

# **Software Engineering GP02 Project**

## **User Interface Specification**

Authors: Thomas Roethenbaugh [tpr3] Will Abbott [wia14]  
Lance Sebastian [lvs1] Jack Book [jab153]  
Config Ref: SE.G02.UISpec  
Date: 27th February 2023  
Version: 1.2.0  
Status: For review

Department of Computer Science  
Aberystwyth University  
Aberystwyth  
Ceredigion  
SY23 3DB  
Copyright © Aberystwyth University 2023

## CONTENTS

CONTENTS .....	2
1. INTRODUCTION .....	3
1.1 Purpose of this Document .....	3
1.2 Scope.....	3
1.3 Objectives.....	3
2. TYPICAL USERS .....	3
2.1 General Users .....	3
Beginner .....	3
Expert .....	3
2.2 Example Users .....	3
3. USE CASES .....	4
3.1 Use cases for the main menu .....	4
3.2 Use cases for the game screen.....	4
3.3 Use Case Diagram.....	6
4. ERROR CONDITIONS .....	7
4.1 Potential use case errors .....	7
REFERENCES .....	8
DOCUMENT HISTORY .....	9

# 1. INTRODUCTION

This document is the User Interface Design Specification, created to guide the development of the program as well as being a visual reference from which the group can all work from. Within the Chess Tutor requirements, it is stated the two main functions of this program are as follows:

1. To help users learn the valid moves in the game of chess, and to enforce those rules. [1]
2. To allow two human players to use the game to play a game of chess. [1]

## 1.1 Purpose of this Document

The purpose of this document is to describe the main use cases of the program and give visual reference to developers for the structure of the project.

## 1.2 Scope

This document is for detailing the use cases of the project as well as listing any error conditions that may arise from those use cases. Each use case will be used to guide the development of the program itself.

This document should be read by all project members and understood thoroughly by both the user interface and systems programmers.

## 1.3 Objectives

The objective of this document is to explain/list the typical users, list all the use cases and further detail how this will affect the development of the project. Finally, this document will go through all the error conditions to help avoid problems when it comes time to code the project.

# 2. TYPICAL USERS

## 2.1 General Users

### Beginner

The beginner will be new to chess, but not new to computers or Windows programs. The user interface will need clear icons for each piece as well as any the moves they can make, as the user may not know what can be done when selecting each piece from the board. There will also need to be a clear indication of which person is currently playing as well as to indicate which color the user is.

### Expert

The expert will have a good understanding of chess and will not require much tutoring to get started with the program. However, this player may have certain expectations when it comes to the user interface, with things like drag and drop being expected when playing. It is also important to not clutter the user interface for this type of user as they would become frustrated with hints that while helpful for beginners, would becoming irritating for a more experienced player.

## 2.2 Example Users

**2.2.1** Mary is long sighted and has never played chess but has a basic knowledge of computers and interacting with Microsoft Windows. They will start a new game and select white. The large icons and simple user interface

will make it easier for Mary to interact with the game and the indications for where each piece can be moved to will be helpful for them.

**2.2.2** Mohammad is a teacher and chess expert, who runs an after-school chess club for the students. The indicated permissive play positions will not hinder the playing style of an expert such as Mohammad. The replay function allows for Mohammad to play a game, and step through each of the moves to explain to the class.

**2.2.3** Elon is an extremely busy person, who very rarely finds time to complete a full game of chess with their friend Jeff. The save function is very useful as it allows them to start and pause their game and return later to finish it.

### **3. USE CASES**

#### **3.1 Use cases for the main menu**

UC01 – Start game – FR1 & FR2

The user will select the ‘Start Game’ button from the menu options. This button will lead to the player selecting their desired start color and show the opponent’s color as well (see UC02). Once the user has completed UC01, they will then be presented with the option to choose start color (either white or black). The user then selects their chosen color by selecting the tick box next to the color they wish to use. The game will assign their chosen color to the user and the other color to their opponent.

UC02 – Load game – FR1 & FR2

On the main menu, the user will select the ‘load game’ button. A file explorer window will appear, and the user will navigate to the folder that the game saves are stored in and select the save file they wish to load. The game will then display the selected chess game on the board, in the state that it was saved in.

UC03 – Replay game – FR2 & FR10

When replaying a game, the user will first select the ‘Replay Game’ button. A file explorer window will appear, and the user will need to navigate to the folder that the game saves are stored in and select the save file they wish to load. The game will then have an option for the user to ‘step through’ each of the replayed game’s moves, as well as another button to step back and go back to the previously made move.

UC04 – Back button – FR9

The player will be able to select a back button after pressing the start game button (UC01) in case they change their mind or click on the wrong button. It will take the user back to the main menu.

UC05 – Quit – FR9

For any available ‘Quit’ button, the user will select the quit button and a dialogue box will appear asking the user if they’re sure they want to quit. Clicking ‘yes’ will quit the game and clicking ‘no’ will close the dialogue box and return to the screen the user was previously on.

#### **3.2 Use cases for the game screen**

UC06 – Select piece – FR4

The user will select a piece when the on-screen prompt notifies them that it is their turn. The user can move the mouse cursor over any of their available pieces and click. Clicking on any of their pieces will show available options for the given piece that the user has selected. This will be highlighted on the screen with dots in the center of the available piece destination cells.

UC07 – Move piece – FR5

As described in UC07, the user will select their chosen piece. To move the piece, they will select the destination square that they wish to move the piece to.

UC08 – Capture piece – FR2 & FR5

When a game is in progress and there is an option for the user to capture their opponent's piece, the square that their opponent's piece is on will be highlighted within a circle as an available move. The user will capture their opponent's piece, and the removed piece will be shown off the board, at the side of the screen, it will remain there for the rest of the game or until it is brought back into play with UC11.

UC09 – See all potential moves for selected piece – FR3 & FR5

As stated in UC07, when the user selects a piece, the available moves will be highlighted on the screen, with dots in the center of either the empty spaces, or the space containing an opposition piece that can be taken.

UC10 – Promoting a pawn – FR2

When a user pawn has successfully moved across the board, the pawn must be promoted to either a bishop, rook, knight, or queen. A box will appear with these options, and the user can pick which piece they wish to promote. Once picked, this piece replaces the pawn at the same location.

UC11 – See all removed pieces – FR2 & FR3

Removed pieces will be displayed at the side of the screen for the entirety of the game unless one of the pieces is replacing a pawn, in which case that piece will be removed from the side of the screen and returned to the board, and the pawn that has been swapped will be moved to the side of the board.

UC12 – Prompt Castling move – FR4

When a castling move is valid and available to the user, the user will select the King and will be shown yellow versions of the pieces in the respective cells that they would move to, if the player was to castle. This will be different from the normal dots that are shown to indicate where the user can move a piece to, as having yellow pieces allows for a more obvious state of play once the move has been made, as well as the distinction of the current game, to the possible move.

UC13 – Prompt En passant – FR4

If a pawn is side by side with an opposing pawn that has just moved two spaces forward (initial two square advance), the player can select that pawn and will be prompted to move it behind the opposing pawn. Removing the opposing pawn from the game and putting the player's pawn in a potentially stronger position. The player will receive this prompt similarly to UC13 with yellow pieces.

UC14 – Check – FR6

If a player has reached a point where their king is in check, the cell the king is in will be highlighted in yellow to indicate to the player that they must move out of check. This can be done by either moving the king itself, or by moving another piece to 'defend' the king. If the player selects the king, in order to take it out of check, any positions that the king would normally be able to move into but cannot in this instance as it would still be in check, will not be shown with a dot in the cell.

UC15 – Checkmate – FR7

If a player has reached a point where their king is in check and the king cannot be saved, the cell the king is located in will be highlighted in red and a popup indicating that the game is over will be displayed to both players. The details of this popup are covered in UC20.

UC16 – Save game – FR11

If the user wants to save the game they are currently playing, they will click the 'Save Game' button. A file explorer window will appear asking the user to select a location they wish to save the game in. They will then need to navigate to their desired location and click save. The game will then be stored in that location and will update after each player's turn.

UC17 – Offer draw – FR8

Either player can be able to offer a draw to their opponent if neither side can win or both teams wish to end the game. There will be a draw button on the sidebar that, when pressed, will prompt the other player to either accept or decline the request for a draw. If the offer is accepted then the game will end in a draw, if not, the game will continue.

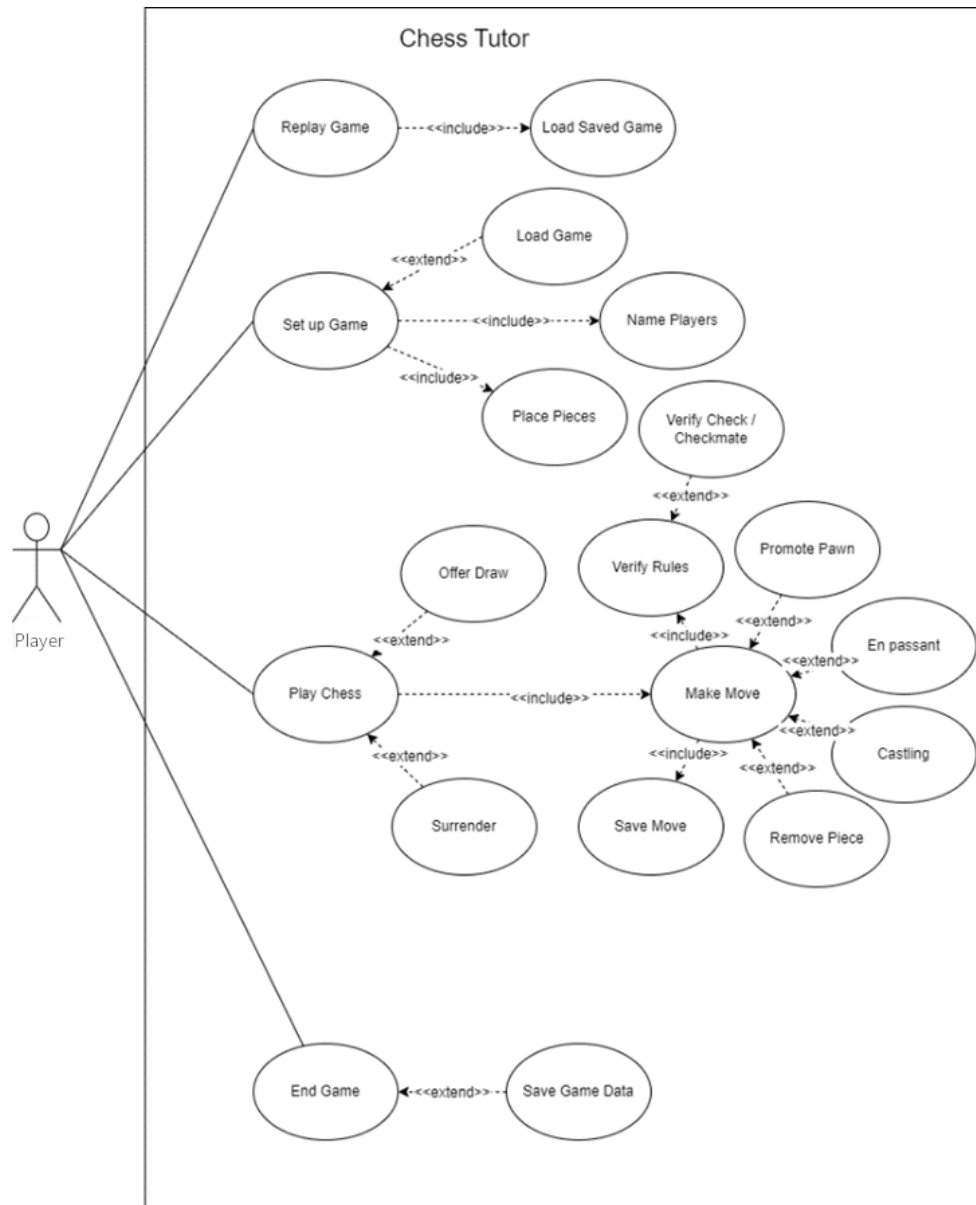
#### UC18 – Resign button – FR8

During any game, either player will be able to resign at any time from a button. When pressed this will end the game in the favor of the opposing player and put them both on the main menu screen.

#### UC19 – End game – FR8

Once the game has been concluded, either through a draw, resignation, or checkmate. There will be a popup prompt detailing the result of the game and a button that once clicked, will take the players back to the main menu.

### 3.3 Use Case Diagram



## 4. ERROR CONDITIONS

### 4.1 Potential use case errors

#### EC01 – Starting a new game

When the user has started a game, they will be prompted to choose their starting color and give each player's name. The player should not be able to start a new game without giving both players a name and selecting each color. If the player does attempt this, there should be a popup that warns the user this is not possible.

#### EC02 - Move piece

If the user attempts to select their opponent's color to move, the piece should not be highlighted, or make any indication of movement if the piece is dragged to a different place.

The user will not be allowed to move the King into check.

The game will not allow for a pawn to move 2 places if it is not the first move for that pawn.

There should be no delay when selecting a piece, this will be an optimization problem.

The pieces move according to the rules defined [2]. If the piece breaks the rules, it must return to its previous position for the user to try again.

If the user attempts to select their opponent's color, the piece should not show any potential moves for that piece as described in UC09. Instead, there should be a warning to the user explaining which color the user currently is, as well as good starting moves for said user.

#### EC03 – Capture piece

If the user captures a piece, the game will remove the piece from the cell and place it at the side of the screen. If the user attempts to capture a piece with an illegal move there should be a warning detailing how that move is illegal and the piece selected will not move.

#### EC04 – Promoting pawn

When the user manages to move a pawn to the other side of the board, thus allowing for a promotion, they should not be able to select the pawn to swap for the pawn on the board.

The user is also forced to promote their pawn. Their turn will not end nor will they be able to select other pieces until they promote their pawn.

#### EC05 - Load game

When the user attempts to load a game to continue playing, if the file selected is invalid, the game will notify them that the file is invalid.

If the file format is incorrect, the program must relay that the file is 'corrupted' and avoid loading the file.

#### EC06 - Replay game

When the user attempts to load a game to replay, if the file selected is invalid, the game will notify them that the file is invalid.

The replay game function should also not be able to run if the file is invalid or the user has not selected a file to replay.

#### EC07 – Quit game

The Quit game button within the current game section of the program, should not immediately quit the game without first prompting the user to save the game. Other quit game buttons that are not within this, are allowed to immediately quit the game.

#### EC08 - Prompt Castling move

The valid moves shown to the player for castling will be different from the normal dots shown. The game will instead show yellow versions of the pieces, showing their new positions if the move is played as well as demonstrating.

If the player attempts to castle the king after it has already moved, castling is not be allowed and the user will be warned by the program why.

**EC09 – Check**

When either the king or the opponent's king is in check, the game should not allow any moves to be made that do not take the king directly out of check, whether this is by moving the king into another position where it is still in check, or moving another piece that does not directly take the king out of check. If the user attempts any move that will not get them out of check the program should warn them they cannot make any move that does not take them out of check.

**EC10 – Prompt En passant**

The prompt shall be similar to EC12, where there shall be a yellow highlighted version of the piece, in the position it will be, once the move has been finished.

**EC11 – Crashing**

In the event of the program crashing, the game will automatically be saved throughout the game, ready to be continued on start-up.

**EC12 – Offer Drawing**

In the event of the draw being ignored (The player makes a move or doesn't do anything in a period) it must be assumed that the user denies the request and the prompt disappears. The prompt cannot stay up forever.

## **REFERENCES**

- [1] Software Engineering Group Projects: General Documentation Standards. C.W. Loftus. SE.QA.03. 2.5 Release
- [2] E. Miscellaneous / 01. Laws of Chess / FIDE Laws of Chess taking effect from 1 January 2023 / FIDE Handbook (2023). Available at: <https://handbook.fide.com/chapter/E012023> (Accessed: 19 February 2023).
- [3] Software Engineering Group Projects – Test Procedure Standards. C.W. Loftus. SE.QA.06. 2.1 Release



## DOCUMENT HISTORY

Version	Issue No.	Date	Changes made to document	Changed by
0.1	N/A	02-02-2023	N/A - original version	TPR3
0.2	N/A	13-02-2023	Added use case specifics	WIA14
0.3	N/A	16-02-2023	Added error cases and use case corrections	WIA14
0.4	N/A	16-02-2023	Added examples users	WIA14, JAB153
0.4.1	N/A	16-02-2023	Minor changes and modifications	WIA14
0.4.2	N/A	17-02-2023	Added Use Case Diagram and minor changes	WIA14, LVS1
0.5.0	N/A	17-02-2023	Added 6 new use cases and changed formatting of error conditions	TPR3
0.5.1	N/A	19-02-2023	Added 4 new error conditions and updated some old error conditions and use cases and added references	LVS1
0.5.2	N/A	21-02-2023	Added updated Use Case Diagram, and removed toggle related UC and EC	LVS1
0.6.0	N/A	22-02-2023	Added config ref number, final version before review	TPR3
0.6.1	N/A	22-02-2023	Changes made during review	TPR3
0.6.2	N/A	24-02-2023	Re-arranged use cases to menu and in-game use cases	TPR3
0.6.3	N/A	25-02-2023	Detailed pawn promotion EC, renamed EC04 to Capture piece, and added resizing window EC	LVS1
0.6.4	N/A	27-02-2023	Changed grammatical errors. Minor changes to Use case specifics	WIA14
0.6.5	N/A	27-02-2023	Minor error corrections	WIA14
1.0.0	N/A	28-02-2023	Release version	TPR3
1.1.0	#40	04-03-2023	Fixed several issues based on feedback, a few problems were left in and need to be followed up with Chris.	TPR3
1.2.0	#40	20-03-2023	Fixed all problems with document according to feedback. Document is now ready for a second review	TPR3