Software Requirements Specification

Jungle game

Group 104, Fall 2025

1. Preface

1.1 Expected Readership: Project Stakeholders, development team, User

1.2 Version History: 1.0

1.3 Updates: N/A

1.4 Rationale: N/A

2. Introduction

The purpose of the JG program is to provide an implementation of the traditional Jungle board game involving 2 players, based on the rules and user stories provided by COMP3211 project description. It aims to support gameplay functions such as starting a new game, setting player names by manual input or random assigning, game saving, reload a saved game session, replay a recorded game, and commands for gameplay including moving a piece, undoing previous movements and stop the game.

3. Glossary

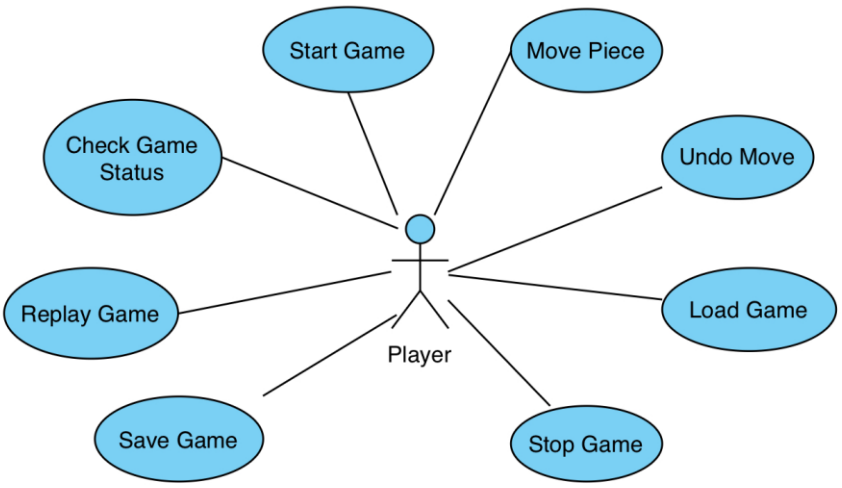
Term	Definition
Board	The 7x9 grid which represents a jungle with special squares including the dens, traps, and rivers. The squares are represented in the format alphabets A to G, numbers 1 to 9. For instance, the uppermost left square is A1.

CLI	Command-line interface
FR	Functional requirement
GUI	Graphical User Interface
JG	The Jungle Game program.
NFR	Non-functional requirement
UFR	User functional requirement

4. User Requirements Definition

4.1 Use Cases

The following use cases are defined for player in the JG system:



4.2 User Requirements

ID	Requirement
UFR1	The system shall initialize a new game with a fresh board and player setup.
UFR2	The system shall allow users to terminate an ongoing game at any time.
UFR3	The system shall support manual input or random generation of player names.
UFR4	The game shall be playable through the command line console.
UFR5	The system shall display game status including number of remaining pieces, piece positions, and the next active player.
UFR6	The system shall allow each user undo at most three previous moves per game.
UFR7	The system shall record player names and move history into a file named

	“record”.
UFR8	The system shall replay a previously recorded game session from the “record” file.
UFR9	The system shall save the current game status to a file named “jungle”.
UFR10	The system shall load a saved game session from the “jungle” file and resume gameplay.

4.3 User non-functional requirements

Usability: The CLI should be easy to use by players so that the instructions are visible to players.

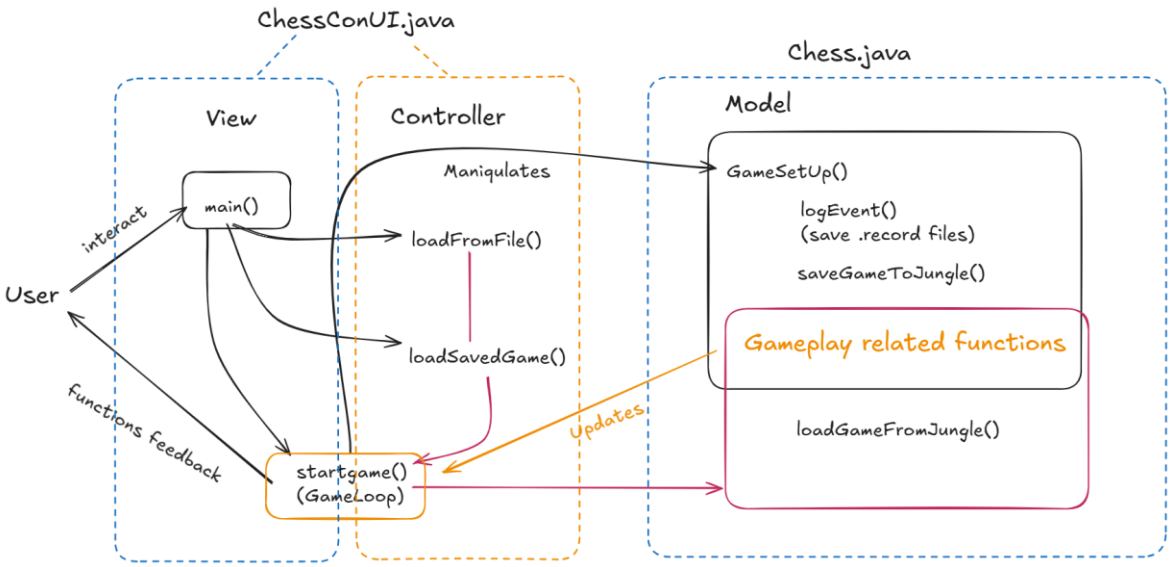
Performance: Game operations such as move, undo, save shall complete within 1 second.

Reliability: The system shall ensure game loading shall not lead to any faults or exceptions during file operations.

5. System Architecture

The MVC (Model-View-Controller) pattern is used in the JG system.

Yellow highlighted ones are reused.



6. System Requirements Specification

6.1 Functional Requirements

6.1.1 Game Initialization and Termination

ID	Requirement
SFR1	The system shall initialize the game board with correct dimensions (7×9 grid) and place all animal pieces in their starting positions within 2 seconds.
SFR2	The system shall allow users to terminate an ongoing game at any time.

6.1.2 Player Management

ID	Requirement
SFR3	The system shall alternate turns between players and track the active player.
SFR4	The player shall move at their turn.

6.1.3 Game Logic and Rules Enforcement

ID	Requirement
SFR5	The system shall validate and execute piece movements according to following Jungle rules: ● Horizontal or vertical movement only ● No entry into own den ● Only the rat is allowed to go into river ● Only lion and tiger may jump over the river ● If there is a rat (whether friendly or enemy) on any of the intervening water squares, no jumping movement is allowed
SFR6	The system shall enforce capturing rules based on the piece rank and special conditions (e.g. rat in water, rat vs elephant, trap capture).
SFR7	The system shall detect win conditions by a piece enters the opponent's den, or all opponent pieces are captured.

6.1.4 Game State Management

ID	Requirement

SFR8	The system shall maintain and update the game state after each move, including piece movements, remaining pieces, current player, remaining undo limit.
SFR9	The system shall allow each user undo at most three previous moves per game.

6.1.5 Interface and Interaction

ID	Requirement
SFR10	The system shall support the user to choose to start a new game, to load a saved game, to replay a recorded game, or exit the system.
SFR11	The system shall support user commands via the command-line console and validate input formats.
SFR12	The system shall display game status including board layout, piece positions, and active player.

6.2 Non-functional Requirements

6.2.1 Performance

ID	Requirement
NFR1	The system shall respond to a user command within 200 ms on a typical developer machine.
NFR2	Auto replay shall maintain timing fidelity 1500 ms.

6.2.2 Reliability

ID	Requirement
NFR3	The save function shall prevent data loss.
NFR4	initializeBoardHistory() size should match usage in movePiece()/undo().

6.2.3 Correctness

ID	Requirement

NFR5	Applying commands read by loadFromFile() should reproduce game state exactly when review mode is used.

6.2.4 Usability

ID	Requirement
NFR8	Prompts and error messages for user commands should be unambiguous (move format, columns A–G).
NFR9	The game command shall accept both uppercase/lowercase and shall trim whitespace.

6.2.5 Maintainability

ID	Requirement
NFR10	Chess review mode shall be implemented to prevent creating log files under maintenance.

6.2.6 Persistence and Auditability

ID	Requirement
NFR11	On write failure, logEvent(…) should retry fallback to a safe log, and should record timestamped backup of the event.