**Test Plan**

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](#_Toc507943933)

[Revisions i](#_Toc507943934)

[1. Introduction 1](#_Toc507943935)

[1.1 Definitions, Acronyms and Abbreviations 4](#_Toc507943936)

[1.2 Test Plan Objectives 6](#_Toc507943937)

[2. Test Strategy 6](#_Toc507943938)

[2.1 System Test 6](#_Toc507943939)

[2.2 Stress/Performance Test 6](#_Toc507943940)

[2.3 Security Test 6](#_Toc507943941)

[2.4 Automated Test 6](#_Toc507943942)

[2.5 Recovery Test 6](#_Toc507943943)

[2.6 Documentation Test 6](#_Toc507943944)

[2.7 Beta Test 6](#_Toc507943945)

[2.8 User Acceptance Test 6](#_Toc507943946)

[3. Environment Requirements 7](#_Toc507943947)

[3.1 Environment 1 7](#_Toc507943948)

[3.2 Environment 2 7](#_Toc507943949)

[3.3 Environment …n 7](#_Toc507943950)

[4. Functions To Be Tested 7](#_Toc507943951)

# Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.0 | Alvin Romualdo | Update sections 1, 4 | 03/03/2018 |
| 1.1 | Alvin Romualdo | Update sections 4 & TABLE 2 | 03/03/2018 |

# Introduction

The purpose of this Test Plan is to provide a detailed description of the implementation of the "Ferret Army Chess (FAC)" software. It defines the parameters and specifics of the 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 and the Software Design Document (SDD). The SRS document can be found included with the final deliverables for this project.

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. |
| XYCoord | x:y coordinate on the game board |

## Test Plan Objectives

The test plan is to verify and validate that the specific limitations for game environment, game rules, and game pieces are adhered to. It shall insure that the planning and components used in the production of the FAC software complies to the agreed protocols as outlined in the Software Requirement Specification (SRS) that has been approved by the client and the Software Design Document (SDD) developed.

# Test Strategy

<*For each of the following, you should explain how you plan on testing your application, provide a brief description, and possibly scenario/example of how you will accomplish that sort of test. Some of these may not pertain to your application>*

## System Test

A system test will be conducted by playing a game of chess. This type of testing will continue until an error is discovered or the team has decided that no flaws exist and the FAC software is ready for distribution.

## Stress/Performance Test

The FAC software requires very little computational resources. No performance or stress testing is needed.

## Security Test

The FAC software is not currently designed to be played via a network. It is a standalone game. No security test is needed.

## Automated Test

J-unit testing will be conducted to verify a minimum of 80% code coverage of working system. J-unit test will be formatted to check the viability of the functions as noted below <see section 4>. Variations of the J-unit shall be constructed to test each piece types minimal moving requirements and an alternate test to check the maximum moving restrictions are met. It will also test to insure that pieces cannot move outside the game board boundaries, cannot move atop once own piece, cannot move through another piece on the board (with the exception of the night).

## Recovery Test

## Documentation Test

## Beta Test

## User Acceptance Test

# Environment Requirements

*<Here you need to define the environment(s) that you will be use for testing. May be P.C. (please give specific details), mobile, browser, etc.>*

## Environment 1

## Environment 2

## Environment …n

# Functions To Be Tested

**4.1 General setup, assignments, general movement, and conditions.**

4.1.1 Game shall display the board after each turn using designated png characters to represent play pieces in either white or black.

example:

Black pieces

- - - - - - - -

- - - - - - - -

- - - - - - - -

- - - - - - - -



 White pieces

4.1.2 Pieces shall be identified from an array:

1. piece[0] = White Pawn 1 (WP1)
2. piece[1] = White Pawn 2 (WP2)
3. piece[2] = White Pawn 3 (WP3)
4. piece[3] = White Pawn 4 (WP4)
5. piece[4] = White Pawn 5 (WP5)
6. piece[5] = White Pawn 6 (WP6)
7. piece[6] = White Pawn 7 (WP7)
8. piece[7] = White Pawn 8 (WP8)
9. piece[8] = White Rook 1 (WR1)
10. piece[9] = White kNight 1 (WN1)
11. piece[10] = White Bishop 1 (WB1)
12. piece[11] = White Queen (WQ)
13. piece[12] = White King (WK)
14. piece[13] = White Bishop 2 (WB2)
15. piece[14] = White kNight 2 (WN2)
16. piece[15] = White Rook 2 (WR2)
17. piece[16] = Black Pawn 1 (WP1)
18. piece[17] = Black Pawn 2 (WP2)
19. piece[18] = Black Pawn 3 (WP3)
20. piece[19] = Black Pawn 4 (WP4)
21. piece[20] = Black Pawn 5 (WP5)
22. piece[21] = Black Pawn 6 (WP6)
23. piece[22] = Black Pawn 7 (WP7)
24. piece[23] = Black Pawn 8 (WP8)
25. piece[24] = Black Rook 1 (WR1)
26. piece[25] = Black kNight 1 (WN1)
27. piece[26] = Black Bishop 1 (WB1)
28. piece[27] = Black Queen (WQ)
29. piece[28] = Black King (WK)
30. piece[29] = Black Bishop 2 (WB2)
31. piece[30] = Black kNight 2 (WN2)
32. piece[31] = Black Rook 2 (WR2)

4.1.3. Game shall require two players. White shall be played by player 1, black shall be played by player 2.

* 1. Computer AIE can be one or both of the players.
  2. In a human vs. computer scenario, computer will play black player.

4.1.4. Game board shall consist of an 8 x 8 tile matrix: x (horizontal) = { 0 – 7 }; y (vertical) = { 0 – 7 }. Any tile identified outside this range will be considered out of bounds.

1. Each index shall be identified as squares and shall not exceed 64 in quantity.
2. example: y coordinate

0 1 2 3 4 5 6 7

0 - - - - - - - -

1 - - - - - - - -

2 - - - - - - - -

3 - - - - - - - -

x coordinate

4 - - - - - - - -

5 - - - - - - - -

6 - - - - - - - -

7 - - - - - - - -

4.1.5. Game shall accept moves for each player via x:y coordinate (XYCoord) of a game piece to XYCoord of an empty square for moving or XYCoord of square occupied by an opponent’s piece for capture.

1. Movement shall not exceed the limits of the 8 x 8 game board and players shall not be able to access a position outside of this range.
2. Movement shall not be allowed onto an occupied tile or pass through an occupied tile.
   * 1. Knights are an exception.
3. Special moves shall be identified by a XYCoord of the highest ranking piece onto a termination square that can only be achieved using a special move (see King: Castling section 4.2.6.d for details).
4. Capture shall only be allowed when a destination square has been chosen that is occupied by an opponents piece (see Capture specifics for each piece for details).
5. Capture of a king shall result in the completion of the game and will identify the winner (see King: Capture section 4.2.6.e for details).

4.1.6. White player shall move first.

**4.2 Movement restrictions:**

**4.2.1 Pawn**

* 1. **White pawn** 
     1. Game start locations
        1. White Pawn 1 (WP1): piece[0] = square[0] [1]
        2. White Pawn 2 (WP2): piece[1] = square[1] [1]
        3. White Pawn 3 (WP3): piece[2] = square[2] [1]
        4. White Pawn 4 (WP4): piece[3] = square[3] [1]
        5. White Pawn 5 (WP5): piece[4] = square[4] [1]
        6. White Pawn 6 (WP6): piece[5] = square[5] [1]
        7. White Pawn 7 (WP7): piece[6] = square[6] [1]
        8. White Pawn 8 (WP8): piece[7] = square[7] [1]
     2. Movement shall be limited to square[x] [y - 1].
        1. Movement cannot occur if square[x] [y - 1] is occupied by another piece
        2. Movement cannot exceed square [x][0]
     3. Exception (two-space initial move): if pawn is located on square [x] [6], then movement allowed will be square[x] [y - 1] or square [x] [y - 2].
        1. This two-space initial move shall be temporarily recorded as true (will last no longer than one turn) to allow GE to track en passant capture for opposing piece (see Special Moves: en passant – Section 1.1.2.1.a.v. for details).
        2. Two-space initial move recording shall be reset to false after one turn.
     4. Capture can only occur if GE identifies piece to be captured as piece[16 - 31]
        1. it must be located on square [x - 1] [y - 1] or square [x + 1] [y - 1] from players piece.
        2. A captured piece shall have its coordinates set to square [-1] [-1]
     5. En passant capture can only occur if
        1. Opponent’s pawn has just made a two-space initial move (as temporarily recorded by GE).
        2. Players pawn must be in square [x] [3].
        3. Piece to be captured must have terminated its move in an adjacent square [x - 1] [3] or on square [x + 1] [3]
        4. Player’s pawn shall terminate movement in square [x - 1] [2] or square [x + 1] [2] as defined by the location of the opponent’s piece.
        5. A captured pawn shall have its coordinates set to square [-1] [-1]
     6. Pawn promotion:
        1. Pawns that arrive at square [x] [0] shall be promoted to any previously captured white major piece (piece choices are limited to the rook, knight, bishop, or queen).
        2. A list of captured pieces will be presented to the user and XYCoord will select chosen replacement.
        3. Replacement piece will immediately replace pawns location on the GB.
        4. Replacement piece will maintain all the attributes assigned for that piece.
  2. **Black pawn**
     1. Game start locations:
        1. Black Pawn 1 (BP1): piece[16] = square[0] [6]
        2. Black Pawn 2 (BP2): piece[17] = square[1] [6]
        3. Black Pawn 3 (BP3): piece[18] = square[2] [6]
        4. Black Pawn 4 (BP4): piece[19] = square[3] [6]
        5. Black Pawn 5 (BP5): piece[20] = square[4] [6]
        6. Black Pawn 6 (BP6): piece[21] = square[5] [6]
        7. Black Pawn 7 (BP7): piece[22] = square[6] [6]
        8. Black Pawn 8 (BP8): piece[23] = square[7] [6]
     2. Movement shall be limited to square[x] [y + 1].
        1. Movement cannot occur if square[x] [y + 1] is occupied by another piece
        2. Movement cannot exceed square [x][7]
     3. Exception: if pawn is located on square [x] [1], then movement allowed will be square[x] [y + 1] or square [x] [y + 2].
        1. This two-space initial move shall be temporarily recorded as true (will last no longer than one turn) to allow GE to track en passant capture for opposing piece (see Special Moves: en passant – Section 1.1.2.1.b.v. for details).
        2. Two-space initial move recording shall be reset to false after one turn.
     4. Capture can only occur if GE identifies piece to be captured as piece[0 - 15]
        1. it must be located on square [x - 1] [y + 1 ] or square [x + 1] [y + 1] from players piece.
        2. A captured piece shall have its coordinates set to square [-1] [-1]
     5. En passant capture can only occur if
        1. Opponent’s pawn has just made a two-space initial move (as temporarily recorded by GE).
        2. Players pawn must be in square [x] [4].
        3. Piece to be captured must have terminated its move in an adjacent square [x - 1] [4] or on square [x + 1] [4].
        4. Player’s pawn shall terminate movement in square [x - 1] [5] or square [x + 1] [5] as defined by the location of the opponent’s piece.
        5. A captured pawn shall have its coordinates set to square [-1] [-1]
     6. Pawn promotion:
        1. Pawns that arrive at square [x] [7] shall be promoted to any previously captured black major piece (piece choices are limited to the rook, knight, bishop, or queen).
        2. A list of captured pieces will be presented to the user and PAC will select chosen replacement.
        3. Replacement piece will immediately replace pawns location on the GB.
        4. Replacement piece will maintain all the attributes assigned for that piece.
  3. Capturing a pawn shall subtract 1 point from opponent’s Reinfeld points.

**4.2.2 Rook**

1. Game start locations.
2. White Rook 1 (WR1): piece[8] = square[0] [7]
3. White Rook 2 (WR2): piece[15] = square[7] [7]
4. Black Rook 1 (BR1): piece[24] = square[0] [0]
5. Black Rook 2 (BR2): piece[31] = square[7] [0]
6. Movement:
7. Movement options shall be determined XYCoord of the rook and XYCoord of the destination space.
8. Qualified moves shall be determined by GE and shall conform to the limits allowed by these movement locations:
   * + 1. square [x] [y += 1]
       2. square [x] [y -= 1]
       3. square [x += 1] [y]
       4. square [x -= 1] [y]
9. Movement shall step through every space from origin square to destination square.
10. Movement shall not be allowed to step through a space occupied by another piece.
11. Movement shall not exceed the game board limits of square [0] [y], square [7] [y], square [x] [0], square [x] [7].
12. Capturing:
13. Capture shall be determined by XYCoord of the rook and XYCoord of an opponent’s piece
14. White rooks, WR1 and WR2 (piece[8] and piece[15]) can only capture piece[16 – 31].
15. Black rooks, BR1 and BR2 (piece[24] and piece[31]) can only capture piece[0 – 15].
16. Movement to destination of piece to be capture shall follow the same restrictions imposed on movement from origin square to destination square.
17. Movement shall step through every space from origin square to destination square.
18. Movement shall not be allowed to step through a space occupied by another piece.
19. Movement shall not exceed the game board limits of square [0] [y], square [7] [y], square [x] [0], square [x] [7].
20. A captured piece shall have its coordinates set to square [-1] [-1]
21. Rook will terminate its movement at the captured piece’s square location.
22. Castling:
23. Castling shall only be allowed for a human player
24. Castling shall only be allowed if this is the rooks first move
25. A bool must be assigned to all rook pieces to identify if rook has moved.
26. At the start of a game, this bool shall be assigned a value of true.
27. After a rook has finalized it’s first move, the bool shall be set to false.
28. Castling shall be allowed if there are no pieces between origin space and destination space:
29. WR1 (piece[8]): square[1] [0], square[2] [0], square[3] [0] must be unoccupied. WR1 will end its move on square [3] [0].
30. WR2 (piece[15]): square[5] [0] and square[6] [0] must be unoccupied. WR2 will end its move on square [5] [0].
31. BR1 (piece[24]: square[1] [7], square[2] [7], square[3] [7] must be unoccupied. BR1 will end its move on square [3] [7].
32. BR2 (piece[31]): square[5] [7] and square[6] [7] must be unoccupied. BR2 will end its move on square [5] [7].
33. Castling shall be restricted to limits imposed on king (see King section 4.2.6.d.)
34. Capturing a rook shall subtract 5 points from opponent’s Reinfeld points.

**4.2.3 Knight**

1. Game start locations.
2. White kNight 1 (WN1): piece[9] = square[1] [7]
3. White kNight 2 (WN2): piece[14] = square[6] [7]
4. Black kNight 1 (BN1): piece[25] = square[1] [0]
5. Black kNight 2 (BN2): piece[30] = square[6] [0]
6. Movement:
7. Movement options shall be determined by XYCoord of the knight and XYCoord of the destination space.
8. Qualified moves shall conform to the limits allowed by these movement locations:
9. square [x + 1] [y + 2]
10. square [x + 2] [y + 1]
11. square [x + 2] [y - 1]
12. square [x + 1] [y - 2]
13. square [x - 1] [y - 2]
14. square [x - 2] [y - 1]
15. square [x - 2] [y + 1]
16. square [x - 1] [y + 2]
17. Movement shall not be restricted by pieces (friend or foe) between origin square to destination square.
18. Movement shall not exceed the game board limits of square [0] [y], square [7] [y], square [x] [0], square [x] [7].
19. Capture:
20. Capture shall be determined by XYCoord of the knight and XYCoord of an opponents piece
21. White knights, WN1 and WN2 (piece[9] and piece[14]) can only capture piece[16 – 31].
22. Black knights, BN1 and BN2 (piece[25] and piece[30]) can only capture piece[0 – 15].
23. Movement to destination capture piece shall follow the same restrictions imposed on normal movement from origin square to destination square as described above.
24. A captured piece shall have its coordinates set to square [-1] [-1]
25. Knight shall replace captured piece on the square formerly occupied by captured piece.
26. Capturing a knight shall subtract 3 points from opponent’s Reinfeld points.

**4.2.4 Bishop**

1. Game start locations.
2. White Bishop 1 (WB1): piece[10] = square[2] [7]
3. White Bishop 2 (WB2): piece[13] = square[5] [7]
4. Black Bishop 1 (BB1): piece[26] = square[2] [0]
5. Black Bishop 2 (BB2): piece[29] = square[5] [0]
6. Movement:
7. Movement options shall be determined by XYCoord of the bishop and XYCoord of the destination space.
8. Qualified moves shall be determined by GE and shall conform to the limits allowed by these movement locations:
9. square [x += 1] [y += 1]
10. square [x += 1] [y -= 1]
11. square [x -= 1] [y -= 1]
12. square [x -= 1] [y += 1]
13. Movement shall step through every space from origin square to destination square.
14. Movement shall not be allowed to step through a space occupied by another piece.
15. Movement shall not exceed the game board limits of square [0] [y], square [7] [y], square [x] [0], square [x] [7].
16. Capturing:
17. Capture shall be determined by XYCoord of the bishop and XYCoord of an opponent’s piece
18. White bishops, WB1 and WB2 (piece[10] and piece[13]) can only capture piece[16 – 31].
19. Black bishops, BB1 and BB2 (piece[26] and piece[29]) can only capture piece[0 – 15].
20. Movement to destination capture piece shall follow the same restrictions imposed on movement from origin square to destination square.
21. Movement shall step through every space from origin square to destination square.
22. Movement shall not be allowed to step through a space occupied by another piece.
23. Movement shall not exceed the game board limits of square [0] [y], square [7] [y], square [x] [0], square [x] [7].
24. A captured piece shall have its coordinates set to square [-1] [-1]
25. Bishop shall terminate its movement at the captured piece’s square location.
26. Capturing a bishop shall subtract 3 points from opponent’s Reinfeld points.

**4.2.5 Queen**

1. Game start locations.
2. White Queen (WQ): piece[11] = square[3] [7]
3. Black Queen (BQ): piece[27] = square[3] [0]
4. Movement:
5. Movement options shall be determined by XYCoord of the queen and XYCoord of the destination space.
6. Qualified moves shall be determined by GE and shall conform to the limits allowed by these movement locations:
7. square [x] [y += 1]
8. square [x += 1] [y += 1]
9. square [x += 1] [y]
10. square [x += 1] [y -= 1]
11. square [x] [y -= 1]
12. square [x -= 1] [y -= 1]
13. square [x -= 1] [y]
14. square [x -= 1] [y += 1]
15. Movement shall step through every space from origin square to destination square.
16. Movement shall not be allowed to step through a space occupied by another piece.
17. Movement shall not exceed the game board limits of square [0] [y], square [7] [y], square [x] [0], square [x] [7].
18. Capturing:
19. Capture shall be determined by XYCoord of the queen and XYCoord of an opponent’s piece
20. The white queen, WQ (piece[11]) can only capture piece[16 – 31].
21. The black queen, BQ (piece[27]) can only capture piece[0 – 15].
22. Movement to destination capture piece shall follow the same restrictions imposed on movement from origin square to destination square.
23. Movement shall step through every space from origin square to destination square.
24. Movement shall not be allowed to step through a space occupied by another piece.
25. Movement shall not exceed the game board limits of square [0] [y], square [7] [y], square [x] [0], square [x] [7].
26. A captured piece shall have its coordinates set to square [-1] [-1]
27. Queen shall terminate its movement at the captured piece’s square location.
28. Capturing a queen shall subtract 9 points from opponent’s Reinfeld points.

**4.2.6 King**

1. Game start locations.
2. White King (WK): piece[12] = square[4] [7]
3. Black King (BK): piece[28] = square[4] [0]
4. Movement:
5. Movement options shall be determined XYCoord of the king and XYCoord of the destination space.
6. Qualified moves shall be determined by GE and shall conform to the limits allowed by these movement locations:
7. square [x] [y + 1]
8. square [x + 1] [y + 1]
9. square [x + 1] [y]
10. square [x + 1] [y - 1]
11. square [x] [y - 1]
12. square [x - 1] [y - 1]
13. square [x - 1] [y]
14. square [x - 1] [y + 1]
15. Movement shall not exceed the game board limits of square [0] [y], square [7] [y], square [x] [0], square [x] [7].
16. King cannot move into a square that will place it in check.
17. Capturing:
18. Capture shall be determined by XYCoord of the king and XYCoord of an opponents piece
19. The white king, WK (piece[12]) can only capture piece[16 – 27] and piece[29 – 31].
20. The black king, BK (piece[28]) can only capture piece[0 – 11] and piece [13 – 15].
21. A king cannot capture another king as it violates the rule that a king cannot move into a square that places it in check.
22. A captured piece shall have its coordinates set to square [-1] [-1]
23. King shall terminate its movement at the captured piece’s square location.
24. King cannot capture a piece if its destination square will result in it being check.
25. Castling
26. Castling shall only be allowed to a human player
27. Castling shall only be allowed if this is the kings first move
28. A bool must be assigned to king pieces to identify if king has moved
29. At the start of the game, this bool will be assigned a value of true.
30. After the king’s first move this bool will be set to false for the remainder of the game.
31. Castling shall be determined by entering XYCoord of the king and XYCoord of final destination of king as determined below.
32. Castling shall be allowed if there are no pieces between origin space and destination space:
33. If castling with WR1: WK (piece[12]): square[1] [0], square[2] [0], square[3] [0] must be unoccupied. WK will end its move on square [2] [0]. Square [2] [0], square [3] [0] cannot be threatened by an opponents piece.
34. If castling with WR2: WK (piece[12]): square[5] [0] and square[6] [0] must be unoccupied. WK will end its move on square [6] [0]. Square [5] [0], square [6] [0] cannot be threatened by an opponents piece.
35. If castling with BR1: BK (piece[28]: square[1] [7], square[2] [7], square[3] [7] must be unoccupied. BK will end its move on square [2] [7]. Square [2] [7], square [3] [7] cannot be threatened by an opponents piece.
36. If castling with BR2: BK (piece[28]): square[5] [7] and square[6] [7] must be unoccupied. BR2 will end its move on square [6] [7]. Square [5] [7], square [6] [7] cannot be threatened by an opponents piece.
37. Castling shall be restricted to limits imposed on rook (see rook section 4.2.2.d.).
38. If king is captured it shall result in the end of the game and winner status is assigned to the capturing side.

**4.2.7 King validation**

1. A king shall not be allowed to move into a space that will automatically place it in check.
   1. A king shall not be allowed to move to a space adjacent to the opponents king.
2. A piece shall not be allowed to move or capture if that move will place the king in check.
3. When castling, a king cannot move across a tile that could place it in check if the king ended its move on that tile.

**4.3 Other Features**

**4.3.1 Stalemate**

1. Stalemate option shall be activated if a king is the only remaining piece for one side on the board.
2. GE will track king’s available moves
3. If a king cannot move into an adjacent space without being placed in check, the GE shall determine that a stalemate must be declared.
4. Adjacent spaces are defined by
   * + - 1. square [x] [y + 1]
         2. square [x + 1] [y + 1]
         3. square [x + 1] [y]
         4. square [x + 1] [y - 1]
         5. square [x] [y - 1]
         6. square [x - 1] [y - 1]
         7. square [x - 1] [y]
         8. square [x - 1] [y + 1]
5. If a stalemate is determined, the GE shall declare a winner based on which side has the highest Reinfeld points.
6. Winner and total Reinfeld points shall be displayed.

**4.3.2 Game Timer**

1. Game timer option shall be available as an option to the user before the game starts at the Settings page.
2. Selecting the game timer option will activate the turn timer option (see Turn Timer section 1.1.2.9.).
3. The game timer shall have a default setting of 60 minutes.
4. The game timer shall have the option to change the setting to any time in 1-minute increments.
5. Game timer shall begin countdown at the start of the game.
6. Game timer shall continually run and be displayed during the course of the game.
7. Capture of the king shall cancel game timer and winner will be declared based on whose king was captured.
8. Upon completion of game time, a winner shall be determined from remaining Reinfeld points.

**4.3.3 Turn Timer**

1. Turn timer option shall be available as an option to the user before the game starts on the Settings page.
2. Turn timer option shall be activated automatically of game timer option is activated.
3. The turn timer shall have a default setting of 5 minutes.
4. The turn timer shall have an option to change the setting to any time in 1-second increments.
5. Turn timer shall reset to selected turn time at the start of each turn then begin a countdown to 0.
6. Turn time shall be displayed during game play.
7. Capture of the king shall cancel turn timer and winner will be declared based on whose king was captured.
8. If a player does not make a move in the allotted turn time, that player shall have declared a forfeit. The opposite player shall be declared the winner.

**4.3.4 Artificial Intelligence Engine**

1. When the computer is chosen to be one or both of the players, it shall randomly choose 3 numbers:
   1. First random number: shall determine the game piece from the array of 15 possible pieces for that color. If it is white’s turn, it shall choose an index number between 0 and 15 to determine which piece it will move.
   2. Second random number: It shall choose a direction of travel from 0 to 7. The direction ‘0’ shall represent a direction in square[x] [y + 1] direction. The remaining directions shall be laid out in a clockwise manner from direction 0. Some pieces shall only be allowed movement in certain directions.
      1. Pawns shall only be allowed movement in one direction, but capture in diagonal directions toward the opposite side of the board.
      2. Rooks can only move in the 0, 2, 4, and 6 directions.
      3. Bishops can only move in the 1, 3, 5, and 7 directions.
      4. Knights, Queens, and Kings can move in all eight directions.
         1. Kights moves are different than other pieces as they move in an L-shaped pattern and they are allowed to jump over pieces to get to their destination.
   3. The third random number: shall determine the distance, in tiles, that a piece can travel.
      1. Rooks, Bishops, and Queens can move the length of the board, 0 – 7 tiles, in their allowed moving directions.
      2. Knights move in a single phase of the turn which shall allow them to jump over pieces. However, this move is still a single move, so distance = 1.
      3. King pieces can only move a distance of one tile.
      4. Pawn pieces can only move 1 tile unless they are on their starting rank then they can move 2 tiles.
2. None of these movements shall allow a piece to go past the boundaries of the game board.
3. None of these movements shall allow a piece to occupy a tile already occupied by another piece, unless it is capturing that other piece. Hence, it will replace the captured piece.
4. None of these movements shall allow a piece to move through a tile occupied by another piece. With the exception of the Knight.
5. AIE shall not be allowed to do any of the “Special Moves”.
6. AIE shall not weigh movements or captures to determine the best course of action.