**Induction draughts project**

**System Requirements Specification**

**999/170**

**Document Reference SRS\_999170.docx**

**7 October 2011**

**Issue 2**

**Copyright Notice**

This document has been prepared by IPL Information Processing Limited (“IPL”) and, subject to any existing rights of other parties, IPL is the owner of the copyright of this document. No part of this document may be copied, reproduced, stored in a retrieval system, disclosed to a third party or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of IPL.

**© IPL Information Processing Limited 2011**

**Authorisation**

|  |  |  |
| --- | --- | --- |
| Approved by: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Software Project Manager | Brian Bassil IPL |
| Approved by: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Training Coordinator | Christine Harrison IPL |
|  |  |  |
|  |  |  |

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Issue | Author | Comments |
| 30/06/2010 | 1 | Brian Bassil | First release |
| 07/10/2011 | 2 | C Harrison | Corrections. |

Contents

[1 Introduction 1](#_Toc262724342)

[2 Applicable Documents 1](#_Toc262724343)

[3 System Overview and Terminology 1](#_Toc262724344)

[4 System Architecture 2](#_Toc262724345)

[5 Detailed Requirements 3](#_Toc262724346)

[5.1 Requirements Notation 3](#_Toc262724347)

[5.2 Functional Requirements 4](#_Toc262724348)

[5.2.1 New Game Dialog 4](#_Toc262724349)

[5.2.2 Computer Player 4](#_Toc262724350)

[5.2.3 Human Player 4](#_Toc262724351)

[5.2.4 Remote Player 4](#_Toc262724352)

[5.2.5 PDN file reading 5](#_Toc262724353)

[5.2.6 In Game Controls 5](#_Toc262724354)

[5.2.7 Game End 5](#_Toc262724355)

[5.2.8 Remote Game control 5](#_Toc262724356)

[6 Deliverables 6](#_Toc262724357)

[7 Supporting Documentation 6](#_Toc262724358)

[7.1 THE RULES OF DRAUGHTS ACCORDING TO THE 1991 EDA HANDBOOK 6](#_Toc262724359)

[7.1.1 The Numbered Board 6](#_Toc262724360)

[7.1.2 ORDER OF PLAY 7](#_Toc262724361)

[7.1.3 THE MOVES 7](#_Toc262724362)

[7.1.4 Ordinary Move of a Man 7](#_Toc262724363)

[7.1.5 Ordinary Move of a King 7](#_Toc262724364)

[7.1.6 Capturing Move of a Man 7](#_Toc262724365)

[7.1.7 Capturing Move of a King 7](#_Toc262724366)

[7.1.8 Capturing In General 7](#_Toc262724367)

[7.1.9 Touching the Pieces 8](#_Toc262724368)

[7.1.10 FALSE, IMPROPER OR ILLEGAL MOVES 8](#_Toc262724369)

[7.1.11 RESULT OF THE GAME 8](#_Toc262724370)

[7.1.12 Definition of a Win 8](#_Toc262724371)

[7.1.13 Definition of a Draw 8](#_Toc262724372)

[7.2 Portable Draughts Notation (PDN) 9](#_Toc262724373)

[7.2.1 TAGS 9](#_Toc262724374)

[8 Glossary 10](#_Toc262724375)

# Introduction

* 1. The system is to play the game of English draughts as defined in 7.1. It will allow any combination of human/computer opponents to play against each other. The game will only allow human players to play legal moves therefore section 7.1.10 will not be applicable.
  2. A GUI has already been developed that must be used in the implementation of the game; this is fully defined in [1].
  3. It is desirable that the game can interface to other draughts games. If implemented this should meet the interface in [1].
  4. The game shall support the reading and writing of Portable Draughts Notation (PDN) files. This will allow the game to be setup with an initial position.

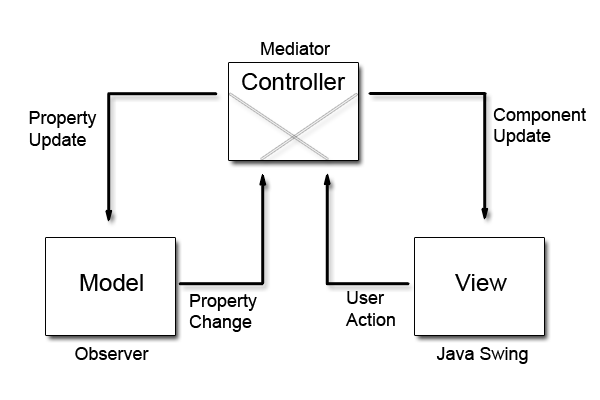
# Applicable Documents

Inclusion of a document within this section does not imply conformance to all of the information within the document. Each requirement listed in the Detailed Requirements section shall explicitly define the extent of any referenced information and its applicability to that requirement.

1. Draughts View Package Specification  
   IPL  
   10/10/2011  
   4

# System Overview and Terminology

* 1. The system will make use of MVC. A definition of a model, view and controller from <http://java.sun.com/developer/technicalArticles/javase/mvc/> states:
* **Model** - The model represents data and the rules that govern access to and updates of this data. In enterprise software, a model often serves as a software approximation of a real-world process.
* **View** - The view renders the contents of a model. It specifies exactly how the model data should be presented. If the model data changes, the view must update its presentation as needed. This can be achieved by using a *push model*, in which the view registers itself with the model for change notifications, or a *pull model*, in which the view is responsible for calling the model when it needs to retrieve the most current data.
* **Controller** - The controller translates the user's interactions with the view into actions that the model will perform. In a stand-alone GUI client, user interactions could be button clicks or menu selections, whereas in an enterprise web application, they appear as GET and POST HTTP requests. Depending on the context, a controller may also select a new view -- for example, a web page of results -- to present back to the user.

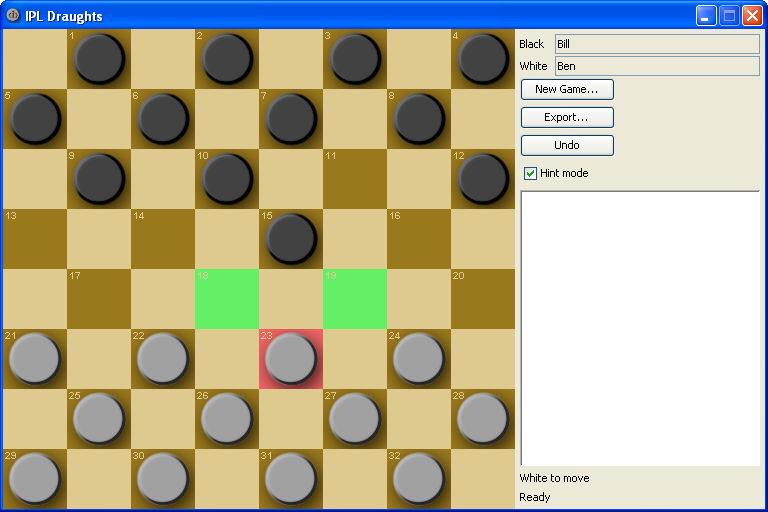


# System Architecture

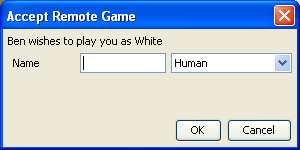
* 1. At startup a new game dialog is presented showing



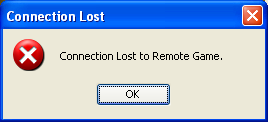
* 1. Once a game has started the main UI will be shown as



* 1. If a request for a remote game is received an accept game dialog will be displayed.



* 1. Errors will be displayed using an error dialog. Accepting any error causes a new game to be initiated.



# Detailed Requirements

## Requirements Notation

* 1. The Requirements Notation is applicable to the information contained in (and only in) the Detailed Requirements section of this specification.
  2. The Requirements Notation applicable to requirements text shall not be applicable to informational text.
  3. Informational text is provided to simply support the requirements and does not imply any additional functionality to that provided by the requirements text. *Information text is indicated by the use of italics.*
  4. The term shall is used to indicate a mandatory requirement..

## Functional Requirements

* 1. When the game is started a new game dialog shall be displayed.

### New Game Dialog

* 1. The game shall allow the user to select whether the computer or a human opponent is controlling the pieces for each side.
  2. The game shall allow the user to select a remote game to play either side of the board.
  3. The game shall allow the loading of a PDN file, as described in 7.2, to setup an initial board position.
  4. The dialog shall allow entry of names for human players.

### Computer Player

* 1. The computer shall only make valid moves as defined in 7.1.
  2. The computer shall make its move within 10 seconds.
  3. A computer player name shall always be displayed as “Computer”.

### Human Player

* 1. The game shall only allow a player to make valid moves as defined in 7.1.
  2. The player shall be allowed unlimited time in which to make their move.
  3. When a player selects a piece the square it sits on shall be highlighted.
  4. A human player name shall be displayed as the name entered in the New Game dialog or Accept Game dialog

#### Move Hints

* 1. The game shall allow the user to enable or disable move hints at any point.
  2. When a piece is selected by a human player the square that piece is on shall be highlighted to indicate it is selected.
  3. When a square is selected and hints are enabled the legal moves a piece can make, as defined in 7.1, will be highlighted.
  4. When legal moves are being highlighted only the first move in a chain shall be highlighted.

*Note that if a capture move is possible then any non capture move is considered illegal so shouldn’t be highlighted.*

* 1. If, after a capture has occurred, a further capture may be made then this shall be highlighted when move hints are enabled.

### Remote Player

* 1. The game shall support connection to a remote instance of the game.
  2. The game shall support being either a server or client for a remote instance of the game.
  3. The remote interface shall use the protocol defined in [1].
  4. The game shall only allow a remote player to make valid moves as defined in 7.1.
  5. Either side of the game may be controlled by a remote player.
  6. The name entered for a remote player on the new game dialog shall be used to identify the remote host.
  7. There shall be at most one remote player.

### PDN file reading

* 1. The game shall report an error if a PDN file it loads does not comply with the syntax as defined in 7.2.
  2. If a PDN file is invalid the game shall not load the file.
  3. The game shall report success if the PDN file is successfully loaded.
  4. PDN file loading shall be supported for all combinations of human, computer and remote player.
  5. The contents of the FEN tag shall be sent to the remote player so that both game instances can be initialised to the same start condition.

### In Game Controls

#### New Game

* 1. The game shall allow the user to start a new game at any point.

#### PDN exporting

* 1. The game shall allow the user to export a PDN file at any point during a game.
  2. The user shall be prompted for the filename and location of the PDN file before an export occurs.
  3. PDN file export will export all moves that have been recorded.
  4. PDN file export will export a FEN tag, with associated Setup tag, that represents the game position.

#### Undo

* 1. An undo button shall be provided that will undo the last human move.
  2. The undo functionality shall allow all moves to be undone sequentially back to the beginning of the game.
  3. The undo functionality shall be disabled when one or both of the players are controlled remotely.
  4. Undo shall not be supported once the game is complete.

### Game End

* 1. The game shall detect when a game has ended, as defined in 7.1.11, and report the result.

### Remote Game control

#### Remote Game Initiation

* 1. When a new game has been started with a remote player a request shall be sent to the computer identified in the dialog.

#### Accept Game dialog

* 1. A dialog shall be displayed when a remote request is received unless the game is in system test mode.
  2. The Accept Game dialog shall allow the player to select either a human or computer player.
  3. The Accept Game dialog shall allow a human player to input their name.

#### Error Handling

* 1. All messages sent to a remote machine shall be acknowledged with an OK/FAIL status as defined in [1].
  2. If a message received over RMI is invalid then a FAIL status shall be returned and an error dialog displayed.
  3. If a FAIL status is received over RMI then an error dialog shall be displayed.
  4. Once an error dialog has been acknowledge the game will revert to its start-up state.
  5. A FAIL status shall be sent if an invalid move has been received.
  6. A FAIL status shall be sent if a message not conforming to [1] is received.
  7. A FAIL status shall be sent if an invalid FEN setup is received.

#### System Test Mode

* 1. If the game is started with the property “SystemTest” set, it shall enter system test mode.
  2. In system test mode the game shall start with a blank board and make the new game button unavailable.
  3. In system test mode the game shall immediately accept a remote game request and play locally as the computer.

# Deliverables

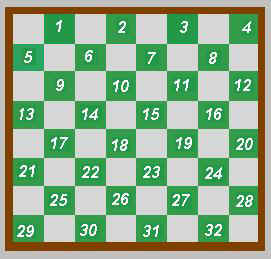
* 1. The deliverable shall consist of a single runnable jar file.

# Supporting Documentation

## THE RULES OF DRAUGHTS ACCORDING TO THE 1991 EDA HANDBOOK

### The Numbered Board

The following is based upon the rules as described (THE RULES OF DRAUGHTS ACCORDING TO THE 1991 EDA HANDBOOK).  
At the start of the game the black pieces occupy squares 1-12 the white pieces occupy squares 21-32

1. DRAUGHTS BOARD AND MEN
   1. The draughts board is square in shape and is divided into 64 squares of equal size, alternately light and dark in colour(technically called black and white).
   2. The board is placed between the two players such that the bottom left-hand corner square is black.
   3. The game is played only on the black squares, which for the purpose of reference are assigned numbers from 1 to 32.
   4. Each player starts with 12 discs, or "men", all of equal size. One player has dark coloured men (called black) and the other has light coloured men (called white). Note... The colours of the men must make a distinct contrast with the colours of the squares of the board.
   5. At the start of play the black men occupy squares 1 to 12 and the white men occupy squares 21 to 32.

### ORDER OF PLAY

* 1. To start the first game the players decide by the toss of a coin which colour they will play. In subsequent games the players alternate colours.
  2. The first move in each game is made by the player with the black men; thereafter the moves are made by each player in turn.

### THE MOVES

* 1. There are fundamentally 4 types of move: the ordinary move of a man, the ordinary move of a king, the capturing move of a man and the capturing move of a king.

### Ordinary Move of a Man

* 1. An ordinary move of a man is its transfer diagonally forward left or right from one square to an immediately neighbouring vacant square.
  2. When a man reaches the farthest row forward (the king-row or crownhead) it becomes a king, and this completes the turn of play. The man is crowned by the opponent, who must place a man of the same colour on top of it before making his own move. (It may be necessary to borrow from another set if no captured man is available for the purpose).

### Ordinary Move of a King

* 1. An ordinary move of a king (crowned man) is from one square diagonally forward or backward, left or right, to an immediately neighbouring vacant square.

### Capturing Move of a Man

* 1. A capturing move of a man is its transfer from one square over a diagonally adjacent and forward square occupied by an opponent`s piece (man or king) and on to a vacant square immediately beyond it. (A capturing move is called a "jump"). On completion of the jump the captured piece is removed from the board.

### Capturing Move of a King

* 1. A capturing move of a king is similar to that of a man, but may be in a forward or backward direction.

### Capturing In General

* 1. If a jump creates an immediate further capturing opportunity, then the capturing move of the piece (man or king) is continued until all the jumps are completed. The only exception is that if a man reaches the king-row by means of a capturing move it then becomes a king but may not make any further jumps in the same turn. At the end of the capturing sequence, all captured pieces are removed from the board, in the order in which they were jumped.
  2. All capturing moves are compulsory, whether offered actively or passively. If there are two or more ways to jump, a player may select any one he wishes, not necessarily that which gains most pieces. Once started, a multiple jump must be carried through to completion.

### Touching the Pieces

* 1. Either player, on intimating his intention to his opponent, is entitled to adjust his own or his opponent`s pieces properly on their squares at any time during the course of the game.
  2. If a player on his turn to move touches a piece he must play that piece, unless he has given an adjustment warning. If the piece is not legally playable, rule .3 applies.
  3. If any part of a playable piece is played over a corner of the square on which it is stationed, the move must be completed in that direction

### FALSE, IMPROPER OR ILLEGAL MOVES

* 1. A player making a false, improper or illegal move shall be cautioned for the first offence, and the move recalled. He shall forfeit the game for any subsequent false, improper or illegal move made in that game. This applies, for example, if a player:
* Omits to capture or to complete a multiple capture. (This supersedes the old "huff" rule).
* On his turn to play touches an unplayable piece.
* Moves a piece, either in an ordinary move or in a capturing move, on to a wrong square.
* Moves an uncrowned man backwards.
* When capturing, removes an opponent`s piece or pieces not in a position to be captured by that move.
* When capturing, inadvertently removes one or more of his own pieces.
* Continues a capturing move through the king-row with a man not already crowned.
* Moves a piece when it is not his turn to play.
  1. If any of the pieces are accidently displaced by the players or through any cause outside their control, the pieces are replaced without penalty and the game is continued.
  2. A player who refuses to adhere to the rules shall immediately forfeit the game.

### RESULT OF THE GAME

* 1. There are only two possible states to define: the win and the draw.

### Definition of a Win

* 1. The game is won by the player who can make the last move; that is, no move is available to the opponent on his turn to play, either because all his pieces have been captured or his remaining pieces are all blocked.
  2. A player also wins if his opponent:
* Resigns at any point;
* Forfeits the game by contravening the rules.

### Definition of a Draw

* 1. The game is drawn if at any stage both players agree on such a result. (This usually occurs when neither side can force a win).
  2. 50-move rule. The game shall be declared drawn if, at any stage of the game, a player can demonstrate to the satisfaction of the referee that both the following conditions hold**:**
* Neither player has advanced an uncrowned man towards the king-row during the previous 50 moves;
* No pieces have been removed from the board during the previous 50 moves.

**Note** For the purposes of this rule, a move shall be said to consist of one black move and one white move.

## Portable Draughts Notation (PDN)

There are a range of variants of PDN available that have a number of minor conflicts. The PDN to be supported by the game must meet the syntax described below, it is a restricted set of PDN 2.0 as described at (~~http://www.nemesis.info/pdn2.txt~~, http://pdn.fmjd.org/introduction.html)

Example PDN file

[Event "Induction"]

[Site "IPL"]

[Date "2010.03.09"]

[Round "1"]

[White "Name"]

[Black "Name"]

[Result "0-2"]

1.32-28 17-22 2.28x17 11x22 3.37-32 6-11 4.41-37 12-17 5.46-41 8-12 6.34-30 2-8 7.30-25 19-23 8.35-30 1-6 9.40-35 13-19 10.31-27 22x31 11.36x27 9-13 12.33-28 4-9 13.41-36 17-22 14.28x17 11x31 15.37x26 23-28 16.32x23 19x28 17.42-37 20-24 18.30x19 14x23 19.37-31 16-21 20.26x17 12x21 21.31-27 21x32 22.38x27 6-11 23.47-42 15-20 24.25x14 10x19 25.39-33 28x39 26.44x33 8-12 27.42-38 23-28 28.33x22 12-17 29.49-44 17x28 30.38-33 28x39 31.44x33 18-22 32.27x18 13x22 33.43-38 19-23 34.38-32 11-17 35.32-27 22x31 36.36x27 9-13 37.45-40 13-18 0-2

### TAGS

* 1. All of the tags are case insensitive.
  2. Each tag will be on a new line.
  3. The Event tag will contain “Induction”.
  4. The Site tag will contain “IPL”.
  5. The date tag will contain the date the game was started in YYYY.MM.DD format.
  6. The round tag will contain “1”.
  7. The White tag will contain the name entered for the white player in the game. This will be white by default.
  8. The Black tag will contain the name entered for the black player in the game. This will be black by default.
  9. The result tag shall contain:
* “1-0” indicating that black has won;
* “0-1” indicating that white has won;
* “1/2-1/2” indicating that the game was a draw;
* “\*” indicating that the game was unfinished, or the result unknown.
  1. The FEN tag will be used to setup an initial position. The syntax of the tag is

[TURN]:[COLOUR1][[K][SQUARE\_NUM][,]...]:[COLOUR2][[K][SQUARE\_NUM][,]...]

* 1. The fen tag is always preceded by a setup tag which will contain “1”. An example is shown below:

[SetUp “1”]

[FEN "B:W18,24,27,28,K10,K15:B12,16,20,K22,K25,K29"]

This setup indicates that black is to go next and the following pieces are on the board:-

* White pieces on squares 18,24,27 and 28;
* White kings on squares 10 and 15;
* Black pieces on squares 12,16 and 20;
* Black kings on squares 22, 25 and 29.

# Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
|  |  |
| FEN | Forsyth-Edwards Notation |
| ICD | Interface Control Document |
| MVC | Model-View-Controller |
| PDN | Portable Draughts Notation |
| RMI | Remote Method Invocation |
| SRS | System Requirements Specification |
|  |  |
|  |  |

**- End of Document -**