**Software Design  
Document**

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 i

Revisions i

1. Introduction 1

1.1 Purpose 1

1.2 System Overview 1

1.3 Definitions, Acronyms and Abbreviations 4

1.4 Supporting Materials 5

1.5 Document Overview 6

2. Architecture 6

2.1 Overview 6

2.2 User Interface (UI) 7

2.3 Game Engine (GE) 7

2.4 Artificial Intelligence Engine (AIE) 7

3. High-Level Design 8

3.1 View / Model Component 1..n 8

# 

# Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Primary Author(s) | Description of Version | Date Completed |
| 1.0 | Alexander Maxwell | Completion of *section 1* including sub sections 1.1, 1.2, 1.3, 1.4, and 1.5. | 02/19/2018 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this Software Design Document (SDD) is to provide a detailed description of the implementation of the "Ferret Army Chess (FAC)" software. It illustrates the design patterns, components, and units involved with creating an interactive version of chess called FAC. This document was generated according to the Software Requirement Specification (SRS) document *version 2.5* agreed upon with the client. The SRS document can be found included with the final deliverables for this project.

## System Overview

The FAC software is broken into three key components, *game engine* (GE), *user interface* (UI), and *artificial intelligence engine* (AIE). These components interact throughout the course of a chess game. The design of FAC largely revolves around the definitions, design, and implementation of GE, UI, and AIE. The architecture of the GE, UI, and AIE will be discussed later in *section 2.*

In accordance with the SRS the FAC software shall have no network functionality. The software shall run locally on desktop / mobile computers with either Windows or macOS operating systems. Due to the locality of the FAC software there will be no external interaction with other systems or actors beyond the *user*. A detailed description of the requirements can be seen in the SRS and will be discussed as required in later sections of this document. The general functional requirements can be seen, in bulleted form, below:

* FAC software shall provide users with three game modes including computer vs. computer, user vs. computer, user vs. user prior to gameplay.
* System shall implement a user interface (UI) allowing the user(s) to select game mode and settings.
* FAC software shall include functionality for limited artificial intelligence engine (AIE). Using randomization, the FAC software’s AIE shall 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 player’s pawn to any other available game piece (GP). An available GP is defined as a GP previously captured by an opponent.
  + Castling – Both Queenside and Kingside.

These special moves shall be available to players only when specific conditions are met. The detailed description of each special move and the specific conditions to be met are covered in-depth in section 4.

* FAC Software shall provide functionality for user(s) to 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 1.

TABLE 1.

*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. |
| 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. |

* 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).
* Users shall have the option to quit an ongoing game at any time. It is not necessary for both players to agree before an individual player quits a game.
* FAC software shall provider user(s) with 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. See section 4.8 for more on stalemate resolution.
  + Turn Time Limit – Set a duration for the length of a single turn.

## Definitions, Acronyms and Abbreviations

TABLE 2.

Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| AIE | Refers to artificial intelligence engine that makes moves for the computer. |
| Bystander | A user who is observing a computer vs. computer game without making a game move. |
| 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. |
| Computer | The system that represents the artificial intelligence of which a user can compete against. |
| FAC | Refers to Ferret Army Chess the software under development. |
| File | The columns of the chessboard that run vertically and are referred to by letters. |
| 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. |
| GM | Refers to game move which is the act of moving a game piece on the Board. |
| GP | Refers to any of the game piece(s) which may be a Pawn, Rook, Bishop, Knight, Queen, or King |
| Major Piece | Refers to specifically to the queen or rook game pieces. |
| Minor Piece | Refers to specifically to the bishop or knight game pieces. |
| Player | A user who has initiated a game against either another user or the computer. |
| Rank | The rows that go from side to side across the chessboard and are referred to by numbers. |
| Reinfeld Value | The numeric value assigned to each game piece the values are as follows: Pawn (1), Bishop (3), Knight (3), Rook (5), Queen (9) |
| SDD | Software Design Document |
| TP | Refers to the Test Plan used to test the functionality of FAC. |
| UI | Refers to interface by which the user interacts with the FAC software. |
| User | A person interacting with the FAC software. |

## Supporting Materials

<Note any references or related materials here.

## Document Overview

This Software Design Document (SDD) is divided into three sections with various subsections. The major sections of this SDD are as follows:

1. Introduction – Structure of the SDD.
2. Architecture – In-depth look at the structure of the FAC software.
3. High Level Design – Overview of each component.

In *section 1. – Introduction*, the purpose, methodology, and reason for this Software Design Document is introduced. In *section 2. – Architecture,* the system design, component design, sub-component design, and class design is discussed in-depth. In *section 3. – High Level Design*, an overview of the structural and functional decomposition of the systems components is discussed. In addition, the general interaction between the system’s components and the *user* is discussed at an abstract level.

# Architecture

## Overview

As previously mentioned the FAC software is broken down into three key components, the *game engine* (GE), user *interface* (UI), and artificial *intelligence engine* (AIE). Each of these primary components can be further sub-divided into several sub-components. An overview of the FAC software at a component/sub-component level can be seen in figure 1.

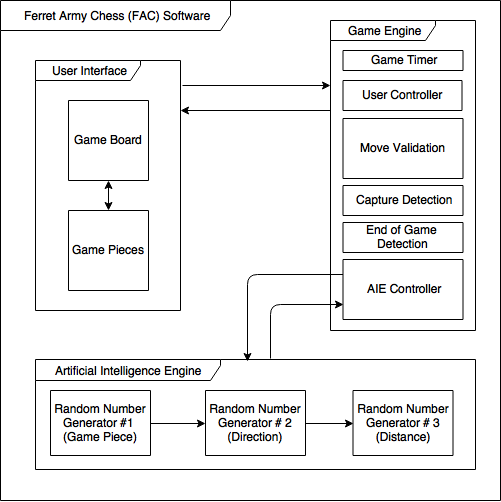


Figure 1. FAC Software Overview

The FAC software was decomposed into the GE, UI, and AIE for several reasons. In order to play a game of chess there are several key components including: *game board* (GB), *game pieces* (GP), *players* (2 *users*), the *computer*, and associated game winning mechanics. Of the key components *users* need only be aware of the GB and GP’s to play FAC. The remaining key components are the “under-the-hood” components supporting the transaction between the GB and GP’s.

Since a *player* shall either be a *human user* or a *computer user* the need arises for the FAC software to support some form of artificial intelligence. In an effort to support modularity it was beneficial to break the *user interface* (UI) elements from the game mechanics including the artificial intelligence engine thus leading to the UI, GE, and AI.

To further help with modularity the AIE was separated from the GE. Since it is highly unlikely that interactions between the UI and GE will change during maintenance the compartmentalization of the AIE leads to greater maintainability. When the AIE ability is improved there will be little refactoring involved since all interaction between the GE and the AIE run through the AIE controller. This means that changes to the AIE will almost be “plug-in-play” since all calls from the GE to the AIE will remain unchanged. The interactions between the GE and AIE will be discussed in detailed later in this document.

## User Interface (UI)

<Describe an element (subsystem, component, etc...) from architecture in further detail. When appropriate, include information on how the element is further broken down and the interactions and relationships between these subcomponents.

## Game Engine (GE)

## Artificial Intelligence Engine (AIE)

The UI has two primary sub-components the game board (GB) which is comprised of 64 individual squares representing every legal space a game piece (GP) may occupy

# High-Level Design

<This section describes in further detail elements discussed in the Architecture. Normally this section would be split into separate documents for different areas of the design.

High-level designs are most effective if they attempt to model groups of system elements from a number of different views.

## View / Model Component 1..n

<Provide a description and diagrams of a system component or set of components that describes a clearly defined view or model of the entire system or a subset of the system.