Software Requirements Specification

for

ChessMate

Version 1.0 approved

Prepared by Mauricio Valdez, Claudio Corona,

Justin leFrance, Nityam Jigyasu, Man Kit Tsui

The Flame Army

03/08/2018

Table of Contents

Table of Contents 2

1. Introduction 3

1.1 Purpose 3

1.2 Glossary 3

1.3 Intended Audience and Reading Suggestions 4

1.4 Document Conventions 4

1.5 References 4

1.6 Product Perspective ........................................................................................................................ 4

1.7 Product Functions …………………………………………………………………………………5

1.8 User Classes and Characteristics ………………………………………………………………….5

1.9 Operating Environment ……………………………………………………………………………5

2. Overall Description 6

2.1 User Documentation 6

2.2 User Interfaces 6

2.3 Hardware Interfaces 6

2.4 Software Interfaces 6

2.5 Communication Interfaces 6

2.6 Text interaction Interfaces 7

3. System Feautres 7

3.1 Functional Requirements 7

3.2 Chaos Chess Mode 7

3.3 Scoreboard 7

3.4 Move History 8

3.5 Speed Chess………………………………………………………………………………………. 8

3.6. Restart Game……………………………………………………………………………………….8

4. Safetey and Security Requirements 8

4.1 Safetey Requirements 8

4.2 Security Requirements 9

1. Introduction
   1. **Purpose**

The purpose of this document is to present a detailed description of a chess game called ChessMate. It will explain the features of the system, the interface of the system, and how the system will interact with the end user.

**1.2 Glossary**

|  |  |
| --- | --- |
| King | The main piece of the game, checkmating this piece is the object of the game. It can move 1 space in any direction. |
| Queen | This piece can move in any number of spaces in any direction as long as no other piece are in its way. |
| Knight | This piece can move 1 space vertically, 2 spaces horizontally, 2 spaces vertically and 1 space horizontally. This piece looks like a horse and it can also jump over other pieces. |
| Rook | Four pieces on the board (2 per team), that may be moved any number squares horizontally or vertically, as long as no other piece blocks its way. |
| Pawn | Eight pieces per team and consequently the piece with the lowest capture value, usually moved one square at a time vertically and capturing diagonally, although it can move 2 spaces vertically if it is its first move. |
| Bishop | 2 pieces per team that may be moved any number squares diagonally, as long as no other piece blocks its way. One piece always remains on White squares and the other always on Black. |
| Chess Board | A board you need to play Chess. Have 64 black and white squares. |
| Chess | A game that consists of two teams on a chessboard with 16 pieces each. |
| En Passant | This is a method by which a pawn that is moved two squares can be captured by an opponent's pawn commanding the square that was passed |
| Castling | to move the king two squares horizontally and bring the appropriate rook to the square the king has passed over. |
| Check | To make a move that puts the opponents King under direct attack. |
| Checkmate | This is a situation in which an opponent’s king is in check and it cannot avoid being captured. This then brings the game to a victorious result. |
| Player or User | A user or a player will be the entity that is playing the chess game. |
| Stalemate | A situation in which a player’s king is not in check, but that player can make no move. This then results is a stalemate, which is a draw. |

The product shall be able to deliver a complete experience of a game of chess with an AI opponent or with another player on the same computer system. The game will allow the user to control chess pieces in two different forms, mouse or user input. The end goal is to allow an experienced chess player to practice certain strategies and move possibilities, it is not intended for a competitive setting.

**1.3 Intended Audience and Reading Suggestions**

The project shall be available to the following audience:

1. User (Professor Diaz and who will use the final edition)

2. Developers (Team FlameArmy)

3. Tester (Team FlameArmy)

4. Documentation writers (Team FlameArmy)

**1.4 Document Conventions**

Bullet points and number lists where used to list certain requirements. Other than that, natural language sentences were used. The keyword "shall" will describe certain requirements the project will deliver.

**1.5 References**

Wikipedia: Understanding the rules of chess in order to design Classes, and Functions.

AI: (Nityam's slack share link detail), in order to create a AI opponent, by professor Diaz's permit, we are allowed to use online resources instead of creating our own AI configuration.

**1.6 Product Perspective**

ChessMate will be the first version of a chess game design. It is indented to work as a standalone system.

**1.7 Product Functions**

This program shall contain the following functions:

1. Display the chess rules pertaining to this system.

2. Move pieces using mouse or user input.

4. Allow the user to play against an AI.

5. Allow the user to see the scoreboard.

6. Give the user an option to play chess under the constrain of a timer.

8. Allow the user to see the move log of a game.

9. Able to determine checkmate correctly.

10. Allow the user to restart the game at any time.

11. Allow the user to input name to track high score on scoreboard.

The program shall meet the following requirements

1. A user interface (ASCII text is fine, sprites, html images, etc.).

2. There must be at least ONE form of AI implementation

3. There must be user input

4. The game must begin from the normal starting position and must complete when it has determined check mate.

5. You MAY use libraries/APIs, but you MAY NOT copy code.

6. All documentation and code must be pushed to GitHub by everyone as they complete work.

**1.8 User Classes and Characteristics**

Intermediate Chess Player: The user must be an experienced chess player. The user must know how to play chess before attempting to play the game since it will not provide the user hints to making moves.

**1.9 Operating Environment**

The game will operate on windows 7 or higher version.

**2.0 Overall Description**

**2.1 User Documentation**

A button will be available to show the rules for any piece the user decides to check, including the moves it can make.

External Interface Requirements

**2.2 User Interfaces**

The chess board will be black and while. Each piece will have the same color of its corresponding team, either black or white.

Any errors that the user makes will be displayed in a text field containing the error summary. If the user does the same error twice, a pop-up screen will be displayed for the user to acknowledge.

**2.3 Hardware Interfaces**

When the User is utilizing the product, the Hardware shall able to:

1. allow the game to support mouse click interaction (left/right click)

2. use mouse to interact with all tab options (start the game, surrender...)

3. use mouse to interact with the chess and the available places to move.

4. use mouse to close the game.

(Maybe)5. use mouse to interact with the message such as surrender (start a new game/continue option on the surrender message)

**2.4 Software Interfaces**

Developing language: Java.

IDE and Version: Eclipse, Oxygen.

**2.5 Communications Interfaces**

No communication interfaces will be necessary since the player will be playing against an AI

or a player in the same system.

**2.6 Text interaction Interfaces(Product logic design test)**

By using chess notation, we display the notation (maybe grid by grid) of the game user input and output to perform the logic of the game and test that the classes and functions are correct.

An idea example:

game start!

move: pawn1 // input

move to: a5 // another input

display: can’t move to their cause " " choose another move (Bool check)

move: pawn 1 // input

move to: a3 // input

display: success, move pawn 1 to a3 // display move

display:

r k b K q b k r

p p p p p p p p

. . . . . . . . . . .

. . . . . . . . . . .

P . . . . . . . . . .

P P P P P P P

R K B Q K B K R

**3.0 System Features**

**3.1 Functional Requirements**

If Player chooses the wrong place to move the piece, due to the "Rule" Class, it will not allow Player to perform such certain move. If Player interacts with the other side pieces, it will not show any interaction.

**3.2 Chaos Chess Mode**

**3.2.1** This feature makes the board pieces spawn at the start of the game in a random fashion. Each piece such as the King could potentially appear in a check position.

**3.2.2** The user will be given the option to play again an AI or a Player. After the player chooses

his opponent, the player will be given the option to play in either normal or chaos mode.

**3.3 Score Board**

**3.3.1** This feature allows the user to see the scoreboard record in win to loss ratio fashion. Only the top 5 are recorded.

**3.3.2**

The user in the main menu can choose to see the record.

**3.4 Move History**

The move history will keep track of the piece movements using the standard chess notation.

**3.4.1**

The player can view a complete move history by choosing "View Move History".

In addition the player can view moves performed by clicking a button "View Movies" while playing the game.

**3.5 Speed Chess**

Only counts players time when they are playing. 2 modes – normal chess clock counting down time left and time used to show how fast the player is playing.

**3.5.1** The player will be given the choice in the main menu to choose what type of timer he or she wishes to use.

**3.6 Restart Game**

The user can restart the game at any time, which will not save any stats from the current game.

**3.6.1** The user can choose to press the "Restart Button" while playing the game. It will be located in the top-mid screen.

**3.7.1** Moving Pieces

The player can choose to move pieces using either the mouse or data input, but not both.

The player can choose by going to the setting tab on the top-left of the main menu.

**4.0 Safety and Security Requriements**

**4.1 Safety Requirements**

If the system is overloaded, the system will turn off automatically.

**4.2 Security Requirements**

The users are advice to use nicknames or only their first names when recording their high scores since their selected names will be being displayed to whoever gets to the scoreboard screen should they beat one of the top 5 scores.