Acceptance Test Plan should be comfortable with their standard OS' interfaces. All parties should be familiar with the user interface of the chess game, as well as with the system documentation and the Software Requirements Specification.

5.3. Problem Reporting

Any problem discovered by either the Client Representative or a Tester must be documented and reported to the Test Team Leader. The problem report will be submitted to the Team Leader and addressed during a periodic or special meeting depending on the severity of the problem.

5.4. Progress Reporting

The Acceptance Test Plan Report will be compiled after the testing process has been completed by the Test Team Leader and submitted to the Team Leader.

6. Test Cases

6.1. Create a User

ID	TC 1
Name	Create a User
Requirement(s)	3.1.1) R1.1.1 and 3.1.1) R1.1.2
Description	The user creates a new user
Precondition(s)	● The user has started the chess software ● The user has clicked ‘New Game’
Action(s)	1. The user enters a **new** username 2. The user clicks ‘ok’
Postcondition(s)	● A new file has been created for the user’s information ● The new user is in the lobby

6.2. Create a User Which Already Exists

ID	TC 2
Name	Create a User Which Already Exists
Requirement(s)	3.1.1) R1.1.1 , R1.1.2
Description	The user attempts to create a new user with an username existing in the system
Precondition(s)	● The user has started the chess software ● The user has clicked ‘New Game’
Action(s)	1. The user enters a username which already exists in the system 2. The user is prompted to choose a different username
Postcondition(s)	● The user has not entered the lobby ● The user is still in the create user screen

6.3. Enter the Lobby

ID	TC 3
Name	Enter the Lobby
Requirement(s)	3.1.1) R1.4
Description	The user enters the lobby with an existing user
Precondition(s)	● The user has started the chess software ● The user has a pre-existing username
Action(s)	1. The user clicks ‘New Game’ and enters the lobby
Postcondition(s)	● The user is in the lobby

6.4. Exit the Lobby

ID	TC 4
Name	Exit the lobby
Requirement(s)	3.1.1) R1.4
Description	The user clicks the ‘Leave Lobby’ button
Precondition(s)	● The user has started the chess software ● The user has entered the lobby
Action(s)	1. The user clicks ‘Leave Lobby’ and exits the lobby
Postcondition(s)	● The user is is on the main menu

6.5. Send a Game Request and Request is Accepted

ID	TC 5
Name	Send a Game Request and Request is Accepted
Requirement(s)	3.1.1) R1.4
Description	The user sends a game request which is accepted by another user
Precondition(s)	● The user has started the chess software ● The user has entered the lobby
Action(s)	1. The user sends a game request to another user 2. The other user accepts the request
Postcondition(s)	● The two users have entered a game together ● The user is white in the game and is allowed to move first

6.6. Send a Game Request and Request is Denied

ID	TC 6
Name	Send a Game Request and Request is Denied
Requirement(s)	3.1.1) R1.4
Description	The user sends a game request which is denied by another user
Precondition(s)	● The user has started the chess software ● The user has entered the lobby
Action(s)	1. The user sends a game request to another user 2. The other user denies the request
Postcondition(s)	● The user remains in the lobby ● The request is marked as denied in the database

6.7. Deny a Game Request

ID	TC 7
Name	Deny a Game Request
Requirement(s)	3.1.1) R1.4
Description	The user denies a game request
Precondition(s)	● The user has started the chess software ● The user has been sent a game request
Action(s)	1. The user clicks ‘Deny’ on the request popup
Postcondition(s)	● The game request popup disappears ● The user remains in the lobby

6.8. Accept a Game Request

ID	TC 8
Name	Accept a Game Request
Requirement(s)	3.1.1) R1.4
Description	The user accepts a game request that is sent by another user
Precondition(s)	● The user has started the chess software ● The user has been sent a game request
Action(s)	1. The user clicks ‘Accept’ on the game request popup
Postcondition(s)	● The game request popup disappears ● The user enters a game with the user who sent the request ● The user is Black and moves second

6.9. Enter a Game With Another User

ID	TC 9
Name	Enter a Game With Another User
Requirement(s)	3.1.1) R1.4
Description	The user enters a game with another user
Precondition(s)	● The user has started the chess software ● The user has sent a game request to another user ● The other user has sent the game request
Action(s)	1. The user is White (because they sent the request) and is allowed to move first 2. The user selects a piece
Postcondition(s)	● The other user is in a game with the other user ● The user is ready to make a move

6.10. Make a Valid Move

ID	TC 10
Name	Make a Move
Requirement(s)	3.1.1) R1.5 and 3.2.3) R3.1
Description	The user makes a move
Precondition(s)	● The user has started the chess software ● The user has entered a game ● It is the user's turn
Action(s)	1. The user selects a piece 2. The user selects a square to move to
Postcondition(s)	● The user has created a move ● The user is allowed to accept or cancel their move

6.11. Make an Invalid Move

ID	TC 11
Name	Make an Invalid Move
Requirement(s)	3.1.1) R1.5 and 3.2.3) R3.1
Description	The user attempts to make an invalid move
Precondition(s)	● The user has started the chess software ● The user has entered a game ● It is the user's turn
Action(s)	1. The user selects a piece 2. The user clicks an invalid square to move to 3. The user is not allowed to select an invalid square
Postcondition(s)	● The user has not created a move ● The user must still create a move

6.12. Confirm a Move

ID	TC 12
Name	Confirm a move
Requirement(s)	3.1.1) R1.5 and 3.2.3) R3.1
Description	The user has moved a piece from one place to another and the system needs to confirm the move
Precondition(s)	● The user has started a game ● It's the user's turn ● The user has selected a move
Action(s)	1. The user clicks the 'Confirm Move' button
Postcondition(s)	● The board has been updated with the new move ● It is no longer the user's turn ● It is the other user's turn

6.13. Cancel a Move

ID	TC 13
Name	Cancel a Move
Requirement(s)	3.1.1) R1.5 and 3.2.3) R3.1
Description	The user cancels a selected move
Precondition(s)	● The user has started a game ● It's the user's turn ● The user has selected a move
Action(s)	1. The user clicks the 'Cancel Move' button
Postcondition(s)	● The user may select another move ● It is still the user's turn

6.14. Try to Make Two Moves in a Row

ID	TC 14
Name	Try to Make Two Moves in a Row
Requirement(s)	3.1.1) R1.5 and 3.2.3) R3.1
Description	The user attempts to make a move when it is not their turn
Precondition(s)	● The user has entered a game ● The last move made was by the user
Action(s)	1. The user attempts to make an additional move after the first
Postcondition(s)	● The user is not able to make the move

6.15. Make a Move While in Check

ID	TC 15
Name	Game Ends in Checkmate
Requirement(s)	3.1.1) R1.5.8
Description	The user attempts to move a piece while their king is in check
Precondition(s)	● A game is in progress ● The player's king is in check ● They attempt to select a piece
Action(s)	1. The user selects a piece 2. They may only move the piece into a position that gets them out of check
Postcondition(s)	● Their king is no longer in the check position

6.16. Game Ends in Checkmate and the User Wins

ID	TC 16
Name	Game Ends in Checkmate and the User Wins
Requirement(s)	3.1.1) R1.5.8
Description	The game ends because the user has put the other user in checkmate
Precondition(s)	● The user has entered a game ● The user has put the other user in checkmate
Action(s)	1. The user views the popup alerting them that they have won 2. The user clicks ‘Ok’ and returns to the main menu
Postcondition(s)	● The user’s record has been updated ● The user is returns to the main menu

6.17. Game Ends in Checkmate and the User Loses

ID	TC 17
Name	Game Ends in Checkmate and the User loses
Requirement(s)	3.1.1) R1.5.8
Description	The user is defeated as the opponent has put their king in checkmate
Precondition(s)	● The game was in progress ● The opponent made a move that put the players king in check
Action(s)	1. The user views the popup alerting them that they have lost 2. The user clicks ‘Ok’ and returns to the main menu
Postcondition(s)	● The user’s record has been updated ● The user is returns to the main menu

6.18. Game Ends in Stalemate

ID	TC 18
Name	Game Ends in Stalemate
Requirement(s)	3.3.2) R1.2.4
Description	Neither player wins as
Precondition(s)	
Action(s)	
Postcondition(s)	● The game ends ● User records are updated

6.19. Game Ends Because User Forfeits

ID	TC 19
Name	Game ends because user forfeits
Requirement(s)	3.3.2) R1.2.4
Description	At any time during a game, a player may forfeit and the game will be counted as a loss
Precondition(s)	● The user has entered a game
Action(s)	1. The user clicks the ‘Forfeit Game’ button
Postcondition(s)	● The user’s record has been updated ● The user is returns to the main menu

6.20. Special Move Tests

6.20.1. Pawn Two-Move Head Start

ID	TC 20.1
Name	Pawn Two-Move Head Start
Requirement(s)	3.1.1) R1.5 and 3.2.3) R3.1
Description	From their starting position, a pawn may move two places forward.
Precondition(s)	• A game has been created and started • A player has selected the pawn piece
Action(s)	1. The user moves a pawn two spaces from its starting pointing
Postcondition(s)	• The pawn is moved to the selected location

6.20.2. Pawn Promotion

ID	TC 20.2
Name	Pawn promotion
Requirement(s)	3.1.1) R1.5 and 3.2.3) R3.1
Description	Upon reaching the other side of the board, a pawn will be “promoted” to a Queen piece
Precondition(s)	• A Game has been started • A pawn has been moved to the opposite end of the board
Action(s)	1. The user moves their pawn piece onto the opposing side of the board
Postcondition(s)	• There is now a Queen in place of the pawn

6.20.3. Castling

ID	TC 20.3
Name	Castling
Requirement(s)	3.1.1) R1.5 and 3.2.3) R3.1
Description	The user performs the castling move
Precondition(s)	● The user has entered a game ● The user is has not moved his king or rook ● There are no pieces between the user's king and rook ● The king is not in check ● The king does not end up in check ● The king does not pass through any square which an enemy piece attacks
Action(s)	1. The user selects their king 2. The user selects the space one over from their unmoved rook 3. The user confirms their move
Postcondition(s)	● The king has moved one space away from the rooks original position ● The rook has moved one space over the king ● It is the other user's turn