Software Engineering Group Project

System Test Specification

|  |  |
| --- | --- |
| Author: | Jasper Crabb [jac127] |
| Config Ref: | SE\_N66\_xxx\_xx |
| Date: | 15th February 2023 |
| Version: | 1.1 |
| 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. System Test Specification 3

REFERENCES 7

DOCUMENT HISTORY 8

# Introduction

## Purpose of this Document

This document will be used as reference to system testing and specifies how we will test the system as a whole.

## Scope

This document’s tests are derived from requirements specification for the software group project [1].

The document’s standard and format are from the test procedure standards for the software group project [2]

## Objectives

The objective of this document is to guide the system tester during testing, so that they know what the system should be able to do and how it should behave when doing so.

# System Test Specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Ref** | **Req being tested** | **Test Content** | **Input** | **Output** | **Pass Criteria** |
| SE-F-001 | FR1 | Check that the user is prompted with a menu on startup | Opening or running the program | A menu is displayed on the screen | The menu is displayed correctly |
| SE-F-002 | FR1 | Check that menu contains an option to start a new game | Opening or running the program | One of the menu options is to start a new game | The menu option is present and works correctly |
| SE-F-003 | FR1 | Check that the players are prompted to enter their names for each colour | The player selects the menu option “start a new game” | The player should be presented with a way to enter two names, one for black and one for white | The player is able to enter a name for both colours and each name is shown correctly with the correct colour |
| SE-F-004 | FR1 | Check that the player has the option to restore a previous game on startup | Opening or running the program | One of the menu options is to restore a previous game | The menu option is present and works correctly |
| SE-F-001 | FR2 | Check that the player names are tracked for the duration of the game | The player selects the menu option “start a new game” or makes a move in game | The player name is stored for the duration of the game | The name is stored in memory and can be corresponded to pieces |
| SE-F-002 | FR2 | Check that the player colour is tracked for the duration of the game | The player selects the menu option “start a new game” or makes a move in game | The colours of each player are stored for each of the players for the duration of the game | The colours are stored in memory corresponded to the correct players, and therefore the correct pieces |
| SE-F-003 | FR2 | Check that program is keeping track of where the players pieces are for the duration of the game | The player selects the menu option “start a new game” or makes a move in game | The pieces and their locations are stored for the duration of the game | The piece positions on the board are stored correctly in memory |
| SE-F-001 | FR3 | Check that program visually indicates which player should move | The player selects the menu option “start a new game” or makes a move in game | The program displays which player should move | The correct player is told to move |
| SE-F-002 | FR3 | Check that the players names are displayed on the screen | The player selects the menu option “start a new game” or makes a move in game | The program displays the player names | The names are displayed correctly |
| SE-F-003 | FR3 | Check that the board is displayed on the screen | The player selects the menu option “start a new game” or makes a move in game | The chess board should be displayed on the screen | The chess board is correctly displayed with no graphical errors |
| SE-F-004 | FR3 | Check that the pieces are displayed | The player selects the menu option “start a new game” or makes a move in game | The pieces should appear on the board | The pieces should be displayed in the correct locations, in the correct colour with no graphical errors |
| SE-F-001 | FR4 | Check if the player can only select their pieces | The player selects the menu option “start a new game” or makes a move in game | The player will be told if they try to move a piece that isn’t theirs | The correct player is prevented from moving the opponents pieces |
| SE-F-002 | FR4 | Check that the player can only move one piece at a time | The player selects the menu option “start a new game” or makes a move in game | The player will be told if the try to move more than one piece | Only one piece can be moved at a time |
| SE-F-001 | FR5 | Check that the possible legal moves are shown to the user | The player selects the menu option “start a new game” or makes a move in game | The program will show the legal moves for that turn to the user | The legal moves shown are valid and correct |
| SE-F-002 | FR5 | Check that the pieces can moved | The player attempts to move a piece | The piece selected will move | The correct piece moves to the correct location |
| SE-F-003 | FR5 | Check that the piece can only be moved if it is a valid move | The player attempts to move a piece | The player will be told if the piece cannot be moved | The program does not allow an illegal move |
| SE-F-004 | FR5 | Check that the player cannot move a piece to an invalid square | The player attempts to move a piece to an invalid square | The player will be told that it is an illegal move | The program does not allow a move to an illegal square |
| SE-F-001 | FR6 | Check that the program detects check and indicates it to the user | The opponent targets the king with one of their pieces | The player is told that they are in check | The program makes sure that the player knows that their king is in check |
| SE-F-001 | FR7 | Check that the program detects checkmate | The opponent puts the players king into checkmate | The player is told that they have lost | The program ends the chess game and stores the winner |
| SE-F-001 | FR8 | Check that program clearly indicates a game over | A condition that causes the game to end is triggered | The program tells the players that the game is over | The program displays a game over screen at the correct time |
| SE-F-002 | FR8 | Check that the program has an option to resign | The player selects the menu option “start a new game” or makes a move in game | There is a button somewhere that allows either player to resign | The player that resigns loses and the game is over |
| SE-F-003 | FR8 | Check that the players can make a draw | The player selects the menu option “start a new game” or makes a move in game | There is a button somewhere that allows either player to ask for a draw | Both players are presented with an option to draw and if accepted the game ends |
| SE-F-004 | FR8 | Check that the program saves the game as it ends | A condition that causes the game to end is triggered | n/a | The entire game and its details are saved to disk |
| SE-F-001 | FR9 | Check that the players can exit the game | The player selects the menu option “start a new game” or makes a move in game | There is a button somewhere that allows either player to quit the game | The game is adjourned and saved for later |
| SE-F-002 | FR9 | Check that the players can save the game | The player selects the menu option “start a new game” or makes a move in game | There is a button somewhere that allows either player to save the game | The entire game and its details are saved to disk |
| SE-F-001 | FR10 | Check that the game can be replayed one step at a time | The player selects the menu option “restore previous game” | The board is changed to its next / previous state | The board displayed is one move ahead or behind the current state and has the correct game state |
| SE-F-002 | FR10 | Check that the player has the option to review the game backwards | The player selects the menu option “restore previous game” | There is a button that allows the player to move backwards through the game | The board displays the previous move from the current game state |
| SE-F-003 | FR10 | Check that the player has the option to review the game forwards | The player selects the menu option “restore previous game” | There is a button that allows the player to move forwards through the game | The board displays the next move from the current game state |
| SE-F-004 | FR10 | Check the program allows the user to exit the replay mode | The player selects the menu option “restore previous game” | There is a button that allows the player to exit the replay mode | The button takes the user back to the start menu of the program |
| SE-F-001 | FR11 | Check the program saves each move made | The player makes a move in game | n/a | A file is updated with the new move that was made, as well as the new board position |
| SE-F-002 | FR11 | Check that the program can restore a game that has been quit | The player selects the menu option “restore previous game” | The is presented with an option to restore the game in question | When the game is selected it is in the correct (previous) state |
| SE-EI-001 | External Interface | Check that a user’s input is acted in on in under a second | Any input | The program responds within 1 second | The output is correct and took less than one second to display |
| SE-EIR-002 | External Interface | Check that the program can run on the IS PCs | n/a | n/a | The program runs and behaves correctly on the IS PCs |

REFERENCES

[1] Software Engineering Group Projects – Chess Tutor Requirements Specification 1.1 (Release)

[2] Software Engineering Group Projects – Test Procedure Standards/2.1 (Release)

DOCUMENT HISTORY

| *Version* | *Issue No.* | *Date* | *Changes made to document* | *Changed by* |
| --- | --- | --- | --- | --- |
| 1.0 | N/A | 15/02/23 | N/A - original version | Jac127 |
| 1.1 | N/A | 22/02/23 | Added required sections and changed test reference numbers | Jac127 |