Software Requirements Specification

for

Ferret Army Chess

Version 1.0 approved

Prepared by A. Maxwell, J. Guerrero, A. Romualdo, J. Cole, M. Yi

Ferret Army

February 2018

Table of Contents

Table of Contents ii

Revision History i

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 Artificial Intelligence Engine 4

4.2 Human vs. Human 4

4.3 Computer vs. Human 4

4.4 Computer vs. Computer 4

4.5 En passant 4

4.6 Pawn promotion………………………………………………………………………….. 4

4.7 Castling 4

4.8 Stalemate Resolution 4

4.9 Game Timer 4

4.10 Turn Timer 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

# 

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| A. Maxwell | 02/05 | Update sections 1.1, 1.2, 1.3, 1.4, and 1.5 | 1.0 |
| J. Cole | 02/06 | Update TABLE 2. section 2.2 | 1.1 |
| A. Maxwell | 02/07 | Update sections 2.1, 2.2, 2.3, 2.4, 2.5 | 1.2 |
| A. Maxwell | 02/08 | Update sections 2.6, 2.7 | 1.3 |
| A. Maxwell | 02/08 | Update sections section 3.1. left blank section 3.2, 3.3, and 3.4 | 1.4 |
| A. Romualdo | 02/09 | Update TABLE 2. section 2.2 | 1.5 |
| A. Maxwell | 02/09 | Update Table of Contents | 1.6 |
| A. Maxwell | 02/09 | Changes to formatting of TABLE 2. Section 3.1 figures modified | 1.7 |
| A. Maxwell | 02/09 | General formatting changes. | 1.8 |
| A. Romualdo | 02/09 | Proofread and highlighting deletions | 1.9 |
| A. Romualdo | 02/10 | Section 4.4 to 4.10 generated | 2.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to provide a detailed description of the requirements for the "Ferret Army Chess (FAC)" software. It will illustrate the purpose and declaration for the development of the FAC software. It also details the system constraints, interface, and expected interaction between the user(s) and the system.

## Document Conventions

## This document is written in accordance with the IEEE Software Engineering Standards Committee document titled "IEEE Recommended Practice for Software Requirements Specifications". This calls for the use of 12pt Times New Roman font, spaced 2.0 points separation between lines, with top margin: 0.6", bottom margin: 0.5", mirror images, inner margin: 0.75", outer margin: 0.75", Gutter: 0.5 ", header and footer 0.3" from edge. Due to readability this document has been formatted with 1” top margins, 0.75” bottom, inner, and outer margins with 1.5 points separation between lines.

## Definitions, Acronyms, and Abbreviations

The intended readers of this document are the client, project managers, and marketing personnel associated with the development and deployment of the Ferret Army Chess (FAC) software. Due to this, section 1.3 has been included to ease in the reading of this document. Important terms, definitions, acronyms, and abbreviations have been included in detail.

TABLE 1.

Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Computer | The system that represents the artificial intelligence of which a user can compete against. |
| User | A person interacting with the FAC software. |
| Player | A user who has initiated a game against either another user or the computer. |
| Bystander | A user who is observing a computer vs. computer game without making a game move. |
| Username | A unique name for the player |
| FAC | Refers to Ferret Army Chess the software under development. |
| AIE | Refers to artificial intelligence engine that makes moves for the computer. |
| UI | Refers to user Interface the system by which the player interacts with the FAC software. |
| GP | Refers to any of the game piece(s) which may be a Pawn, Rook, Bishop, Knight, Queen, or King |

## Product Scope

The "Ferret Army Chess (FAC)" software is Ferret Army's take on the classic turn-based game of chess founded in northern India around 6th century AD. It will feature traditional chess moves between the user(s) and/or the games artificial intelligence engine.

The FAC software will initially only support local gameplay on desktop/mobile computers with the eventual goal of transitioning it into a web-based application to support online gameplay and platform independence.

Prior to gameplay, the software shall feature an option to the player to initiate either local user vs. user, user vs. computer, or computer vs. computer gameplay.

Furthermore, the FAC software shall assign the “Reinfeld values” for pieces captured during gameplay to resolve possible stalemates scenarios.

## References

# [1] "830-1998 - IEEE Recommended Practice for Software Requirements Specifications - IEEE Standard", Ieeexplore.ieee.org, October 20, 1998. [Online]. Available: http://ieeexplore.ieee.org/document/720574/?reload=true&arnumber=720574. [Accessed: 04- Feb- 2018].

“Making Wise Captures.” Chess Corner - Chess Tutorial - Making Wise Captures, www.chesscorner.com/tutorial/basic/capture/capture.htm. [Accessed: 09-Feb-2018]

The Chess Clock, www.chesscorner.com/tutorial/chess\_clock/chess\_clock.htm.[Accessed: 10-Feb-2018]

# Overall Description

## Product Perspective

The FAC software will be a new self-contained application comprised of three primary components the *user interface* (UI), *game engine* (GE), and *artificial intelligence engine* (AIE). Of these, the UI is comprised of two sub-components the *game board* (GB), and thrity-two *game pieces* (GP) which player(s) uses during the course of a game. The general relationship between each major component and sub-components can be seen in figure 1.

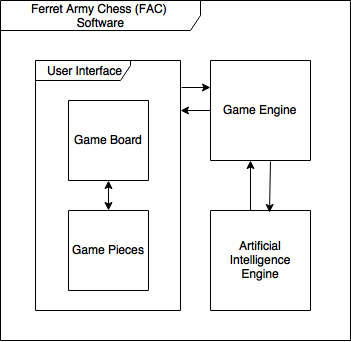


Figure 1. Component Overview

In the process of a user using the FAC software, there is a constant interfacing between the UI, GE, and AIE. When a user interacts with the GB this interaction is handled by the GE, which depending on the *game mode* (GM) may make calls to the AIE for a randomized move representing the *game move* (GM) of the *computer*. In the event of a user. vs. user game the GE will initiate no calls to the AIE.

Since the FAC software is initially going to be designed for desktop/mobile platforms there will be no strong emphasis on restraining resources; however, when FAC is eventually adapted into a web-based application a series of optimizations will be applied to constrain resources according to hardware requirements of the web-server.

## Product Functions

With the FAC software the players, either a user or the computer, shall be able to have the ability to perform several key interactions. Those interactions are cataloged below, in bulleted form, in figure 2.

* User(s) shall be able to select from three game modes including computer vs. computer, user vs. computer, user vs. user prior to gameplay.
* FAC software shall include functionality for limited artificial intelligence engine (AIE). Using randomization, the FAC’s AIE will move game pieces (GP) around the game board (GB).
* In addition to the standard chess moves the FAC software shall allow for three special moves including:
  + En passant – Special pawn Capture move.
  + Pawn Promotion – From a Pawn to any other Game Piece (GP)
  + Castling – Both Weak Side and Strong Side.

These features shall only be available when or while specific conditions are met.

The specifics of each special move and what is entailed with these moves will be covered in-depth in section 4.

* User(s) shall be able to move any game piece (GP) according to that GP’s specific game move (GM) attribute. These moves are detailed in-depth in TABLE 2. shown on page 5.
* In addition, the GB shall include game coordinates, so a user can submit moves using a coordinate on the GB. This will be represented as numbers for the rank (horizontal coordinates) and letters for the file (vertical coordinates).
* System shall implement a user interface (UI) allowing the user(s) to select game mode and settings.

TABLE 2.

*Game Pieces* and their associated *Game Moves*

|  |  |  |
| --- | --- | --- |
| **Game Piece** | **Game Move** | **Capture** |
| Pawn | **Forward 1 space**  **Forward 2 spaces** (Starting move only) – Movement cannot cause collision with another piece.  If the option for pawn promotion is chosen, then player can choose the piece the pawn will be promoted to after it reaches the last row of the opposing players side.  **Restrictions:**  Movement cannot extend past the edge of the game | Left Diagonal 1 space  Right Diagonal 1 space  **Special case:**  en passant capture – left or right diagonal 1 space  (see specifics in section 4.5) |
| Rook | **Forward 1-7 spaces**  **Backward 1-7 spaces**  **Left 1-7 spaces**  **Right 1-7 spaces**  **Restrictions:**  Movement is unrestricted until another game piece is encountered, or edge of game board is reached.  **Special case:**  Simultaneous movement with King is allowed for castling (see specifics detailed in section 4.7). | Same as game move until an opponent’s piece is captured. |
| Knight | **Vertical 1 space** and **3 spaces** either **horizontally left** or **horizontally right**  **Horizontally 1 space** and **3 spaces vertically** either **up** or **down**.  **Restrictions:**  Movement is unrestricted. Another game piece will not prevent movement.  Movement cannot exceed the perimeter of the board. | Can only capture at the termination of the move. |
| Bishop | **Diagonal 1-7 spaces** on the game pieces color of origin  **Restrictions:**  Movement is unrestricted until another game piece is encountered, or end of game board is reached. | Same as game move until an opponent’s piece is captured. |
| Queen | **Diagonally 1-7 spaces**  **Vertically 1-7 spaces**  **Horizontally 1-7 spaces**  **Restrictions:**  Movement is unrestricted until another game piece is encountered, or end of game board is reached. | Same as game move until an opponent’s piece is captured. |
| King | **Diagonally 1 space**  **Vertically 1 space**  **Horizontally 1 space**  **Restrictions:**  Cannot move into a position that will place it within 1 space of the opponents King.  Cannot move into a position that will place it in check.  Movement cannot exceed the perimeter of the board.  **Special case:**  Castling will allow movement greater than 1 space along the 1st rank (see specifics in section 4.7). | Same as Game Move until an opponent’s piece is captured. Must not be in check when Capture completed. |

* User(s) shall have the ability to enable a timer to control the flow of the game. The time shall have functionality to support the following operations:
  + Game Time Limit – Set duration for the length of a single game.
    - Winner will be determined by the total points accumulated from the capture of the opposing teams game pieces.
  + Turn Time Limit – Set a duration for the length of a single turn.
* Users shall have the option to quit an ongoing game at any time. It is not necessary for both Player’s to agree before an individual Player quits a game.

## User Classes and Characteristics

There are three users that will interact with the system: *player(s)* of FAC, *bystander(s)* of FAC, and AIE represented as the *computer.* Each of these users have different interactions with the system and as such have their own unique requirements.

A *user* may interact with FAC as both *bystander* and a *player*. A *player* may start a game with another *player* or the *computer*. The *player* will have options select their opponent (*computer* or *user*), enable *game* timer and *turn timer*, and select a *username* prior to starting a new game.

A *bystander* is a *player* that has chosen to setup and watch a *computer* vs. *computer* game*.* The *bystander* will have no ability to interfere with the game and cannot make moves for the *computer*.

The *computer* is the manifestation of the AIE. It has no ability to change game settings, end the current game, or change a *player’s* moves in any way. The *computer* will interact with the FAC software only when requested by the *game engine* (GE).

## Operating Environment

Since the FAC software’s target platform is desktop and mobile computers the software must be able to run on any current operating system regardless of manufacturer. This means the FAC software must support both Microsoft and Apple Inc. operating systems at the very least.

Being operating system independent will promote a wider player base to draw from once the software is adapted into a web-based application. Once converted into a web-based application FAC will include support for mobile platforms in addition to the current support for desktop and mobile computers.

Due the scope of this document detailing the current build of FAC and not the future web-based application no description or detail of the web implementation will be included. The conversion documentation will be included in another software requirement specification once development reaches the refactoring stage.

## Design and Implementation Constraints

Due to the *operating environment* requirements laid out in *section 2.5* the FAC software will need to be developed in a platform independent programming language to accommodate being operating system independent. The FAC software shall make use of any programming language libraries, API’s, SDK’s, or frameworks as long as the software scales to multiple platforms with no degradation in appearance or performance.

The efficiency of the *artificial intelligence engine* (AIE) is a potential constraint hindering performance of the *game engine* (GE). The AIE is responsible for providing all *computer* moves as requested by the GE. Therefore, the need for clean efficient algorithms behind the AIE is important to overall system performance.

## User Documentation

Limited user support documentation will be provided with the FAC software upon delivery. However, to support refactoring, feature addition, and maintenance of FAC software by a future development team the software design document (SDD), test plan (TP), test cases and test result will be included as deliverables.

## Assumptions and Dependencies

One of the main assumptions made in this *software requirement specification* (SRS) document is that the chosen programming language will have and extensive library of tools to support rapid development through code reuse. A problem could arise if the selected language does not have extensive libraries for UI, data structures, and eventually web based interactions.

Additionally, for the *user interface* (UI) this document assumes that an off the shelf framework can be used to implement the *game board* (GB), *game pieces* (GP), and all necessary UI menus. This means the development team can have limited UI or *user experience* (UX) when implementing FAC. Should no UI framework exist to work with the chosen programming language this could potentially delay production until UI professionals are brought on board or another programming language is selected that supports such a framework.

# External Interface Requirements

## User Interfaces

A first-time user of the FAC shall see a *Landing Page* with two menu options, *Play Ferret Chess* and *Settings* when he/she launches the FAC software, see figure 2. When a user selects *Play Ferret Chess* the system shall take the user to the *Game Setup Page*, see figure 3.

|  |  |
| --- | --- |
|  |  |
| Figure 2. Landing Page | Figure 3. Game Setup Page |

The *Game Setup Page* shall allow the user to set a *username*, select *game mode*, and configure *game timer* and *turn timer* as well as begin a game. Once a user selects *Play* the system shall take the user to the *Game Page*, see figure 4.

When a user selects one of the available GP (Pawn, Rook, Bishop, Knight, Queen, King) the system shall allow the user to place selected GP according to that piece’s *game move* (GM). If a user selects *quit game* the system shall immediately end the game and take the user back to the *Landing Page*. If user selects *settings* the user will be taken *Settings Page*, see figure 5.

|  |  |
| --- | --- |
|  | PLACE HOLDER IMAGE |
| Figure 4. Game Page | Figure 5. Settings Page |

When a user reaches the *Settings Page* either via the *Landing Page* or by clicking *settings* on the *Game Page* the user will have options available to select default settings which include: *game timer* and *turn timer.*

On the *Game Page* the user shall have the ability to move *game pieces* (GP) on the *game board* (GB). Additionally, they shall be able to keep track of the *game timer*, *turn timer*,and *remaining pieces*. Furthermore, the user shall have the option to change *settings* and *quit game*.

## Hardware Interfaces

Since the FAC software is a standalone application designed to be installed on desktop and mobile computers it does not have any direct hardware interfaces. The game engine, artificial intelligence engine, and user interface are managed by the FAC application. Its use of the central processing unit (CPU), random access memory (RAM), and data storage is managed by the underlying operating system of the desktop or mobile computer.

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

LEFT BLANK INTENTIONALLY (Requires team discussion)

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

The communication between the different components of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for both the FAC software and the desktop and mobile computers it runs on.

Considering the FAC software is an application designed to be run locally on desktop and mobile computers it will have limited communication interfaces. Initially there will be no support for network gameplay. As previously mentioned

*LEFT BLANK INTENTIONALLY (No Information to include)*

# System Features

## Artificial Intelligence Engine (AIE)

4.1.1 Description and Priority

This high priority system feature is responsible for responding to requests from the *game engine* (GE). It will provide all computer *game moves* (GM) during a *computer* vs. *user* and *computer* vs. *computer* game modes. The *benefit* of implementing this feature is 9, the *penalty* for not implementing is 9, the *cost* of implementation is 4, while the risk of implementation is 5.

4.1.2 Stimulus/Response Sequences

The user has no direct interactions with the AIE. All request to the AIE come from the GE. When a user moves a *game piece* (GP) the GE responds to this request and queries the AIE for a move. Once calculated the AIE passes back the move to the GE to reflect on the *game board* (GB).

4.1.3 Functional Requirements

REQ-1: The AIE must be implemented for a user to play the *user* vs. *computer* or the *computer* vs. *computer* game modes this means also that the GE needs to be configured as well due to its interfacing between the *user interface* and the AIE. Due to the AIE being a standalone engine it handles no input or output validation. This is handled by the GE.

REQ-2: When the GE requests a move from the AIE the AIE will calculate a move and return it back to the GE. The GE will validate that the move can be made on the *game board* (GB). If it cannot then the GE will reject the move and request another from the AIE. This process will continue until the AIE passes a valid move to the GE. Upon receiving a valid move the GE will reflect the move on the GB.

REQ-3: The AIE does not have to be an agile and responsive artificial intelligence. For the purposes of this project the AIE may respond in completely random and unpredictable ways. This includes leaving GP’s in peril, capturing non-essential GP’s, playing to its own detriment, moving in non-strategic ways.

## Human vs. Human (Game Mode)

4.2.1 Description and Priority

This high priority system feature allows a *user* to challenge another *user* locally to a game of chess using the FAC software. Each *player* will take turns, making GM’s using their available GP’s. This game mode ends when a *stalemate* or *checkmate* is reached. Please refer to TABLE 2. on page 2 for definitions of the terms *stalemate* and *checkmate*. The *benefit* of implementing this feature is 9, the *penalty* for not implementing is 9, the *cost* of implementation is 8, while the risk of implementation is 8.

4.2.2 Stimulus/Response Sequences

The *human* vs. *human* game mode will consist of multiple interactions between each *player* and the GE. Once a *player* confirms their move the GE will record that move and update the GB to reflect the most recent move. This includes removing an opponent’s pieces if the last move resulted in a *capture*. Play will continue like this with the GE handling moves from each *player* in succession until one *player* causes *checkmate* on the other *player* or game *stalemate* is reached. For more information on *stalemate resolution* refer to *section 4.8*.

4.2.3 Functional Requirements

REQ-1: In order for *user* vs. *user* to function correctly the UI must be fully established.

REQ-2: The GE shall be able to respond fully to *users* request to move GP’s on the GB.

REQ-3: The GE shall validate all incoming GM’s and ensure that the current GB’s status can support the *users* submitted GM.

REQ-4: A winner shall be determined by the GE, which will check after each GM whether a player is in checkmate or the game as a whole has reached *stalemate*. If *stalemate* is reached it will be handled in accordance with the requirements set forth in *section* 4.8.

## Computer vs. Human (Game Mode)

4.3.1 Description and Priority

This high priority system feature allows a *user* to challenge the *computer* to a game of chess using the FAC software. Each *player* will take turns, making GM’s using their available GP’s. This game mode ends when a *stalemate* or *checkmate* is reached. Please refer to Glossary for definitions of the terms *stalemate* and *checkmate*. The *benefit* of implementing this feature is 9, the *penalty* for not implementing is 9, the *cost* of implementation is 8, while the risk of implementation is 8.

4.3.2 Stimulus/Response Sequences

The *computer* vs. *human* game mode shall consist of multiple interactions between the *player* and the GE, and the *computer* and the GE. Once a *player* confirms their move the GE shall record that move and update the GB to reflect the most recent move. This includes removing an opponent’s pieces if the last move resulted in a *capture*. After a human *player’s* move is recorded on the GB the GE shall make a request to the AIE to fetch a counter move. The process for handling this counter move is detailed in *section 4.1*. Play shall continue like this with the GE handling moves from the human *player* and *computer* in succession until either the *computer* or *player* causes *checkmate* on the other *player* or game *stalemate* is reached. For more information on *stalemate resolution* refer to *section 4.8*.

4.3.3 Functional Requirements

REQ-1: In order for *computer* vs. *user* to function correctly the UI must be fully established.

REQ-2: The GE shall respond fully to *users* request to move GP’s on the GB.

REQ-3: The GE shall have the ability to fetch new moves from the AIE, validate fetched moves, and reflect legal moves on the GB.

REQ-4: The GE shall validate all incoming GM’s and ensure that the current GB’s status can support the *users* submitted GM.

REQ-5: A winner must be determined by the GE, which shall check after each GM whether a player is in checkmate or the game as a whole has reached *stalemate*. If *stalemate* is reached it shall be handled in accordance with the requirements set forth in *section* 4.8

**4.4 Computer vs. Computer (Game Mode)**

4.4.1 Description and Priority

This high priority system feature allows a *user* to force the computer to play itself in a game of chess using the FAC software. The AIE shall play both sides independently and shall not be exposes to the strategies utilized by the opposing side. All GP’s shall be limited to the movement and capture restrictions outlined in Table 2. The opposing AIE shall take turns moving their pieces. This game mode ends when a *stalemate* or *checkmate* is reached. Please refer to Glossary for definitions of the terms *stalemate* and *checkmate*. The *benefit* of implementing this feature is 9, the *penalty* for not implementing is 9, the *cost* of implementation is 8, while the risk of implementation is 8.

4.4.2 Stimulus/Response Sequences

The *computer* vs. *computer* game mode shall consist of multiple interactions between the *computer* and the GE. The GE representing the white side, shall make a request to the AIE to fetch a move. The process for handling this counter move is detailed in *section 4.1*. The moves shall be limited to the specifications noted in Table 2. After this first initial move, the GE shall make another request to the AIE to fetch a counter move for the opposing black team. Again, the process for this shall be detailed in *section 4.1*. The opposing sides shall continue to alternate turns until one side forces a *checkmate* or a *stalemate* is reached. For more information on *stalemate resolution* refer to *section 4.8.*

4.4.3 Functional Requirements

REQ-1: *Computer* vs. *computer* function requires the UI to be established to the point that the process can be chosen and started.

REQ-2: The GE shall have the ability to fetch new moves from the AIE, validate fetched moves, and reflect legal moves on the GB.

REQ-3: If a more complex AIE is implemented, it shall require strategies to be hidden from the opposing side.

REQ-4: A winner must be determined by the GE, which shall check after each GM whether a side is in checkmate or the game as a whole has reached *stalemate*. If *stalemate* is reached it will be handled in accordance with the requirements set forth in *section* 4.8

**4.5 En passant (Special Move)**

4.5.1 Description and Priority

This is a low priority move which allows a pawn in the 5th rank to capture a passing opponents pawn that is using the two-space initial move. The capturing pawn shall only capture at an angle and shall occupy the space directly behind the captured pawn. This is an option available for the player to use and shall require a UI button to activate. The GE shall determine if the move has met the criteria and shall activate the button for player use. There shall be no indication to the player that the button is active. This option shall not be available for use by the *computer* in *computer* vs. *human* or *computer* vs. *computer* game mode. The *benefit* of implementing this feature is 6, the *penalty* for not implementing is 4, the *cost* of implementation is 6, while the risk of implementation is 8.

4.5.2 Stimulus/Response Sequences

The en passant option shall be a readily available UI during game play. It shall only be active to the player when the opposing pawn has performed the two-step initial move, the opposing pawn is in a file adjacent to the capturing pawn, and the capturing pawn is in it’s fifth rank. The UI shall not display any indication it is active and shall only stay active for the turn after the opposing pawn has moved. The GE shall determine the appropriate move required as specified in TABLE 2. The captured pawn shall be removed from the board and one point shall be awarded to the capturing side.

4.5.3 Functional Requirements

REQ-1: Opposing pawn must have made the two-space initial move on the previous turn.

REQ-2: The capturing pawn must be on the fifth *rank.*

REQ-3: The opposing pawn must be on an adjacent *file* to the capturing pawn.

REQ-4: The GE shall move the capturing pawn into the *rank* and *file* directly behind the opposing pawns location on the GB.

REQ-5: The GE shall remove the opposing pawn from the GB.

REQ- 6: The GE shall award the capturing side with one point.

**4.6 Pawn promotion (Special Move)**

4.6.1 Description and Priority

This is a low priority move which allows a pawn that reaches the opposing 8th *rank* to be promoted to either a *major* (consisting of the queen and rook) or a *minor* piece (consisting of the bishop or knight). This is an option available for the player to use and shall require a UI button to activate. The GE shall determine if the criteria for pawn promotion has been met and shall activate the pawn promotion button for the player to use. This can be a button that is highlighted to advertise that a special move is available. This option shall not be available for by the computer in *computer vs. human* or *computer vs. computer* game mode. Upon activation, a UI displaying the optional pieces that the pawn can be traded for shall be presented. Once an appropriate replacement piece has been chosen, the pawn shall be replaced on the GB with the chosen piece and shall acquire all movement restrictions attributed to the new piece. The *benefit* of implementing this feature is 9, the *penalty* for not implementing is 9, the *cost* of implementation is 9, while the risk of implementation is 9.

4.6.2 Stimulus/Response Sequences

Pawn promotion shall only be available when the GE has determined that special criteria has been met to activate this option. Upon activation, the Pawn Promotion UI will display. Once chosen, this option will prompt a window to that displays all the available minor and major pieces to chose from. A UI shall be displayed next to each piece to allow the player to chose which GP he/she wishes his/her pawn to be promoted to. The pawn will be replaced with the piece chosen. The replacement piece will adopt all the movement restrictions limited to that type of piece (see TABLE 2). Movement for this piece will commence on that player’s next turn. When the replacement piece is determined, the Pawn Promotion option will be turned off.

4.6.3 Functional Requirements

REQ-1: Pawns must reach the opposing players 8th rank before UI for Pawn Promotion is activated.

REQ-2: Pawn promotion option shall be highlighted.

REQ-3: A list of pieces the pawn can be traded for shall be displayed.

REQ-4: Pawn shall be removed from the GB.

REQ-5: The GE shall replace the pawn with the chosen piece.

REQ- 6: The new piece shall have all the move restrictions displayed in TABLE 2 for that type of piece.

REQ-7: Movement for new piece shall commence on the next available turn.

REQ-8: The Pawn Promotion UI shall be removed from the options.

**4.7 Castling (Special Move)**

4.7.1 Description and Priority

This is a high priority move which allows a king to reposition its location quickly on the GB. This special move will require the movement restrictions noted in TABLE 2 to be ignored. This is an option available for the player to use and shall require a UI button to activate. The GE shall determine if the criteria for castling has been met and shall activate the castling button for the player to use. This can be a button that is highlighted to advertise that this special move is available. It shall remain highlighted and available for players use until certain criterion is violated. If the option for castling is available on both the kingside and queenside, another UI will display to allow the player to choose which direction to go. This option shall not be available for by the computer in *computer vs. human* or *computer vs. computer* game mode. When this option is chosen, the GE will relocate the king and rook to the appropriate locations on the GB. The *benefit* of implementing this feature is 9, the *penalty* for not implementing is 9, the *cost* of implementation is 9, while the risk of implementation is 9.

4.7.2 Stimulus/Response Sequences

The Castling option shall only be available when the GE has determined that special criteria has been met to activate this option and other criteria has not yet been violated. Upon activation, the Castling UI will display. If an option to castle on both the kingside and queenside is available, then a secondary UI shall be provided to allow the user to choose a side to castle on. Once chosen, the king and rook will be relocated to their appropriate positions.

Kingside castle: rook is moved to the position originally occupied by the knight while the king is moved to the position originally occupied by the bishop.

Queenside castle: rook is moved to the position originally occupied by the queen while the king is moved to the position originally occupied by the bishop.

Once this option has been completed the castling UI will be permanently disabled for that side.

4.7.3 Functional Requirements

REQ-1: The king has not previously moved from its start position.

REQ-2: The rook has not previously moved from its start position.

REQ-3: There are no pieces between the king and the rook.

REQ-4: The king is not currently in check.

REQ-5: The king does not pass through a space that would put it in check.

REQ- 6: The king is not in check at the end of the move.

REQ-7: If REQ-1 or REQ-2 have been violated then the castling UI cannot be activated.

REQ-8: If this special move is used, the castling UI will be permanently disabled for the duration of the game for that player.

**4.8 Stalemate Resolution (Special Condition)**

4.8.1 Description and Priority

This is a medium priority option which will determine if a stalemate has been reached. Upon determination of a stalemate, the point system of captured pieces will determine the winning side. Throughout the game, the AIE shall continually analyze the position of the GP and will halt play and declare stalemate if certain criterion is met. Once stalemate has been determined, the game will cease, and the point value of captured pieces will be displayed. The *benefit* of implementing this feature is 7, the *penalty* for not implementing is 7, the *cost* of implementation is 3, while the risk of implementation is 3.

4.8.2 Stimulus/Response Sequences

Stalemate resolution monitoring shall only be active when one player is down to only one piece, the king. The AIE shall constantly monitor the options for that piece to determine if a stalemate must be called. If the conditions for stalemate has been deduced, the game will end and the points accumulated for captured pieces will be displayed for both sides. The conditions for stalemate are:

1) One player is down to only the king remaining and cannot move into any position without placing itself in check.

2) Both players are down to only one piece each, the king. Since kings cannot come closer than 2 spaces and their capturing range is 1 space then a resolution can never be reached.

3) The game timer runs down.

4.8.3 Functional Requirements

REQ-1: One player has only a king left on the GB and cannot move without placing itself in check.

REQ-2: Both players are down to only one piece each.

REQ-3: Game Timer (see section 4.9 for details) has timed out.

REQ-4: Points for captured pieces will be displayed.

**4.9 Game Timer (Pre-Game Option)**

4.9.1 Description and Priority

This is a low priority option which sets a specific amount of time for gameplay. The option will allow a player or players to set a maximum amount of game time to play the game. This could be useful if the player has a limited amount of time to interact with this program. This option shall require the Turn Timer, described in section 4.10, to also be active. The activation of the Turn Timer will prevent one player from dominating the game with excessively long turns. Prior to the start of the game, the option for setting a game timer shall be offered as a UI. If chosen, a UI for the Turn Timer shall also be displayed. Game Timer shall have a default setting of 60 minutes for all the moves but shall allow any time adjustment in minutes that is mutually chosen by both players. Upon time out of the game timer the game will cease and the accumulated points from the captured pieces on both sides will be displayed. The *benefit* of implementing this feature is 1, the *penalty* for not implementing is 1, the *cost* of implementation is 5, while the risk of implementation is 5.

4.9.2 Stimulus/Response Sequences

Game Timer is an option chosen before the game begins. It is a stand-alone system that is separate from the GE or the AIE. The UI for the Game Timer will be active prior to game play only. If chosen it will default to 60 minutes but will allow a custom time to be entered in minutes. The Game Timer will prompt the activation of the Turn Timer, detailed in section 4.10. The Game Timer is a countdown timer that begins when the game begins. Upon countdown termination, the game will end and the point score for the captured pieces from both sides will be displayed.

4.9.3 Functional Requirements

REQ-1: *Game Timer* UI shall only be displayed on the Game Setup screen prior to the start of the game (see section 3.1, Figure 2).

REQ-2: *Game Timer* shall have a default setting of 60 minutes.

REQ-3: *Game Timer* can be adjusted by 1 minute intervals.

REQ-4: Count down shall be displayed in minutes:seconds.

REQ-5: Activation of the *Game Timer* shall prompt activation of the *Turn Timer* (see section 4.10 for details on *Turn Timer*).

REQ-6: *Game Timer* shall commence countdown when the game is started.

REQ-7: Upon completion of *Game Timer,* the game shall cease and the points for all captured pieces shall be displayed.

**4.10 Turn Timer (Pre-Game Option)**

4.10.1 Description and Priority

This is a medium priority option to prevent games from continuing for too long and forces players to commit to an action. The option will allow the players to limit the amount of time they spend “thinking.” It is an optional feature presented in the *Game Setup* *screen* (see section 3.1, Figure 2). If the *Game Timer* option (see section 4.9) is chosen, then *Turn Timer* shallalso be used. The default time for the *Turn Timer* will be 1 minute. This time can be adjusted when both parties agree to a turn-time. The *Turn Timer* shall allow times to be adjusted by either minutes, seconds or both. The *Turn Timer* will activate at the start of the game. It will be displayed somewhere on the *game page* (see section 2, TABLE 2). It shall notice communication and movement from the GE on the GB to identify when to restart the countdown for the other player. Allowing a *Turn Timer* to countdown to zero will result in forfeiture of the game. The *benefit* of implementing this feature is 5, the *penalty* for not implementing is 5, the *cost* of implementation is 5, while the risk of implementation is 5.

4.10.2 Stimulus/Response Sequences

Turn Timer is an option chosen before the game begins. It is a stand-alone system that is separate from the GE or the AIE. The UI for the Game Timer will be active prior to game play only. If chosen it will default to 5 minutes but will allow a custom time to be entered in minutes and seconds. The Turn Timer will be displayed as a window on the Game Page, as shown in section 3.1, Figure 3. The Turn Timer is a countdown timer that resets at the start of each turn. If the Turn Timer is allowed to countdown, that player is considered to have forfeit the game and game will cease.

4.10.3 Functional Requirements

REQ-1: The *Turn Timer* UI shall only be displayed on the Game Setup screen prior to the start of the game (see section 3.1, Figure 2).

REQ-2: *Turn Timer* shall have a default setting of 5 minutes.

REQ-3: *Turn Timer* can be adjusted by minute or second intervals.

REQ-4: Count down shall be displayed in minutes:seconds.

REQ-6: *Turn Timer* shall reset and commence countdown at the beginning of each turn for each player.

REQ-7: If the *Turn Timer* is allowed to reach 0, that player will be considered to have forfeit the game and the game will end.

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

This glossary is a duplication of TABLE 1. from section 1.3, which outlines the definitions, acronyms, and abbreviations used throughout this software requirement specification document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Computer | The system that represents the artificial intelligence of which a user can compete against. |
| User | A person interacting with the FAC software. |
| Player | A user who has initiated a game against either another user or the computer. |
| Bystander | A user who is observing a computer vs. computer game without making a game move. |
| FAC | Refers to Ferret Army Chess the software under development. |
| AIE | Refers to artificial intelligence engine that makes moves for the computer. |
| UI | Refers to user Interface the system by which the player interacts with the FAC software. |
| GP | Refers to any of the game piece(s) which may be a Pawn, Rook, Bishop, Knight, Queen, or King |
| GM | Refers to game move which is the act of moving a game piece on the Board. |
| GB | Refers to the Game Board which is comprised of an 8 square by 8 square board with alternating colors which total 64 possible squares a game piece may occupy. |
| GE | Refers to the game engine which is collectively the code that runs the game pieces, game moves, and Game Board. |
| TP | Refers to the Test Plan used to test the functionality of FAC. |
| Capture | The act of a player removing another player’s game piece by replacing their opponent’s game piece with the attacking game piece thus Capturing said game piece. |
| Check | Refers to a game move where a player’s King is under attack from another player whether a user or the computer. |
| Checkmate | Refers to a game move where a player’s King has no remaining moves where said game piece is not under attack from another player whether a user or the computer. |
| rank | The rows that go from side to side across the chessboard and are referred to by numbers. |
| file | The columns of the chessboard that run vertically and are referred to by letters. |
| Major piece | Referring to the queen or rook. |
| Minor piece | Referring to the bishop or knight. |

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>