Group 9 - Test Specification / 1.0 (Draft)

Group 9 - Test Specification

Authors : Craymon Chan(ckc2@aber.ac.uk)

Jim Brown(jib16@aber.ac.uk)

Config Ref: Test-Spec-Docu-GP9

Date : 21 February 2023

Version : 1.1

Status : Draft

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

Aberystwyth University / Computer Science

Page 1 of 00

Group 9 - Test Specification / 1.0 (Draft)

# Contents

1. **Introduction**

1.1 Purpose of This Document 3

1.2 Scope 3

1.3 Objective 3

1. **Test Specfication** 3
2. **References**  3
3. **Document Change History**  3

Aberystwyth University / Computer Science

Page 1 of 00

Group 9 - Test Specification / 1.0 (Draft)

Introduction

Purpose of this document

This document contains all the tests ran on the Chess Tutor project and will be used to confirm what functional requirements are passed and failed.

Scope

This document describes every possible system test for the entire chess tutor project, serving as a guidance for creators to create a refined and immaculate procedure of tests that helps with keeping track of validity functionable software.

This document should be read by all project members. It is assumed that the reader is already familiar with the **QA Plan [1]**.

Objectives

The Objective of this document :

1. Provides a detailed description of how the testing will be running when it has been fully developed, to allow accurate estimation of the final project.
2. Further refining the testing approach and identify features to be covered by the design and all of its associated tests.
3. Specifies the purpose of a specific test while identifies the required inputs and expected results, procedures for executing the tests, and configuration of fail or pass criteria in order to determine acceptance of the project.

Aberystwyth University / Computer Science

Page 1 of 00

Group 9 - Test Specification / 1.0 (Draft)

Test Specification

| Test Ref | Req 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 ‘Restore previous game’ works properly | Click ‘Restore previous game’ | Previous game is restored | Previous game is restored successfully |
| SE-F-003 | 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-004 | 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-005 | FR3 | Unambiguous board representation | New game and move a piece | Board clearly displayed | Unambiguous board and pieces represented |
| SE-F-006 | 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-000 | 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-000 | 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-000 | 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-007 | FR5 | Check whether Pawn Pieces are moving legally | Clicking on Pawn Pieces  To only move vertically one or two steps | Pawn Pieces  should only move one or at most two steps | Pawn Pieces moved correctly |
| SE-F-000 | 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 | Pawn Pieces could only capture enemy pieces diagonally and then landing on its designated location | Pawn Pieces attacked correctly |
| SE-F-000 | 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 | 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-000 | 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-000 | 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-008 | FR5 | Check whether Bishop Pieces are moving legally | Clicking on Bishop Pieces to only move diagonally as many steps as desired | Bishop Pieces should only move diagonally | Bishop Pieces moved correctly |
| SE-F-009 | FR5 | Test functionality of Bishop Pieces for capturing enemy pieces | Bishop Pieces could only move diagonally in any steps to capture enemy pieces | Bishop Pieces could only capture enemy pieces diagonally and then landing on its designated location | Pawn Pieces attacked correctly |
| SE-F-000 | 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-010 | 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 | Knight Pieces should only move as in a ‘L shape’ | Knight Pieces moved correctly  (top side) |
| SE-F-011 | 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 | Knight Pieces should only move as in a ‘L shape’ | Knight Pieces moved correctly  (bottom side) |
| SE-F-012 | 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 landing on the enemy pieces to capture them | Knight Pieces could only capture enemy pieces after attacking and then landing on its designated location | Knight Pieces attacked  Correctly  (top side) |
| SE-F-013 | 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 landing on the enemy pieces to capture them | Knight Pieces could only capture enemy pieces after attacking and then landing on its designated location | Knight Pieces attacked correctly  (bottom side) |
| SE-F-000 | 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-014 | FR5 | Check whether Rook Pieces are moving legally as in vertically | Clicking on Rook Pieces to only move vertically for as far as desired | Rook Pieces  should only move vertically as in a straight line | Rook Pieces moved correctly  (vertically) |
| SE-F-015 | FR5 | Check whether Rook Pieces are moving legally as in horizontally | Clicking on Rook Pieces to only move horizontally for as far as desired | Rook Pieces should only move horizontally as in a straight line | Rook Pieces moved correctly  (horizontally) |
| SE-F-016 | FR5 | Test functionality of Rook Pieces for capturing enemy pieces vertically | Rook Pieces could only move vertically for as many spaces as desired to capture enemy pieces | 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-017 | FR5 | Test functionality of Rook Pieces for capturing enemy pieces horizontally | Rook Pieces could only move horizontally for as many spaces as desired to capture enemy pieces | 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-000 | 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-018 | 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 | Queen Pieces could move into any space desired | Queen Piece moved correctly |
| SE-F-019 | 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 to capture any enemy pieces | 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-000 | 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-020 | FR5 | Check whether King Piece is moving legally | Clicking on King Piece to only move one space in any direction | King Piece should only move one space in any direction | King Piece moved correctly |
| SE-F-021 | 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 | 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-000 | 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 | Kind piece should not move and flash red |
| SE-F-000 | 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-000 | 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-000 | 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-000 | 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-022 | 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 with a colour to indicate it as in a checked state | The player is indicated that the opponent’s king is in check |
| SE-F-000 | 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 user’s king with a colour to indicate it as in a checked state | The player is indicated that the player’s king is in check |
| SE-F-023 | 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 |
|  | 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 |
|  | 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 |
|  | 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-000 | 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-000 | 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-024 | FR8 | Checkmate | One player is checkmated | Game finishes | Game ends and players asked if they want to save the game |
| SE-F-025 | FR8 | Resign | One player has resigned the game | Game finishes as one player has resigned | Game ends and players asked if they want to save the game |
| SE-F-026 | FR8 | Agree to finish (Draw) | Both players agreed to end the game | Game finished | Game ends and players asked if they want to save the game |
|  | 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-027 | FR9 | Game quit | One player quits game | Game finishes with save prompt | Game closes and players asked if they want to save the game |
|  | Fr9 | Exit game without quiting | One player force closes the game | Game is saved for later | Game is saved although it was force closed |
| SE-F-028 | FR10 | Game replay | Player clicks replay game | Game runs through in replay mode | Players can go forward or back in the replay |
| SE-F-029 | FR11 | Game saved while playing | Move made | Game is saved to a file to later be either restored or replayed | If program crashes the game can be restored. If game is saved it can be replayed |

Aberystwyth University / Computer Science

Page 1 of 00

Group 9 - Test Specification / 1.0 (Draft)

REFERENCES

Document history

| Version | Date | Changes made to document | Changed by |
| --- | --- | --- | --- |
| 1.0 | 10/02/2023 | Added tests for each functional requirement | jib16 |
| 1.1 | 20/02/2023 | Trimmed out the structure of the test specification document | ckc2 |

Aberystwyth University / Computer Science

Page 1 of 00