Software Engineering Group Project - UI Specification

Author(s): Tyler Lewis [tyw1@aber.ac.uk]
Sean Hobson [seh25@aber.ac.uk]
Jack Thompson [jat92@aber.ac.uk]

Config Ref:	UI-Spec-Docu-GP9
Date:	15 February 2023
Version:	0.4
Status:	Draft

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

CONTENTS

1. INTRODUCTION

1.1 Purpose of this document

This document's purpose is to describe the specifications for the User Interface. These specifications will detail how to meet the criteria set out by SE.QA.04 [1] and how to create and design the interface effectively.

It also aims to support the document that should be supplied alongside this one, the UI Presentation (UI-Spec-Pres-GP9.pptx).

1.2 Scope

This document explains the specifications for creating the User Interface. This document should be read by all team members, most notably any members that are producing content related to the user interface.

It is assumed that the reader is already aware of the details laid out in SE.QA.04 [1].

1.3 Objectives

The objective of this document is to:

1. Identify the potential users of the system, and discuss how best to cater to each user's needs.
2. Describe the variety of use cases that need to be considered, and describe the steps involved to satisfy each use case.
3. Consider the possible error conditions that may arise during operation of the system, and how to overcome these in a way that follows good UI design practice.

2. TYPICAL USERS

This section will discuss the different types of typical users that will be interacting with the system via the user interface.

2.1 Beginner

Beginners that are new to Chess or have little experience are likely to be a significant part of the user base, and are the focus of the design. As a “Chess Tutor” program, the system will be looking to educate players on the different moves that can be made at any given stage.

Tony is a teenager that wants to learn how to play chess. He has only played once before, so is aware of the very basic concept. Tony wants to learn about how each of the pieces move, in what situations they can capture other pieces, and under what circumstances they cannot be moved.

2.2 Intermediate

Intermediate users that have some experience in playing chess and understand the basics of the game may want to advance their skills and improve tactics and planning. Intermediate users will require less help with the basics, and more assistance with advanced techniques.

Sam is an intermediate player that has already learnt the basics of chess after playing with their friend for many months. Sam wants some assistance to improve and build upon their current abilities, but doesn’t want to waste time and effort relearning the basics of the game.

2.3 Experienced

Experienced users that have significant experience will want minimal interruption when playing a game, and are content with little to no assistance. They will likely want to practise independently of the system against another player.

Beth is an experienced player, and knows the game of chess well. She enjoys challenging people to games and competing against others. Being able to play the game quickly and make independent decisions is important to Beth.

2.4 Casual

Casual players can have any level of experience, and are less concerned about improving or learning and would prefer to simply have fun. Being able to use the system easily, quickly, and dynamically is important to casual players.

Alaw is a casual chess player that spends a lot of time travelling on public transport. He would like to play chess whilst he travels, and sometimes feels like practising to improve, at other times he simply wants to play a game of chess against himself or a friend.

3. USE CASES

The following section will discuss the various use cases relating to the system and its users.

UC-0.0: Launching The Application (FR1)

Upon launching the application, the user will be greeted with a “main menu” screen. This main menu screen will contain all of the core parts of the application, “New Game”, “Load Game”, and “View Game”. The buttons will be labelled to make their purpose obvious and to draw the users focus towards each of the buttons.

UC-0.1: Starting a New Game (FR1, FR2)

After reaching the main menu, the user can select to begin a new game (UC-0.0). Upon selecting the “New Game” option, the user will be prompted to enter the names for both players. After confirming the names, the user will then be shown the new board with pieces ready to be played, as well as the players colours.

UC-0.2: Restoring a Previous Game (FR1, FR11)

After reaching the main menu, the user can select to load a game (UC-0.0). When selecting the “Load Game” option, the user is then shown a screen that lists the save files in a predetermined directory. Once the user has selected the save file to load into the application, the board will be set, and the pieces placed in the correct positions according to the save file, and then the game will be ready to play.

UC-1.0: Selecting a Piece (FR4)

The user will select a piece by clicking on its graphical representation in the relevant window. Once selected, the piece will glow/move/shake to indicate that the user has selected it. The board will then highlight the possible movements that the selected piece can make, or which of the opponents pieces may be captured by the selected piece.

UC-1.1: Moving a Piece (FR5)

To move a piece, the player will be able to click on any of the highlighted sections of the board. Once the move has completed, the user will see that the selected piece has now moved positions.

UC-1.2: Capturing a Piece (FR5)

To capture an opponent’s piece, the player will first select their own piece, and any valid pieces available to capture will be highlighted. After selecting the target piece, the user's piece will reappear in the new position, and the opponent's piece will be removed from the board, signifying a successful capture.

UC-1.3: Changing Piece Selection (FR4)

After the player has selected a piece, they may choose to select another piece by simply clicking on another piece of their own. The selected piece will then switch from the former to the new selection. This will be shown by the piece's highlight changing so that the newly selected piece is now highlighted.

UC-2.0: Quitting a Game (FR9)

At any point during the game, the player may choose to exit and return to the main menu. To do this, the player can select the “Quit Game” button and will then be prompted to confirm their choice to quit. Upon confirmation, the game will save its state to be loaded at another point in time, and the user will be returned to the main menu screen.

UC-3.0: Reviewing a Previous Game (FR10)

After reaching the main menu, the user can select to view a previous game (UC-0.0). Once the user has selected the relevant save file to load into the application, the board will be set, and the pieces placed at their beginning positions. Buttons will appear on the screen to show “Step Forward” or “Step Backwards”.

UC-3.1: Stepping Forward (FR10)

When the user selects the “Step Forward” button, the next change on the game board will be executed, showing the move according to the save file.

UC-3.2: Stepping Backwards (FR10)

When the user selects the “Step Backwards” button, the last change on the game board will be reverted, showing the previous state of the board according to the save file.

UC-4.0: End Of Game (FR8)

Either player may choose to end the game at any time by clicking a “End Game” button. Upon clicking this button, the user will be notified that the game will be saved for replay at a future point.

UC-4.1: Detecting Check (FR6)

When the program detects that a king is in check, it will notify the user by highlighting the attacking piece(s), and the player's king.

UC-4.2: Detecting Checkmate (FR7)

If the program concludes that a player is in checkmate, the game will end and the program will display which of the players has won. The user will then be notified that the game will be saved for future viewing.

4. ERROR CONDITIONS

The following error conditions aim to address how to handle unintentional scenarios or illegal moves/states.

EC-0.0: Attempted Illegal Move

After the user has selected a piece (UC-1.0), they will be shown valid positions to move the piece to. If the player attempts to select a position that is invalid for any reason, the application will highlight the target position chosen by the user in red, rather than a positive colour. The highlight will then fade, and the board will not have changed.

EC-0.1: Attempted Illegal Capture

After the user has selected a piece (UC-1.0), they will be shown valid positions to move the piece to. If the player attempts to capture an opponent's piece that cannot be captured for any reason, the application will highlight the target piece chosen by the user in red, rather than a positive colour. The highlight will then fade, and the board will not have changed.

EC-0.2: Attempted Illegal Castling

After the user has selected the piece to move (the Castle or King), it will be highlighted as described in UC-1.0. If the user then selects a Castle or King (whichever is to be moved) that cannot be castled, the second selected piece will highlight in red, showing that this is an invalid move.

EC-1.0: Resuming a Corrupted or Invalid Game

If the user attempts to view a previously saved game file that is corrupted or contains invalid data and replay it, the game will be unaware of this until it encounters the error. Upon triggering this error, the user will be notified that there was an issue viewing the game, and a button to acknowledge this will be shown. Upon acknowledging the message, the user will be returned to the main menu.

REFERENCES

[1] QA Document SE.QA.04 - User Interface Specification Standards

DOCUMENT HISTORY

Version	Date	Changes made to document	Changed by
0.1	15/02/23	Document created.	TYW1
0.2	20/02/23	Incl. references to functional requirements.	TYW1
0.3	20/02/23	Updated authors of document. General formatting.	TYW1
0.4	22/02/23	Removed duplicate use case (UC-0.3).	TYW1