Software Engineering Group Project - Test Specification

| Author(s): | Craymon Chan [ckc2@aber.ac.uk]  Jim Brown [jib16@aber.ac.uk] |
| --- | --- |
| Config Ref: | Test-Spec-GP9 |
| Date: | 07 March 2023 |
| Version: | 0.5 |
| Status: | Review |

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

# CONTENTS

[**CONTENTS 2**](#_h1evqmk3o7mz)

[**1. INTRODUCTION 3**](#_u9fr9t8g3lds)

[1.1 Purpose of this document 3](#_1eyjjgkh8wf4)

[1.2 Scope 3](#_zhtapykiyhsd)

[1.3 Objectives 3](#_eyio7d7yvzia)

[**2. TEST SPECIFICATION 4**](#_raieuj923qb)

[**3. REFERENCES 19**](#_3iq0azfr34na)

[**4. DOCUMENT HISTORY 19**](#_m4u7vnsmocd5)

# 1. INTRODUCTION

## 1.1 Purpose of this document

This document describes all aspects of testing to be performed on the Chess Tutor project and ensures that functional requirements perform as expected.

## 1.2 Scope

This document is relevant to all team members that are involved in the development of the software (“the code”). This document helps to describe and outline the required modules that must be tested to ensure completion of the project.

It is assumed that the reader is already aware of the details laid out in the following documentation:

* SE.QA.RS-CS22120 - Chess Tutor Requirements Specification [1]
* SE.QA.06 - Test Procedure Standards [2]
* UI-Spec-Docu-GP9 - User Interface Specification [3]
* UI-Spec-Pres-GP9 - User Interface Presentation [4]

## 1.3 Objectives

The objective of the document is to:

1. Provide a detailed description of how testing will function throughout the project to ensure that all requirements are met, and function as expected.
2. Serve as a guide to assist development of the final solution, and provide criteria against which to test said solution.
3. Describe the purposes of various tests while identifying the required inputs and expected outputs, procedures for executing tests, and configuration of fail or pass criteria in order to determine acceptance of the project.

# 2. TEST SPECIFICATION

| **Test Ref** | **Requirement Being tested** | **Test Content** | **Input** | **Output** | **Pass Criteria** |
| --- | --- | --- | --- | --- | --- |
| **SE-F-001** | FR1 | Check whether new game works properly | Click ‘New Game’ | New game is started | New game is started |
| **SE-F-002** | FR1 | Check whether ‘Load Game’ works properly | Click ‘Load Game’ | Previous game is loaded | Previous game is loaded successfully |
| **SE-F-003** | FR1 | Check whether ‘View Game’ works properly | Click ‘View Game’ | Allows player to view their previous game | Previous game is enabled to view |
| **SE-F-004** | FR1 | Check whether the players entered their names after being prompted | Insert players’ names into the provided tab | Player’s names are displayed | Players names are confirmed |
| **SE-F-005** | FR1 | Check whether invalid names are allowed | When players entered their names | Invalid names are not displayed | Players names are invalid |
| **SE-F-006** | FR2 | Check whether White side is at the bottom, whilst Black side is at the top | Entering names for black and white sides | Board display as white side as bottom whilst black side at top | Players will be shown that white side is at the bottom whilst black side is at the top |
| **SE-F-007** | FR2 | Both players names, pieces and colour is tracked correctly | Makes a move | Updated board | The players name should match the pieces and colour |
| **SE-F-008** | FR3 | Tutor keeps track of current starting player as in white side | Starting the game | White side pieces ready to be moved | Starting player and pieces must be from white side |
| **SE-F-009** | FR3 | Tutor keeps track of current player and shows visually whos turn it is | New turn | Changed turn | Changes turn and has a visual to show whos turn it is |
| **SE-F-010** | FR3 | Unambiguous board representation | New game and move a piece | Board clearly displayed | Unambiguous board and pieces represented |
| **SE-F-011** | FR4 | Only current player can move there pieces | Move piece from other player and current player | Players moving pieces | Only current player can move there coloured pieces |
| **SE-F-012** | FR4 | Check if piece selection works | Click piece that is available to move | Available piece moves are shown | All available positions should be shown to move to |
| **SE-F-013** | FR4 | Check if piece selection works | Check piece that cannot move as it is pinned | Available piece moves are shown | No available moves are shown that would move piece so it’s no longer blocking check |
| **SE-F-014** | FR4 | Check if piece selection works | Check piece that cannot move as it is check | Available piece moves are shown | No available moves are shown that aren’t preventing check (ie: capturing the checking piece) |
| **SE-F-015** | FR4 | Check if piece reselection works | Select a piece and then reselect another piece | The new piece should be selected and the old piece should deselect | The new piece's moves should not be shown and the old piece should not have its moves shown. |
| **SE-F-016** | FR5 | Check whether Pawn Pieces are moving legally | Clicking on Pawn Pieces  To only move vertically one or two steps as any legal moving spaces are highlighted | Pawn Pieces  should only move one or at most two steps | Pawn Pieces moved correctly |
| **SE-F-017** | FR5 | Test functionality of Pawn Pieces for capturing pieces | Pawn Pieces could only move towards enemy pieces as in diagonally to the left or right while any enemy pieces that are allowed to be captured are highlighted | Pawn Pieces could only capture enemy pieces diagonally and then landing on its designated location | Pawn Pieces attacked correctly |
| **SE-F-018** | FR5 | Test Pawn Pieces applying En Passant rule which only occurs when a pawn captures a horizontally adjacent enemy pawn piece that made an initial two square advance | Pawn Pieces could only capture enemy pawn pieces if it advances two square which its position is adjacent to the player’s pawn piece and the space is highlighted | Pawn Pieces could only capture enemy pawn pieces if it met the requirements of En Passant | Pawn Pieces applying the En Passant rule |
| **SE-F-019** | FR5 | Check if players intend to move Pawn Pieces into restricted spaces | Players attempting to move Pawn Pieces into illegal spaces | Pawn Pieces return back to its initial moving space and flashes red | Pawn Pieces flashes red and is not moved |
| **SE-F-020** | FR5 | Check Pawns cannot capture illegally | Click Pawn and then attempt to move across the board | Pawn piece flashes red | Pawn piece flashes red and is not moved |
| **SE-F-021** | FR5 | Check whether Bishop Pieces are moving legally | Clicking on Bishop Pieces to only move diagonally as many steps as desired as any legal moving spaces are highlighted | Bishop Pieces should only move diagonally | Bishop Pieces moved correctly |
| **SE-F-022** | FR5 | Test functionality of Bishop Pieces for capturing enemy pieces | Bishop Pieces could only move diagonally in any steps while any enemy pieces that are allowed to be captured are highlighted | Bishop Pieces could only capture enemy pieces diagonally and then landing on its designated location | Pawn Pieces attacked correctly |
| **SE-F-023** | FR5 | Test functionality of Bishop making illegal move | Move bishop piece horizontally | Bishop piece flashes red | Bishop piece flashes red and is not moved |
| **SE-F-024** | FR5 | Check whether Knight Pieces are moving legally on the top side | Clicking on Knight Pieces to only move one space left or right horizontally and then two spaces up or down vertically as any legal moving spaces are highlighted | Knight Pieces should only move as in a ‘L shape’ | Knight Pieces moved correctly  (top side) |
| **SE-F-025** | FR5 | Check whether Knight Pieces are moving legally on the bottom side | Clicking on Knight Pieces to only move two spaces left or right horizontally and then one space up or down vertically as any legal moving spaces are highlighted | Knight Pieces should only move as in a ‘L shape’ | Knight Pieces moved correctly  (bottom side) |
| **SE-F-026** | FR5 | Test functionality of Knight Pieces for capturing enemy pieces on the top side | Knight Pieces could only move one space left or right horizontally and then two spaces up or down vertically while the enemy pieces that are allowed to be captured are highlighted | Knight Pieces could only capture enemy pieces after attacking and then landing on its designated location | Knight Pieces attacked  Correctly  (top side) |
| **SE-F-027** | FR5 | Test functionality of Knight Pieces for capturing enemy pieces on the bottom side | Knight Pieces could only move two spaces left or right horizontally and then one space up or down vertically while the enemy pieces that are allowed to be captured are highlighted | Knight Pieces could only capture enemy pieces after attacking and then landing on its designated location | Knight Pieces attacked correctly  (bottom side) |
| **SE-F-028** | FR5 | Test functionality of knight piece making illegal move | Move knight to across the board to a square it shouldn’t be able to go | Knight piece should not be moved | Knight piece should flash red and not move |
| **SE-F-029** | FR5 | Check whether Rook Pieces are moving legally as in vertically | Clicking on Rook Pieces to only move vertically for as far as desired as any legal moving spaces are highlighted | Rook Pieces  should only move vertically as in a straight line | Rook Pieces moved correctly  (vertically) |
| **SE-F-030** | FR5 | Check whether Rook Pieces are moving legally as in horizontally | Clicking on Rook Pieces to only move horizontally for as far as desired as any legal moving spaces are highlighted | Rook Pieces should only move horizontally as in a straight line | Rook Pieces moved correctly  (horizontally) |
| **SE-F-031** | FR5 | Test functionality of Rook Pieces for capturing enemy pieces vertically | Rook Pieces could only move vertically for as many spaces as desired while the enemy pieces that are allowed to be captured are highlighted | Rook Pieces could only capture enemy pieces in a vertical direction and then landing on its designated location | Rook Pieces attacked correctly  (vertically) |
| **SE-F-032** | FR5 | Test functionality of Rook Pieces for capturing enemy pieces horizontally | Rook Pieces could only move horizontally for as many spaces as desired while the enemy pieces that are allowed to be captured are highlighted | Rook Pieces could only capture enemy pieces in a horizontal direction and then landing on its designated location | Rook Pieces attacked correctly  (horizontally) |
| **SE-F-033** | FR5 | Test functionality of rook moving illegally | Attempt to move rook piece diagonally | Rook piece flashes red and isn't moved | Rook piece flashes red and isn’t moved |
| **SE-F-034** | FR5 | Check whether Queen Piece is moving legally | Clicking on Queen Piece to move in any directions and any distance as in vertically, horizontally and diagonally as any legal moving spaces are highlighted | Queen Pieces could move into any highlighted space desired | Queen Piece moved correctly |
| **SE-F-035** | FR5 | Test functionality of Queen Piece for capturing enemy pieces | Queen Piece could move in any direction and any distance regardless of vertical, horizontal or diagonal while any enemy pieces that are allowed to be captured are highlighted | Queen Piece could move in any direction and any distance regardless of vertical, horizontal or diagonal to capture any enemy pieces and then landing on the designated location | Queen Piece attacked correctly |
| **SE-F-036** | FR5 | Test functionality of queen piece making illegal move | Move queen piece to a position is shouldn’t be able to move | Queen piece should flash red and not move | Queen piece should flash red and not move |
| **SE-F-037** | FR5 | Check whether King Piece is moving legally | Clicking on King Piece to only move one space in any direction as any legal moving spaces are highlighted | King Piece should only move one space in any direction | King Piece moved correctly |
| **SE-F-038** | FR5 | Test functionality of King Piece for capturing enemy pieces | King Piece only moves one space in any direction as vertical, horizontal or diagonal to capture enemy pieces while any enemy pieces that are allowed to be captured are highlighted | King Piece could only move in one space in any direction as vertical, horizontal or diagonal to capture enemy pieces and then landing on the designated location | King Piece attacked correctly |
| **SE-F-039** | FR5 | Test functionality of king piece moving illegally | Move king piece to across the board to a square it shouldn’t be able to move | King piece should flash red and not move | King piece should not move and flash red |
| **SE-F-040** | FR5 | Test functionality of king piece moving itself into check | Attempt to move king piece to a square that is attacked by opponents piece | King should flash red and not move | King piece should flash red and not move as it’s an illegal move |
| **SE-F-041** | FR5 | Test functionality of kingside castling when allowed | Castling when neither the king or rook has moved yet and there is space between on the kings side | The king should move 2 spaces right from players perspective and the rook should be one square to the left of king | Should be king and rook next to each other king on the left side of rook from the player's perspective. |
| **SE-F-042** | FR5 | Test functionality of queenside castle when allowed | Castling when neither the king or rook has moved yet and there is space between on the queens side | The king should move 2 squares to the left from players perspective and the rook should be one square to the right of king | Should be king and rook next to each other with the rook on the right side of the king |
| **SE-F-043** | FR5 | Test functionality of kingside castle when not allowed because king or castle previously moved | Attempt castling when there is space to castle but the king or rook have already moved | It shouldn’t work and the king should flash red | The king flashes red and no move is made |
| **SE-F-044** | FR5 | Test functionality of queenside castle when not allowed because king or castle previously moved | Attempt castling when there is space to castle but the king or rook have already moved | It shouldn’t work and the king should flash red | The king flashes red and no move is made |
| **SE-F-045** | FR5 | Test functionality of kingside castle when not allowed because there are pieces in the way | Attempt castling when there is no space to castle however the king and rook haven't moved | It shouldn’t work and the king should flash red | The king flashes red and no move is made |
| **SE-F-046** | FR5 | Test functionality of attempting to castle when in check | Click king and then attempt either king or queenside castle | It shouldn’t work and king should flash red | It shouldn’t work and king should flash red |
| **SE-F-047** | FR5 | Test functionality of Pawn Pieces which are eligible for promotion | Clicking on Pawn Pieces which have advanced until the starting point of the enemy | display a window for players to allow them choose which piece to promote to from a Pawn Piece into a Queen, Knight, Bishop or Rook Piece | Pawn Pieces are valid to undergo the promotion process |
| **SE-F-048** | FR5 | Detects any Pawn Pieces which are on the starting line of the enemy where their promotion is mandatory | Each of the Pawn Pieces on the starting line of the enemy has the ability to promote to a Queen, Knight, Bishop or Rook Piece | Indicates the player that a Pawn Piece is ready for promotion | Players must choose between the pieces to promote from a pawn |
| **SE-F-049** | FR5 | Test error condition if piece is moved off the board | Check pieces can only stay on the allocated board by clicking on a piece and then off the board | The piece should flash red and not move. | The piece should flash red and not move, the players turn continues as the move was invalid |
| **SE-F-050** | FR6 | Tutor detects whether if the opponent’s king is currently in stalemate state | It is the opponent’s turn but player’s King Piece does not have any legal moves left | Display on the board as the opponent’s King Piece isn’t allowed to move | The player is indicated that the opponent’s King Piece is in stalemate |
| **SE-F-051** | FR6 | Tutor detects whether if the player’s king is currently in stalemate state | It is the Player’s turn but player’s King Piece does not have any legal moves left | Display on the board as the player’s King Piece isn’t allowed to move | The player is indicated that the player’s King Piece is in stalemate |
| **SE-F-052** | FR6 | Tutor detects whether if the opponent’s king is currently in the checked state | Only appears when the opponent’s King Piece is being attacked | Display opponent’s king and the attacking piece with a different colour to indicate it as in a checked state | The player is indicated that the opponent’s king is in check |
| **SE-F-053** | FR6 | Tutor detects whether if the player’s king is currently in the checked state | Only appears when the player’s King Piece is being attacked | Display player’s king and the attacking piece with a different colour to indicate it as in a checked state | The player is indicated that the player’s king is in check |
| **SE-F-054** | FR7 | Tutor detects whether if the opponent’s king is in the state of checkmate | Only appears when the opponent’s King Piece is in checked state and do not have any legal moves left | Display opponent’s king with a colour which indicates that it is in checkmate and the player has won the game | The player is indicated that the opponent’s king is in checkmate |
| **SE-F-055** | FR7 | Tutor detects whether checkmate is avoidable by moving piece to block check | Put a player in check and then have the only option to block the checkmate | Piece is moved and checkmate blocked | Checkmate is blocked and game continues |
| **SE-F-056** | FR7 | Tutor detects whether checkmate is avoidable by moving piece to capture attacking piece | Put a player in check and have the only option to capture the attacking piece to avoid checkmate | Piece is moved and opponents piece is captured, checkmate avoided | Checkmate is blocked and game continues |
| **SE-F-057** | FR7 | Tutor detects whether checkmate is avoidable by moving king out of checkmate | Put a player in check and have the only option to move the king to avoid checkmate | King is moved and checkmate is avoided | Checkmate is blocked and game continues |
| **SE-F-058** | FR7 | Tutor detects whether if the player’s king is in the state of checkmate | Only appears when the player’s King Piece is in checked state and do not have any legal moves left | Display player’s king with a colour which indicates that it is in checkmate and the opponent has won the game | The player is indicated that the player’s king is in checkmate |
| **SE-F-059** | FR8 | Stalemate | One player is in stalemate | Game finishes in a draw | Game ends and players asked if they want to save the game |
| **SE-F-061** | FR8 | Checkmate | One player is checkmated | Game finishes | Game ends and players asked if they want to save the game |
| **SE-F-062** | FR8 | Resign | Click ‘Resign’ and player is notified that the game will be saved for replay at a future point | Game finishes as one player has resigned | Game ends and players are notified that the game is saved for future purposes |
| **SE-F-063** | FR8 | Agree to finish (Offer Draw) | Click ‘Offer Draw’ and prompt to the opponent for agreement | Game finished and ends in a draw | Game ends in a draw as both players agree to a draw |
| **SE-F-064** | FR8 | Disagree to finish  (Offer Draw) | Click ‘Offer Draw’ and prompted to the opponent for disagreement | Game continues as the player takes the turn while the ‘Offer Draw’ button disappeared | Game continues as one of the players disagree to a draw |
| **SE-F-065** | FR8 | Resign the game after game is won (checkmate) | Resign the game after you have already won | Nothing as game is not running | Game does not allow you to resign as you have already won the game |
| **SE-F-066** | FR9 | Game quit | Click ‘ Save and Quit’ | Game finishes with save prompt | Game closes and players asked if they want to save the game then returned to main menu |
| **SE-F-067** | FR9 | Exit game without quitting | One player force closes the game | Game is saved for later | Game is saved although it was force closed |
| **SE-F-068** | FR10 | Game replay | Player clicks replay game | Game runs through in replay mode | Replay mode opens |
| **SE-F-069** | FR10 | Replay step forward | Click ‘Step Forward’ | Replay goes forward | Replay rolls forward |
| **SE-F-070** | FR10 | Replay step backwards | Click ‘Step Backwards’ | Replay goes backwards | Replay rolls backwards |
| **SE-F-071** | FR10 | Replaying invalid step forward | Player reached the end as the most recent move | ‘Step Forward’ button does not appear | Unable to replay steps forward |
| **SE-F-072** | FR10 | Replaying invalid step backwards | Player reached the beginning as the first move | ‘Step Backwards’ button does not appear | Unable to replay steps backwards |
| **SE-F-073** | FR10 | Attempt to replay game when there is nothing to replay | Click replay game | Program will popup and say there are no games to replay | Program will popup and say there are no games to replay |
| **SE-F-074** | FR11 | Game saved while playing | Move made | Game is saved to a file to later be either restored or replayed | If the program crashes the game can be restored. If game is saved it can be replayed |
| **SE-F-075** | FR11 | Game reloaded | Previous unfinished game selected and reloaded | Game is loaded from a file and is restored | Game is restored as it was when previously exited with the same players turn and board position |
| **SE-F-076** | FR11 | Non-Existent Game | First time the program is run there will not be previously saved games or files | ‘Continue’ button is not shown | Unable to replay an non-existent game or file |

# 

# 3. REFERENCES

| [1] | QA Document SE.QA.RS-CS22120 - Chess Tutor Requirements Specification |
| --- | --- |
| [2] | QA Document SE.QA.06 - Test Procedure Standards |
| [3] | UI-Spec-Docu-GP9 - User Interface Specification |
| [4] | UI-Spec-Pres-GP9 - User Interface Presentation |

# 4. DOCUMENT HISTORY

| Version | Issue No. | Date | Changes made to document | Changed by |
| --- | --- | --- | --- | --- |
| 0.1 | N/A | 10/02/23 | Added tests for each FR. | JIB16 |
| 0.2 | N/A | 15/02/23 | Added/modified additional tests. | CKC2 |
| 0.3 | N/A | 20/02/23 | Reformatted document structure. | CKC2 |
| 0.4 | N/A | 02/03/23 | Added error conditions. | JIB16 |
| 0.5 | N/A | 07/03/23 | Reformatted document. Prep for review. | TYW1 |