

---

# **Software Requirements Specification**

**for**

# **Chesstackle**

**Version 1.0 approved**

**Prepared by Snake Army (Group 1)  
Burmeister Uruchurtu, Alec  
Blanco, Bruno  
Lockman, Cory  
Lemmon, Dylan  
Currie Buckner, Ty Baron**

**Texas State University**

**01/28/2018**

# Table of Contents

<b>Table of Contents .....</b>	<b>ii</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions .....	1
1.4 Product Scope .....	1
1.5 References.....	1
<b>2. Overall Description.....</b>	<b>2</b>
2.1 Product Perspective.....	2
2.2 Product Functions .....	2
2.3 User Classes and Characteristics .....	2
2.4 Operating Environment.....	2
2.5 Design and Implementation Constraints .....	2
2.6 User Documentation .....	2
2.7 Assumptions and Dependencies .....	2
<b>3. External Interface Requirements .....</b>	<b>3</b>
3.1 User Interfaces .....	3
3.2 Functional Requirements .....	3
<b>4. System Features .....</b>	<b>5</b>
4.1 Available Moves .....	5
4.2 Taken Pieces .....	5
4.3 Pawn Promotion.....	5
4.4 AI Difficulties .....	5
4.5 Game Sounds .....	5
4.6 Restart/Surrender .....	6
4.7 Play Timer.....	6
4.8 Board Rotation.....	6
<b>Appendix A: Glossary.....</b>	<b>7</b>
<b>Appendix B: Analysis Models.....</b>	<b>7</b>

# 1 Introduction

## 1.1 Purpose

The purpose of this project is to better understand how to use the Waterfall process by creating a Chess game as a team project.

## 1.2 Document Conventions

To maintain order, keep every section of text separated by a space.

## 1.3 Intended Audience and Reading Suggestions

This document is intended to be viewed by those directly involved with the process of creating the software. This includes members of the Snake Army team and the 'client' Mr. Jason Diaz.

Snake Army members should begin reading this document from section 2, referring to section 1.5 for explanations on relevant issues, when applicable.

Mr. Jason Diaz should read the document in its entirety and attempt to locate Easter eggs, or just because effort was expended in creating this document.

## 1.4 Product Scope

This program is a chess game simulator. 1-2 players, human or AI, will take turns moving and taking pieces attempting to mate the other's King. The goal of this software is to show Snake Army superiority over all others.

## 1.5 References

1. Bodlaender, Hans. "The Rules of Chess." The Chess Variant Pages, [www.chessvariants.com/d.chess/chess.html](http://www.chessvariants.com/d.chess/chess.html)
2. "Chessboard." Wikipedia, Wikimedia Foundation, 24 Jan. 2018, [en.wikipedia.org/wiki/Chessboard](http://en.wikipedia.org/wiki/Chessboard).

## 2 Overall Description

### 2.1 Product Perspective

This program will be a new, self-contained product designed to show the importance of the Waterfall method and design patterns.

### 2.2 Product Functions

- The Software shall allow the User to choose the number of human players participating in the game. Choosing 0 players means 2 AI opponents will play a match against each other, 1 player means 1 human player and 1 AI player, 2 players means 2 human players playing against each other.
- The game shall continue to run until one player mates the other's King, presses the 'surrender' button, or presses the 'restart' button.
- Each unique piece shall act as described in Section 3.1.
- The program shall start a match from the default starting position, with the white piece (if only 1 player is chosen, the player shall go before the AI).
- The program shall keep track of the both color's King's check and mate.

### 2.3 User Classes and Characteristics

After delivery, only player users shall access the program, where they will choose to play or watch a match between 2 AI opponents. The AI opponent shall only make legal chess moves, and will not be able to surrender, restart, or otherwise quit the game/program.

### 2.4 Operating Environment

The program shall operate on Windows 10, utilizing the Java Version 8 update 161.

### 2.5 Design and Implementation Constraints

Any computer which attempts to run the software shall run Windows 10, with Java 9.0.1, or backwards compatible version and a working mouse. The Snake Army group shall not be responsible for any of the software once it has been delivered. Changes requested shall be made through official channels where a new cost evaluation shall be generated.

### 2.6 User Documentation

No documentation other than the Software Requirement Specification, System Design Document, and associated code files. Manuals or tutorials, on-line or otherwise shall not be prepared.

### 2.7 Assumptions and Dependencies

The version of Windows 10 used is compatible with the required or compatible Java version (current Version 8).

## 3 External Interface Requirements

### 3.1 User Interfaces

The software shall have a user interface upon which the user may click (using a mouse) and make choices of how many players that they would like to play with.

### 3.2 Functional Requirements

1. The software shall have a user interface (ASCII text is fine, sprites, html images, etc.).
2. The software shall have at least ONE form of AI implementation.
  - AI shall attempt to make intelligent decisions.
  - Defend King or move King if he's threatened.
  - Attack King when given an opportunity.
3. The software shall be user input driven.
  - Select a piece by clicking it, then click a highlighted square to move it.
4. The software shall begin from a main menu.
  - The user will choose number of players (0-2).
  - More specifics can be added to increase the document size.
  - The game begins with the white pieces moving first and the black pieces second.
5. The software shall be able to allow only one move per turn.
6. The software shall be able to determine a check mate and end the game.
  - When a player is in check, and he cannot make a move such that after the move, the King is not in check, then he is mated.
7. Each piece shall move and attack only in their legal ways.
  - a. Rook
    - The Rook attacks in the same way it moves.
    - The Rook moves in a straight line, horizontally or vertically in the + direction.
    - The Rook shall not jump over other pieces, that is: all squares between the square where the Rook starts its move and where the Rook ends its move must be empty.
  - b. Bishop
    - The Bishop attacks in the same way it moves.
    - The Bishop moves in a straight diagonal line, in the X direction.
    - The Bishop shall also not jump over other pieces.
  - c. Queen
    - The Queen attacks in the same way it moves.
    - The Queen has the combined moves of the Rook and the Bishop, i.e., the Queen may move in any straight line, horizontal, vertical, or diagonal.
    - The Queen shall also not jump over other pieces.
  - d. Knight
    - The Knight makes a move that consists of first one step in a horizontal or vertical direction, and then one step diagonally in an outward direction.

- e. Pawn
    - When a pawn has not moved at all, the pawn may choose to move one square forward, or two squares forward. Otherwise it shall only move one square directly forward.
    - Pawns shall only attack diagonally forward.
    - The pawn shall choose between attacking and moving.
    - If a pawn makes it to the opposing side of the chess board, it shall be promoted to any piece of choice.
  - f. King
    - The King shall move one square in any direction, so long as it does not put him under attack.
    - If the King has not moved, the Rook has not moved, the space between the King and the Rook contains no other pieces, and the King is not in check nor is it moving into check, the King shall have the opportunity to “castle” itself. Which lets the King move two spaces towards the selected Rook and moves the Rook to the square next to the King, on its opposite side.
8. The square where any piece ends their move shall not contain a piece of the same color.
  9. All pieces, when the square where their attack ends contains a piece of the opponent, then the opponent’s piece is taken.
    - Taken pieces are removed from the board, and out of play for the rest of the game. (Taking is not compulsory.)
    - Pieces that are captured shall be visible in a display on the right side of the game board.
  10. Each piece shall move in a contained area (the board) and shall not move off the board under any circumstances (including claiming a piece of the opposite color).
  11. Each turn (Excluding AI), the player(s) shall be granted an option of forfeiting the match if they wish to do so.
  12. Each time a player (Including AI) moves a piece, the game shall not allow the opposing player to make a move until the previous piece is moved on the board.
  13. The spaces on the chess board shall be labeled on an XY axis where the origin begins at the bottom-left corner.
    - The Y axis shall be labeled 1-8 starting from the origin.
    - The X axis shall be labeled a-h starting from the origin.

## 4 System Features

### 4.1 Available Moves

#### 4.1.1 Description and Priority

When selecting a piece on your turn, the squares where this piece may move shall light up indicating valid move. This will be a high priority feature and will be implemented as soon as the basic chess game is working.

#### 4.1.2 Stimulus/Response Sequences

When it is the player's turn, the player will click on the piece they wish to move, and all available move locations for such piece will appear in a different coloration to indicate where the player may finish their move with the chosen piece.

### 4.2 Graveyard

#### 4.2.1 Description and Priority

As the game progresses, a list of the pieces each color has taken shall be kept, to show what each side has lost up until that point in the game. This will be a high priority that may be implemented with the basic chess game.

#### 4.2.2 Stimulus/Response Sequences

When a piece is taken, a representation of the piece (name or shape) shall be shown on the side of the board.

### 4.3 Pawn Promotion

#### 4.3.1 Description and Priority

When selecting a pawn reaches the opponent's end row, it may promote (transform) into any other piece. This will be a high priority feature and will be implemented as soon as the basic chess game is working.

#### 4.3.2 Stimulus/Response Sequences

When a player's pawn reaches the opponent's end row, a menu shall appear to allow the player to choose the pawn's promotion piece.

### 4.4 AI Difficulties

#### 4.4.1 Description and Priority

When attempting to play the AI, the player shall get to choose between difficulties for the AI opponent. This is a medium priority and will be implemented after the basic AI is implemented.

#### 4.4.2 Stimulus/Response Sequences

When selecting 1 player, a second option shall appear to allow the difficulty of the AI to be selected.

### 4.5 Game Sounds

#### 4.5.1 Description and Priority

Upon making a move, taking a piece, winning or losing (vs AI only), the game shall make a sound. This is low priority and will be implemented only if time allows.

#### 4.5.2 Stimulus/Response Sequences

Upon completing a move, taking a piece, winning or losing the game, a sound will be generated to indicate the action has been taken.

#### 4.6 Restart/Surrender

##### 4.6.1 Description and Priority

Buttons shall be available to restart or end the current game. This is a high priority and shall be implemented alongside the basic chess game.

##### 4.6.2 Stimulus/Response Sequences

Upon pressing the surrender button, the game immediately ends, with the player whose turn it is losing.

#### 4.7 Play Timer

##### 4.7.1 Description and Priority

Upon the start of a player's turn, a timer shall show how long the player has taken to complete their turn. This is a low priority feature and shall be implemented if time allows.

##### 4.7.2 Stimulus/Response Sequences

None.

#### 4.8 Board Rotation

##### 4.8.1 Description and Priority

The player whose turn it is shall always have their pieces on the lower part of the board. This is a medium priority feature and will be implemented prior to the AI difficulty implementation.

##### 4.8.2 Stimulus/Response Sequences

At the start of the player's turn (only in 2 human player mode), the board shall rotate to show the current player's pieces at the bottom of the screen. This feature shall be turned off if there is only 1 human player, as they will always be on the bottom of the screen.



## Appendix A: Glossary

No acronyms currently used in this document.

## Appendix B: Analysis Models

Add Use cases or possible State Diagram

### Possible use for the SDD?

