Software Engineering Group Project
Use Case Document

Author:	Archie Malvern [arm36]
Config. Ref.:	UseCaseGroup18
Date:	2023-02-20
Version:	0.4
Status:	Draft

Department of Computer Science,
Aberystwyth University,
Aberystwyth,
Ceredigion, SY23 3DB,
U.K.

CONTENTS

1 INTRODUCTION

1.1 Purpose of this Document

This document aims to specify the use cases of the chess tutor program, to assist in the design of it's user interface.

1.2 Scope

Document includes each step different types of users would take during operation of the program, hypothetical background information for said users, and potential error conditions that should be considered in the design of the user interface. This document was produced following the User Interface Specification Standards [1], and is based on the LATEXtemplate [2].

1.3 Objectives

The objective of this document is to assist in the design and development of the user interface for the chess tutor, ensuring that a wide variety of use cases and error conditions are accounted for.

2 TYPICAL USERS

2.1 Casual Chess Players

This includes players that are completely new to the game or do not play often. These players would want to learn the rules of the game, or at least gain a better understanding. Two subclasses of this general class could be users that are completely new and need to learn the rules from scratch, and users that have played the game before and have a rough understanding of the rules, but may need a reminder or may not be familiar with certain obscure rules. Below is a hypothetical example of a typical user of this class:
Sarah and Gareth are both casual chess players. Sarah has played before, and has a rough understanding of the game's rules, while Gareth has never played before. Sarah uses the tutor's display to quickly remind herself which moves are valid with which pieces, and to learn about new manoeuvres that are otherwise quite obscure. Gareth carefully looks at the display to work out which moves he can use with which pieces to memorise the roles of each different piece.

2.2 Advanced Chess Players

This includes players that play the game frequently, and are highly competent at chess. These players would want to use the program as a standard chess program, perhaps using it to remind themselves of some more obscure rules if need be. Below is a hypothetical example of a typical user of this class:
Morgan and Sam are two competitive chess players. They use the program to play a game of chess against each other, then use the replay functions to analyse their strategies and improve their performance in future matches. They also upload their configuration files online for other players to view and take inspiration for their own chess matches.

3 USE CASES

3.1 UC-1.1 Starting a game

One player opens the program and clicks the *New Game* menu option. One player enters their name and picks whether they're playing white or black, with the option of a random choice and whether they would like to enable tutorials or not, and clicks *Next*. There are also accessibility options for colour-blindness on these screens. The other player types their name and selects the opposite colour to the first player, as the colour the first player picked is blanked out. The players are then presented with a chess board interface.

3.2 UC-1.2 Playing a game

The player who has selected white goes first in accordance to the rules of chess, and selects a piece. The potential spaces said piece can move are highlighted in blue, while any pieces that can be taken have a green background (with the exception of certain colour-blind modes). The player can then click on one of these spaces to move the piece. If the player selects an invalid space, the piece will not move, and a warning for an invalid move will be displayed. The player's name is shown in the corner, with the font colour matching that of the colour of their pieces, alongside the turn number. When a player is in check, it highlights the background of the king piece alongside the piece putting it in check in red (with the exception of certain colour-blind modes); the same is done for checkmate, but the game is ended and the word *Checkmate* is displayed on screen, alongside the name of the winner, i.e. "*Checkmate! Mary Wins!*".

3.3 UC-1.3 Continuing a game

A player can click the *Continue* menu option when the program starts to view a list of previously saved games. The player can select one of these files and click *Load* to load the game back to it's exact previous state, or select *Erase* to delete a save.

3.4 UC-1.4 Viewing a previous game

A player can select the *Replay* option to view a list of previous games. The player can then select one and click the *View* button to open a chess board interface with a *Pause*, *Play*, *Previous* and *Next* button. The *Play* button plays each turn in order, the *Pause* button pauses on the current turn, the *Previous* button rewinds to a previous turn and the *Next* button advances to the next turn.

3.5 UC-1.5 Exiting the game

The player can exit a running game by clicking on the cross icon in the top right corner of the program window. A pop up dialogue will ask the player to confirm they would like to quit, and if the *Confirm* button is clicked, the program will cease.

4 ERROR CONDITIONS

4.1 EC-1.1 The player attempts to move a piece to an invalid space

This error is triggered when the player tries to make an invalid move. The piece snaps back into it's previous position. A message is not necessary for this error.

4.2 EC-1.2 Two players pick the same name

When two players try to pick the same name, an error message is displayed telling the user to pick another name in order to avoid conflicts.

4.3 EC-1.3 File reading error

If the save file being read in is invalid, an error message is displayed then the program ceases to read in the file and returns to the menu.

4.4 EC-1.4 Player tries go to a turn that does not exist

If the player selects *Next* on the last turn of playback, or *Previous* on the first turn, an error message is displayed to inform the user that such a turn does not exist.

REFERENCES

[1] *Software Engineering Group Projects* User Interface Specification Standards. C. W. Loftus, SE.QA.04. 1.4 Release.

[2] *Software Engineering Group Project* LaTeX Document Example. N.W. Hardy, C.W. Loftus, SE-N66-TEST. 1.4 Release.

DOCUMENT HISTORY

Version	Issue No.	Date	Changes made to Document	Changed by
0.1	N/A	2023-02-07	Initial draft document	arm36
0.2	N/A	2023-02-07	Added an additional use case	arm36
0.3	N/A	2023-02-14	Added some early error conditions	arm36
0.4	N/A	2023-02-20	Made the document more consistent with the presentation	arm36