Group 17

Test Specification

|  |  |
| --- | --- |
| Author: | George Cooper [gwc1] |
| Config Ref: | SE\_GP17\_TestSpecificiation\_01 |
| Date: | 26th February 2023 |
| Version: | 0.5 |
| Status: | Draft |

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Copyright © Aberystwyth University 2021

CONTENTS

CONTENTS 2

1. Introduction 3

1.1 Purpose of this Document 3

1.2 Scope 3

1.3 Objectives 3

2. TEST SPECIFICATION 3

REFERENCES 3

DOCUMENT HISTORY 3

# Introduction

## Purpose of this Document

The purpose of this document is to describe all system tests to be executed for the Chess Tutor Project for the Software Engineering Group Project 2022/23, designed and implemented by Group 17. This document will provide a comprehensive testing guide that will be used to produce a Test Report.

## Scope

This document features a comprehensive testing guide which describes the test reference, the functional requirement the test covers, the content of the test, the inputs, expected outputs, pass criteria and any draft notes. This document follows the format covered in SE.QA.06 and SE.QA.02.

This document should be read by all project members. It is assumed that the reader is already familiar with the QA Plan and UI Specification.

## Objectives

This document is intended to:

* Provide a testing plan to be used by Group 17 to test the group project application for 2022/23 (Chess Tutor).
* Provide reproducible tests, the results of which will be entered into a Test Report to be submitted along with the Final Report.
* Provide tests that cover all functional requirements, external interface requirements, performance requirements and design constraints (outlined in SE.QA.06).

# TEST SPECIFICATION

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Ref** | **Requirement** | **Test Content** | **Input** | **Output** | **Pass Criteria** | **Draft Notes** |
| **SE-F-001** | FR1 | Menu Prompts are displayed upon loading the application | Open the "Chess Tutor" application | All menu prompts are displayed on the screen | All menu prompts are displayed without issue |  |
| **SE-F-002** | FR1 | "New Game" prompt can be selected | Select "New Game" prompt | Player setup screen appears for a new game | "New Game" prompt can be selected without issue |  |
| **SE-F-003** | FR1 | Player setup prompts are displayed | Select "New Game" prompt | Setup prompts are displayed on the screen | All player setup prompts are displayed without issue |  |
| **SE-F-004** | FR1 | New Game setup allows player name entries within character limit | Enter a valid player name ("Dustin") within the character limit | User can progress through the setup | User is able to progress with the valid player’s name | Decide on character limit |
| **SE-F-005** | FR1 | New Game setup does not allow an empty name entry | Enter no value into the player’s name ("") | User is prompted to enter a valid player name. | User is unable to progress with the invalid player’s name |  |
| **SE-F-006** | FR1 | New Game setup does not allow player name entries above the character limit | Enter a valid player name ("…") over the character limit | User is prompted to enter a valid player name | User is unable to progress with the invalid player’s name | Decide on Character limit |
| **SE-F-007** | FR1 | New game setup allows one player to select a colour | Select a piece colour for one player | Selected player is given the selected colour. Another player is given the opposing colour | User can progress with the chosen colours |  |
| **SE-F-008** | FR1 | New game setup does not allow an "Empty" colour choice | Do not select any colour for either player | User is prompted to select a colour option for a player | User is unable to progress with the invalid colour option |  |
| **SE-F-009** | FR1 | "Play/Confirm" prompt can be selected | Select "Play/Confirm" prompt with valid setup Inputs (SE-F-004, SE-F-007) | The new game initialises and the chess board is displayed. | "Play/Confirm" prompt can be selected without issue | Decide on "Play/confirm" prompt label |
| **SE-F-010** | FR1 | "Load Game" prompt can be selected if a save file is available | Select "Load Game" prompt with valid save files available | The "Load Game" menu is displayed on the screen | "Load Game" prompt can be selected without issue |  |
| **SE-F-011** | FR1 | "Load Game" prompt cannot be selected if no save files are available | Select "Load Game" prompt with no save files available | User is prompted that no save files are available | "Load Game" menu does not load, and error message is displayed |  |
| **SE-F-012** | FR1 | "Exit" prompt can be selected without issue | Select "Exit" prompt | "Chess Tutor" application closes | The application closes |  |
| **SE-F-013** | FR1 | "Back/Return" prompt can be selected without issue | Select "Back/Return" prompt | The previous screen is displayed | The user is navigated back to the previous screen | Decide on "Back/Return" prompt label |
| **SE-F-014** | FR2 | Check player name entries are kept track of in save file | Setup a new game (SE-F-009) | Name entries can be observed | Name entries are stored within the game |  |
| **SE-F-015** | FR2 | Check player colour is kept track of in save file | Setup a new game (SE-F-009) and choose colour | Colour data is stored in save file | Colour data is present in file and correct |  |
| **SE-F-016** | FR2 | Check the players pieces are kept track of (initial state) | Setup a new game (SE-F-009) and save the game | Player pieces still in play for each player stored in save file | All player pieces should be stored in save file as still in play as none have been taken |  |
| **SE-F-017** | FR2 | Check the players pieces are kept track of after a piece is taken | Setup a new game (SE-F-009), take a piece and save the game | Player pieces still in play for each player stored in save file | All pieces still in play should be stored in the save file |  |
| **SE-F-018** | FR2 | Check the initial position of the pieces is correct | Setup a new game (SE-F-009), save the game | Player piece position stored in the save file | The initial positions of the pieces are correct |  |
| **SE-F-019** | FR3 | Check white moves first | Setup a new game (SE-F-009) | The indicator should show it is whites move | Program indicates it is whites turn |  |
| **SE-F-020** | FR3 | Check program indicates whose turn it is | Setup a new game (SE-F-009) | The indicator should change at the end of each turn and show the which player needs to move next | The player indicated to make the next move should be correct |  |
| **SE-F-021** | FR3 | Check the pieces are presented in an unambiguous way | Setup a new game (SE-F-009) | The board should be displayed with the correct textures for each piece | Check that the white and black textures are correct and easy to tell apart |  |
| **SE-F-022** | FR4 | Check that only the current player's pieces can be selected | Setup a new game (SE-F-009), try and select a black piece, then try and select a white piece | The program should note display/allow the player to select a black piece as it is white's turn. The white piece should be selected. | Black piece is not allowed to be selected; the white piece should be allowed to be selected |  |
| **SE-F-023** | FR4 | Check that the player can select different pieces during their turn | Setup a new game (SE-F-009), try and select a white piece, then try and select another white piece. | The program should display the first piece as selected, when the second piece is selected, the first piece should become unselected. | Both pieces should show as selected. |  |
| **SE-F-024** | FR4 | Check the pieces change appearance when selected | Setup a new game (SE-F-009), select a white piece | The white piece should be highlighted when selected. | White piece highlight should display correctly. |  |
| **SE-F-025** | FR4 | Check the player can’t select a piece after they’ve moved another piece that turn. | Setup a new game (SE-F-009), move a piece, then try and select another piece. | Error message informs the player they have already moved a piece. | The program should not allow the user to select a piece after they’ve moved one on their turn. |  |
| **SE-F-026** | FR5 | Check that only the current player can move a piece | Start a new game, try and move a black piece. | Error message should inform the player that only white can move | You should only be able to move white’s pieces as white moves first |  |
| **SE-F-027** | FR5 | Check that all moves are displayed when a pawn is selected | Select a pawn piece during a game, check that all moves are accounted for including moving 2 spaces during a pawn’s first move and en passant | The program should highlight all moves | All moves are displayed |  |
| **SE-F-028** | FR5 | Check all pawn moves are legal | Select a pawn piece during a game, check that all moves are accounted for including moving 2 spaces during a pawn’s first move and en passant | All highlighted moves should be legal | Only legal moves are displayed |  |
| **SE-F-029** | FR5 | Check that all legal moves are displayed when a rook is selected | Select a rook piece during a game | The program should highlight all moves | All moves are displayed |  |
| **SE-F-030** | FR5 | Check all rook moves are legal | Select a rook piece during a game | All highlighted moves should be legal | Only legal moves are displayed |  |
| **SE-F-031** | FR5 | Check that all legal moves are displayed when a knight is selected | Select a knight piece during a game | The program should highlight all moves | All moves are displayed |  |
| **SE-F-032** |  | Check all knight moves are legal | Select a knight piece during a game | All highlighted moves should be legal | Only legal moves are displayed |  |
| **SE-F-033** | FR5 | Check that all legal moves are displayed when a bishop is selected | Select a bishop piece during a game | The program should highlight all moves | All moves are displayed |  |
| **SE-F-034** | FR5 | Check all bishop moves are legal | Select a bishop piece during a game | All highlighted moves should be legal | Only legal moves are displayed |  |
| **SE-F-035** | FR5 | Check that all legal moves are displayed when a queen is selected | Select a queen piece during a game | The program should highlight all moves | All moves are displayed |  |
| **SE-F-036** | FR5 | Check all queen moves are legal | Select a queen piece during a game | All highlighted moves should be legal | Only legal moves are displayed |  |
| **SE-F-037** | FR5 | Check that all legal moves are displayed when a king is selected | Select a king piece during a game | The program should highlight all moves | All moves are displayed |  |
| **SE-F-038** | FR5 | Check all king moves are legal | Select a king piece during a game | All highlighted moves should be legal | Only legal moves are displayed |  |
| **SE-F-039** | FR5 | Check left side castling rule is implemented correctly | Perform a left-side castle | The board should update to reflect the move | The move should work as explained in the chess rules [1] |  |
| **SE-F-040** | FR5 | Check right side castling rule is implemented correctly | Perform a right-side castle | The board should update to reflect the move | The move should work as explained in the chess rules [1] |  |
| **SE-F-041** | FR5 | Check en passant rule is implemented correctly | Perform an en passant move | The board should update to reflect the move | The move should work as explained in the chess rules [1] |  |
| **SE-F-042** | FR5 | Check pieces can’t move off the board | Attempt to move a piece of the board | An error message informing the player they can not move a piece outside the board should appear | The program should not allow the player to move a piece off the board |  |
| **SE-F-043** | FR6 | Check that check is correctly detected for the white king | Put a player into check as black | A red square should appear around the check king | The program should detect the player is in check |  |
| **SE-F-044** | FR6 | Check that check is correctly detected for the black king | Put a player into check as white | A red square should appear around the check king | The program should detect the player is in check |  |
| **SE-F-045** | FR6 | Check that a red square appears when the white king is put into check | Put a player into check as black | A red square should appear around the check king | A red square should appear around the check king |  |
| **SE-F-046** | FR6 | Check that a red square appears when the black king is put into check | Put a player into check as white | A red square should appear around the check king | A red square should appear around the check king |  |
| **SE-F-047** | FR7 | The checkmate function is called when the white King is in checkmate | Person playing white has checkmated the black King | The game will end | The game ends and cannot be continued |  |
| **SE-F-048** | FR7 | The checkmate function is called when the black King is in checkmate | Person playing black has checkmated the white King | The game will end | The game ends and cannot be continued |  |
| **SE-F-049** | FR7 | A red square will appear around the White King when checkmated | The Black player checkmates the White player | The square the White King is in will be highlighted in red | The square the White King is on turns red and the game ends |  |
| **SE-F-050** | FR7 | A red square will appear around the Black King when checkmated | The White player checkmates the Black player | The square the Black King is in will be highlighted in red | The square the Black King is on turns red and the game ends |  |
| **SE-F-051** | FR8 | The game ends when the program detects White has checkmated the Black King | White player checkmates the Black King | The game will end | The game ends and cannot be continued |  |
| **SE-F-052** | FR8 | The game ends when the program detects Black has checkmated the White King | Black player checkmates the White King | The game will end | The game ends and cannot be continued |  |
| **SE-F-053** | FR8 | The name of the winning player is displayed in the 'Game Over' screen | One player checkmate's the other's King | The 'Game Over' screen will display the correct players name | The winner's name is displayed |  |
| **SE-F-054** | FR8 | The option to start a new game will appear in the "Game Over" screen | One player checkmate's the other's King | The 'Game Over' screen will display the option to start a new game | A function button to start a new game is shown |  |
| **SE-F-055** | FR8 | The game ends when White chooses to resign | The White player pauses the game and chooses to resign | The game will be saved and end | The game is saved and can be continued when the players decide to resume it |  |
| **SE-F-056** | FR8 | The game ends when Black chooses to resign | The Black player pauses the game and chooses to resign | The game will be saved and end | The game is saved and can be continued when the players decide to resume it |  |
| **SE-F-057** | FR8 | White can call a draw and Black can accept | The White player pauses and calls a draw, and the Black player can accept | The draw is accepted and the game ends | A 'Game Over' screen shows that the game ended in a draw |  |
| **SE-F-058** | FR8 | Black can call a draw and White can accept | The Black player pauses and calls a draw, and the White player can accept | The draw is accepted and the game ends | A 'Game Over' screen shows that the game ended in a draw |  |
| **SE-F-059** | FR8 | White can call a draw and Black can decline | The White player pauses and calls a draw, and the Black player can decline | The game continues | The game will be un-paused and continue |  |
| **SE-F-060** | FR8 | Black can call a draw and White can decline | The Black player pauses and calls a draw, and the White player can decline | The game continues | The game will be un-paused and continue |  |
| **SE-F-061** | **FR9** | White player can choose to quit the game at any time | The White player pauses the game and chooses to quit | The game will end | The menu to choose whether to save the game or not appears |  |
| **SE-F-062** | **FR9** | Black player can choose to quit the game at any time | The Black player pauses the game and chooses to quit | The game will end | The menu to choose whether to save the game or not appears |  |
| **SE-F-063** | **FR9** | The menu to choose to save the current game or not appears | Black player chooses to quit the game | The correct menu appears | The correct menu appears |  |
| **SE-F-064** | **FR9** | The menu to choose to save the current game or not appears | White player chooses to quit the game | The correct menu appears | The correct menu appears |  |
| **SE-F-065** | **FR9** | The White player can select the option to save the game | White player chooses to save the game | The game is saved | The game data appears in the .txt file |  |
| **SE-F-066** | **FR9** | The Black player can select the option to save the game | Black player chooses to save the game | The game is saved | The game data appears in the .txt file |  |
| **SE-F-067** | **FR9** | The White player can select the option to not save the game | The White player chooses to not save the game | The game is not saved, and the player is returned to the main menu | The game data does not appear in the .txt save file |  |
| **SE-F-068** | **FR9** | The Black player can select the option to not save the game | The Black player chooses to not save the game | The game is not saved, and the player is returned to the main menu | The game data does not appear in the .txt save file |  |
| **SE-F-069** | **FR9** | Can the White player exit the pause menu | The White player chooses to exit the pause menu | The pause menu disappears | The game is resumed |  |
| **SE-F-070** | **FR9** | Can the Black player exit the pause menu | The Black player chooses to exit the pause menu | The pause menu disappears | The game is resumed |  |
| **SE-F-071** | **FR10** | Does the option to replay a previous game appear | The player opens the main menu | The option to replay a previous game appears | The option to replay a previous game appears |  |
| **SE-F-072** | **FR10** | Are all saved games shown | The player selects the option to replay a game | The list of saved games is shown | All saved games are shown | UI Spec needs to include how the list of saved games will look. |
| **SE-F-073** | **FR10** | Can the player select a game to load | The player clicks on a past game to be loaded | The game loads correctly | All the pieces are in the correct positions |  |
| **SE-F-074** | **FR10** | Are all of the pieces in the correct positions | A player chooses to load the game | The game is loaded | All the pieces are in the correct positions |  |
| **SE-F-075** | **FR10** | Are all of the pieces that were removed from the board still removed | A player chooses to load the game | The game is loaded | All the pieces that were removed are still removed |  |
| **SE-F-076** | **FR10** | Does the game load the JavaFX for if the White King is in check | A player chooses to load the game | The game is loaded | The JavaFX for a King in check is loaded for the White King |  |
| **SE-F-077** | **FR10** | Does the game load the JavaFX for if the Black King is in check | A player chooses to load the game | The game is loaded | The JavaFX for a King in check is loaded for the Black King |  |
| **SE-F-078** | **FR10** | Does the game load the correct player names assigned to the correct colors | A player chooses to load the game | The game is loaded | The players will have the same names as when they last played the game |  |
| **SE-F-079** | **FR10** | Can the White player choose to move forwards and backwards through the game on their turn | The White player chooses to move forwards or backwards | The game undoes or redoes a turn based on the White players input | The action performed matches the correct input | UI Spec needs to include how the option to undo/redo a turn will look like |
| **SE-F-080** | **FR10** | Can the Black player choose to move forwards and backwards through the game on their turn | The Black player chooses to move forwards or backwards | The game undoes or redoes a turn based on the Black players input | The action performed matches the correct input | UI Spec needs to include how the option to undo/redo a turn will look like |
| **SE-F-081** | **FR10** | Can the White player choose to move forwards and backwards through the game when it's not their turn | The White player chooses to undo or redo an action when it is the Black player's turn | Nothing happens | The White player is unable to undo/redo an action while it is the Black player's turn | UI Spec needs to include how the option to undo/redo a turn will look like |
| **SE-F-082** | **FR10** | Can the Black player choose to move forwards and backwards through the game when it's not their turn | The Black player chooses to undo or redo an action when it is the White player's turn | Nothing happens | The Black player is unable to undo/redo an action while it is the White player's turn | UI Spec needs to include how the option to undo/redo a turn will look like |
| **SE-F-083** | **FR11** | The game is saved in its entirety when a player chooses to quit | A player quits the game | The game data is written to a .txt file | The correct data is written and saved to the .txt file |  |
| **SE-F-084** | **FR11** | The game is saved and does not overwrite another saved game | A player quits the game | The game data is written to a .txt file | The game is added to a .txt file and all other game saves are still in the same file |  |
| **SE-F-085** | **FR11** | Is the game saved after every player turn | A player performs a move | The next player can begin their turn | The game data has been updated in the .txt file |  |
| **SE-F-086** | **FR11** | If the program is closed during a game, was the last move saved | A player performs a move and then closes the game without pausing and choosing to quit | The game is closed and must be reopened | The previous game is shown in the list of saved games, and everything s correct when loaded (refer to previous tests for FR10 and FR11) |  |
| **SE-F-087** | **FR11** | If the program crashes during a game, was the last move saved | The player performs a move and the game crashes | The game is closed and must be reopened | The previous game is shown in the list of saved games, and everything is correct when loaded (refer to previous tests for FR10 and FR11) |  |

REFERENCES

[1] Wikipedia (2022), “Rules of chess” (Online) https://en.wikipedia.org/wiki/Rules\_of\_chess  
(Accessed 26th February 2023)

[2] QA Document SE.QA.06 – Test Procedure Standards.

[3] QA Document SE.QA.02 – General Documentation Standards.

[4] User Interface Presentation – SE\_GP17\_UIPresentation\_01

[5] User Interface Specification – SE\_GP12\_UISpecification\_01

DOCUMENT HISTORY

| *Version* | *Issue No.* | *Date* | *Changes made to document* | *Changed by* |
| --- | --- | --- | --- | --- |
| 0.1 | 10 | 26/02/23 | N/A - original draft | GWC1 |
| 0.2 | 10 | 27/02/23 | Added tests for Functional Requirements 7, 8, 9, 10 and 11 | KIF11 |
| 0.3 | 10 | 27/02/23 | Filled in section 1 | GWC1 |
| 0.4 | 10 | 28/02/23 | Removed duplicate tests | GWC1 |
| 0.5 | 10 | 28/02/23 | Added more content to Introduction. Added References 2-5. Fixed spelling errors and sections with wrong font/font size. | KIF11 |
|  |  |  |  |  |