**controller**

**com.ipl.training.induction.draughts.controller**

**Package Specification**

**Project Reference 999/170**

**Document Reference draughts\_controller\_ps**

**22 November 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**

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Issue | Author | Comments |
| 10/06/2010 | 1 | R.Miskin | Issued |
| 22/11/2011 | 2 | J Poole, C Harrison | Updated to reflect new Java standard, including diagrams. |

Contents

[1 Package Identity 1](#_Toc305414994)

[2 Description 1](#_Toc305414995)

[3 Interface 2](#_Toc305414996)

[4 Structure 3](#_Toc305414997)

[4.1 Package overview 3](#_Toc305414998)

[4.2 Package Detail 4](#_Toc305414999)

[5 Element Descriptions and Interfaces 5](#_Toc305415000)

[5.1 PlayerData 5](#_Toc305415001)

[5.1.1 Public Methods 5](#_Toc305415002)

[5.2 Enumeration PlayerData.PlayerType 8](#_Toc305415003)

[5.2.1 Public Methods 8](#_Toc305415004)

[5.2.2 Package Access Methods 8](#_Toc305415005)

[5.2.3 Public Fields 8](#_Toc305415006)

[5.2.4 Package Access Fields 8](#_Toc305415007)

[5.3 GameData 8](#_Toc305415008)

[5.3.1 Public Methods 9](#_Toc305415009)

[5.4 Enumeration GameState 12](#_Toc305415010)

[5.4.1 Public Methods 12](#_Toc305415011)

[5.4.2 Package Access Methods 12](#_Toc305415012)

[5.4.3 Public Fields 12](#_Toc305415013)

[5.5 Enumeration Square 13](#_Toc305415014)

[5.5.1 Public Methods 13](#_Toc305415015)

[5.5.2 Package Access Methods 14](#_Toc305415016)

[5.5.3 Public Fields 14](#_Toc305415017)

[5.5.4 Package Access Fields 14](#_Toc305415018)

[5.6 SquareData 14](#_Toc305415019)

[5.6.1 Public Methods 15](#_Toc305415020)

[5.7 Interface IDraughtsController 17](#_Toc305415021)

[5.7.1 Public Methods 18](#_Toc305415022)

[5.7.2 Public Fields 22](#_Toc305415023)

[5.8 DraughtsController 22](#_Toc305415024)

[5.8.1 Public Methods 23](#_Toc305415025)

[5.8.2 Public Fields 28](#_Toc305415026)

[5.9 Interface IDraughtsModel 28](#_Toc305415027)

[5.9.1 Nested classes 28](#_Toc305415028)

[5.9.2 Methods 28](#_Toc305415029)

[5.9.3 Public Fields 31](#_Toc305415030)

[5.9.4 Package Access Fields 32](#_Toc305415031)

[5.10 Interface IDraughtsView 32](#_Toc305415032)

[5.10.1 Public Methods 32](#_Toc305415033)

[5.11 Enumeration IDraughtsView.Type 33](#_Toc305415034)

[5.11.1 Public Methods 33](#_Toc305415035)

[5.11.2 Package Access Methods 33](#_Toc305415036)

[5.11.3 Public Fields 33](#_Toc305415037)

[5.11.4 Package Access Fields 33](#_Toc305415038)

[5.12 DraughtsChangeEvent 34](#_Toc305415039)

[5.12.1 Public Methods 34](#_Toc305415040)

[5.12.2 Package Access Methods 35](#_Toc305415041)

[5.12.3 Public Fields 35](#_Toc305415042)

[5.12.4 Package Access Fields 35](#_Toc305415043)

[6 Resource Requirements 35](#_Toc305415044)

[7 Test Plan 35](#_Toc305415045)

[8 Scenarios 35](#_Toc305415046)

[9 Supporting Documentation 35](#_Toc305415047)

# Package Identity

Package Name - controller  
Package Identity - com.ipl.training.induction.draughts.controller

# Description

The draughts controller package contains the mvc controller class, and interfaces required between the controller/model and controller/view.

# Interface

The controller interfaces with the model and the views. The controller:

* Send events to the views (ui) to inform them of an update they may wish to render.
* Accepts calls from a view to indicate user action with the view
* Sends property updates to the model
* Accepts property updates from the model.



# Structure

## Package overview

The overview shows how the controller interacts with the views and model through interfaces and/or direct interaction.



## Package Detail



# Element Descriptions and Interfaces

## PlayerData

Stores the name and type (HUMAN/COMPUTER/REMOTE) of a player. There will be one for each of the white and black players. For remote players the name will be the hostname of the remote machine.

|  |  |
| --- | --- |
| Attribute | Value |
| Visibility | Public |
| Modifiers | Final |
| Extends | - |
| Implements | - |

### Public Methods

#### PlayerData

Constructor to create a PlayerData object. Throws IllegalArgumentException if either parameter is null.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | IllegalArgumentException |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| name | String | The name of this player |
| type | PlayerType | The type of this player |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - |  |

#### getName

Return the name of this player.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| String | The name of this player |

#### getType

Return the type of this player (human/computer/remote).

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| PlayerType | The type of this player. |

#### equals

This methods overrides Object.equals(). It is essential that it obeys the general contract for java.lang.Object.equals() as described in the javadoc.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | Object.equals() |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| obj | Object | The Object to use in the comparison |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| boolean | True if obj is a PlayerData with the same name and type as this PlayerData. Otherwise false. |

#### hashCode

This methods overrides Object.hashCode(). It is essential that it obeys the general contract for java.lang.Object.hashCode() as described in the javadoc.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | Object.hashCode() |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| int | A hash value for this PlayerData. |

#### toString

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | Object.toString() |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| String | A representation of this PlayerData. |

## Enumeration PlayerData.PlayerType

PlayerData contains information on the type of a player (Human, Computer or Remote) and the name of the player. The type of the player will be stored as a public enum named PlayerType with fields Human, Computer and Remote.

### Public Methods

#### toString

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | Object.toString() |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| String | A representation of this PlayerData. |

### Package Access Methods

None

### Public Fields

|  |  |
| --- | --- |
| Name | Description |
| COMPUTER | Indicates the computer will play this side. String representation will be “Computer”. |
| HUMAN | Indicates a human will play this side. String representation will be “Human”, |
| REMOTE | Indicates a remote player will play this side. String representation will be “Remote” |

### Package Access Fields

None

## GameData

GameData contains PlayerData information for the Black and White players and a FEN tag describing a board layout. If the FEN tag is empty, then the game should be started with pieces placed in the default locations. There are two constructor for GameData, one that takes a String representing a FEN tag and another that take a java.io.Reader which represents a PDN file from which a FEN tag will be extracted.

|  |  |
| --- | --- |
| Attribute | Value |
| Visibility | Public |
| Modifiers | Final |
| Extends | - |
| Implements | - |

### Public Methods

#### GameData(PlayerData, PlayerData, Reader)

Constructs a new GameData with players specified by the two PlayerData parameters and a java.io.Reader that represents the PDN data to import. If the Reader is null it is ignored, otherwise the PDN data is read and the FEN tag is stored.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| black | PlayerData | The black player data |
| white | PlayerData | The white player data |
| pdn | java.io.Reader | A Reader that is linked to the PDN data to import. |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### GameData(PlayerData, PlayerData, String)

Constructs a new GameData with players specified by the two PlayerData parameters and a String representing a FEN tag file. If the String is null it is ignored.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| black | PlayerData | The black player data |
| white | PlayerData | The white player data |
| fen | String | A string representing a FEN tag |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### getBlackPlayer

Returns the PlayerData describing the black player.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| PlayerData | PlayerData describing the black player. |

#### getFEN

Returns the FEN tag that this GameData was constructed with, or an empty String if none was specified.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| String | The FEN tag that the GameData was constructed with, or an empty String. |

#### getHost

Returns a String representing the remote host to use for a network game. An empty String is returned if there is no remote host.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| String | PlayerData describing the white player. |

#### getWhitePlayer

Returns the PlayerData describing the white player.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| PlayerData | PlayerData describing the white player. |

#### isNetworkGame

Returns true if one of the players is remote, otherwise false.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| boolean | true if one of the players is remote, otherwise false |

## Enumeration GameState

GameState is an enumeration describing the possible states of a game which are IN\_PROGRESS, DRAW , BLACK\_WIN and WHITE\_WIN.

|  |  |
| --- | --- |
| Attribute | Value |
| Visibility | Public |
| Modifiers | - |
| Extends | - |
| Implements | - |

### Public Methods

None

### Package Access Methods

None

### Public Fields

|  |  |
| --- | --- |
| Name | Description |
| IN\_PROGRESS | The game is still in progress |
| BLACK\_WIN | The Black player won the game |
| WHITE\_WIN | The White player won the game |
| DRAW | The game ended in a draw |

## Enumeration Square

Square is an enumeration representing the possible states of a square

|  |  |
| --- | --- |
| Attribute | Value |
| Visibility | Public |
| Modifiers | - |
| Extends | - |
| Implements | - |

### Public Methods

#### isBlack()

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| boolean | Returns true for BLACK\_PIECE and BLACK \_KING, otherwise false. |

#### isWhite()

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| boolean | Returns true for WHITE\_PIECE and WHITE\_KING, otherwise false. |

#### isKing()

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| boolean | Returns true for BLACK\_KING and WHITE\_KING, otherwise false. |

### Package Access Methods

None

### Public Fields

|  |  |
| --- | --- |
| Name | Description |
| EMPTY | There is no piece in the square. |
| BLACK\_PIECE | The square contains a black piece. |
| BLACK\_KING | The square contains a black king. |
| WHITE\_PIECE | The square contains a white piece. |
| WHITE\_KING | The square contains a white king. |

### Package Access Fields

None

## SquareData

This class describes the contents of a square on board.

|  |  |
| --- | --- |
| Attribute | Value |
| Visibility | Public |
| Modifiers | - |
| Extends | - |
| Implements | - |

### Public Methods

#### SquareData()

Constructs a square with an ID (1 – 32) and its contents.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| squareID | int | The id (1-32) of the square this object represents |
| contents | Square | The contents of the Square |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### getContents()

Return the contents of the Square data (WHITE\_PIECE, BLACK\_PIECE, WHITE\_KING, BLACK\_KING or EMPTY).

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| Square | The contents of the square. |

#### getSquareID()

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| int | The id of the square the square data is for (1-32) |

#### equals

This methods overrides Object.equals(). It is essential that it obeys the general contract for java.lang.Object.equals() as described in the javadoc.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | Object.equals() |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| obj | Object | The Object to use in the comparison |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| boolean | True if obj is a SquareData with the same id and contents as this SquareData. Otherwise false. |

#### hashCode

This methods overrides Object.hashCode(). It is essential that it obeys the general contract for java.lang.Object.hashCode() as described in the javadoc.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | Object.hashCode() |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| int | A hash value for this SquareData. |

#### toString

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | Object.toString() |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| String | A representation of this SquareData. |

## Interface IDraughtsController

IDraughtsController is an interface used to export all functionality used by a view or model. It is used to remove the direct coupling between the view to controller and model to controller.

|  |  |
| --- | --- |
| Attribute | Value |
| Visibility | Public |
| Modifiers | - |
| Extends | - |
| Implements | - |

### Public Methods

#### addView

Registers a view with the DraughtsController. Only registered views will receive updates from the controller.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| view | IDraughtsView | View to be added to the controller |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### setModel

Sets the model that the DraughtsController is observing.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| model | IDraughtsModel | The model to connect to this controller |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### squareClicked

Indicates to controller that click has occurred on a square. For use by IDraughtsView implementations. This method will filter out clicks not made by a valid player. For example if the computer is controlling the black player and black is to move then no clicks will be considered valid.

The controller shall instruct the model to propagate the click if it was received from a local view.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| view | IDraughtsView | The view the click came from. |
| squareID | int | The id of the square (1 to 32) |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| boolean | True for a valid square – i.e. one that can be selected. |

#### newGame

When called fires a NEW\_GAME event to all registered views.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### resetBoard

Resets the board to its default layout, clearing all stored data. This will force a new game dialog to be displayed.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| gameData | GameData | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### connectionFailed

Called when a connection to the remote machine has occurred. This condition will start a new game.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### error

Called when as unrecoverable error has occurred. This condition will start a new game.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| errorMsg | String | Error string |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### export

Export the game to a PDN file.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| writer | Writer | The writer to use when exporting PDN data |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### undo

This is called when a move has been undone. This is passed down to the model.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

### Public Fields

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Modifiers | Contents | Description |
| SET\_SQUARE | String | static final | "SetSquare" | The contents of a square has changed |
| SET\_SELECTED | String | static final | "SetSelected" | Set the square to be a white piece |
| SET\_HINTS | String | static final | "SetHints" | Set the squares to highlight |
| STATUS | String | static final | "SetStatus" | Set the status message |
| PLAYER | String | static final | "SetPlayer" | Current player |
| NEW\_GAME | String | static final | "NewGame" | Set request a new game |
| ERROR | String | Static final | “Error” | An error has been detected |
| REMOTE | String | static final | "Remote" | Connect to a remote host |
| SEND\_CLICK | String | static final | "SendClick" | Sent to indicate that a 'click' should be performed. (Used for RMI view) |
| GAME\_DATA\_UPDATED | String | static final | "GameDataUpdated" | Sent to indicate that there has been an update to the game data e.g. player names |
| ADD\_HISTORY | String | static final | "AddHistory" | Completed move that can be displayed in history |
| REMOVE\_HISTORY | String | static final | "RemoveHistory" | Move that has been undone so should be removed from history |
| GAME\_END | String | static final | “GameEnd” | The game has completed. |
| DISCONNECT | String | static final | “Disconnect” | A player has disconnected from a remote game. |

## DraughtsController

DraughtsController is a concrete implementation of IDraughtsController. DraughtsController is a singleton as the design is based upon there only being a single controller. DraughtsController takes the part of the controller in the Model-View-Controller system. It acts as an Observer with a single instance of IDraughtsModel acting as the Subject.

Multiple IDraughtsView objects can register with the DraughtsController to be notified when changes to the IDraughtsModel occur. The IDs of the events that are passed to the IDraughtsView objects are defined as public fields in DraughtsController.

Each IDraughtsView can interact with the DraughtsController via public methods. The DraughtsController determines if these interactions should be passed on to the DraughtsModel. The DraughtsController maintains an association between each IDraughtsView and zero, one or two players to determine if interactions should be passed to the IDraughtsModel. The DraughtsController does not provide public access the IDraughtsModel.

|  |  |
| --- | --- |
| Attribute | Value |
| Visibility | Public |
| Modifiers | - |
| Extends | - |
| Implements | IDraughtsController, java.beans.PropertyChangeListener |

### Public Methods

#### getInstance

Returns the single instance of a DraughtsController.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | static |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| IDraughtsController | The single instance of a DraughtsController |

#### addView

Registers a view with the DraughtsController. Only registered views will receive updates from the controller.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | IDraughtsView.addView |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| view | IDraughtsView | View to be added to the controller |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### setModel

Sets the model that the DraughtsController is observing.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | IDraughtsView.setModel |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| model | IDraughtsModel | The model to connect to this controller |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### squareClicked

Indicates to controller that click has occurred on a square. For use by IDraughtsView implementations. This method will filter out clicks not made by a valid player. For example if the computer is controlling the black player and black is to move then no clicks will be considered valid.

The controller shall instruct the model to propagate the click if it was received from a local view.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | IDraughtsView.squareClicked |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| view | IDraughtsView | The view the click came from. |
| squareID | int | The id of the square (1 to 32) |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### propertyChange

This method is used to implement the PropertyChangeListener interface. Any model changes will be sent to this controller through the use of this method. This method will then fire a property change to all registered views.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | PropertyChangeListener.propertyChange |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### newGame

When called fires a NEW\_GAME event to all registered views.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | IDraughtsView.newGame |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### resetBoard

Resets the board to its default layout, clearing all stored data. This will force a new game dialog to be displayed.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | IDraughtsView.resetBoard |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| gameData | GameData | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### connectionFailed

Called when a connection to the remote machine has occurred. This condition will start a new game.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | IDraughtsView.connectionFailed |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### error

Called when as unrecoverable error has occurred. This condition will start a new game.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | IDraughtsView.error |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| errorMsg | String | Error string |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### export

Export the game to a PDN file.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | IDraughtsView.export |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| writer | Writer | The writer to use when exporting PDN data |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### undo

This is called when a move has been undone. This is passed down to the model.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | IDraughtsView.undo |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

### Public Fields

None.

## Interface IDraughtsModel

IDraughtsModel is the public interface that must be implemented by all classes that can be registered with the DraughtsController.

|  |  |
| --- | --- |
| Attribute | Value |
| Visibility | Public |
| Modifiers | - |
| Extends | - |
| Implements | - |

### Nested classes

#### enum PlayerColor

Used to indicate the Black and White players, contains two fields BLACK and WHITE.

### Methods

#### addPropertyChangeListener

Registers a property change listener with the DraughtsModel. All registered listeners will be notified about property change events.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| listener | PropertyChangeListener | The property change listener to register |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### errorMsg

Used to inform the model that an error has occurred. This causes property change events to be fired, one to display an error message and another to trigger a new game.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | IDraughtsModel.errorMsg() |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| errorMsg | String | The error message to display |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### export

Writes a PDN file containing the current board layout and the game history to the file specified by file.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | IDraughtsModel.export() |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| file | java.io.File | The file to write the PDN data to |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### getCurrentPlayer

Returns either IDraughtsModel.PlayerColor.BLACK or IDraughtsModel.PlayerColor.WHITE indicating which player is currently taking a move.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | IDraughtsModel.getCurrentPlayer() |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| PlayerColor | The value returned indicates whose turn it. |

#### newGame

Sets up the DraughtsModel according to the supplied GameData object. If the GameData object specifies a PDN file then this is used to layout the board, otherwise the default layout is used.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| gameData | GameData | Initial data to use when setting up the game |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### setClick

Used to indicate to the model that a square has been ‘clicked’ by the user.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| squareID | Integer | The square that has been clicked |
| propagate | boolean | Whether or not a SEND\_CLICK event should be fired. |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### undo

Instructs the model to undo the last human move.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | IDraughtsModel.undo() |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

### Public Fields

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Attributes | Value | Description |
| BLACK\_COMPUTER\_PLAYER | String | final | "BLACK\_COMPUTER\_PLAYER" | Name of system property specifying the class to load for the black computer player. |
| WHITE\_COMPUTER\_PLAYER | String | final | "WHITE\_COMPUTER\_PLAYER" | Name of system property specifying the class to load for the white computer player. |

### Package Access Fields

None

## Interface IDraughtsView

IDraughtsView is a public interface that must be implemented by all classes that will be able to register with DraughtsController and be notified of updates to the registered IDraughtsModel. IDraughtsView.getType() allows the DraughtsController to query a view to see if it is local or remote.

|  |  |
| --- | --- |
| Attribute | Value |
| Visibility | Public |
| Modifiers | - |
| Extends | - |
| Implements | - |

### Public Methods

#### getType

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| IDraughtsView.Type | The type of this view Local or Remote. |

#### modelPropertyChange

Called by the controller when it needs to pass along a property change from a model.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| event | PropertyChangeEvent | The property change event, the name of which is defined within IDraughtsController |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### registerController

Register this view with a controller so that it will be sent view events.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| controller | IDraughtsController | The controller that this view is registered with |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

## Enumeration IDraughtsView.Type

IDraughtsView.Type is an enumeration which describes the possible types of View, e.g. local or remote.

### Public Methods

None

### Package Access Methods

None

### Public Fields

|  |  |
| --- | --- |
| Name | Description |
| LOCAL | The view is local. |
| REMOTE | The view represents a remote interface. |

### Package Access Fields

None

## DraughtsChangeEvent

DraughtsChangeEvent is an extension of PropertyChangeEvent that has a more user friendly toString method.

|  |  |
| --- | --- |
| Attribute | Value |
| Visibility | Public |
| Modifiers | Final |
| Extends | java.beans.PropertyChangeEvent |
| Implements | - |

### Public Methods

#### DraughtsChangeEvent

Calls the super constructor passing all parameters.

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | - |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| source | Object | The source of the event |
| propertyName | String | The name of the property that has changed |
| oldValue | Object | The old value of the property |
| newValue | Object | The new value of the property |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| - | - |

#### toString

|  |  |
| --- | --- |
| Attribute | Value |
| Modifiers | - |
| Overrides | Object.toString() |
| Throws | - |

**Inputs**

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| - | - | - |

**Outputs**

|  |  |
| --- | --- |
| Type | Description |
| String | A String that describes this event. i.e.  "Name: " + getPropertyName() + " Old: " + getOldValue() + " New: " + getNewValue() |

### Package Access Methods

None

### Public Fields

None

### Package Access Fields

None

# Resource Requirements

None.

# Test Plan

There will be no formal package level testing. Each class will be unit tested and then the entire system will undergo system level testing.

# Scenarios

To be added post refactoring of the model so that it can support an intelligent computer engine.

# Supporting Documentation

None.

**- End of Document -**