

User Manual

SimuChess v1.0

Authors: Crystal Software (Team 19)

Jesse Corrales-Lozano

Mohammed Haque

Jose Alberto Padilla

Sean Michael Santarsiero

Ralph Hieu Cao Tran

User's Manual

Table of Contents

	<u>Page #</u>
1.0 Computer Chess.....	4
1.1 Usage Scenario.....	4-8
1.2 Goals.....	8
1.3 Features.....	8-9
2.0 Installation.....	9
2.1 System requirements.....	9
2.2 Setup and Configuration.....	9
2.3 Uninstalling.....	9
3.0 Chess Program Functions and Features.....	10-11
3.1 Chess Piece Movement.....	10
3.1.1 Moving a chess piece.....	10
3.1.2 Special Movements.....	10
3.2 Surrender Match.....	11
3.3 Main Menu.....	11
3.3.1 Start Game.....	11
• Choosing Player(Black or White)	
3.3.2 Quit Game.....	11
3.3.3 Previous Game Log.....	11
4.0 Back Matter.....	11-13
4.1 Copyright.....	12
4.2 Error Messages.....	12
4.3 Index.....	13

Glossary

Pieces:

- **Piece:** Any of the chess figures used in play (**King, Queen, Knight, Bishop, Rook, Pawn**). The Queen, Bishop, and Rook can move to any square as long as they are not obstructed by other pieces.
- **Bishop:** The chess piece that moves diagonally.
- **King:** The most important piece of the game. It can move 1 square at a time in any direction.
- **Knight:** The chess piece that moves in an “L” (2 squares horizontally and 1 square vertically or 1 square horizontally and 2 squares vertically).
- **Pawn:** The chess piece can only move forward 1 square at a time and **capture** diagonally by 1 square. If the pawn has not been moved yet, it can move 2 squares on its first move.
- **Queen:** The most powerful piece in the game. It can move straight in all 8 directions.
- **Rook:** The chess piece that can move horizontally or vertically.

Game:

- **Algebraic Notation:** A method for recording chess moves using the **rank** and **file** of the squares as coordinates.
- **Attack:** To move a **piece** to a square where you could **capture** an opponent's piece on the next move.
- **Capture:** To remove an opponent's **piece** from the board by occupying its square with a piece of your own via a legal move.
- **Castling:** To move your unmoved **King** 2 squares to the right (kingside castling) or left (queenside castling) and your closest unmoved **Rook** to the other side of the **King**. The squares between the two pieces must be empty in order to castle.
- **Check:** An **attack** on the **King**.
- **Checkmate:** An **attack** on the **King** where there is no way for your opponent to finish his turn and no longer have the **King** under **attack**. This ends the game.
- **En Passant:** Capturing a **pawn** that has moved 2 spaces with a **pawn** that could have captured it if it had only moved 1 space. If the **capture** is not done on the turn after the **pawn** moves 2 spaces, it can no longer be captured this way.
- **File:** The vertical columns of squares on the board, lettered a-h with “a” always on White's left (Black's right).
- **Illegal Move:** A move that either moves a **piece** in an illegal manner, or results in an illegal position.
- **Promote:** When a **pawn** reaches the other side of the board (8th **rank**), it must promote to any other piece besides the **King** immediately.

- **Rank:** The horizontal rows of squares on the board, numbered 1st-8th starting from White's side.
- **Repetition:** A type of draw where the same position is reached 3 times with the same player to move. Does not require the same moves and can occur at any point in the game.
- **Stalemate:** A type of draw where the player to move is not in check, but none of his pieces can move.

1.0 Computer Chess

This section is meant to provide examples of different scenarios, and information on what our goals are for the user and features in the game.

1.1 Usage Scenario

Here is an image of the board that will be displayed on the screen once the user chooses to begin:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Posted below is a scenario in which the user chooses to be white and moves their white pawn in position e2 to position e4 to depict what the user

should expect to see when moving their piece:

	+-----+-----+-----+-----+-----+-----+-----+-----+																
8		bR		bN		bB		bQ		bK		bB		bN		bR	
	+-----+-----+-----+-----+-----+-----+-----+-----+																
7		bP		bP		bP		bP		bP		bP		bP		bP	
	+-----+-----+-----+-----+-----+-----+-----+-----+																
6																	
	+-----+-----+-----+-----+-----+-----+-----+-----+																
5																	
	+-----+-----+-----+-----+-----+-----+-----+-----+																
4								wP									
	+-----+-----+-----+-----+-----+-----+-----+-----+																
3																	
	+-----+-----+-----+-----+-----+-----+-----+-----+																
2		wP		wP		wP		wP				wP		wP		wP	
	+-----+-----+-----+-----+-----+-----+-----+-----+																
1		wR		wN		wB		wQ		wK		wB		wN		wR	
	+-----+-----+-----+-----+-----+-----+-----+-----+																
		a		b		c		d		e		f		g		h	

Another scenario is posted in which the user performs the special move “castle.” Potential setup of the board before castling:

	+-----+-----+-----+-----+-----+-----+-----+-----+								
8		bR				bK			
	+-----+-----+-----+-----+-----+-----+-----+-----+								
7		bP		bP		bP			
	+-----+-----+-----+-----+-----+-----+-----+-----+								
6						bP			
	+-----+-----+-----+-----+-----+-----+-----+-----+								
5									
	+-----+-----+-----+-----+-----+-----+-----+-----+								
4									
	+-----+-----+-----+-----+-----+-----+-----+-----+								
3						wP			
	+-----+-----+-----+-----+-----+-----+-----+-----+								
2								wP	
	+-----+-----+-----+-----+-----+-----+-----+-----+								
1						wK			
	+-----+-----+-----+-----+-----+-----+-----+-----+								
		a		b		c		d	
						e		f	
								g	
									h

Potential setup of the board after castling:

	+-----+-----+-----+-----+-----+-----+-----+-----+							
8			bK	bR				
	+-----+-----+-----+-----+-----+-----+-----+-----+							
7	bP	bP	bP					
	+-----+-----+-----+-----+-----+-----+-----+-----+							
6				bP				
	+-----+-----+-----+-----+-----+-----+-----+-----+							
5								
	+-----+-----+-----+-----+-----+-----+-----+-----+							
4								
	+-----+-----+-----+-----+-----+-----+-----+-----+							
3					wP			
	+-----+-----+-----+-----+-----+-----+-----+-----+							
2						wP	wP	wP
	+-----+-----+-----+-----+-----+-----+-----+-----+							
1						wR	wK	
	+-----+-----+-----+-----+-----+-----+-----+-----+							
	a	b	c	d	e	f	g	h

The final scenario is posted in which the user performs the special move “En Passant.” Potential setup of the board when attempting En passant:

8			bK	bR				
7					bP		bP	
6								
5						wP		
4								
3								
2								
1							wK	wR
	a	b	c	d	e	f	g	h

1.2 Goals

The goal of Modern Chess is to provide a chess program that abides by the general rules of chess and allows a user to play against an AI opponent to test their chess skills.

1.3 Features

- Can challenge an AI or another human

- Can choose what color to play as (White always goes first)
- Pawns can be promoted if they reach the opposite side

2.0 Installation

Information of the minimum system requirements, how to setup and configure, and how to uninstall the program.

2.1 System requirements

CPU: Intel(R) Xeon(R) Processor E5-2660 v4 @ 2.00GHz

OS: Linux (CentOS release 6.9)

Memory: 4 GB RAM

Graphics: 512 MB Video Memory

Storage: 1MB

2.2 Setup and Configuration

How to install SimuChess:

1. Download the Chess_V1.0.tar.gz package.

2. "Untar" this downloaded package.

```
tar -xvzf Chess_V1.0.tar.gz
```

3. Run the game by typing "./Chess" (omitting quotes) into the command line while in the folder *bin*.

```
./Chess
```

4. Begin playing!

2.3 Uninstalling

To uninstall SimuChess, the user may simply delete the tar package.

3.0 Chess Program Functions and Features

General information that explains each of the functions available in the system and describes their intended use.

3.1 Chess Piece Movement

This section will describe how the user is able to move a chess piece and how to access special movements.

3.1.1 Moving a Chess Piece

Chess board is an 8x8 Matrix with columns represented alphabetically with the bottom left corner being 'a' and the right most corner being column 'h'. Rows are represented by numbers with the bottom row being '1' and the top being '8'. Pieces on the board will be represented by a tag containing the first letter of the name of the piece and a number that differentiates multiple pieces of the same kind. An example of this would be a pawn with a tag of "P1". Movement requires two sets of input, the first designates the spot on the board that contains the piece the user desires to move. The program will verify the piece and prompt the user to designate where on the matrix that the piece will be moving to. An example of these inputs would be "h8", which designates the top right corner on the board for the program inspect.

3.1.2 Special Movements

Special moves for this program consist of "Castling", "Promotion" and "En Passant". Refer to the glossary for additional information on each.

These moves follow the official rules of chess and require no additional user input. A user will follow the steps in **3.1.1** and the computer will determine whether you have met the requirements for these moves and validate this movement.

3.2 Surrender Match

To end a chess match during the middle of the game the user will input “quit” in the command line. This will then return the user back to start menu.

3.3 Main Menu

This section will describe the functions in the main menu and how to operate them.

3.3.1 Start Game

This option will allow the user to decide whether they want to be Black or White, with the main difference being that White always goes first. Also, they must choose if they want to go against an AI or another human, and if they choose AI then they must choose its difficulty level. The user can also decide if they want to set a timer. Once these options have been chosen, the game will begin.

3.3.2 Quit Game

During a game, an option will be provided that will allow the user to quit the current game and will then be redirected to the main menu.

3.3.3 Previous Game Log

A log of the most previous game will be stored so that the player can view a replay of the match.

4.0 Back Matter

Copyright, any potential error messages, and an index are included in this section.

4.1 Copyright

© 2018 Crystal Software. This work is the intellectual property of Crystal Software. Permission is granted for this material to be shared for non-commercial, educational purposes, provided that this copyright statement appears on the reproduced materials and notice is given that the copying is by permission of Crystal Software. To disseminate other or to republish requires written permissions from Crystal Software. All rights reserved.

4.2 Error Messages

Error messages are designed to keep gameplay fair for all players. If user receives error message for input then that input will be unacknowledged in gameplay and an error free input will need to be re-submitted to continue game.

Code	Meaning	Solution
Invalid Move	User input for selected space on board has triggered flags for illegal movements.	Verify the movement follows Chess official rules guidelines for particular chess piece. In addition, verify space is not already populated with ally chess piece.

Invalid Selection	User input has triggered flags for invalid selection of chess piece.	Verify desired space indeed contains ally chess piece and not opponent's piece.
Wait	User input has been submitted during opponent's turn.	Verify input is submitted during player's own turn.
Invalid Tile	User has entered a space off the chess board	Verify space is within the 64 squares available on the chess board.

4.3 Index

Castle/Castling 6-7, 10

En Passant 8,10

Features 10

Invalid 12

Glossary 2

Graphics 9

Log 11

Matrix 10

Memory 9

Pieces 2

Storage 9

Surrender 11

User's Manual 2

Wait 12