

# My Project

Generated by Doxygen 1.9.3

<b>1 Class Documentation</b>	<b>1</b>
1.1 AnswerCallback . . . . .	1
1.1.1 Detailed Description . . . . .	1
1.2 AppSocket . . . . .	2
1.2.1 Detailed Description . . . . .	2
1.2.2 Constructor & Destructor Documentation . . . . .	2
1.2.3 Member Function Documentation . . . . .	3
1.2.4 Member Data Documentation . . . . .	6
1.3 ArchivedGameInfo . . . . .	6
1.3.1 Detailed Description . . . . .	7
1.3.2 Constructor & Destructor Documentation . . . . .	7
1.3.3 Member Function Documentation . . . . .	7
1.4 Arg . . . . .	8
1.4.1 Detailed Description . . . . .	8
1.4.2 Constructor & Destructor Documentation . . . . .	8
1.4.3 Member Function Documentation . . . . .	9
1.4.4 Member Data Documentation . . . . .	11
1.5 ArgsUtil . . . . .	11
1.5.1 Detailed Description . . . . .	11
1.5.2 Member Function Documentation . . . . .	11
1.6 ArgType Enum Reference . . . . .	13
1.6.1 Detailed Description . . . . .	13
1.6.2 Constructor & Destructor Documentation . . . . .	13
1.6.3 Member Data Documentation . . . . .	14
1.7 ArrUtils . . . . .	15
1.7.1 Detailed Description . . . . .	15
1.7.2 Member Function Documentation . . . . .	15
1.8 BasicMove . . . . .	16
1.8.1 Detailed Description . . . . .	16
1.8.2 Constructor & Destructor Documentation . . . . .	17
1.8.3 Member Function Documentation . . . . .	17
1.9 BitData . . . . .	20
1.9.1 Detailed Description . . . . .	20
1.9.2 Member Data Documentation . . . . .	20
1.10 Board . . . . .	21
1.10.1 Detailed Description . . . . .	21
1.10.2 Constructor & Destructor Documentation . . . . .	21
1.10.3 Member Function Documentation . . . . .	22
1.10.4 Member Data Documentation . . . . .	25
1.11 MyJFrame.BooleanClosing . . . . .	25
1.11.1 Detailed Description . . . . .	25
1.11.2 Member Function Documentation . . . . .	25

1.12 Callback< T > . . . . .	26
1.12.1 Detailed Description . . . . .	26
1.12.2 Member Function Documentation . . . . .	26
1.13 CastlingRights . . . . .	27
1.13.1 Detailed Description . . . . .	27
1.13.2 Constructor & Destructor Documentation . . . . .	27
1.13.3 Member Function Documentation . . . . .	28
1.13.4 Member Data Documentation . . . . .	31
1.14 MyJFrame.Closing< T > . . . . .	31
1.14.1 Detailed Description . . . . .	32
1.14.2 Member Function Documentation . . . . .	32
1.14.3 Member Data Documentation . . . . .	33
1.15 Col . . . . .	33
1.15.1 Detailed Description . . . . .	34
1.15.2 Constructor & Destructor Documentation . . . . .	34
1.15.3 Member Function Documentation . . . . .	36
1.15.4 Member Data Documentation . . . . .	42
1.16 Condition . . . . .	43
1.16.1 Detailed Description . . . . .	43
1.16.2 Class Documentation . . . . .	43
1.16.3 Constructor & Destructor Documentation . . . . .	44
1.16.4 Member Function Documentation . . . . .	44
1.17 Config< V > . . . . .	49
1.17.1 Detailed Description . . . . .	49
1.17.2 Constructor & Destructor Documentation . . . . .	49
1.17.3 Member Function Documentation . . . . .	51
1.17.4 Member Data Documentation . . . . .	52
1.18 ConfirmDialogs . . . . .	52
1.18.1 Detailed Description . . . . .	53
1.18.2 Member Function Documentation . . . . .	53
1.19 CreatedGame . . . . .	54
1.19.1 Detailed Description . . . . .	54
1.19.2 Constructor & Destructor Documentation . . . . .	54
1.19.3 Member Function Documentation . . . . .	55
1.20 CustomStatement . . . . .	55
1.20.1 Detailed Description . . . . .	55
1.20.2 Constructor & Destructor Documentation . . . . .	55
1.20.3 Member Function Documentation . . . . .	56
1.21 DBRequest . . . . .	56
1.21.1 Detailed Description . . . . .	56
1.21.2 Class Documentation . . . . .	56
1.21.3 Constructor & Destructor Documentation . . . . .	57

1.21.4 Member Function Documentation . . . . .	58
1.21.5 Member Data Documentation . . . . .	59
1.22 DBResponse . . . . .	59
1.22.1 Detailed Description . . . . .	59
1.22.2 Class Documentation . . . . .	59
1.22.3 Constructor & Destructor Documentation . . . . .	60
1.22.4 Member Function Documentation . . . . .	60
1.22.5 Member Data Documentation . . . . .	62
1.23 Delete . . . . .	62
1.23.1 Detailed Description . . . . .	63
1.23.2 Constructor & Destructor Documentation . . . . .	63
1.23.3 Member Function Documentation . . . . .	63
1.24 Direction Enum Reference . . . . .	63
1.24.1 Detailed Description . . . . .	64
1.24.2 Member Function Documentation . . . . .	65
1.24.3 Member Data Documentation . . . . .	66
1.25 EnvManager . . . . .	69
1.25.1 Detailed Description . . . . .	69
1.25.2 Member Function Documentation . . . . .	69
1.26 ErrorHandler< E extends MyError > . . . . .	70
1.26.1 Detailed Description . . . . .	70
1.26.2 Member Function Documentation . . . . .	70
1.27 EstablishedGameInfo . . . . .	71
1.27.1 Detailed Description . . . . .	71
1.27.2 Constructor & Destructor Documentation . . . . .	71
1.27.3 Member Function Documentation . . . . .	72
1.27.4 Member Data Documentation . . . . .	73
1.28 Evaluation . . . . .	73
1.28.1 Detailed Description . . . . .	74
1.28.2 Constructor & Destructor Documentation . . . . .	74
1.28.3 Member Function Documentation . . . . .	75
1.28.4 Member Data Documentation . . . . .	79
1.29 EvaluationParameters Enum Reference . . . . .	79
1.29.1 Detailed Description . . . . .	80
1.29.2 Member Data Documentation . . . . .	80
1.30 GameInfo . . . . .	81
1.30.1 Detailed Description . . . . .	81
1.30.2 Constructor & Destructor Documentation . . . . .	81
1.30.3 Member Function Documentation . . . . .	82
1.30.4 Member Data Documentation . . . . .	84
1.31 GameStatus . . . . .	84
1.31.1 Detailed Description . . . . .	85

1.31.2 Member Function Documentation . . . . .	85
1.32 GameState.GameStatusType Enum Reference . . . . .	91
1.32.1 Detailed Description . . . . .	91
1.32.2 Constructor & Destructor Documentation . . . . .	91
1.32.3 Member Function Documentation . . . . .	92
1.32.4 Member Data Documentation . . . . .	92
1.33 HandledThread . . . . .	93
1.33.1 Detailed Description . . . . .	93
1.33.2 Constructor & Destructor Documentation . . . . .	93
1.33.3 Member Function Documentation . . . . .	94
1.34 MyJFrame.MyAdapter.HeldDown . . . . .	95
1.34.1 Detailed Description . . . . .	95
1.34.2 Member Function Documentation . . . . .	95
1.35 IDsGenerator . . . . .	95
1.35.1 Detailed Description . . . . .	95
1.35.2 Member Function Documentation . . . . .	95
1.36 StrUtils.IterationThingy< T > . . . . .	96
1.36.1 Detailed Description . . . . .	96
1.36.2 Member Function Documentation . . . . .	96
1.37 Location Enum Reference . . . . .	97
1.37.1 Detailed Description . . . . .	97
1.37.2 Constructor & Destructor Documentation . . . . .	98
1.37.3 Member Function Documentation . . . . .	98
1.37.4 Member Data Documentation . . . . .	104
1.38 LoginInfo . . . . .	104
1.38.1 Detailed Description . . . . .	105
1.38.2 Constructor & Destructor Documentation . . . . .	105
1.38.3 Member Function Documentation . . . . .	105
1.39 LoginType Enum Reference . . . . .	108
1.39.1 Detailed Description . . . . .	108
1.39.2 Member Function Documentation . . . . .	108
1.39.3 Member Data Documentation . . . . .	108
1.40 Math Enum Reference . . . . .	109
1.40.1 Detailed Description . . . . .	110
1.40.2 Member Function Documentation . . . . .	110
1.40.3 Member Data Documentation . . . . .	113
1.41 Message . . . . .	114
1.41.1 Detailed Description . . . . .	115
1.41.2 Constructor & Destructor Documentation . . . . .	115
1.41.3 Member Function Documentation . . . . .	116
1.41.4 Member Data Documentation . . . . .	133
1.42 MessageCallback . . . . .	134

1.42.1 Detailed Description . . . . .	134
1.42.2 Member Function Documentation . . . . .	134
1.43 MessagesHandler . . . . .	134
1.43.1 Detailed Description . . . . .	135
1.43.2 Constructor & Destructor Documentation . . . . .	135
1.43.3 Member Function Documentation . . . . .	136
1.43.4 Member Data Documentation . . . . .	142
1.44 MinimaxMove . . . . .	143
1.44.1 Detailed Description . . . . .	143
1.44.2 Constructor & Destructor Documentation . . . . .	143
1.44.3 Member Function Documentation . . . . .	144
1.45 Move . . . . .	146
1.45.1 Detailed Description . . . . .	147
1.45.2 Class Documentation . . . . .	148
1.45.3 Constructor & Destructor Documentation . . . . .	148
1.45.4 Member Function Documentation . . . . .	149
1.46 MoveAnnotation . . . . .	157
1.46.1 Detailed Description . . . . .	157
1.46.2 Member Function Documentation . . . . .	157
1.47 Move.MoveFlag Enum Reference . . . . .	158
1.47.1 Detailed Description . . . . .	159
1.47.2 Constructor & Destructor Documentation . . . . .	159
1.47.3 Member Function Documentation . . . . .	159
1.47.4 Member Data Documentation . . . . .	160
1.48 MyError . . . . .	161
1.48.1 Detailed Description . . . . .	161
1.48.2 Constructor & Destructor Documentation . . . . .	161
1.48.3 Member Function Documentation . . . . .	162
1.49 MyJButton . . . . .	163
1.49.1 Detailed Description . . . . .	163
1.49.2 Constructor & Destructor Documentation . . . . .	163
1.49.3 Member Function Documentation . . . . .	165
1.50 MyJFrame . . . . .	166
1.50.1 Detailed Description . . . . .	167
1.50.2 Constructor & Destructor Documentation . . . . .	167
1.50.3 Member Function Documentation . . . . .	167
1.50.4 Member Data Documentation . . . . .	168
1.51 MyThread . . . . .	168
1.51.1 Detailed Description . . . . .	169
1.51.2 Constructor & Destructor Documentation . . . . .	169
1.51.3 Member Function Documentation . . . . .	169
1.52 ObjCallback< T > . . . . .	170

1.52.1 Detailed Description . . . . .	170
1.53 Piece Enum Reference . . . . .	170
1.53.1 Detailed Description . . . . .	171
1.53.2 Member Function Documentation . . . . .	171
1.53.3 Member Data Documentation . . . . .	173
1.54 PieceType Enum Reference . . . . .	175
1.54.1 Detailed Description . . . . .	175
1.54.2 Member Function Documentation . . . . .	176
1.54.3 Member Data Documentation . . . . .	179
1.55 PlayerColor Enum Reference . . . . .	181
1.55.1 Detailed Description . . . . .	181
1.55.2 Constructor & Destructor Documentation . . . . .	181
1.55.3 Member Function Documentation . . . . .	182
1.55.4 Member Data Documentation . . . . .	183
1.56 PreMadeRequest . . . . .	184
1.56.1 Detailed Description . . . . .	184
1.56.2 Member Function Documentation . . . . .	184
1.56.3 Member Data Documentation . . . . .	185
1.57 Question . . . . .	186
1.57.1 Detailed Description . . . . .	186
1.57.2 Class Documentation . . . . .	186
1.57.3 Constructor & Destructor Documentation . . . . .	187
1.57.4 Member Function Documentation . . . . .	187
1.57.5 Member Data Documentation . . . . .	189
1.58 RegEx . . . . .	190
1.58.1 Detailed Description . . . . .	191
1.58.2 Constructor & Destructor Documentation . . . . .	191
1.58.3 Member Function Documentation . . . . .	191
1.58.4 Member Data Documentation . . . . .	193
1.59 RequestBuilder . . . . .	195
1.59.1 Detailed Description . . . . .	196
1.59.2 Constructor & Destructor Documentation . . . . .	196
1.59.3 Member Function Documentation . . . . .	197
1.59.4 Member Data Documentation . . . . .	201
1.60 Selection . . . . .	202
1.60.1 Detailed Description . . . . .	202
1.60.2 Class Documentation . . . . .	202
1.60.3 Constructor & Destructor Documentation . . . . .	203
1.60.4 Member Function Documentation . . . . .	204
1.61 CastlingRights.Side Enum Reference . . . . .	205
1.61.1 Detailed Description . . . . .	206
1.61.2 Constructor & Destructor Documentation . . . . .	206

1.61.3 Member Function Documentation . . . . .	206
1.61.4 Member Data Documentation . . . . .	207
1.62 GameState.SpecificStatus Enum Reference . . . . .	208
1.62.1 Detailed Description . . . . .	208
1.62.2 Constructor & Destructor Documentation . . . . .	208
1.62.3 Member Function Documentation . . . . .	209
1.62.4 Member Data Documentation . . . . .	209
1.63 SQLStatement . . . . .	210
1.63.1 Detailed Description . . . . .	211
1.63.2 Constructor & Destructor Documentation . . . . .	211
1.63.3 Member Function Documentation . . . . .	211
1.63.4 Member Data Documentation . . . . .	212
1.64 Square . . . . .	212
1.64.1 Detailed Description . . . . .	213
1.64.2 Constructor & Destructor Documentation . . . . .	213
1.64.3 Member Function Documentation . . . . .	213
1.64.4 Member Data Documentation . . . . .	215
1.65 StatusResponse . . . . .	215
1.65.1 Detailed Description . . . . .	215
1.65.2 Constructor & Destructor Documentation . . . . .	215
1.65.3 Member Function Documentation . . . . .	216
1.66 MyJFrame.StringClosing . . . . .	217
1.66.1 Detailed Description . . . . .	217
1.66.2 Member Function Documentation . . . . .	217
1.67 StrUtils . . . . .	218
1.67.1 Detailed Description . . . . .	218
1.67.2 Member Function Documentation . . . . .	218
1.68 SwitchCase . . . . .	228
1.68.1 Detailed Description . . . . .	228
1.68.2 Constructor & Destructor Documentation . . . . .	228
1.68.3 Member Function Documentation . . . . .	229
1.69 Table Enum Reference . . . . .	230
1.69.1 Detailed Description . . . . .	230
1.69.2 Constructor & Destructor Documentation . . . . .	230
1.69.3 Member Function Documentation . . . . .	231
1.69.4 Member Data Documentation . . . . .	231
1.70 TableDBResponse . . . . .	232
1.70.1 Detailed Description . . . . .	232
1.70.2 Constructor & Destructor Documentation . . . . .	232
1.70.3 Member Function Documentation . . . . .	233
1.70.4 Member Data Documentation . . . . .	236
1.71 ThreadsManager . . . . .	236



1.71.1 Detailed Description . . . . .	236
1.71.2 Member Function Documentation . . . . .	236
1.72 ThrowingCallback< T > . . . . .	237
1.72.1 Detailed Description . . . . .	237
1.72.2 Member Function Documentation . . . . .	237
1.73 UnfinishedGame . . . . .	238
1.73.1 Detailed Description . . . . .	238
1.73.2 Constructor & Destructor Documentation . . . . .	238
1.73.3 Member Function Documentation . . . . .	239
1.73.4 Member Data Documentation . . . . .	239
1.74 Update . . . . .	239
1.74.1 Detailed Description . . . . .	240
1.74.2 Constructor & Destructor Documentation . . . . .	240
1.74.3 Member Function Documentation . . . . .	240
1.75 ValueBtn< T > . . . . .	240
1.75.1 Detailed Description . . . . .	240
1.75.2 Constructor & Destructor Documentation . . . . .	241
1.76 PreMadeRequest.VariationCreator . . . . .	241
1.76.1 Detailed Description . . . . .	241
1.76.2 Member Function Documentation . . . . .	241
1.77 VoidCallback . . . . .	242
1.77.1 Detailed Description . . . . .	242
1.77.2 Member Function Documentation . . . . .	242
<b>Index</b>	<b>243</b>

## 1 Class Documentation

### 1.1 AnswerCallback

#### Additional Inherited Members

#### 1.1.1 Detailed Description

Answer callback - an answer callback.

#### Author

Bezalel Avrahami ( bezalel3250@gmail.com)

## 1.2 AppSocket

### Public Member Functions

- **AppSocket** (String ip, int port) throws IOException
- **AppSocket** (Socket socket) throws IOException
- void **close** ()
- void **close** ( **MyError** err)
- void **interruptListener** ( **MyError** err)
- void **requestMessage** ( **Message** requestMsg, **MessageCallback** onRes)
- **MessagesHandler** **getMessagesHandler** ()
- void **setMessagesHandler** ( **MessagesHandler** messagesHandler)
- void **respond** ( **Message** msg, **Message** respondingTo)
- synchronized void **writeMessage** ( **Message** msg)
- boolean **isConnected** ()
- String **getLocalAddress** ()
- String **getRemoteAddress** ()
- boolean **isClosed** ()
- **Message** **requestMessage** ( **Message** requestMsg)
- void **stopReading** ()

### Protected Member Functions

- void **handledRun** ()

### Protected Attributes

- final Socket **msgSocket**

### Additional Inherited Members

#### 1.2.1 Detailed Description

App socket - represents a communications socket able to send and receive messages from the client to the server and vice versa.

#### Author

Bezalel Avrahami ( [bezalel13250@gmail.com](mailto:bezalel13250@gmail.com))

#### 1.2.2 Constructor & Destructor Documentation

**1.2.2.1 AppSocket()** [1/2]    **AppSocket** (   
          String ip,   
          int port ) throws IOException

Instantiates a new App socket.

## Parameters

<i>ip</i>	the ip
<i>port</i>	the port

## Exceptions

<i>IOException</i>	the io exception
--------------------	------------------

**1.2.2.2 AppSocket()** [2/2]    **AppSocket** (  
          Socket *socket* ) throws **IOException**

Instantiates a new App socket.

## Parameters

<i>socket</i>	the socket
---------------	------------

## Exceptions

<i>IOException</i>	the io exception
--------------------	------------------

## 1.2.3 Member Function Documentation

**1.2.3.1 close()** [1/2]    void close ( )

Close.

**1.2.3.2 close()** [2/2]    void close (  
          **MyError** *err* )

Close.

## Parameters

<i>err</i>	the error
------------	-----------

**1.2.3.3 getLocalAddress()**    String getLocalAddress ( )

Gets local address.

**Returns**

the local address

**1.2.3.4 getMessagesHandler()** `MessagesHandler getMessagesHandler ( )`

Gets messages handler.

**Returns**

the messages handler

**1.2.3.5 getRemoteAddress()** `String getRemoteAddress ( )`

Gets remote address.

**Returns**

the remote address

**1.2.3.6 handledRun()** `void handledRun ( ) [protected]`

Handled run.

Reimplemented from **MyThread** (p. 169).

**1.2.3.7 interruptListener()** `void interruptListener (   
    MyError err )`

Interrupt listener.

**Parameters**

<i>err</i>	the err to interrupt with
------------	---------------------------

**1.2.3.8 isClosed()** `boolean isClosed ( )`

Is closed boolean.

## Returns

the boolean

**1.2.3.9 isConnected()** `boolean isConnected ( )`

Is connected boolean.

## Returns

the boolean

**1.2.3.10 requestMessage()** [1/2] `Message requestMessage ( Message requestMsg )`

sending request and blocking til res

## Parameters

<i>requestMsg</i>	= "can i have x message?"
-------------------	---------------------------

## Returns

response

**1.2.3.11 requestMessage()** [2/2] `void requestMessage ( Message requestMsg, MessageCallback onRes )`

Request message.

## Parameters

<i>requestMsg</i>	the request msg
<i>onRes</i>	the on res

**1.2.3.12 respond()** `void respond ( Message msg, Message respondingTo )`

Respond.

**Parameters**

<i>msg</i>	the msg
<i>respondingTo</i>	the responding to

**1.2.3.13 setMessagesHandler()** `void setMessagesHandler ( MessagesHandler messagesHandler )`

Sets messages handler.

**Parameters**

<i>messagesHandler</i>	the messages handler
------------------------	----------------------

**1.2.3.14 stopReading()** `void stopReading ( )`

Stop reading.

**1.2.3.15 writeMessage()** `synchronized void writeMessage ( Message msg )`

Write message.

**Parameters**

<i>msg</i>	the msg
------------	---------

## 1.2.4 Member Data Documentation

**1.2.4.1 msgSocket** `final Socket msgSocket [protected]`

The Msg socket.

## 1.3 ArchivedGameInfo

**Public Member Functions**

- **ArchivedGameInfo** (String **gameId**, String **creatorUsername**, String **opponentUsername**, GameSettings **gameSettings**, String winner, Stack< **Move** > moveStack)
- String **getWinner** ()
- String **toString** ()

## Additional Inherited Members

### 1.3.1 Detailed Description

Archived game info.

#### Author

Bezalel Avrahami ( bezalel13250@gmail.com)

### 1.3.2 Constructor & Destructor Documentation

#### 1.3.2.1 ArchivedGameInfo() ArchivedGameInfo (

```
String gameId,
String creatorUsername,
String opponentUsername,
GameSettings gameSettings,
String winner,
Stack< Move > moveStack )
```

Instantiates a new Archived game info.

#### Parameters

<i>gameId</i>	the game id
<i>creatorUsername</i>	the creator username
<i>opponentUsername</i>	the opponent username
<i>gameSettings</i>	the game settings
<i>winner</i>	the winner
<i>moveStack</i>	the move stack

### 1.3.3 Member Function Documentation

#### 1.3.3.1 getWinner() String getWinner ( )

Gets winner.

#### Returns

the winner

### 1.3.3.2 toString() `String toString ()`

To string string.

#### Returns

the string

Reimplemented from **EstablishedGameInfo** (p. 73).

## 1.4 Arg

### Public Member Functions

- **Arg** ( **ArgType** argType)
- **Arg** ( **ArgType** argType, **Config**<?> config)
- **Arg** ( **ArgType** argType, boolean escape, **Config**<?> config)
- boolean **isUserInput** ()
- void **setUserInput** (boolean userInput)
- boolean **equals** (Object o)
- String **toString** ()
- String **createVal** (Object val)

### Public Attributes

- final String **replnStr**
- final boolean **escape**
- final **ArgType** **argType**
- final **Config**<?> **config**

### 1.4.1 Detailed Description

**Arg** (p. 8) - .

#### Author

Bezalel Avrahami ( bezalel3250@gmail.com)

### 1.4.2 Constructor & Destructor Documentation

#### 1.4.2.1 Arg() [1/3] `Arg ( ArgType argType )`

Instantiates a new **Arg** (p. 8).



## Parameters

<i>argType</i>	the arg type
----------------	--------------

**1.4.2.2 Arg()** [2/3]    **Arg** (  
    **ArgType** *argType*,  
    **Config**<?> *config* )

Instantiates a new **Arg** (p. 8).

## Parameters

<i>argType</i>	the arg type
<i>config</i>	the config

**1.4.2.3 Arg()** [3/3]    **Arg** (  
    **ArgType** *argType*,  
    boolean *escape*,  
    **Config**<?> *config* )

Instantiates a new **Arg** (p. 8).

## Parameters

<i>argType</i>	the arg type
<i>escape</i>	the escape
<i>config</i>	the config

### 1.4.3 Member Function Documentation

**1.4.3.1 createVal()** String createVal (  
    Object *val* )

Create val string.

## Parameters

<i>val</i>	the val
------------	---------

**Returns**

the string

**1.4.3.2 equals()** `boolean equals (   
Object o )`

Equals boolean.

**Parameters**

<i>o</i>	the o
----------	-------

**Returns**

the boolean

**1.4.3.3 isUserInput()** `boolean isUserInput ( )`

Is user input boolean.

**Returns**

the boolean

**1.4.3.4 setUserInput()** `void setUserInput (   
boolean userInput )`

Sets user input.

**Parameters**

<i>userInput</i>	the user input
------------------	----------------

**1.4.3.5 toString()** `String toString ( )`

To string string.

**Returns**

the string

### 1.4.4 Member Data Documentation

#### 1.4.4.1 `argType` `final ArgType argType`

The **Arg** (p. 8) type.

#### 1.4.4.2 `config` `final Config<?> config`

The **Config** (p. 49).

#### 1.4.4.3 `escape` `final boolean escape`

The **Escape**.

#### 1.4.4.4 `replnStr` `final String repInStr`

The **Rep in str**.

## 1.5 ArgsUtil

### Public Member Functions

- **OptionalArg equalsSign** (String preEqualStr)
- **OptionalArg plainTextIgnoreCase** (String str)
- record **OptionalArg** (String str)

### Static Public Member Functions

- static **ArgsUtil create** (String[] args)

### 1.5.1 Detailed Description

The type Args utils.

Author

Bezalel Avrahami ( bezalel3250@gmail.com)

### 1.5.2 Member Function Documentation

#### 1.5.2.1 `create()` `static ArgsUtil create (String[] args) [static]`

Create args util.

**Parameters**

<i>args</i>	the args
-------------	----------

**Returns**

the args util

**1.5.2.2 equalsSign()**    **OptionalArg** equalsSign (   
                              String *preEqualStr* )

Equals sign optional arg. for any arg of this format: preEqualStr=argval%

**Parameters**

<i>preEqualStr</i>	the pre equal str
--------------------	-------------------

**Returns**

the optional arg value(assuming there is one) argval% in the example above

**1.5.2.3 OptionalArg()**    record OptionalArg (   
                              String *str* )

Optional arg.

**Author**

Bezalel Avrahami ( bezalel13250@gmail.com)

**1.5.2.4 plainTextIgnoreCase()**    **OptionalArg** plainTextIgnoreCase (   
                              String *str* )

Plain text ignore case optional arg.

**Parameters**

<i>str</i>	the str
------------	---------

## Returns

the optional arg

## 1.6 ArgType Enum Reference

### Public Member Functions

- **ArgType** ()
- **ArgType** (boolean **isUserInput**)

### Public Attributes

- **Date**
- **DateRange**
- **Text**
- **ServerAddress**
- **Number**
- **Username**
- **Password**
- **Url**
- **PictureUrl**
- final boolean **isUserInput**

#### 1.6.1 Detailed Description

**Arg** (p. 8) type - argument type.

#### Author

Bezalel Avrahami ( bezalel3250@gmail.com)

#### 1.6.2 Constructor & Destructor Documentation

##### 1.6.2.1 ArgType() [1/2] ArgType ( )

Instantiates a new **Arg** (p. 8) type.

##### 1.6.2.2 ArgType() [2/2] ArgType ( boolean *isUserInput* )

Instantiates a new **Arg** (p. 8) type.

#### Parameters

<i>isUserInput</i>	the is user input
--------------------	-------------------

### 1.6.3 Member Data Documentation

#### 1.6.3.1 **Date** `Date`

Date arg type.

#### 1.6.3.2 **DateRange** `DateRange`

Date range arg type.

#### 1.6.3.3 **isUserInput** `final boolean isUserInput`

The Is user input.

#### 1.6.3.4 **Number** `Number`

Number arg type.

#### 1.6.3.5 **Password** `Password`

Password arg type.

#### 1.6.3.6 **PictureUrl** `PictureUrl`

Picture url arg type.

#### 1.6.3.7 **ServerAddress** `ServerAddress`

Server address arg type.

#### 1.6.3.8 **Text** `Text`

Text arg type.

#### 1.6.3.9 **Url** `Url`

Url arg type.

#### 1.6.3.10 **Username** `Username`

Username arg type.

## 1.7 ArrUtils

### Static Public Member Functions

- static< T > T[] **concat** (T[] array1, T... array2)
- static< T > ArrayList< T > **createList** ( **ObjCallback**< T > objCreator, int size)
- static< T > T **exists** (T[] arr, int... index)

#### 1.7.1 Detailed Description

Array Utility Class.

#### Author

Bezalel Avrahami ( bezalel3250@gmail.com)

#### 1.7.2 Member Function Documentation

**1.7.2.1 concat()** static< T > T[] concat (   
     T[] array1,   
     T... array2 ) [static]

credit

#### Parameters

< T >	the type parameter
array1	the array 1
array2	the array 2

#### Returns

t []

**1.7.2.2 createList()** static< T > ArrayList< T > createList (   
     **ObjCallback**< T > objCreator,   
     int size ) [static]

Create list array list.

#### Parameters

< T >	the type parameter
objCreator	the obj creator
size	the size

**Returns**

the array list

```
1.7.2.3 exists() static< T > T exists (
    T[] arr,
    int... index ) [static]
```

Exists t.

**Parameters**

<i>&lt;T&gt;</i>	the type parameter
<i>arr</i>	the arr
<i>index</i>	the index

**Returns**

the t

## 1.8 BasicMove

**Public Member Functions**

- **BasicMove** ( **BasicMove** other)
- **BasicMove** ( **Location** movingFrom, **Location** movingTo)
- **BasicMove** (String move)
- void **flip** ()
- **Location** **getMovingFrom** ()
- void **setMovingFrom** ( **Location** movingFrom)
- **Location** **getMovingTo** ()
- void **setMovingTo** ( **Location** movingTo)
- int **hashCode** ()
- boolean **equals** (Object o)
- String **toString** ()
- String **getBasicMoveAnnotation** ()
- **BasicMove** **cp** ()

**Static Public Member Functions**

- static **BasicMove**[] **createBatch** (Location... locs)
- static **BasicMove** **getFlipped** ( **BasicMove** basicMove)

### 1.8.1 Detailed Description

Basic move - represents a basic move. with a source and a destination.

**Author**

Bezalel Avrahami ( bezalel13250@gmail.com)



## 1.8.2 Constructor & Destructor Documentation

### 1.8.2.1 BasicMove() [1/3] BasicMove ( BasicMove other )

Copy constructor.

#### Parameters

<i>other</i>	the other
--------------	-----------

### 1.8.2.2 BasicMove() [2/3] BasicMove ( Location movingFrom, Location movingTo )

Instantiates a new Basic move.

#### Parameters

<i>movingFrom</i>	the moving from
<i>movingTo</i>	the moving to

### 1.8.2.3 BasicMove() [3/3] BasicMove ( String move )

Instantiates a new Basic move.

#### Parameters

<i>move</i>	the move
-------------	----------

## 1.8.3 Member Function Documentation

### 1.8.3.1 cp() BasicMove cp ( )

copies this move.

#### Returns

the new copy

**1.8.3.2 createBatch()** `static BasicMove[] createBatch ( Location... locs ) [static]`

Create batch basic move [ ].

**Parameters**

<i>locs</i>	the locs
-------------	----------

**Returns**

the basic move [ ]

**1.8.3.3 equals()** `boolean equals ( Object o )`

Equals boolean.

**Parameters**

<i>o</i>	the o
----------	-------

**Returns**

the boolean

Reimplemented in **Move** (p. 150).

**1.8.3.4 flip()** `void flip ( )`

Flips the source and destination.

**1.8.3.5 getBasicMoveAnnotation()** `String getBasicMoveAnnotation ( )`

Gets basic move annotation.

**Returns**

the basic move annotation

**1.8.3.6 getFlipped()** `static BasicMove getFlipped ( BasicMove basicMove ) [static]`

Gets a copy of the provided move with the source and destination flipped

## Parameters

<i>basicMove</i>	the original move
------------------	-------------------

## Returns

the flipped move

**1.8.3.7 getMovingFrom()** `Location getMovingFrom ( )`

Gets moving from. aka source

## Returns

the moving from

**1.8.3.8 getMovingTo()** `Location getMovingTo ( )`

Gets moving to.

## Returns

the moving to

**1.8.3.9 hashCode()** `int hashCode ( )`

Hash code int.

## Returns

the int

**1.8.3.10 setMovingFrom()** `void setMovingFrom (   
Location movingFrom )`

Sets moving from.

## Parameters

<i>movingFrom</i>	the moving from
-------------------	-----------------

**1.8.3.11 setMovingTo()** `void setMovingTo (   
                  Location movingTo )`

Sets moving to.

Parameters

<i>movingTo</i>	the moving to
-----------------	---------------

**1.8.3.12 toString()** `String toString ( )`

To string string.

Returns

the string

Reimplemented in **Move** (p. 157).

## 1.9 BitData

### Static Public Attributes

- static final long **notAFile**
- static final long **notHFile**
- static final long **everything**

### 1.9.1 Detailed Description

Bit data - utility class meant for storing useful board constants. (in bitboard format of course)

Author

Bezalel Avrahami ( [bezalel3250@gmail.com](mailto:bezalel3250@gmail.com))

### 1.9.2 Member Data Documentation

**1.9.2.1 everything** `final long everything [static]`

The constant everything. the whole board

### 1.9.2.2 notAFile `final long notAFile [static]`

The constant notAFile. the whole board but the A file

### 1.9.2.3 notHFile `final long notHFile [static]`

The constant notHFile. the whole board but the H file

## 1.10 Board

### Public Member Functions

- **Board** ( **Board** other)
- **Board** ()
- void **setPiece** ( **Location** loc, **Piece** piece)
- **Square** **getSquare** ( **Location** loc)
- **Board** (String fen)
- void **fenSetup** (String fen)
- **Square**[] **getRow** (int row, boolean flipLocs)
- **Piece** **getPiece** ( **Location** loc, boolean notNull)
- **Piece** **getPiece** ( **Location** loc)
- void **print** ()
- Iterator< **Square** > **iterator** ()
- void **setSquareEmpty** ( **Location** loc)
- boolean **isSquareEmpty** ( **Location** loc)
- String **toString** ()

### Static Public Member Functions

- static **Board** **startingPos** ()

### Static Public Attributes

- static final String **startingFen**
- static final **Board** **example**

### 1.10.1 Detailed Description

**Board** (p. 21) - represents the logic board of the model

Author

Bezalel Avrahami ( `bezalel13250@gmail.com`)

### 1.10.2 Constructor & Destructor Documentation

#### 1.10.2.1 **Board**() [1/3] `Board ( Board other )`

Instantiates a new **Board** (p. 21).

**Parameters**

<i>other</i>	the other
--------------	-----------

**1.10.2.2 Board()** [2/3] `Board ( )`

Instantiates a new **Board** (p. 21).

**1.10.2.3 Board()** [3/3] `Board (`  
`String fen )`

Instantiates a new **Board** (p. 21).

**Parameters**

<i>fen</i>	the fen
------------	---------

**1.10.3 Member Function Documentation****1.10.3.1 fenSetup()** `void fenSetup (`  
`String fen )`

Fen setup.

**Parameters**

<i>fen</i>	the fen
------------	---------

**1.10.3.2 getPiece()** [1/2] `Piece getPiece (`  
`Location loc )`

Gets piece.

**Parameters**

<i>loc</i>	the loc
------------	---------

**Returns**

the piece

**1.10.3.3 getPiece()** [2/2] **Piece** getPiece (   
        **Location** *loc*,  
        **boolean** *notNull* )

Gets piece.

Parameters

<i>loc</i>	the loc
<i>notNull</i>	the not null

Returns

the piece

**1.10.3.4 getRow()** **Square[]** getRow (   
        **int** *row*,  
        **boolean** *flipLocs* )

Get row square [ ].

Parameters

<i>row</i>	the row
<i>flipLocs</i>	the flip locs

Returns

the square [ ]

**1.10.3.5 getSquare()** **Square** getSquare (   
        **Location** *loc* )

Gets square.

Parameters

<i>loc</i>	the loc
------------	---------

Returns

the square

**1.10.3.6 isSquareEmpty()** `boolean isSquareEmpty (`  
    **Location** *loc* `)`

Is square empty boolean.

**Parameters**

<i>loc</i>	the loc
------------	---------

**Returns**

the boolean

**1.10.3.7 iterator()** `Iterator< Square > iterator ( )`

Iterator iterator.

**Returns**

the iterator

**1.10.3.8 print()** `void print ( )`

Print.

**1.10.3.9 setPiece()** `void setPiece (`  
    **Location** *loc*,  
    **Piece** *piece* `)`

Sets piece.

**Parameters**

<i>loc</i>	the loc
<i>piece</i>	the piece

**1.10.3.10 setSquareEmpty()** `void setSquareEmpty (`  
    **Location** *loc* `)`

Sets square empty.



## Parameters

<i>loc</i>	the loc
------------	---------

**1.10.3.11 startingPos()** `static Board startingPos ( ) [static]`

Starting pos board.

## Returns

the board

**1.10.3.12 toString()** `String toString ( )`

To string string.

## Returns

the string

## 1.10.4 Member Data Documentation

**1.10.4.1 example** `final Board example [static]`

The constant example.

**1.10.4.2 startingFen** `final String startingFen [static]`

The constant startingFen.

## 1.11 MyJFrame.BooleanClosing

### Public Member Functions

- default Boolean **show** ()
- default boolean **checkClosingVal** (Boolean val)
- void **closing** ()

### Additional Inherited Members

#### 1.11.1 Detailed Description

The interface Boolean closing.

#### 1.11.2 Member Function Documentation

**1.11.2.1 checkClosingVal()** `default boolean checkClosingVal ( Boolean val )`

Check closing val boolean.

**Parameters**

<code>val</code>	the val
------------------	---------

**Returns**

the boolean

**1.11.2.2 closing()** `void closing ( )`

**Closing** (p. 31).

**1.11.2.3 show()** `default Boolean show ( )`

Show boolean.

**Returns**

the boolean

## **1.12 Callback< T >**

### **Public Member Functions**

- `void callback (T obj)`

#### **1.12.1 Detailed Description**

**Callback** (p. 26) - an object callback.

**Parameters**

<code>&lt; T &gt;</code>	the object's type
--------------------------	-------------------

**Author**

Bezalel Avrahami ( `bezalel3250@gmail.com` )

#### **1.12.2 Member Function Documentation**

**1.12.2.1 callback()** `void callback (`  
`T obj )`

**Callback** (p. 26).

## Parameters

<i>obj</i>	the obj
------------	---------

## 1.13 CastlingRights

### Classes

- enum **Side**

### Public Member Functions

- **CastlingRights** (byte rights)
- **CastlingRights** ()
- **CastlingRights** ( **CastlingRights** other)
- void **enableCastling** ( **PlayerColor** playerColor, **Side** side)
- String **toString** ()
- boolean **isEnabled** ( **PlayerColor** playerColor, **Side** side)
- byte **disableCastling** ( **PlayerColor** playerColor, **Side** side)
- byte **getRights** ()
- void **enable** (byte b)
- boolean **hasAny** ( **PlayerColor** playerColor)
- byte **getPlayersCastling** ( **PlayerColor** playerColor)

### Static Public Member Functions

- static **CastlingRights** **createFromStr** (String castlingAbilityStr)
- static void **main** (String[] args)
- static **PlayerColor** **whosCastling** (byte castlingRights)

### Static Public Attributes

- static final String **NO\_CASTLING\_ABILITY**

#### 1.13.1 Detailed Description

Castling rights - .

#### Author

Bezalel Avrahami ( bezalel13250@gmail.com)

#### 1.13.2 Constructor & Destructor Documentation

##### 1.13.2.1 **CastlingRights()** [1/3] **CastlingRights** ( byte rights )

Instantiates a new Castling rights.

## Parameters

<i>rights</i>	the rights
---------------	------------

**1.13.2.2 CastlingRights()** [2/3]    **CastlingRights** ( )

Instantiates a new Castling rights.

**1.13.2.3 CastlingRights()** [3/3]    **CastlingRights** (  
    **CastlingRights** *other* )

Instantiates a new Castling rights.

## Parameters

<i>other</i>	the other
--------------	-----------

**1.13.3 Member Function Documentation****1.13.3.1 createFromStr()**    **static CastlingRights** createFromStr (  
    String *castlingAbilityStr* )    [static]

Create from str castling rights.

## Parameters

<i>castlingAbilityStr</i>	the castling ability str
---------------------------	--------------------------

## Returns

the castling rights

**1.13.3.2 disableCastling()**    **byte** disableCastling (  
    **PlayerColor** *playerColor*,  
    **Side** *side* )

Disable castling byte.

## Parameters

<i>playerColor</i>	the player color
<i>side</i>	the side

## Returns

the byte

**1.13.3.3 enable()** `void enable (`  
    `byte b )`

Enable.

## Parameters

<i>b</i>	the b
----------	-------

**1.13.3.4 enableCastling()** `void enableCastling (`  
    `PlayerColor playerColor,`  
    `Side side )`

Enable castling.

## Parameters

<i>playerColor</i>	the player color
<i>side</i>	the side

**1.13.3.5 getPlayersCastling()** `byte getPlayersCastling (`  
    `PlayerColor playerColor )`

Gets players castling.

## Parameters

<i>playerColor</i>	the player color
--------------------	------------------

## Returns

the players castling

**1.13.3.6 getRights()** `byte getRights ( )`

Gets rights.

**Returns**

the rights

**1.13.3.7 hasAny()** `boolean hasAny (   
          PlayerColor playerColor )`

Has any boolean.

**Parameters**

<i>playerColor</i>	the player color
--------------------	------------------

**Returns**

the boolean

**1.13.3.8 isEnabled()** `boolean isEnabled (   
          PlayerColor playerColor,   
          Side side )`

Is enabled boolean.

**Parameters**

<i>playerColor</i>	the player color
<i>side</i>	the side

**Returns**

the boolean

**1.13.3.9 main()** `static void main (   
          String[] args ) [static]`

The entry point of application.

## Parameters

<i>args</i>	the input arguments
-------------	---------------------

**1.13.3.10 toString()** `String toString ( )`

To string string.

## Returns

the string

**1.13.3.11 whosCastling()** `static PlayerColor whosCastling ( byte castlingRights ) [static]`

Whos castling player color.

## Parameters

<i>castlingRights</i>	the castling rights
-----------------------	---------------------

## Returns

the player color

**1.13.4 Member Data Documentation****1.13.4.1 NO\_CASTLING\_ABILITY** `final String NO_CASTLING_ABILITY [static]`

The constant NO\_CASTLING\_ABILITY.

**1.14 MyJFrame.Closing< T >****Public Member Functions**

- default void **tryClose** ()
- T **show** ()
- boolean **checkClosingVal** (T val)
- void **closing** (T val)

## Public Attributes

- String **title**
- String **header**
- ImageIcon **icon**

### 1.14.1 Detailed Description

The interface **Closing** (p. 31).

#### Parameters

<code>&lt;T&gt;</code>	the type parameter
------------------------	--------------------

### 1.14.2 Member Function Documentation

**1.14.2.1 checkClosingVal()** `boolean checkClosingVal ( T val )`

Check closing val boolean.

#### Parameters

<code>val</code>	the val
------------------	---------

#### Returns

the boolean

**1.14.2.2 closing()** `void closing ( T val )`

**Closing** (p. 31).

#### Parameters

<code>val</code>	the val
------------------	---------

**1.14.2.3 show()** `T show ( )`

Show t.



**Returns**

the t

**1.14.2.4 tryClose()** default void tryClose ( )

Try close.

**1.14.3 Member Data Documentation****1.14.3.1 header** String header

The constant header.

**1.14.3.2 icon** ImageIcon icon

The constant icon.

**1.14.3.3 title** String title

The constant title.

**1.15 Col****Public Member Functions**

- **Col** ( **Col** col)
- **Col** (String colName, String alias)
- **Col** (String colName)
- String **label** ()
- void **wrap** ()
- **Col** **time** ()
- **Col** **date** ()
- void **setWrapped** (boolean wrapped)
- **Col** **as** ()
- **Col** **as** (String alias)
- void **setColName** (String colName)
- String **colName** ()
- **Col** **of** ( **Table** table)
- **Col** **of** (String ofWhom)
- **Col** **replace** (String replacing, String replaceWith)
- String **nested** ()
- **Col** **math** ( **Math** operation, Object value)
- **Col** **math** ( **Math** operation, Object value, boolean changeSelf)
- String **toString** ()

### Static Public Member Functions

- static **Col count** (String **as**)
- static **Col count** (String **as**, Object countWhat)
- static **Col countIf** (String **as**, **Condition** condition)
- static CustomCol **sum** (String **as**, Col... colsToSum)
- static **Col switchCase** (String **as**, SwitchCase... cases)

### Static Public Attributes

- static final **Col GameID**
- static final **Col SavedGame**
- static final **Col Password**
- static final **Col Player1**
- static final **Col Player2**
- static final **Col Username**
- static final **Col Winner**
- static final **Col PlayerToMove**
- static final **Col CreatedDateTime**
- static final **Col ProfilePic**

#### 1.15.1 Detailed Description

**Col** (p.33) - represents a column. either existing column in the db (the constant columns **GameID** (p.42), **SavedGame** (p.42) ...) or created columns.

#### Author

Bezalel Avrahami ( bezalel3250@gmail.com)

#### 1.15.2 Constructor & Destructor Documentation

##### 1.15.2.1 **Col()** [1/3] **Col** (     **Col col** )

Instantiates a new **Col** (p.33).

#### Parameters

<i>col</i>	the col
------------	---------

##### 1.15.2.2 **Col()** [2/3] **Col** (     String *colName*,     String *alias* )

Instantiates a new **Col** (p. 33).

**Parameters**

<i>colName</i>	the col name
<i>alias</i>	the alias

**1.15.2.3 Col()** [3/3]    **Col** (  
          String *colName* )

Instantiates a new **Col** (p. 33).

**Parameters**

<i>colName</i>	the col name
----------------	--------------

**1.15.3 Member Function Documentation**

**1.15.3.1 as()** [1/2]    **Col** as ( )

As own name. practically keeps the name's case.

**Returns**

the col

**1.15.3.2 as()** [2/2]    **Col** as (  
          String *alias* )

creates a new column with the given alias as its alias

**Parameters**

<i>alias</i>	the alias
--------------	-----------

**Returns**

the col

**1.15.3.3 colName()** `String colName ( )`

**Col** (p. 33) name string.

**Returns**

the string

**1.15.3.4 count()** [1/2] `static Col count (String as ) [static]`

Count col. counts every row (\*)

**Parameters**

<i>as</i>	the alias
-----------	-----------

**Returns**

the col

**1.15.3.5 count()** [2/2] `static Col count (String as, Object countWhat ) [static]`

Count col.

**Parameters**

<i>as</i>	the alias
<i>countWhat</i>	the count what

**Returns**

the col

**1.15.3.6 countIf()** `static Col countIf (String as, Condition condition ) [static]`

Count if col. only counts if the given condition is true

**Parameters**

<i>as</i>	the as
<i>condition</i>	the condition

**Returns**

the col

**1.15.3.7 date()** `col date ( )`

Date col.

**Returns**

a new col representing datetime

**1.15.3.8 label()** `String label ( )`

Label string.

**Returns**

the string

**1.15.3.9 math()** [1/2] `col math (`  
`Math operation,`  
`Object value )`

**Math** (p. 109) col.

**Parameters**

<i>operation</i>	the operation
<i>value</i>	the value

**Returns**

the col

**1.15.3.10 math()** [2/2] Col math (   
     **Math** operation,   
     Object value,   
     boolean changeSelf )

**Math** (p. 109) col.

Parameters

<i>operation</i>	the operation
<i>value</i>	the value
<i>changeSelf</i>	the change self

Returns

the col

**1.15.3.11 nested()** String nested ( )

Nested string.

Returns

the string

**1.15.3.12 of()** [1/2] Col of (   
     String ofWhom )

new col like this that belongs to the given owner

Parameters

<i>ofWhom</i>	the of whom
---------------	-------------

Returns

the col

**1.15.3.13 of()** [2/2] Col of (   
     **Table** table )

new col like this that belongs to the given table

**Parameters**

<i>table</i>	the table
--------------	-----------

**Returns**

the col

**1.15.3.14 replace()** `col replace (`  
    *String replacing,*  
    *String replaceWith )*

Replace col.

**Parameters**

<i>replacing</i>	the replacing
<i>replaceWith</i>	the replace with

**Returns**

the col

**1.15.3.15 setColName()** `void setColName (`  
    *String colName )*

Sets col name.

**Parameters**

<i>colName</i>	the col name
----------------	--------------

**1.15.3.16 setWrapped()** `void setWrapped (`  
    *boolean wrapped )*

Sets wrapped.

**Parameters**

<i>wrapped</i>	the wrapped
----------------	-------------



**1.15.3.17 sum()** `static CustomCol sum (`  
    `String as,`  
    `Col... colsToSum ) [static]`

Sum cols.

**Parameters**

<i>as</i>	the alias
<i>colsToSum</i>	the cols to sum

**Returns**

the custom col

**1.15.3.18 switchCase()** `static Col switchCase (`  
    `String as,`  
    `SwitchCase... cases ) [static]`

Switch case col.

**Parameters**

<i>as</i>	the as
<i>cases</i>	the cases

**Returns**

the col

**1.15.3.19 time()** `Col time ( )`

Time col.

**Returns**

a new col representing time

**1.15.3.20 toString()** `String toString ( )`

To string string.

**Returns**

the string

**1.15.3.21 wrap()** `void wrap ( )`

Wrap.

**1.15.4 Member Data Documentation****1.15.4.1 CreatedDateTime** `final Co1 CreatedDateTime [static]`

The constant CreatedDateTime.

**1.15.4.2 GameID** `final Co1 GameID [static]`

The constant GameID.

**1.15.4.3 Password** `final Co1 Password [static]`

The constant Password.

**1.15.4.4 Player1** `final Co1 Player1 [static]`

The constant Player1.

**1.15.4.5 Player2** `final Co1 Player2 [static]`

The constant Player2.

**1.15.4.6 PlayerToMove** `final Co1 PlayerToMove [static]`

The constant PlayerToMove.

**1.15.4.7 ProfilePic** `final Co1 ProfilePic [static]`

The constant ProfilePic.

**1.15.4.8 SavedGame** `final Co1 SavedGame [static]`

The constant SavedGame.

**1.15.4.9 Username** `final Co1 Username [static]`

The constant Username.

**1.15.4.10 Winner** `final Col Winner [static]`

The constant Winner.

**1.16 Condition****Classes**

- enum **Relation**

**Public Member Functions**

- **Condition** (String str, Object... parms)
- **Condition noNulls** ()
- **Condition add** ( **Condition** condition, **Relation** relation)
- **Condition add** ( **Condition** condition, **Relation** relation, boolean **wrap**)
- void **wrap** ()
- **Condition and** ( **Condition** condition)
- String **getStr** ()
- void **setStr** (String str)
- String **toString** ()

**Static Public Member Functions**

- static **Condition equals** (Object col, Object value)
- static **Condition math** (Object col, @MagicConstant(stringValues={">", ">=", "<", "<="}) String operation, Object value)
- static **Condition between** (Object col, Object start, Object end)
- static **Condition notEquals** (Object col, Object value)

**1.16.1 Detailed Description**

**Condition** (p. 43) - represents a condition.

**Author**

Bezalel Avrahami ( [bezalel13250@gmail.com](mailto:bezalel13250@gmail.com))

**1.16.2 Class Documentation**

**1.16.2.1 enum ver14::SharedClasses::DBActions::Condition::Relation Relation** (p. ??) - relations between conditions.

**Author**

Bezalel Avrahami ( [bezalel13250@gmail.com](mailto:bezalel13250@gmail.com))

#### Enumerator

AND	And relation.
OR	Or relation.

### 1.16.3 Constructor & Destructor Documentation

**1.16.3.1 Condition()** `Condition (`  
    `String str,`  
    `Object... parms )`

Instantiates a new **Condition** (p. 43).

#### Parameters

<i>str</i>	the str
<i>parms</i>	the parms

### 1.16.4 Member Function Documentation

**1.16.4.1 add()** [1/2] `Condition add (`  
    `Condition condition,`  
    `Relation relation )`

Add condition.

#### Parameters

<i>condition</i>	the condition
<i>relation</i>	the relation

#### Returns

the condition

**1.16.4.2 add()** [2/2] `Condition add (`  
    `Condition condition,`  
    `Relation relation,`  
    `boolean wrap )`

Add condition.

## Parameters

<i>condition</i>	the condition
<i>relation</i>	the relation
<i>wrap</i>	the wrap

## Returns

THIS condition

**1.16.4.3 and()** `Condition and ( Condition condition )`

wraps

## Parameters

<i>condition</i>	the condition
------------------	---------------

## Returns

condition

**1.16.4.4 between()** `static Condition between ( Object col, Object start, Object end ) [static]`

Between condition.

## Parameters

<i>col</i>	the col
<i>start</i>	the start
<i>end</i>	the end

## Returns

the condition

**1.16.4.5 equals()** `static Condition equals ( Object col, Object value ) [static]`

Equals condition.

## Parameters

<i>col</i>	the col
<i>value</i>	the value

## Returns

the condition

**1.16.4.6** `getStr()` `String getStr ( )`

Gets str.

## Returns

the str

**1.16.4.7** `math()` `static Condition math (`  
`Object col,`  
`@MagicConstant(stringValues={">", ">=", "<", "<="}) String operation,`  
`Object value ) [static]`

Math condition.

## Parameters

<i>col</i>	the col
<i>operation</i>	the operation
<i>value</i>	the value

## Returns

the condition

**1.16.4.8** `noNulls()` `Condition noNulls ( )`

No nulls condition.

## Returns

the condition

**1.16.4.9 notEquals()** `static Condition notEquals (`  
    `Object col,`  
    `Object value ) [static]`

Not equals condition.

**Parameters**

<i>col</i>	the col
<i>value</i>	the value

**Returns**

the condition

**1.16.4.10 setStr()** `void setStr (`  
    `String str )`

Sets str.

**Parameters**

<i>str</i>	the str
------------	---------

**1.16.4.11 toString()** `String toString ( )`

To string string.

**Returns**

the string

**1.16.4.12 wrap()** `void wrap ( )`

Wrap.



## 1.17 Config< V >

### Public Member Functions

- **Config** ()
- **Config** (String **description**)
- **Config** (String **description**, boolean **canUseDefault**, Described< V > defaultValue)
- **Config** (String **description**, V defVal)
- **Config** (String **description**, V defVal, String defDesc)
- **Config** (String **description**, Described< V > defaultValue)
- void **addSuggestion** (Described< V > suggestion)
- ArrayList< Described< V > > **getValuesSuggestion** ()
- Described< V > **getDescribedDefault** ()
- V **getDefault** ()
- String **toString** ()
- String **getDefaultDesc** ()

### Public Attributes

- final boolean **canUseDefault**
- final String **description**

#### 1.17.1 Detailed Description

**Config** (p. 49) - an argument's configuration.

#### Parameters

< V >	the type
-------	----------

#### Author

Bezalel Avrahami ( bezalel3250@gmail.com)

#### 1.17.2 Constructor & Destructor Documentation

##### 1.17.2.1 Config() [1/6] Config ( )

Instantiates a new **Config** (p. 49).

##### 1.17.2.2 Config() [2/6] Config ( String *description* )

Instantiates a new **Config** (p. 49).

**Parameters**

<i>description</i>	the description
--------------------	-----------------

**1.17.2.3 Config()** [3/6]   **Config** (  
    String *description*,  
    boolean *canUseDefault*,  
    Described< V > *defaultValue* )

Instantiates a new **Config** (p. 49).

**Parameters**

<i>description</i>	the description
<i>canUseDefault</i>	the can use default
<i>defaultValue</i>	the default value

**1.17.2.4 Config()** [4/6]   **Config** (  
    String *description*,  
    V *defVal* )

Instantiates a new **Config** (p. 49).

**Parameters**

<i>description</i>	the description
<i>defVal</i>	the def val

**1.17.2.5 Config()** [5/6]   **Config** (  
    String *description*,  
    V *defVal*,  
    String *defDesc* )

Instantiates a new **Config** (p. 49).

**Parameters**

<i>description</i>	the description
<i>defVal</i>	the def val
<i>defDesc</i>	the def desc

**1.17.2.6 Config()** [6/6] **Config** (  
    String *description*,  
    Described< V > *defaultValue* )

Instantiates a new **Config** (p. 49).

**Parameters**

<i>description</i>	the description
<i>defaultValue</i>	the default value

### 1.17.3 Member Function Documentation

**1.17.3.1 addSuggestion()** void addSuggestion (  
    Described< V > *suggestion* )

Add suggestion.

**Parameters**

<i>suggestion</i>	the suggestion
-------------------	----------------

**1.17.3.2 getDefault()** V getDefault ( )

Gets default.

**Returns**

the default

**1.17.3.3 getDefaultDesc()** String getDefaultDesc ( )

Gets default desc.

**Returns**

the default desc

**1.17.3.4** **getDescribedDefault()** `Described< V > getDescribedDefault ( )`

Gets described default.

**Returns**

the described default

**1.17.3.5** **getValuesSuggestion()** `ArrayList< Described< V > > getValuesSuggestion ( )`

Gets values suggestion.

**Returns**

the values suggestion

**1.17.3.6** **toString()** `String toString ( )`

To string string.

**Returns**

the string

**1.17.4 Member Data Documentation****1.17.4.1** **canUseDefault** `final boolean canUseDefault`

The Can use default.

**1.17.4.2** **description** `final String description`

The Description.

**1.18 ConfirmDialogs****Static Public Member Functions**

- static boolean **confirm** (Component parent, String title, String message, ImageIcon icon)
- static void **main** (String[] args)
- static String **confirm** (Component parent, String title, String header, String message, ImageIcon icon, String initialValue)

### 1.18.1 Detailed Description

The utility class Confirm dialogs.

### 1.18.2 Member Function Documentation

**1.18.2.1 confirm() [1/2]** static String confirm (  
    Component *parent*,  
    String *title*,  
    String *header*,  
    String *message*,  
    ImageIcon *icon*,  
    String *initialValue* ) [static]

Confirm string.

#### Parameters

<i>parent</i>	the parent
<i>header</i>	the header
<i>message</i>	the message
<i>title</i>	the title
<i>icon</i>	the icon
<i>initialValue</i>	the initial value

#### Returns

the string

**1.18.2.2 confirm() [2/2]** static boolean confirm (  
    Component *parent*,  
    String *title*,  
    String *message*,  
    ImageIcon *icon* ) [static]

Confirm boolean.

#### Parameters

<i>parent</i>	the parent
<i>title</i>	the title
<i>message</i>	the message
<i>icon</i>	the icon

**Returns**

the boolean

**1.18.2.3 main()** `static void main (`  
`String[] args ) [static]`

The entry point of application.

**Parameters**

<i>args</i>	the input arguments
-------------	---------------------

## 1.19 CreatedGame

**Public Member Functions**

- **CreatedGame** (String **gameId**, String **creatorUsername**, GameSettings **gameSettings**)
- String **getGameDesc** ()

**Additional Inherited Members**

### 1.19.1 Detailed Description

Created game.

**Author**

Bezalel Avrahami ( bezalel3250@gmail.com)

### 1.19.2 Constructor & Destructor Documentation

**1.19.2.1 CreatedGame()** `CreatedGame (`  
`String gameId,`  
`String creatorUsername,`  
`GameSettings gameSettings )`

Instantiates a new Created game.

**Parameters**

<i>gameId</i>	the game id
<i>creatorUsername</i>	the creator username
<i>gameSettings</i>	the game settings

### 1.19.3 Member Function Documentation

#### 1.19.3.1 `getGameDesc()` `String getGameDesc ( )`

Gets game desc.

##### Returns

the game desc

Reimplemented from **GameInfo** (p. 82).

## 1.20 CustomStatement

### Public Member Functions

- **CustomStatement** (DBRequest.Type **type**, @Language("SQL") String statement)

### Protected Member Functions

- String **createStatement** ()

### Additional Inherited Members

#### 1.20.1 Detailed Description

Custom statement - a custom sql statement.

##### Author

Bezalel Avrahami ( bezalel13250@gmail.com)

#### 1.20.2 Constructor & Destructor Documentation

##### 1.20.2.1 **CustomStatement()** `CustomStatement (` `DBRequest.Type type,` `@Language("SQL") String statement )`

Instantiates a new Custom statement.

**Parameters**

<i>type</i>	the type
<i>statement</i>	the statement

**1.20.3 Member Function Documentation****1.20.3.1 createStatement()** `String createStatement ( ) [protected]`

Create statement string.

**Returns**

the string

Reimplemented from **SQLStatement** (p. 211).

**1.21 DBRequest****Classes**

- enum **Type**

**Public Member Functions**

- **DBRequest** ( **SQLStatement** sqlStatement)
- **DBRequest** ( **Type** type, String request, **RequestBuilder** builder)
- **DBRequest** ( **SQLStatement** sqlStatement, **RequestBuilder** builder)
- **RequestBuilder** **getBuilder** ()
- **DBRequest** **getSubRequest** ()
- void **setSubRequest** ( **DBRequest** subRequest)
- String **toString** ()
- String **getRequest** ()

**Public Attributes**

- final **Type** type

**1.21.1 Detailed Description**

Db request - a database request.

**Author**

Bezalel Avrahami ( bezalel13250@gmail.com)

**1.21.2 Class Documentation****1.21.2.1 enum ver14::SharedClasses::DBActions::DBRequest::DBRequest::Type Type** (p. ??) - .**Author**

Bezalel Avrahami ( bezalel13250@gmail.com)



## Enumerator

Query	Query type.
Update	Update type.

## 1.21.3 Constructor &amp; Destructor Documentation

**1.21.3.1 DBRequest()** [1/3] `DBRequest (`  
`SQLStatement sqlStatement )`

Instantiates a new Db request.

## Parameters

<i>sqlStatement</i>	the sql statement
---------------------	-------------------

**1.21.3.2 DBRequest()** [2/3] `DBRequest (`  
`Type type,`  
`String request,`  
`RequestBuilder builder )`

Instantiates a new Db request.

## Parameters

<i>type</i>	the type
<i>request</i>	the request
<i>builder</i>	the builder

**1.21.3.3 DBRequest()** [3/3] `DBRequest (`  
`SQLStatement sqlStatement,`  
`RequestBuilder builder )`

Instantiates a new Db request.

## Parameters

<i>sqlStatement</i>	the sql statement
<i>builder</i>	the builder

#### 1.21.4 Member Function Documentation

##### 1.21.4.1 **getBuilder()** `RequestBuilder getBuilder ( )`

Gets builder.

###### Returns

the builder

##### 1.21.4.2 **getRequest()** `String getRequest ( )`

Gets request.

###### Returns

the request

##### 1.21.4.3 **getSubRequest()** `DBRequest getSubRequest ( )`

Gets sub request.

###### Returns

the sub request

##### 1.21.4.4 **setSubRequest()** `void setSubRequest ( DBRequest subRequest )`

Sets sub request.

###### Parameters

<i>subRequest</i>	the sub request
-------------------	-----------------

##### 1.21.4.5 **toString()** `String toString ( )`

To string string.

## Returns

the string

## 1.21.5 Member Data Documentation

1.21.5.1 `type final Type type`

The `Type` (p. ??).

## 1.22 DBResponse

## Classes

- enum `Status`

## Public Member Functions

- `DBRequest getRequest ()`
- `Status getStatus ()`
- `boolean isSuccess ()`
- `abstract boolean isAnyData ()`
- `DBResponse getAddedRes ()`
- `void setAddedRes (DBResponse addedRes)`
- `abstract DBResponse clean ()`
- `void print ()`
- `String toString ()`

## Protected Member Functions

- `DBResponse (Status status, DBRequest request)`

## Protected Attributes

- `final Status status`
- `final DBRequest request`
- `DBResponse addedRes`

## 1.22.1 Detailed Description

Db response - .

## Author

Bezalel Avrahami ( bezalel13250@gmail.com)

## 1.22.2 Class Documentation

1.22.2.1 `enum ver14::SharedClasses::DBActions::DBResponse::DBResponse::Status Status` (p. ??) - .

## Author

Bezalel Avrahami ( bezalel13250@gmail.com)

**Enumerator**

ERROR	Error status.
SUCCESS	Success status.

**1.22.3 Constructor & Destructor Documentation**

**1.22.3.1 DBResponse()** `DBResponse (`  
    `Status status,`  
    `DBRequest request )` [protected]

Instantiates a new Db response.

**Parameters**

<i>status</i>	the status
<i>request</i>	the request

**1.22.4 Member Function Documentation**

**1.22.4.1 clean()** `abstract DBResponse clean ( )` [abstract]

Clean db response.

**Returns**

the db response

Reimplemented in **StatusResponse** (p.216), and **TableDBResponse** (p.233).

**1.22.4.2 getAddedRes()** `DBResponse getAddedRes ( )`

Gets added res.

**Returns**

the added res

**1.22.4.3 getRequest()** `DBRequest getRequest ( )`

Gets request.

**Returns**

the request

**1.22.4.4 getStatus()** `Status getStatus ( )`

Gets status.

**Returns**

the status

**1.22.4.5 isAnyData()** `abstract boolean isAnyData ( ) [abstract]`

Is any data boolean.

**Returns**

the boolean

Reimplemented in **StatusResponse** (p. 216), and **TableDBResponse** (p. 235).

**1.22.4.6 isSuccess()** `boolean isSuccess ( )`

Is success boolean.

**Returns**

the boolean

**1.22.4.7 print()** `void print ( )`

Print.

**1.22.4.8 setAddedRes()** `void setAddedRes (   
DBResponse addedRes )`

Sets added res.

**Parameters**

<i>addedRes</i>	the added res
-----------------	---------------

**1.22.4.9 toString()** `String toString ( )`

To string string.

**Returns**

the string

Reimplemented in **TableDBResponse** (p. 236).

**1.22.5 Member Data Documentation****1.22.5.1 addedRes** `DBResponse addedRes [protected]`

The Added res.

**1.22.5.2 request** `final DBRequest request [protected]`

The Request.

**1.22.5.3 status** `final Status status [protected]`

The **Status** (p. ??).

**1.23 Delete****Public Member Functions**

- **Delete** ( **Table** deletingFrom, **Condition** condition)

**Protected Member Functions**

- String **createStatement** ()

## Additional Inherited Members

### 1.23.1 Detailed Description

**Delete** (p. 62) - deletion statement.

#### Author

Bezalel Avrahami ( bezalel13250@gmail.com)

### 1.23.2 Constructor & Destructor Documentation

**1.23.2.1 Delete()** `Delete (`  
     **Table** *deletingFrom*,  
     **Condition** *condition* )

Instantiates a new **Delete** (p. 62).

#### Parameters

<i>deletingFrom</i>	the deleting from
<i>condition</i>	the condition

### 1.23.3 Member Function Documentation

**1.23.3.1 createState()** `String createState ( )` [protected]

Create statement string.

#### Returns

the string

Reimplemented from **SQLStatement** (p. 211).

## 1.24 Direction Enum Reference

### Public Member Functions

- **Direction[]** `getCombination ()`
- **Direction** `perspective ( PlayerColor playerColor)`
- abstract **Direction** `opposite ()`

## Static Public Member Functions

- static **Direction** **getRelative** ( **Location** loc1, **Location** loc2)
- static **Direction** **getDirectionByOffset** (int **offset**)

## Public Attributes

- **U**
- **D**
- **L**
- **R**
- **U\_U**
- **D\_D**
- **U\_R**
- **U\_L**
- **D\_R**
- **D\_L**
- **U\_U\_R**
- **U\_U\_L**
- **U\_R\_R**
- **U\_L\_L**
- **D\_D\_R**
- **D\_D\_L**
- **D\_R\_R**
- **D\_L\_L**
- final long **andWith**
- final int **offset**
- final int **asInt**
- final **Direction[]** **combination**

## Static Public Attributes

- static final int **NUM\_OF\_DIRECTIONS**
- static final int **NUM\_OF\_KNIGHT\_DIRECTIONS**
- static final int **NUM\_OF\_DIRECTIONS\_WO\_KNIGHT**
- static final **Direction[]** **ALL\_DIRECTIONS**
- static final List< **Direction** > **ALL\_USED\_DIRECTIONS**
- static final **PlayerColor** **normalPerspective**

### 1.24.1 Detailed Description

**Direction** (p. 63) - represents a moving direction on a board. sort of like a vector. has an **offset** (p. 68) that is added to a certain location or bitboard, in order to achieve movement in that direction. the general direction map looks like this:

-9	-8	-7
-1	loc	1
7	8	9



## Author

Bezalel Avrahami ( bezalel3250@gmail.com)

## 1.24.2 Member Function Documentation

### 1.24.2.1 getCombination() `Direction[] getCombination ( )`

Get combination direction [ ].

## Returns

the direction [ ]

### 1.24.2.2 getDirectionByOffset() `static Direction getDirectionByOffset ( int offset ) [static]`

Gets direction by offset.

## Parameters

<code>offset</code>	the offset
---------------------	------------

## Returns

the direction by offset

### 1.24.2.3 getRelative() `static Direction getRelative ( Location loc1, Location loc2 ) [static]`

Gets relative.

## Parameters

<code>loc1</code>	the loc 1
<code>loc2</code>	the loc 2

## Returns

the relative

#### 1.24.2.4 **opposite()** `abstract Direction opposite ( ) [abstract]`

the Opposite direction to this one.

##### Returns

the direction

#### 1.24.2.5 **perspective()** `Direction perspective ( PlayerColor playerColor )`

gets the correct perspective for the provided player color. this is necessary because for example: a white pawn push(**U** (p. 68)) is the exact opposite of a black pawn push (**D** (p. 67)). so the perspective needs to be in relation to the moving color.

##### Parameters

<i>playerColor</i>	the player color
--------------------	------------------

##### Returns

the direction

### 1.24.3 Member Data Documentation

#### 1.24.3.1 **ALL DIRECTIONS** `final Direction [ ] ALL DIRECTIONS [static]`

The All directions.

#### 1.24.3.2 **ALL\_USED DIRECTIONS** `final List< Direction> ALL_USED DIRECTIONS [static]`

The All used directions.

#### 1.24.3.3 **andWith** `final long andWith`

some directions need to filter false positives. for example: moving left one square from the left-most column, will overflow to the previous row. to fix this problem some directions have a andWith value they have to perform a bitwise and with, after every offset. to cancel the false positives. in the left direction example, the andWith is the whole board but the right-most column

#### 1.24.3.4 **asInt** `final int asInt`

an int for quick access by index.

**1.24.3.5 combination** `final Direction [] combination`

some

**1.24.3.6 D** `D`

one square down the board.

**1.24.3.7 D\_D** `D_D`

two squares down.

**1.24.3.8 D\_D\_L** `D_D_L`

two squares down and one square left.

**1.24.3.9 D\_D\_R** `D_D_R`

two squares down and one square right.

**1.24.3.10 D\_L** `D_L`

one square down and one square left.

**1.24.3.11 D\_L\_L** `D_L_L`

two squares left and one square down.

**1.24.3.12 D\_R** `D_R`

one square down and one square right.

**1.24.3.13 D\_R\_R** `D_R_R`

two squares right and one square down.

**1.24.3.14 L** `L`

one square left.

**1.24.3.15 normalPerspective** `final PlayerColor normalPerspective [static]`

the perspective the offset is correct for. if the moving piece's color is not this value the direction need to be flipped.

**1.24.3.16 NUM\_OF DIRECTIONS** `final int NUM_OF DIRECTIONS [static]`

The constant NUM\_OF DIRECTIONS.

**1.24.3.17 NUM\_OF DIRECTIONS\_WO\_KNIGHT** `final int NUM_OF DIRECTIONS_WO_KNIGHT [static]`

The constant NUM\_OF DIRECTIONS\_WO\_KNIGHT.

**1.24.3.18 NUM\_OF\_KNIGHT DIRECTIONS** `final int NUM_OF_KNIGHT DIRECTIONS [static]`

The constant NUM\_OF\_KNIGHT DIRECTIONS.

**1.24.3.19 offset** `final int offset`

The actual offset.

**1.24.3.20 R** `R`

one square right.

**1.24.3.21 U** `U`

one square up the board.

**1.24.3.22 U\_L** `U_L`

one square up and one square left.

**1.24.3.23 U\_L\_L** `U_L_L`

two squares left and one square up.

**1.24.3.24 U\_R** `U_R`

one square up and one square right.

**1.24.3.25 U\_R\_R** `U_R_R`

two squares right and one square up.

**1.24.3.26 U\_U** `U_U`

two squares up.

**1.24.3.27 U\_U\_L** U\_U\_L

two squares up and one square left.

**1.24.3.28 U\_U\_R** U\_U\_R

two squares up and one square right.

**1.25 EnvManager****Public Member Functions**

- void **handledErr** ( **MyError** err)
- void **criticalErr** ( **MyError** err)

**1.25.1 Detailed Description**

Env manager - an object that can handle errors as they occur.

**Author**

Bezalel Avrahami ( bezalel3250@gmail.com)

**1.25.2 Member Function Documentation****1.25.2.1 criticalErr()** void criticalErr ( **MyError** err )

notifies manager of an un-handleable error. the manager must shut down everything

**Parameters**

<i>err</i>	the error thrown
------------	------------------

**1.25.2.2 handledErr()** void handledErr ( **MyError** err )

notifies manager of a managed error

**Parameters**

<i>err</i>	the error thrown
------------	------------------

## 1.26 ErrorHandler< E extends MyError >

### Public Member Functions

- void **handle** ( **MyError** err)

### Static Public Member Functions

- static boolean **ignore** (ThrowingRunnable runnable)

#### 1.26.1 Detailed Description

The interface Error handler.

##### Parameters

<E>	the type parameter
-----	--------------------

#### 1.26.2 Member Function Documentation

**1.26.2.1 handle()** void handle (   
 **MyError** err )

Handle.

##### Parameters

<i>err</i>	the err
------------	---------

**1.26.2.2 ignore()** static boolean ignore (   
 ThrowingRunnable *runnable* ) [static]

Ignore boolean.

##### Parameters

<i>runnable</i>	the runnable
-----------------	--------------

##### Returns

true if the runnable threw, false otherwise

## 1.27 EstablishedGameInfo

### Public Member Functions

- Date **getCreatedAt** ()
- void **setCreatedAt** (Date **createdAt**)
- Stack< **Move** > **getMoveStack** ()
- String **toString** ()
- String **getGameDesc** ()

### Public Attributes

- final String **opponentUsername**

### Protected Member Functions

- **EstablishedGameInfo** (String **gameId**, String **creatorUsername**, String **opponentUsername**, GameSettings **gameSettings**, Stack< **Move** > **moveStack**)

### Protected Attributes

- Date **createdAt**

### Additional Inherited Members

#### 1.27.1 Detailed Description

Established game info.

#### Author

Bezalel Avrahami ( bezalel13250@gmail.com)

#### 1.27.2 Constructor & Destructor Documentation

**1.27.2.1 EstablishedGameInfo()**    **EstablishedGameInfo** (

```

    String gameId,
    String creatorUsername,
    String opponentUsername,
    GameSettings gameSettings,
    Stack< Move > moveStack ) [protected]
```

Instantiates a new Established game info.

**Parameters**

<i>gameId</i>	the game id
<i>creatorUsername</i>	the creator username
<i>opponentUsername</i>	the opponent username
<i>gameSettings</i>	the game settings
<i>moveStack</i>	the move stack

**1.27.3 Member Function Documentation****1.27.3.1 getCreatedAt()** `Date getCreatedAt ( )`

Gets created at.

**Returns**

the created at

**1.27.3.2 getGameDesc()** `String getGameDesc ( )`

Gets game desc.

**Returns**

the game desc

Reimplemented from **GameInfo** (p. 82).

**1.27.3.3 getMoveStack()** `Stack< Move > getMoveStack ( )`

Gets move stack.

**Returns**

the move stack

**1.27.3.4 setCreatedAt()** `void setCreatedAt (   
Date createdAt )`

Sets created at.



## Parameters

<i>createdAt</i>	the created at
------------------	----------------

**1.27.3.5 toString()** `String toString ( )`

To string string.

## Returns

the string

Reimplemented from **GameInfo** (p. 83).

Reimplemented in **ArchivedGameInfo** (p. 7).

**1.27.4 Member Data Documentation****1.27.4.1 createdAt** `Date createdAt [protected]`

The Created at.

**1.27.4.2 opponentUsername** `final String opponentUsername`

The Opponent username.

**1.28 Evaluation****Public Member Functions**

- **Evaluation** ( **GameStatus** gameStatus, **PlayerColor** evaluationFor)
- **Evaluation** (int eval, **GameStatus** gameStatus, **PlayerColor** evaluationFor)
- **Evaluation** ( **PlayerColor** evaluationFor)
- **Evaluation** ( **Evaluation** other)
- void **addDetail** ( **EvaluationParameters** parm, int value)
- void **assertNotGameOver** ()
- Integer **getEvaluationDepth** ()
- void **setEvaluationDepth** (Integer evaluationDepth)
- boolean **isGameOver** ()
- boolean **isCheck** ()
- boolean **isGreaterThan** ( **Evaluation** other)
- int **getEval** ()
- void **setEval** (int eval)
- **GameStatus** **getGameStatus** ()
- boolean **equals** (Object o)
- String **toString** ()
- **PlayerColor** **getEvaluationFor** ()
- **Evaluation** **setPerspective** ( **PlayerColor** playerColor)
- void **flipEval** ()
- void **print** ()
- float **convertFromCentipawns** ()
- record **EvaluationDetail** ( **EvaluationParameters** parm, double eval) implements Serializable

### Static Public Member Functions

- static **Evaluation** **book** ()

### Static Public Attributes

- static final int **TIE\_EVAL**
- static final int **WIN\_EVAL**
- static final int **LOSS\_EVAL**

## 1.28.1 Detailed Description

**Evaluation** (p. 73).

#### Author

Bezalel Avrahami ( bezalel3250@gmail.com)

## 1.28.2 Constructor & Destructor Documentation

**1.28.2.1 Evaluation()** [1/4] **Evaluation** (  
    **GameStatus** *gameStatus*,  
    **PlayerColor** *evaluationFor* )

Instantiates a new **Evaluation** (p. 73).

#### Parameters

<i>gameStatus</i>	the game status
<i>evaluationFor</i>	the evaluation for

**1.28.2.2 Evaluation()** [2/4] **Evaluation** (  
    int *eval*,  
    **GameStatus** *gameStatus*,  
    **PlayerColor** *evaluationFor* )

Instantiates a new **Evaluation** (p. 73).

#### Parameters

<i>eval</i>	the eval
<i>gameStatus</i>	the game status
<i>evaluationFor</i>	the evaluation for

### 1.28.2.3 `Evaluation()` [3/4] `Evaluation (` `PlayerColor evaluationFor )`

Instantiates a new **Evaluation** (p. 73).

#### Parameters

<i>evaluationFor</i>	the evaluation for
----------------------	--------------------

### 1.28.2.4 `Evaluation()` [4/4] `Evaluation (` `Evaluation other )`

Instantiates a new **Evaluation** (p. 73).

#### Parameters

<i>other</i>	the other
--------------	-----------

## 1.28.3 Member Function Documentation

### 1.28.3.1 `addDetail()` `void addDetail (` `EvaluationParameters parm,` `int value )`

Add detail.

#### Parameters

<i>parm</i>	the parm
<i>value</i>	the value

### 1.28.3.2 `assertNotGameOver()` `void assertNotGameOver ( )`

Assert not game over.

### 1.28.3.3 `book()` `static Evaluation book ( ) [static]`

Book evaluation.

**Returns**

the evaluation

**1.28.3.4 convertFromCentipawns()** `float convertFromCentipawns ( )`

Convert from centipawns float.

**Returns**

the float

**1.28.3.5 equals()** `boolean equals (   
Object o )`

Equals boolean.

**Parameters**

<i>o</i>	the o
----------	-------

**Returns**

the boolean

**1.28.3.6 EvaluationDetail()** `record EvaluationDetail (   
EvaluationParameters parm,   
double eval )`

**Evaluation** (p. 73) detail.

**Author**

Bezalel Avrahami ( bezalel3250@gmail.com)

**1.28.3.7 flipEval()** `void flipEval ( )`

Flip eval.

**1.28.3.8 getEval()** `int getEval ( )`

Gets eval.

Returns

the eval

**1.28.3.9 getEvaluationDepth()** `Integer getEvaluationDepth ( )`

Gets evaluation depth.

Returns

the evaluation depth

**1.28.3.10 getEvaluationFor()** `PlayerColor getEvaluationFor ( )`

Gets evaluation for.

Returns

the evaluation for

**1.28.3.11 getGameStatus()** `GameStatus getGameStatus ( )`

Gets game status.

Returns

the game status

**1.28.3.12 isCheck()** `boolean isCheck ( )`

Is check boolean.

Returns

the boolean

**1.28.3.13 isGameOver()** `boolean isGameOver ( )`

Is game over boolean.

Returns

the boolean

**1.28.3.14 isGreaterThan()** `boolean isGreaterThan (   
Evaluation other )`

Is greater than boolean.

**Parameters**

<i>other</i>	the other
--------------	-----------

**Returns**

the boolean

**1.28.3.15 print()** `void print ( )`

Print.

**1.28.3.16 setEval()** `void setEval (`  
`int eval )`

Sets eval.

**Parameters**

<i>eval</i>	the eval
-------------	----------

**1.28.3.17 setEvaluationDepth()** `void setEvaluationDepth (`  
`Integer evaluationDepth )`

Sets evaluation depth.

**Parameters**

<i>evaluationDepth</i>	the evaluation depth
------------------------	----------------------

**1.28.3.18 setPerspective()** `Evaluation setPerspective (`  
`PlayerColor playerColor )`

Sets perspective.

**Parameters**

<i>playerColor</i>	the player color
--------------------	------------------

**Returns**

the perspective

**1.28.3.19 toString()** `String toString ( )`

To string string.

**Returns**

the string

**1.28.4 Member Data Documentation****1.28.4.1 LOSS\_EVAL** `final int LOSS_EVAL [static]`

The constant LOSS\_EVAL.

**1.28.4.2 TIE\_EVAL** `final int TIE_EVAL [static]`

The constant TIE\_EVAL.

**1.28.4.3 WIN\_EVAL** `final int WIN_EVAL [static]`

The constant WIN\_EVAL.

**1.29 EvaluationParameters Enum Reference****Public Attributes**

- **MATERIAL**
- **PIECE\_TABLES**
- **KING\_SAFETY**
- **HANGING\_PIECES**
- **SQUARE\_CONTROL**
- **MOVEMENT\_ABILITY**
- **FORCE\_KING\_TO\_CORNER**
- **EG\_WEIGHT**
- **STOCKFISH\_SAYS**
- final double **weight**

### 1.29.1 Detailed Description

**Evaluation** (p. 73) parameters - all evaluation parameters. (some are unused).

Author

Bezalel Avrahami ( bezalel3250@gmail.com)

### 1.29.2 Member Data Documentation

#### 1.29.2.1 EG\_WEIGHT EG\_WEIGHT

Eg weight evaluation parameters.

#### 1.29.2.2 FORCE\_KING\_TO\_CORNER FORCE\_KING\_TO\_CORNER

Force king to corner evaluation parameters.

#### 1.29.2.3 HANGING\_PIECES HANGING\_PIECES

Hanging pieces evaluation parameters.

#### 1.29.2.4 KING\_SAFETY KING\_SAFETY

King safety evaluation parameters.

#### 1.29.2.5 MATERIAL MATERIAL

Material evaluation parameters.

#### 1.29.2.6 MOVEMENT\_ABILITY MOVEMENT\_ABILITY

Movement ability evaluation parameters.

#### 1.29.2.7 PIECE\_TABLES PIECE\_TABLES

Piece tables evaluation parameters.

#### 1.29.2.8 SQUARE\_CONTROL SQUARE\_CONTROL

Square control evaluation parameters.



**1.29.2.9 STOCKFISH\_SAYS** `STOCKFISH_SAYS`

Stockfish says evaluation parameters.

**1.29.2.10 weight** `final double weight`

The parameter's weight

**1.30 GamelInfo****Public Member Functions**

- boolean **isCreator** (String username)
- String **ID** ()
- boolean **equals** (Object o)
- String **toString** ()
- abstract String **getGameDesc** ()
- **PlayerColor** **getJoiningPlayerColor** ()
- **PlayerColor** **getStartingColor** ()

**Static Public Member Functions**

- static **GamelInfo** **example** ()

**Public Attributes**

- final String **gameId**
- final String **creatorUsername**
- final GameSettings **gameSettings**

**Protected Member Functions**

- **GamelInfo** (String **gameId**, String **creatorUsername**, GameSettings **gameSettings**)

**1.30.1 Detailed Description**

Game info.

Author

Bezalel Avrahami ( `bezalel3250@gmail.com`)

**1.30.2 Constructor & Destructor Documentation**

**1.30.2.1 GamelInfo()** `GamelInfo (`  
`String gameId,`  
`String creatorUsername,`  
`GameSettings gameSettings )` [protected]

Instantiates a new Game info.

**Parameters**

<i>gameId</i>	the game id
<i>creatorUsername</i>	the creator username
<i>gameSettings</i>	the game settings

**1.30.3 Member Function Documentation**

**1.30.3.1 equals()** `boolean equals (   
Object o )`

Equals boolean.

**Parameters**

<i>o</i>	the o
----------	-------

**Returns**

the boolean

**1.30.3.2 example()** `static GameInfo example ( ) [static]`

Example game info.

**Returns**

the game info

**1.30.3.3 getGameDesc()** `abstract String getGameDesc ( ) [abstract]`

Gets game desc.

**Returns**

the game desc

Reimplemented in **CreatedGame** (p. 55), and **EstablishedGameInfo** (p. 72).

**1.30.3.4 getJoiningPlayerColor()** `PlayerColor getJoiningPlayerColor ( )`

Gets joining player color.

**Returns**

the joining player color

**1.30.3.5 getStartingColor()** `PlayerColor getStartingColor ( )`

Gets starting color.

**Returns**

the starting color

**1.30.3.6 ID()** `String ID ( )`

Id string.

**Returns**

the string

**1.30.3.7 isCreator()** `boolean isCreator (   
String username )`

Is creator boolean.

**Parameters**

<i>username</i>	the username
-----------------	--------------

**Returns**

the boolean

**1.30.3.8 toString()** `String toString ( )`

To string string.

#### Returns

the string

Reimplemented in **ArchivedGameInfo** (p. 7), and **EstablishedGameInfo** (p. 73).

### 1.30.4 Member Data Documentation

#### 1.30.4.1 creatorUsername `final String creatorUsername`

The Creator username.

#### 1.30.4.2 gameId `final String gameId`

The Game id.

#### 1.30.4.3 gameSettings `final GameSettings gameSettings`

The Game settings.

## 1.31 GameStatus

### Classes

- enum **GameStatusType**
- enum **SpecificStatus**

### Public Member Functions

- void **setCustomStr** (String customStr)
- boolean **isDisconnected** ()
- **Location** **getCheckedKingLoc** ()
- **PlayerColor** **getWinningColor** ()
- **GameStatusType** **getGameStatusType** ()
- void **setInCheck** ( **Location** checkedKingLoc)
- boolean **isCheck** ()
- String **getDetailedStr** ()
- String **getDetailedStr** (Map< **PlayerColor**, String > playerUsernamesMap)
- boolean **isGameOver** ()
- String **toString** ()
- void **setDepth** (int depth)
- **SpecificStatus** **getSpecificStatus** ()

### Static Public Member Functions

- static **GameStatus** **checkmate** ( **PlayerColor** winningPlayerColor, **Location** matedKing)
- static **GameStatus** **gameGoesOn** ()
- static **GameStatus** **tieByAgreement** ()
- static **GameStatus** **stalemate** ()
- static **GameStatus** **fiftyMoveRule** ()
- static **GameStatus** **serverStoppedGame** (String cause)
- static **GameStatus** **threeFoldRepetition** ()
- static **GameStatus** **insufficientMaterial** ()
- static **GameStatus** **playerDisconnected** ( **PlayerColor** disconnectedPlayer, boolean isVsAi)
- static **GameStatus** **timedOut** ( **PlayerColor** timedOutPlayer, boolean isSufficientMaterial)
- static **GameStatus** **playerResigned** ( **PlayerColor** resignedPlayer)

#### 1.31.1 Detailed Description

Game status - represents a game status .

##### Author

Bezalel Avrahami ( bezalel13250@gmail.com)

#### 1.31.2 Member Function Documentation

**1.31.2.1 checkmate()** static **GameStatus** checkmate (   
 **PlayerColor** winningPlayerColor,   
 **Location** matedKing ) [static]

Checkmate game status.

##### Parameters

<i>winningPlayerColor</i>	the winning player color
<i>matedKing</i>	the mated king

##### Returns

the game status

**1.31.2.2 fiftyMoveRule()** static **GameStatus** fiftyMoveRule ( ) [static]

Fifty move rule game status.

##### Returns

the game status

**1.31.2.3 gameGoesOn()** `static GameState gameGoesOn ( ) [static]`

Game goes on game status.

**Returns**

the game status

**1.31.2.4 getCheckedKingLoc()** `Location getCheckedKingLoc ( )`

Gets checked king loc.

**Returns**

the checked king loc

**1.31.2.5 getDetailedStr()** `[1/2] String getDetailedStr ( )`

Gets detailed str.

**Returns**

the detailed str

**1.31.2.6 getDetailedStr()** `[2/2] String getDetailedStr ( Map< PlayerColor, String > playerUsernamesMap )`

Gets detailed str.

**Parameters**

<i>playerUsernamesMap</i>	the player usernames map
---------------------------	--------------------------

**Returns**

the detailed str

**1.31.2.7 getGameStatusType()** `GameStatusType getGameStatusType ( )`

Gets game status type.

**Returns**

the game status type

**1.31.2.8 getSpecificStatus()** `SpecificStatus` `getSpecificStatus ( )`

Gets specific status.

**Returns**

the specific status

**1.31.2.9 getWinningColor()** `PlayerColor` `getWinningColor ( )`

Gets winning color.

**Returns**

the winning color

**1.31.2.10 insufficientMaterial()** `static GameStatus` `insufficientMaterial ( )` `[static]`

Insufficient material game status.

**Returns**

the game status

**1.31.2.11 isCheck()** `boolean` `isCheck ( )`

Is check boolean.

**Returns**

the boolean

**1.31.2.12 isDisconnected()** `boolean isDisconnected ( )`

Is disconnected boolean.

**Returns**

the boolean

**1.31.2.13 isGameOver()** `boolean isGameOver ( )`

Is game over boolean.

**Returns**

the boolean

**1.31.2.14 playerDisconnected()** `static GameStatus playerDisconnected (   
 PlayerColor disconnectedPlayer,   
 boolean isVsAi ) [static]`

Player disconnected game status.

**Parameters**

<i>disconnectedPlayer</i>	the disconnected player
<i>isVsAi</i>	the is vs ai

**Returns**

the game status

**1.31.2.15 playerResigned()** `static GameStatus playerResigned (   
 PlayerColor resignedPlayer ) [static]`

Player resigned game status.

**Parameters**

<i>resignedPlayer</i>	the resigned player
-----------------------	---------------------



**Returns**

the game status

**1.31.2.16 serverStoppedGame()** `static GameStatus serverStoppedGame (`  
`String cause ) [static]`

Server stopped game game status.

**Parameters**

<i>cause</i>	the cause
--------------	-----------

**Returns**

the game status

**1.31.2.17 setCustomStr()** `void setCustomStr (`  
`String customStr )`

Sets custom str.

**Parameters**

<i>customStr</i>	the custom str
------------------	----------------

**1.31.2.18 setDepth()** `void setDepth (`  
`int depth )`

Sets depth.

**Parameters**

<i>depth</i>	the depth
--------------	-----------

**1.31.2.19 setInCheck()** `void setInCheck (`  
`Location checkedKingLoc )`

Sets in check.

**Parameters**

<i>checkedKingLoc</i>	the checked king loc
-----------------------	----------------------

**1.31.2.20 stalemate()** `static GameState stalemate ( ) [static]`

Stalemate game status.

**Returns**

the game status

**1.31.2.21 threeFoldRepetition()** `static GameState threeFoldRepetition ( ) [static]`

Three fold repetition game status.

**Returns**

the game status

**1.31.2.22 tieByAgreement()** `static GameState tieByAgreement ( ) [static]`

Tie by agreement game status.

**Returns**

the game status

**1.31.2.23 timedOut()** `static GameState timedOut (`  
    **PlayerColor** *timedOutPlayer*,  
    boolean *isSufficientMaterial* ) `[static]`

Timed out game status.

**Parameters**

<i>timedOutPlayer</i>	the timed out player
<i>isSufficientMaterial</i>	the is sufficient material

**Returns**

the game status

**1.31.2.24 toString()** *String toString ( )*

To string string.

**Returns**

the string

**1.32 GameState.GameStatusType Enum Reference****Public Member Functions**

- **GameStateType** (String **annotation**)
- **GameStateType** (String **annotation**, String **gameOverStr**)
- boolean **isGameOver** ( )

**Public Attributes**

- **TIE**
- **CHECK**
- **GAME\_GOES\_ON**
- **WIN\_OR\_LOSS**
- **UNFINISHED**
- final String **annotation**
- final String **gameOverStr**

**1.32.1 Detailed Description**

Game status type .

**Author**

Bezalel Avrahami ( bezalel3250@gmail.com)

**1.32.2 Constructor & Destructor Documentation****1.32.2.1 GameStateType()** [1/2] **GameStateType** ( *String annotation* )

Instantiates a new Game status type.

**Parameters**

<i>annotation</i>	the annotation
-------------------	----------------

**1.32.2.2 GameStateType()** [2/2]    **GameStateType** (  
    String *annotation*,  
    String *gameOverStr* )

Instantiates a new Game status type.

**Parameters**

<i>annotation</i>	the annotation
<i>gameOverStr</i>	the game over str

**1.32.3 Member Function Documentation**

**1.32.3.1 isGameOver()**    boolean isGameOver ( )

Is game over.

**Returns**

    true if is game over. false otherwise

**1.32.4 Member Data Documentation**

**1.32.4.1 annotation**    final String annotation

The game status annotation.

**1.32.4.2 CHECK**    CHECK

Check game status type.

**1.32.4.3 GAME\_GOES\_ON**    GAME\_GOES\_ON

Game goes on game status type.

**1.32.4.4 gameOverStr** `final String gameOverStr`

game over str

**1.32.4.5 TIE** `TIE`

Tie game status type.

**1.32.4.6 UNFINISHED** `UNFINISHED`

Unfinished game status type.

**1.32.4.7 WIN\_OR\_LOSS** `WIN_OR_LOSS`

Win or loss game status type.

## 1.33 HandledThread

### Public Member Functions

- **HandledThread** ()
- **HandledThread** (ThrowingRunnable runnable)
- void **setRunnable** (ThrowingRunnable runnable)

### Static Public Member Functions

- static **HandledThread runInHandledThread** (ThrowingRunnable runnable)

### Protected Member Functions

- void **handledRun** () throws Throwable

### 1.33.1 Detailed Description

The type Handled thread.

### 1.33.2 Constructor & Destructor Documentation

**1.33.2.1 HandledThread()** [1/2] `HandledThread ( )`

Instantiates a new Handled thread.

**1.33.2.2 HandledThread()** [2/2] `HandledThread (   
 ThrowingRunnable runnable )`

Instantiates a new Handled thread.

**Parameters**

<i>runnable</i>	the runnable
-----------------	--------------

**1.33.3 Member Function Documentation****1.33.3.1 handledRun()** `void handledRun ( ) throws Throwable [protected]`

Handled run.

**Exceptions**

<i>Throwable</i>	the throwable
------------------	---------------

Reimplemented from **MyThread** (p. 169).

**1.33.3.2 runInHandledThread()** `static HandledThread runInHandledThread (   
 ThrowingRunnable runnable ) [static]`

Run in handled thread handled thread.

**Parameters**

<i>runnable</i>	the runnable
-----------------	--------------

**Returns**

the handled thread

**1.33.3.3 setRunnable()** `void setRunnable (   
 ThrowingRunnable runnable )`

Sets runnable.

**Parameters**

<i>runnable</i>	the runnable
-----------------	--------------

## 1.34 MyJFrame.MyAdapter.HeldDown

### Public Member Functions

- void **startPress** ()
- void **endPress** ()
- int **key** ()

#### 1.34.1 Detailed Description

The interface Held down.

#### 1.34.2 Member Function Documentation

##### 1.34.2.1 **endPress()** void endPress ( )

End press.

##### 1.34.2.2 **key()** int key ( )

Key int.

##### Returns

the int

##### 1.34.2.3 **startPress()** void startPress ( )

Start press.

## 1.35 IDsGenerator

### Public Member Functions

- synchronized String **generate** ()
- boolean **canUseId** (String id)

#### 1.35.1 Detailed Description

IDs generator.

##### Author

Bezalel Avrahami ( bezalel3250@gmail.com)

#### 1.35.2 Member Function Documentation

##### 1.35.2.1 **canUseId()** boolean canUseId (String id )

Can use id boolean.

**Parameters**

<i>id</i>	the id
-----------	--------

**Returns**

the boolean

**1.35.2.2 generate()** `synchronized String generate ( )`

Generate id.

**Returns**

the string

**1.36 StrUtils.IterationThingy< T >****Public Member Functions**

- **T iteration** (int i, boolean isLast)

**1.36.1 Detailed Description**

Iteration thingy.

**Author**

Bezalel Avrahami ( `bezalel3250@gmail.com` )

**1.36.2 Member Function Documentation****1.36.2.1 iteration()** `T iteration (`  
`int i,`  
`boolean isLast )`

Iteration t.

**Parameters**

<i>i</i>	the
<i>isLast</i>	the is last



## Returns

the t

## 1.37 Location Enum Reference

### Public Member Functions

- **Location** ()
- String **matrixStr** ()
- String **getColString** ()
- **Location** **flip** ()
- boolean **isBlackSquare** ()
- boolean **isWhiteSquare** ()
- String **toString** ()
- int **getMaxDistance** ( **Location** other)
- String **getRowString** ()
- int **hash** ( **PieceType** pieceType)

### Static Public Member Functions

- static String **locsMatStr** ()
- static **Location**[][] **locsMat** ()
- static String **createMatIndicesStr** ()
- static **Location** **getLoc** ( **Location** loc, **Direction** direction)
- static **Location** **getLoc** ( **Location** loc, int numOfMult, **Direction** direction)
- static **Location** **getLoc** ( **Location** loc, int add)
- static **Location** **getLoc** (int locIndex)
- static **Location** **valueOf** (int locIndex)
- static **Location** **getLoc** (String str)
- static **Location** **getLoc** (int row, int col)
- static **Location** **getLoc** (int row, int col, boolean flip)
- static int **flip** (int num)
- static boolean **isInBounds** ( **Location** loc)

### Public Attributes

- final long **asLong**
- final int **asInt**
- final int **row**
- final int **col**

#### 1.37.1 Detailed Description

**Location** (p.97) - an enum consisting of 64 values representing all 64 squares on the board. used to access squares on the board  
an enum is used for performance reasons.

#### Author

Bezalel Avrahami ( bezalel13250@gmail.com)

## 1.37.2 Constructor & Destructor Documentation

### 1.37.2.1 `Location()` `Location ( )`

Instantiates a new **Location** (p. 97).

## 1.37.3 Member Function Documentation

### 1.37.3.1 `createMatIndicesStr()` `static String createMatIndicesStr ( )` `[static]`

Create mat indices str string.

#### Returns

the string

### 1.37.3.2 `flip()` `[1/2]` `Location flip ( )`

Flip location.

#### Returns

the location

### 1.37.3.3 `flip()` `[2/2]` `static int flip (` `int num )` `[static]`

Flip int.

#### Parameters

<i>num</i>	the num
------------	---------

#### Returns

the int

**1.37.3.4 getColString()** `String getColString ( )`

Gets col string.

**Returns**

the col string

**1.37.3.5 getLoc()** [1/7] `static Location getLoc (`  
`int locIndex ) [static]`

Gets location corresponding to the locIndex provided (0..63)

**Parameters**

<i>locIndex</i>	the locIndex
-----------------	--------------

**Returns**

the location if the provided index is inside the bounds(0...63). null otherwise

**1.37.3.6 getLoc()** [2/7] `static Location getLoc (`  
`int row,`  
`int col ) [static]`

Gets loc.

**Parameters**

<i>row</i>	the row
<i>col</i>	the col

**Returns**

the loc

**1.37.3.7 getLoc()** [3/7] `static Location getLoc (`  
`int row,`  
`int col,`  
`boolean flip ) [static]`

Gets loc.

**Parameters**

<i>row</i>	the row
<i>col</i>	the col
<i>flip</i>	the flip

**Returns**

the loc

```
1.37.3.8  getLoc() [4/7]  static Location getLoc (
    Location loc,
    Direction direction )  [static]
```

Gets the location relative to loc in the direction

**Parameters**

<i>loc</i>	the loc
<i>direction</i>	the direction

**Returns**

the location if the calculated index is inside the bounds(0...63). null otherwise

```
1.37.3.9  getLoc() [5/7]  static Location getLoc (
    Location loc,
    int add )  [static]
```

Gets the location that is exactly add squares from loc NOTE: add should be in bitboard format

**Parameters**

<i>loc</i>	the loc
<i>add</i>	the number of squares to add

**Returns**

the location if the calculated index is inside the bounds(0...63). null otherwise

**See also**

ver14.Model.Bitboard

**1.37.3.10 getLoc()** [6/7] `static Location getLoc (`  
    `Location loc,`  
    `int numOfMult,`  
    `Direction direction ) [static]`

Gets the location relative to loc in the direction given and the distance is determined by the numOfMult

#### Parameters

<i>loc</i>	the loc
<i>numOfMult</i>	the num of mult
<i>direction</i>	the direction

#### Returns

the location if the calculated index is inside the bounds(0...63). null otherwise

**1.37.3.11 getLoc()** [7/7] `static Location getLoc (`  
    `String str ) [static]`

gets the location corresponding to the provided location string.

#### Parameters

<i>str</i>	the square's coordinate according to the Algebraic notation
------------	---

#### Returns

the loc if given str is valid. null otherwise

#### See also

...

**1.37.3.12 getMaxDistance()** `int getMaxDistance (`  
    `Location other )`

Gets max distance.

#### Parameters

<i>other</i>	the other
--------------	-----------

**Returns**

the max distance

**1.37.3.13 getRowString()** `String getRowString ( )`

Gets row string.

**Returns**

the row string

**1.37.3.14 hash()** `int hash (   
 PieceType pieceType )`

Hash int.

**Parameters**

<i>pieceType</i>	the piece type
------------------	----------------

**Returns**

the int

**1.37.3.15 isBlackSquare()** `boolean isBlackSquare ( )`

Is black square boolean.

**Returns**

the boolean

**1.37.3.16 isInBounds()** `static boolean isInBounds (   
 Location loc ) [static]`

Is in bounds boolean.

**Parameters**

<i>loc</i>	the loc
------------	---------

**Returns**

the boolean

**1.37.3.17 isWhiteSquare()** `boolean isWhiteSquare ( )`

Is white square boolean.

**Returns**

the boolean

**1.37.3.18 locsMat()** `static Location[][] locsMat ( ) [static]`

Locs mat location [ ] [ ].

**Returns**

the location [ ] [ ]

**1.37.3.19 locsMatStr()** `static String locsMatStr ( ) [static]`

Locs mat str string.

**Returns**

the string

**1.37.3.20 matrixStr()** `String matrixStr ( )`

Matrix str string.

**Returns**

the string

**1.37.3.21 toString()** `String toString ( )`

To string string.

**Returns**

the string

**1.37.3.22 valueOf()** `static Location valueOf (   
int locIndex ) [static]`

Value of location.

**Parameters**

<i>locIndex</i>	the locIndex
-----------------	--------------

**Returns**

the location if the provided index is inside the bounds(0...63). null otherwise

**1.37.4 Member Data Documentation****1.37.4.1 asInt** `final int asInt`

the calculated index

**1.37.4.2 asLong** `final long asLong`

a long value with a bit set on this location

**1.37.4.3 col** `final int col`

this location's column

**1.37.4.4 row** `final int row`

this location's row

**1.38 LoginInfo****Public Member Functions**

- **LoginInfo** ()
- **LoginInfo** (String username, String password, **LoginType** loginType)
- **LoginInfo** ( **LoginType** loginType)
- String **getProfilePic** ()
- void **setProfilePic** (String profilePic)
- boolean **isGuest** ()
- **LoginType** **getLoginType** ()
- void **setLoginType** ( **LoginType** loginType)
- String  **()**
- void **setUsername** (String username)
- String **getPassword** ()
- void **setPassword** (String password)
- String **toString** ()
- boolean **asUser** ()
- void **initDebugLoginValues** ()



### 1.38.1 Detailed Description

Login info.

Author

Bezalel Avrahami ( bezalel3250@gmail.com)

### 1.38.2 Constructor & Destructor Documentation

#### 1.38.2.1 LoginInfo() [1/3] LoginInfo ( )

Instantiates a new Login info.

#### 1.38.2.2 LoginInfo() [2/3] LoginInfo (

```
String username,  
String password,  
LoginType loginType )
```

Instantiates a new Login info.

Parameters

<i>username</i>	the username
<i>password</i>	the password
<i>loginType</i>	the login type

#### 1.38.2.3 LoginInfo() [3/3] LoginInfo (

```
LoginType loginType )
```

Instantiates a new Login info.

Parameters

<i>loginType</i>	the login type
------------------	----------------

### 1.38.3 Member Function Documentation

#### 1.38.3.1 asUser() boolean asUser ( )

As user boolean.

**Returns**

the boolean

**1.38.3.2 getLoginType()** `LoginType getLoginType ( )`

Gets login type.

**Returns**

the login type

**1.38.3.3 getPassword()** `String getPassword ( )`

Gets password.

**Returns**

the password

**1.38.3.4 getProfilePic()** `String getProfilePic ( )`

Gets profile pic.

**Returns**

the profile pic

**1.38.3.5 getUsername()** `String getUsername ( )`

Gets username.

**Returns**

the username

**1.38.3.6 initDebugLoginValues()** `void initDebugLoginValues ( )`

Init debug login values.

**1.38.3.7 isGuest()** `boolean isGuest ( )`

Is guest boolean.

**Returns**

the boolean

**1.38.3.8 setLoginType()** `void setLoginType (   
LoginType loginType )`

Sets login type.

## Parameters

<i>loginType</i>	the login type
------------------	----------------

**1.38.3.9 setPassword()** `void setPassword (`  
`String password )`

Sets password.

## Parameters

<i>password</i>	the password
-----------------	--------------

**1.38.3.10 setProfilePic()** `void setProfilePic (`  
`String profilePic )`

Sets profile pic.

## Parameters

<i>profilePic</i>	the profile pic
-------------------	-----------------

**1.38.3.11 setUsername()** `void setUsername (`  
`String username )`

Sets username.

## Parameters

<i>username</i>	the username
-----------------	--------------

**1.38.3.12 toString()** `String toString ( )`

To string string.

## Returns

the string

## 1.39 LoginType Enum Reference

### Public Member Functions

- boolean **asUser** ()
- String **toString** ()

### Public Attributes

- LOGIN
- REGISTER
- GUEST
- CANCEL
- NOT\_SET\_YET

#### 1.39.1 Detailed Description

Login Type.

#### Author

Bezalel Avrahami ( bezalel3250@gmail.com)

#### 1.39.2 Member Function Documentation

##### 1.39.2.1 **asUser()** `boolean asUser ( )`

As user boolean.

#### Returns

the boolean

##### 1.39.2.2 **toString()** `String toString ( )`

To string string.

#### Returns

the string

#### 1.39.3 Member Data Documentation

**1.39.3.1 CANCEL** CANCEL

Cancel.

**1.39.3.2 GUEST** GUEST

Guest.

**1.39.3.3 LOGIN** LOGIN

Login.

**1.39.3.4 NOT\_SET\_YET** NOT\_SET\_YET

Not set yet.

**1.39.3.5 REGISTER** REGISTER

Register.

**1.40 Math Enum Reference****Public Member Functions**

- **Col execute** ( **Col col**, Object value)
- **Col execute** ( **Col col**, Object value, boolean changeSelf)
- String **strSource** ()

**Static Public Member Functions**

- static String **nullIf0** (Object val)
- static String **formatNum** (Object num)
- static String **strVal** (Object val)
- static String **formatNum** (Object num, String format)
- static String **str** (Object obj)
- static String **asFloat** (Object num)

**Public Attributes**

- **Plus**
- **Mult**
- **Div**
- **Col col**

**Protected Member Functions**

- void **zeroIfNull** ()
- abstract void **apply** (Object value)
- void **simpleOperation** (Object a, String operation, Object b)

### 1.40.1 Detailed Description

**Math** (p. 109) - allows for math actions on columns and some math-related utilities for columns.

Author

Bezalel Avrahami ( bezalel3250@gmail.com)

### 1.40.2 Member Function Documentation

**1.40.2.1 apply()** `abstract void apply (`  
`Object value ) [abstract], [protected]`

Apply.

Parameters

<i>value</i>	the value
--------------	-----------

**1.40.2.2 asFloat()** `static String asFloat (`  
`Object num ) [static]`

As float string.

Parameters

<i>num</i>	the num
------------	---------

Returns

the string

**1.40.2.3 execute()** [1/2] `col execute (`  
`col col,`  
`Object value )`

Execute col.

Parameters

<i>col</i>	the col
<i>value</i>	the value

**Returns**

the col

**1.40.2.4 execute()** [2/2] `Col execute (`  
    `Col col,`  
    `Object value,`  
    `boolean changeSelf )`

Execute col.

**Parameters**

<i>col</i>	the col
<i>value</i>	the value
<i>changeSelf</i>	the change self

**Returns**

the col

**1.40.2.5 formatNum()** [1/2] `static String formatNum (`  
    `Object num ) [static]`

Format num string.

**Parameters**

<i>num</i>	the num
------------	---------

**Returns**

the string

**1.40.2.6 formatNum()** [2/2] `static String formatNum (`  
    `Object num,`  
    `String format ) [static]`

Format num string.

**Parameters**

<i>num</i>	the num
<i>format</i>	the format

**Returns**

the string

**1.40.2.7 nullIf0()** `static String nullIf0 (`  
`Object val ) [static]`

Null if 0 string.

**Parameters**

<i>val</i>	the val
------------	---------

**Returns**

the string

**1.40.2.8 simpleOperation()** `void simpleOperation (`  
`Object a,`  
`String operation,`  
`Object b ) [protected]`

Simple operation.

**Parameters**

<i>a</i>	the a
<i>operation</i>	the operation
<i>b</i>	the b

**1.40.2.9 str()** `static String str (`  
`Object obj ) [static]`

Str string.

**Parameters**

<i>obj</i>	the obj
------------	---------

**Returns**

the string



**1.40.2.10 strSource()** `String strSource ( )`

Str source string.

**Returns**

the string

**1.40.2.11 strVal()** `static String strVal (   
Object val ) [static]`

Str val string.

**Parameters**

<code>val</code>	the val
------------------	---------

**Returns**

the string

**1.40.2.12 zeroIfNull()** `void zeroIfNull ( ) [protected]`

Zero if null.

**1.40.3 Member Data Documentation****1.40.3.1 col** `Col col`

The **Col** (p.33).

**1.40.3.2 Div** `Div`

The Div.

**1.40.3.3 Mult** `Mult`

The Mult.

#### 1.40.3.4 Plus Plus

The Plus.

### 1.41 Message

#### Public Member Functions

- **Message** (MessageType messageType)
- **Message** (MessageType messageType, String subject, String respondingToMsgId)
- **Message** (MessageType messageType, **Message** respondingTo)
- **Message** (MessageType messageType, String subject)
- void **setDbRequest** ( **DBRequest** dbRequest)
- **MyError** **getError** ()
- void **setError** ( **MyError** error)
- **DBRequest** **getDBRequest** ()
- ArrayList< String > **getUsernameSuggestions** ()
- void **setUsernameSuggestions** (ArrayList< String > usernameSuggestions)
- **DBResponse** **getDBResponse** ()
- void **setDBResponse** ( **DBResponse** requestedStats)
- Stack< **Move** > **getMoveStack** ()
- void **setMoveStack** (Stack< **Move** > moveStack)
- SyncedListItems<?>[] **getSyncedLists** ()
- void **setSyncedLists** (SyncedListItems<?>... syncedLists)
- String **getRespondingToMsgId** ()
- void **setRespondingToMsgId** (String respondingToMsgId)
- void **setRespondingTo** ( **Message** msg)
- boolean **isResponse** ()
- String **getOtherPlayer** ()
- void **setOtherPlayer** (String otherPlayer)
- **Question** **getQuestion** ()
- void **setQuestion** ( **Question** question)
- Boolean **getAvailable** ()
- void **setAvailable** (Boolean available)
- String **getUsername** ()
- void **setUsername** (String username)
- GameSettings **getGameSettings** ()
- void **setGameSettings** (GameSettings gameSettings)
- ArrayList< **Move** > **getPreMoves** ()
- void **setPreMoves** (ArrayList< **Move** > preMoves)
- String **getSubject** ()
- **PlayerColor** **getPlayerColor** ()
- void **setPlayerColor** ( **PlayerColor** playerColor)
- **Move** **getMove** ()
- void **setMove** ( **Move** move)
- GameTime **getGameTime** ()
- void **setGameTime** (GameTime gameTime)
- **GameStatus** **getGameStatus** ()
- void **setGameStatus** ( **GameStatus** gameStatus)
- MovesList **getPossibleMoves** ()
- void **setPossibleMoves** (MovesList possibleMoves)
- **Board** **getBoard** ()
- void **setBoard** ( **Board** board)
- **LoginInfo** **getLoginInfo** ()
- void **setLoginInfo** ( **LoginInfo** loginInfo)
- MessageType **getMessageType** ()
- boolean **isSubject** ()

### Static Public Member Functions

- static **Message** **askForLogin** ()
- static **Message** **returnLogin** ( **LoginInfo** loginInfo, **Message** respondingTo)
- static **Message** **welcomeMessage** (String str, **LoginInfo** loginInfo)
- static **Message** **initGame** ( **Board** board, String opponent, **PlayerColor** player, **GameTime** gameTime, Stack< **Move** > moveStack)
- static **Message** **throwError** ( **MyError** error)
- static **Message** **updateByMove** ( **Move** move, **GameTime** gameTime)
- static **Message** **waitForYourTurn** (String waitingForName, **GameTime** gameTime)
- static **Message** **waitForMatch** ()
- static **Message** **gameOver** ( **GameStatus** gameStatus)
- static **Message** **askForGameSettings** (SyncedItems joinableGames, SyncedItems resumableGames)
- static **Message** **returnGameSettings** (GameSettings gameSettings, **Message** respondingTo)
- static **Message** **interrupt** ()
- static **Message** **bye** (String subject)
- static **Message** **error** (String err)
- static **Message** **returnMove** ( **Move** move, **Message** respondingTo)
- static **Message** **answerQuestion** ( **Question.Answer** answer, **Message** respondingTo)
- static **Message** **askQuestion** ( **Question** question)
- static **Message** **checkUsernameAvailability** (String username)
- static **Message** **returnUsernameNotAvailable** (ArrayList< String > usernameSuggestions, **Message** request)
- static **Message** **returnUsernameAvailable** ( **Message** request)
- static **Message** **dbRequest** ( **DBRequest** dbRequest)
- static **Message** **returnDBResponse** ( **DBResponse** DBResponse, **Message** request)
- static **Message** **syncLists** (SyncedItems<?>... syncedLists)

### Public Attributes

- final String **messageID**

#### 1.41.1 Detailed Description

**1.41.1.1 Message.** by Ilan Peretz ( ilanperets@gmail.com) 10/11/2021

#### 1.41.2 Constructor & Destructor Documentation

**1.41.2.1 Message()** [1/4] **Message** (   
 *MessageType messageType* )

Instantiates a new **Message** (p. 114).

##### Parameters

<i>messageType</i>	the message type
--------------------	------------------

**1.41.2.2 Message()** [2/4]    **Message** (  
    *MessageType messageType*,  
    *String subject*,  
    *String respondingToMsgId* )

Instantiates a new **Message** (p. 114).

**Parameters**

<i>messageType</i>	the message type
<i>subject</i>	the subject
<i>respondingTo↔ MsgId</i>	the responding to msg id

**1.41.2.3 Message()** [3/4]    **Message** (  
    *MessageType messageType*,  
    **Message** *respondingTo* )

Instantiates a new **Message** (p. 114).

**Parameters**

<i>messageType</i>	the message type
<i>respondingTo</i>	the responding to

**1.41.2.4 Message()** [4/4]    **Message** (  
    *MessageType messageType*,  
    *String subject* )

Instantiates a new **Message** (p. 114).

**Parameters**

<i>messageType</i>	the message type
<i>subject</i>	the subject

### 1.41.3 Member Function Documentation

**1.41.3.1 answerQuestion()** `static Message answerQuestion (`  
    `Question.Answer answer,`  
    `Message respondingTo ) [static]`

Answer question message.

Parameters

<i>respondingTo</i>	the responding to
---------------------	-------------------

Returns

the message

**1.41.3.2 askForGameSettings()** `static Message askForGameSettings (`  
    `SyncedItems joinableGames,`  
    `SyncedItems resumableGames ) [static]`

Ask for game settings message.

Parameters

<i>joinableGames</i>	the joinable games
<i>resumableGames</i>	the resumable games

Returns

the message

**1.41.3.3 askForLogin()** `static Message askForLogin ( ) [static]`

Ask for login message.

Returns

the message

**1.41.3.4 askQuestion()** `static Message askQuestion (`  
    `Question question ) [static]`

Ask question message.

**Parameters**

<i>question</i>	the question
-----------------	--------------

**Returns**

the message

**1.41.3.5** **bye()** `static Message bye (`  
`String subject ) [static]`

Bye message.

**Parameters**

<i>subject</i>	the subject
----------------	-------------

**Returns**

the message

**1.41.3.6** **checkUsernameAvailability()** `static Message checkUsernameAvailability (`  
`String username ) [static]`

Check username availability message.

**Parameters**

<i>username</i>	the username
-----------------	--------------

**Returns**

the message

**1.41.3.7** **dbRequest()** `static Message dbRequest (`  
`DBRequest dbRequest ) [static]`

Db request message.

**Parameters**

<i>dbRequest</i>	the db request
------------------	----------------

**Returns**

the message

**1.41.3.8 error()** `static Message error (`  
`String err ) [static]`

Error message.

**Parameters**

<i>err</i>	the err
------------	---------

**Returns**

the message

**1.41.3.9 gameOver()** `static Message gameOver (`  
`GameStatus gameStatus ) [static]`

Game over message.

**Parameters**

<i>gameStatus</i>	the game status
-------------------	-----------------

**Returns**

the message

**1.41.3.10 getAvailable()** `Boolean getAvailable ( )`

Gets available.

**Returns**

the available

**1.41.3.11 getBoard()** `Board getBoard ( )`

Gets board.

**Returns**

the board

**1.41.3.12 getDBRequest()** `DBRequest getDBRequest ( )`

Gets db request.

**Returns**

the db request

**1.41.3.13 getDBResponse()** `DBResponse getDBResponse ( )`

Gets db response.

**Returns**

the db response

**1.41.3.14 getError()** `MyError getError ( )`

Gets error.

**Returns**

the error

**1.41.3.15 getGameSettings()** `GameSettings getGameSettings ( )`

Gets game settings.

**Returns**

the game settings



**1.41.3.16** **getGameStatus()** `GameStatus` `getGameStatus ( )`

Gets game status.

**Returns**

the game status

**1.41.3.17** **getGameTime()** `GameTime` `getGameTime ( )`

Gets game time.

**Returns**

the game time

**1.41.3.18** **getLoginInfo()** `LoginInfo` `getLoginInfo ( )`

Gets login info.

**Returns**

the login info

**1.41.3.19** **getMessageType()** `MessageType` `getMessageType ( )`

Gets message type.

**Returns**

the message type

**1.41.3.20** **getMove()** `Move` `getMove ( )`

Gets move.

**Returns**

the move

**1.41.3.21 getMoveStack()** `Stack< Move > getMoveStack ( )`

Gets move stack.

Returns

the move stack

**1.41.3.22 getOtherPlayer()** `String getOtherPlayer ( )`

Gets other player.

Returns

the other player

**1.41.3.23 getPlayerColor()** `PlayerColor getPlayerColor ( )`

Gets player color.

Returns

the player color

**1.41.3.24 getPossibleMoves()** `MovesList getPossibleMoves ( )`

Gets possible moves.

Returns

the possible moves

**1.41.3.25 getPreMoves()** `ArrayList< Move > getPreMoves ( )`

Gets pre moves.

Returns

the pre moves

**1.41.3.26** `getQuestion()` `Question` `getQuestion ( )`

Gets question.

**Returns**

the question

**1.41.3.27** `getRespondingToMsgId()` `String` `getRespondingToMsgId ( )`

Gets responding to msg id.

**Returns**

the responding to msg id

**1.41.3.28** `getSubject()` `String` `getSubject ( )`

Gets subject.

**Returns**

the subject

**1.41.3.29** `getSyncedList()` `SyncedListItems<?>[]` `getSyncedList ( )`

Get synced lists synced items [ ].

**Returns**

the synced items [ ]

**1.41.3.30** `getUsername()` `String` `getUsername ( )`

Gets username.

**Returns**

the username

**1.41.3.31 getUsernameSuggestions()** `ArrayList< String > getUsernameSuggestions ( )`

Gets username suggestions.

**Returns**

the username suggestions

**1.41.3.32 initGame()** `static Message initGame (`  
    `Board board,`  
    `String opponent,`  
    `PlayerColor player,`  
    `GameTime gameTime,`  
    `Stack< Move > moveStack ) [static]`

Init game message.

**Parameters**

<i>board</i>	the board
<i>opponent</i>	the opponent
<i>player</i>	the player
<i>gameTime</i>	the game time
<i>moveStack</i>	the move stack

**Returns**

the message

**1.41.3.33 interrupt()** `static Message interrupt ( ) [static]`

Interrupt message.

**Returns**

the message

**1.41.3.34 isResponse()** `boolean isResponse ( )`

Is response boolean.

**Returns**

the boolean

**1.41.3.35 isSubject()** `boolean isSubject ( )`

Is subject boolean.

**Returns**

the boolean

**1.41.3.36 returnDBResponse()** `static Message returnDBResponse ( DBResponse DBResponse, Message request ) [static]`

Return db response message.

**Parameters**

<i>DBResponse</i>	the db response
<i>request</i>	the request

**Returns**

the message

**1.41.3.37 returnGameSettings()** `static Message returnGameSettings ( GameSettings gameSettings, Message respondingTo ) [static]`

Return game settings message.

**Parameters**

<i>gameSettings</i>	the game settings
<i>respondingTo</i>	the responding to

**Returns**

the message

**1.41.3.38 returnLogin()** `static Message returnLogin ( LoginInfo loginInfo, Message respondingTo ) [static]`

Return login message.

**Parameters**

<i>loginInfo</i>	the login info
<i>respondingTo</i>	the responding to

**Returns**

the message

```
1.41.3.39 returnMove() static Message returnMove (
    Move move,
    Message respondingTo ) [static]
```

Return move message.

**Parameters**

<i>move</i>	the move
<i>respondingTo</i>	the responding to

**Returns**

the message

```
1.41.3.40 returnUsernameAvailable() static Message returnUsernameAvailable (
    Message request ) [static]
```

Return username available message.

**Parameters**

<i>request</i>	the request
----------------	-------------

**Returns**

the message

```
1.41.3.41 returnUsernameNotAvailable() static Message returnUsernameNotAvailable (
    ArrayList< String > usernameSuggestions,
    Message request ) [static]
```

Return username not available message.

## Parameters

<i>usernameSuggestions</i>	the username suggestions
<i>request</i>	the request

## Returns

the message

**1.41.3.42 setAvailable()** `void setAvailable ( Boolean available )`

Sets available.

## Parameters

<i>available</i>	the available
------------------	---------------

**1.41.3.43 setBoard()** `void setBoard ( Board board )`

Sets board.

## Parameters

<i>board</i>	the board
--------------	-----------

**1.41.3.44 setDbRequest()** `void setDbRequest ( DBRequest dbRequest )`

Sets db request.

## Parameters

<i>dbRequest</i>	the db request
------------------	----------------

**1.41.3.45 setDBResponse()** `void setDBResponse ( DBResponse requestedStats )`

Sets db response.

**Parameters**

<i>requestedStats</i>	the requested stats
-----------------------	---------------------

**1.41.3.46 setError()** `void setError (   
          MyError error )`

Sets error.

**Parameters**

<i>error</i>	the error
--------------	-----------

**1.41.3.47 setGameSettings()** `void setGameSettings (   
          GameSettings gameSettings )`

Sets game settings.

**Parameters**

<i>gameSettings</i>	the game settings
---------------------	-------------------

**1.41.3.48 setGameStatus()** `void setGameStatus (   
          GameStatus gameStatus )`

Sets game status.

**Parameters**

<i>gameStatus</i>	the game status
-------------------	-----------------

**1.41.3.49 setGameTime()** `void setGameTime (   
          GameTime gameTime )`

Sets game time.

**Parameters**

<i>gameTime</i>	the game time
-----------------	---------------



**1.41.3.50 setLoginInfo()** `void setLoginInfo (`  
    **LoginInfo** *loginInfo* )

Sets login info.

Parameters

<i>loginInfo</i>	the login info
------------------	----------------

**1.41.3.51 setMove()** `void setMove (`  
    **Move** *move* )

Sets move.

Parameters

<i>move</i>	the move
-------------	----------

**1.41.3.52 setMoveStack()** `void setMoveStack (`  
    Stack< **Move** > *moveStack* )

Sets move stack.

Parameters

<i>moveStack</i>	the move stack
------------------	----------------

**1.41.3.53 setOtherPlayer()** `void setOtherPlayer (`  
    String *otherPlayer* )

Sets other player.

Parameters

<i>otherPlayer</i>	the other player
--------------------	------------------

**1.41.3.54 setPlayerColor()** `void setPlayerColor (`  
    **PlayerColor** *playerColor* )

Sets player color.

Parameters

<i>playerColor</i>	the player color
--------------------	------------------

**1.41.3.55 setPossibleMoves()** `void setPossibleMoves ( MovesList possibleMoves )`

Sets possible moves.

Parameters

<i>possibleMoves</i>	the possible moves
----------------------	--------------------

**1.41.3.56 setPreMoves()** `void setPreMoves ( ArrayList< Move > preMoves )`

Sets pre moves.

Parameters

<i>preMoves</i>	the pre moves
-----------------	---------------

**1.41.3.57 setQuestion()** `void setQuestion ( Question question )`

Sets question.

Parameters

<i>question</i>	the question
-----------------	--------------

**1.41.3.58 setRespondingTo()** `void setRespondingTo ( Message msg )`

Sets responding to.

## Parameters

<i>msg</i>	the msg
------------	---------

**1.41.3.59 setRespondingToMsgId()** void setRespondingToMsgId (   
String *respondingToMsgId* )

Sets responding to msg id.

## Parameters

<i>respondingToMsgId</i>	the responding to msg id
--------------------------	--------------------------

**1.41.3.60 setSyncedList()** void setSyncedList (   
SyncedList<?>... *syncedList* )

Sets synced lists.

## Parameters

<i>syncedList</i>	the synced lists
-------------------	------------------

**1.41.3.61 setUsername()** void setUsername (   
String *username* )

Sets username.

## Parameters

<i>username</i>	the username
-----------------	--------------

**1.41.3.62 setUsernameSuggestions()** void setUsernameSuggestions (   
ArrayList< String > *usernameSuggestions* )

Sets username suggestions.

## Parameters

<i>usernameSuggestions</i>	the username suggestions
----------------------------	--------------------------

**1.41.3.63 syncLists()** `static Message syncLists (`  
    `SyncedListItems<?>... syncedLists ) [static]`

Sync lists message.

**Parameters**

<i>syncedListItems</i>	the synced lists
------------------------	------------------

**Returns**

the message

**1.41.3.64 throwError()** `static Message throwError (`  
    `MyError error ) [static]`

Throw error message.

**Parameters**

<i>error</i>	if null interrupts
--------------	--------------------

**Returns**

the message

**1.41.3.65 updateByMove()** `static Message updateByMove (`  
    `Move move,`  
    `GameTime gameTime ) [static]`

Update by move message.

**Parameters**

<i>move</i>	the move
<i>gameTime</i>	the game time

**Returns**

the message

**1.41.3.66 waitForMatch()** `static Message waitForMatch ( ) [static]`

Wait for match message.

**Returns**

the message

**1.41.3.67 waitForYourTurn()** `static Message waitForYourTurn (`  
    `String waitingForName,`  
    `GameTime gameTime ) [static]`

Wait for your turn message.

**Parameters**

<i>waitingForName</i>	the waiting for name
<i>gameTime</i>	the game time

**Returns**

the message

**1.41.3.68 welcomeMessage()** `static Message welcomeMessage (`  
    `String str,`  
    `LoginInfo loginInfo ) [static]`

Welcome message message.

**Parameters**

<i>str</i>	the str
<i>loginInfo</i>	the login info

**Returns**

the message

## 1.41.4 Member Data Documentation

**1.41.4.1 messageID** `final String messageID`

The **Message** (p. 114) id.

## 1.42 MessageCallback

### Public Member Functions

- void **onMsg** ( **Message** message)

#### 1.42.1 Detailed Description

Message callback - a message callback.

#### Author

Bezalel Avrahami ( bezalel3250@gmail.com)

#### 1.42.2 Member Function Documentation

**1.42.2.1 onMsg()** void onMsg (   
 **Message** message )

On msg.

#### Parameters

<i>message</i>	the message
----------------	-------------

## 1.43 MessagesHandler

### Public Member Functions

- **MessagesHandler** ( **AppSocket** socket)
- **MessageCallback** **onCancelQuestion** ()
- void **interruptBlocking** ( **MyError** err)
- **Message** **blockTilRes** ( **Message** request)
- void **noBlockRequest** ( **Message** request, **MessageCallback** onRes)
- void **receivedMessage** ( **Message** message)
- void **prepareForDisconnect** ()
- void **onAnyMsg** ( **Message** message)
- final void **onDisconnected** ()
- **MessageCallback** **onLogin** ()
- **MessageCallback** **onResign** ()
- **MessageCallback** **onAddTime** ()
- **MessageCallback** **onOfferDraw** ()
- **MessageCallback** **onWelcomeMessage** ()
- **MessageCallback** **onGetGameSettings** ()
- **MessageCallback** **onWaitForMatch** ()
- **MessageCallback** **onInitGame** ()

- **MessageCallback** **onWaitTurn** ()
- **MessageCallback** **onGetMove** ()
- **MessageCallback** **onUpdateByMove** ()
- **MessageCallback** **onGameOver** ()
- **MessageCallback** **onError** ()
- **MessageCallback** **onQuestion** ()
- **MessageCallback** **onBye** ()
- **MessageCallback** **onUsernameAvailability** ()
- **MessageCallback** **onDBRequest** ()
- **MessageCallback** **onDBResponse** ()
- **MessageCallback** **onUpdateSyncedList** ()
- **MessageCallback** **onInterrupt** ()
- **MessageCallback** **onIsAlive** ()
- **MessageCallback** **onAlive** ()

### Protected Member Functions

- void **onAnyDisconnection** ()
- void **onPlannedDisconnect** ()
- void **onUnplannedDisconnect** ()
- **MyError.DisconnectedError** **createDisconnectedError** ()

### Protected Attributes

- final **AppSocket** **socket**

#### 1.43.1 Detailed Description

The type Messages handler.

Author

Bezael Avrahami ( [bezalel3250@gmail.com](mailto:bezalel3250@gmail.com) )

#### 1.43.2 Constructor & Destructor Documentation

##### 1.43.2.1 MessagesHandler() **MessagesHandler** ( **AppSocket** *socket* )

Instantiates a new Messages handler.

Parameters

<i>socket</i>	the socket
---------------	------------

### 1.43.3 Member Function Documentation

**1.43.3.1 blockTilRes()** `Message blockTilRes (`  
`Message request )`

Block til res message.

#### Parameters

<i>request</i>	the request
----------------	-------------

#### Returns

the message

**1.43.3.2 createDisconnectedError()** `MyError.DisconnectedError createDisconnectedError ( ) [protected]`

Create disconnected error my error . disconnected error.

#### Returns

the my error . disconnected error

**1.43.3.3 interruptBlocking()** `void interruptBlocking (`  
`MyError err )`

Interrupt blocking.

#### Parameters

<i>err</i>	the err
------------	---------

**1.43.3.4 noBlockRequest()** `void noBlockRequest (`  
`Message request,`  
`MessageCallback onRes )`

No block request.



## Parameters

<i>request</i>	the request
<i>onRes</i>	the on res

**1.43.3.5 onAddTime()** **MessageCallback** onAddTime ( )

On add time message callback.

## Returns

the message callback

**1.43.3.6 onAlive()** **MessageCallback** onAlive ( )

On alive message callback.

## Returns

the message callback

**1.43.3.7 onAnyDisconnection()** **void** onAnyDisconnection ( ) [protected]

On any disconnection.

**1.43.3.8 onAnyMsg()** **void** onAnyMsg (   
 **Message** *message* )

On any msg.

## Parameters

<i>message</i>	the message
----------------	-------------

**1.43.3.9 onBye()** **MessageCallback** onBye ( )

On bye message callback.

**Returns**

the message callback

**1.43.3.10 onCancelQuestion()    MessageCallback onCancelQuestion ( )**

On cancel question message callback.

**Returns**

the message callback

**1.43.3.11 onDBRequest()    MessageCallback onDBRequest ( )**

On db request message callback.

**Returns**

the message callback

**1.43.3.12 onDBResponse()    MessageCallback onDBResponse ( )**

On db response message callback.

**Returns**

the message callback

**1.43.3.13 onDisconnected()    final void onDisconnected ( )**

On disconnected.

**1.43.3.14 onError()    MessageCallback onError ( )**

On error message callback.

**Returns**

the message callback

**1.43.3.15 onGameOver()** **MessageCallback** onGameOver ( )

On game over message callback.

**Returns**

the message callback

**1.43.3.16 onGetGameSettings()** **MessageCallback** onGetGameSettings ( )

On get game settings message callback.

**Returns**

the message callback

**1.43.3.17 onGetMove()** **MessageCallback** onGetMove ( )

On get move message callback.

**Returns**

the message callback

**1.43.3.18 onInitGame()** **MessageCallback** onInitGame ( )

On init game message callback.

**Returns**

the message callback

**1.43.3.19 onInterrupt()** **MessageCallback** onInterrupt ( )

On interrupt message callback.

**Returns**

the message callback

**1.43.3.20 onIsAlive()** `MessageCallback onIsAlive ( )`

On is alive message callback.

**Returns**

the message callback

**1.43.3.21 onLogin()** `MessageCallback onLogin ( )`

On login message callback.

**Returns**

the message callback

**1.43.3.22 onOfferDraw()** `MessageCallback onOfferDraw ( )`

On offer draw message callback.

**Returns**

the message callback

**1.43.3.23 onPlannedDisconnect()** `void onPlannedDisconnect ( ) [protected]`

On planned disconnect.

**1.43.3.24 onQuestion()** `MessageCallback onQuestion ( )`

On question message callback.

**Returns**

the message callback

**1.43.3.25 onResign()** **MessageCallback** onResign ( )

On resign message callback.

**Returns**

the message callback

**1.43.3.26 onUnplannedDisconnect()** **void** onUnplannedDisconnect ( ) [protected]

On unplanned disconnect.

**1.43.3.27 onUpdateByMove()** **MessageCallback** onUpdateByMove ( )

On update by move message callback.

**Returns**

the message callback

**1.43.3.28 onUpdateSyncedList()** **MessageCallback** onUpdateSyncedList ( )

On update synced list message callback.

**Returns**

the message callback

**1.43.3.29 onUsernameAvailability()** **MessageCallback** onUsernameAvailability ( )

On username availability message callback.

**Returns**

the message callback

**1.43.3.30 onWaitForMatch()** **MessageCallback** onWaitForMatch ( )

On wait for match message callback.

**Returns**

the message callback

**1.43.3.31 onWaitTurn()** **MessageCallback** onWaitTurn ( )

On wait turn message callback.

**Returns**

the message callback

**1.43.3.32 onWelcomeMessage()** **MessageCallback** onWelcomeMessage ( )

On welcome message message callback.

**Returns**

the message callback

**1.43.3.33 prepareForDisconnect()** **void** prepareForDisconnect ( )

Prepare for disconnect.

**1.43.3.34 receivedMessage()** **void** receivedMessage (   
 **Message** *message* )

Received message.

**Parameters**

<i>message</i>	the message
----------------	-------------

**1.43.4 Member Data Documentation**

**1.43.4.1 socket** final **AppSocket** socket [protected]

The Socket.

## 1.44 MinimaxMove

### Public Member Functions

- **MinimaxMove** ( **Move** move, **Evaluation** moveEvaluation)
- **MinimaxMove** ( **Evaluation** moveEvaluation)
- **MinimaxMove** ( **MinimaxMove** other)
- boolean **isDeeperAndBetterThan** ( **MinimaxMove** other)
- int **getMoveDepth** ()
- **Move** **getMove** ()
- void **setMove** ( **Move** move)
- **Evaluation** **getMoveEvaluation** ()
- void **setMoveEvaluation** ( **Evaluation** moveEvaluation)
- int **hashCode** ()
- boolean **equals** (Object o)
- String **toString** ()
- String **getShortPrintingStr** ()
- int **compareTo** ( **MinimaxMove** o)

### 1.44.1 Detailed Description

Minimax move.

Author

Bezalel Avrahami ( bezalel13250@gmail.com)

### 1.44.2 Constructor & Destructor Documentation

**1.44.2.1 MinimaxMove()** [1/3] **MinimaxMove** (  
     **Move** move,  
     **Evaluation** moveEvaluation )

Instantiates a new Minimax move.

Parameters

<i>move</i>	the move
<i>moveEvaluation</i>	the move evaluation

#### 1.44.2.2 MinimaxMove() [2/3] MinimaxMove ( Evaluation *moveEvaluation* )

Instantiates a new Minimax move.

##### Parameters

<i>moveEvaluation</i>	the move evaluation
-----------------------	---------------------

#### 1.44.2.3 MinimaxMove() [3/3] MinimaxMove ( MinimaxMove *other* )

Instantiates a new Minimax move.

##### Parameters

<i>other</i>	the other
--------------	-----------

### 1.44.3 Member Function Documentation

#### 1.44.3.1 compareTo() int compareTo ( MinimaxMove *o* )

Compare to int.

##### Parameters

<i>o</i>	the o
----------	-------

##### Returns

the int

#### 1.44.3.2 equals() boolean equals ( Object *o* )

Equals boolean.

##### Parameters

<i>o</i>	the o
----------	-------



**Returns**

the boolean

**1.44.3.3 getMove()** `Move getMove ( )`

Gets move.

**Returns**

the move

**1.44.3.4 getMoveDepth()** `int getMoveDepth ( )`

Gets move depth.

**Returns**

the move depth

**1.44.3.5 getMoveEvaluation()** `Evaluation getMoveEvaluation ( )`

Gets move evaluation.

**Returns**

the move evaluation

**1.44.3.6 getShortPrintingStr()** `String getShortPrintingStr ( )`

Gets short printing str.

**Returns**

the short printing str

**1.44.3.7 hashCode()** `int hashCode ( )`

Hash code int.

**Returns**

the int

**1.44.3.8 isDeeperAndBetterThan()** `boolean isDeeperAndBetterThan ( MinimaxMove other )`

Is deeper and better than given minimax move.

**Parameters**

<i>other</i>	the other
--------------	-----------

**Returns**

the boolean

**1.44.3.9 setMove()** `void setMove (   
 Move move )`

Sets move.

**Parameters**

<i>move</i>	the move
-------------	----------

**1.44.3.10 setMoveEvaluation()** `void setMoveEvaluation (   
 Evaluation moveEvaluation )`

Sets move evaluation.

**Parameters**

<i>moveEvaluation</i>	the move evaluation
-----------------------	---------------------

**1.44.3.11 toString()** `String toString ( )`

To string string.

**Returns**

the string

## 1.45 Move

**Classes**

- enum **MoveFlag**
- enum **ThreefoldStatus**

## Public Member Functions

- **Move** ( **Location** movingFrom, **Location** movingTo, **PieceType** capturingPieceType)
- **Move** ( **Location** movingFrom, **Location** movingTo)
- **Move** ( **ThreefoldStatus** threefoldStatus)
- **Move** ( **Move** other)
- MovesList **getCreatorList** ()
- void **setCreatorList** (MovesList creatorList)
- void **setMoveAnnotation** (String moveAnnotation)
- byte **getDisabledCastling** ()
- void **setDisabledCastling** (byte disabledCastling)
- void **setThreefoldOption** ()
- **PlayerColor** **getMovingColor** ()
- void **setMovingColor** ( **PlayerColor** movingPlayerColor)
- int **getPrevFullMoveClock** ()
- void **setPrevFullMoveClock** (int prevFullMoveClock)
- int **getPrevHalfMoveClock** ()
- void **setPrevHalfMoveClock** (int prevHalfMoveClock)
- **PieceType** **getPromotingTo** ()
- void **setPromotingTo** ( **PieceType** promotingTo)
- **BasicMove** **getIntermediateMove** ()
- void **setIntermediateMove** ( **BasicMove** intermediateMove)
- boolean **isCheck** ()
- **Evaluation** **getMoveEvaluation** ()
- void **setMoveEvaluation** ( **Evaluation** moveEvaluation)
- **PieceType** **getCapturingPieceType** ()
- boolean **isReversible** ()
- void **setReversible** (boolean reversible)
- boolean **isCapturing** ()
- void **setCapturing** ( **PieceType** pieceType)
- boolean **equals** (Object o)
- String **toString** ()
- String **getAnnotation** ()
- boolean **strictEquals** ( **Move** move)
- **MoveFlag** **getMoveFlag** ()
- void **setMoveFlag** ( **MoveFlag** moveFlag)
- int **compareTo** ( **Move** o)
- **Location** **getEnPasantLoc** ()
- void **setEnPasantLoc** ( **Location** epsnLoc)

## Static Public Member Functions

- static **Move** **castling** ( **Location** movingFrom, **Location** movingTo, CastlingRights.Side side)
- static **Move** **threefoldClaim** ()
- static **Move** **flipMove** ( **Move** move)
- static **Move** **copyMove** ( **Move** move)

### 1.45.1 Detailed Description

**Move** (p. 146) - represents a "heavy" move. with a lot of info.

Author

Bezalel Avrahami ( bezalel13250@gmail.com)

## 1.45.2 Class Documentation

**1.45.2.1** `enum ver14::SharedClasses::Game::Moves::Move::ThreefoldStatus` Threefold status - represents a threefold draw status.

### Author

Bezalel Avrahami ( `bezalel3250@gmail.com`)

### Enumerator

<code>CAN_CLAIM</code>	Can claim threefold status.
<code>CLAIMED</code>	Claimed threefold status.
<code>NONE</code>	None threefold status.

## 1.45.3 Constructor & Destructor Documentation

**1.45.3.1** `Move()` [1/4] `Move` (  
    `Location` *movingFrom*,  
    `Location` *movingTo*,  
    `PieceType` *capturingPieceType* )

Instantiates a new **Move** (p. 146).

### Parameters

<i>movingFrom</i>	the moving from
<i>movingTo</i>	the moving to
<i>capturingPieceType</i>	the capturing piece type

**1.45.3.2** `Move()` [2/4] `Move` (  
    `Location` *movingFrom*,  
    `Location` *movingTo* )

Instantiates a new **Move** (p. 146).

### Parameters

<i>movingFrom</i>	the moving from
<i>movingTo</i>	the moving to

**1.45.3.3 Move()** [3/4] **Move** (  
**ThreefoldStatus** *threefoldStatus* )

Instantiates a new **Move** (p. 146).

#### Parameters

<i>threefoldStatus</i>	the threefold status
------------------------	----------------------

**1.45.3.4 Move()** [4/4] **Move** (  
**Move** *other* )

Instantiates a new **Move** (p. 146).

#### Parameters

<i>other</i>	the other
--------------	-----------

## 1.45.4 Member Function Documentation

**1.45.4.1 castling()** static **Move** castling (  
**Location** *movingFrom*,  
**Location** *movingTo*,  
CastlingRights.Side *side* ) [static]

creates a Castling move.

#### Parameters

<i>movingFrom</i>	the moving from
<i>movingTo</i>	the moving to
<i>side</i>	the castling side

#### Returns

the move

**1.45.4.2 compareTo()** int compareTo (  
**Move** *o* )

Compare to another move.

**Parameters**

<i>o</i>	the other move
----------	----------------

**Returns**

the comparison result

**1.45.4.3 copyMove()** `static Move copyMove ( Move move ) [static]`

Copy move.

**Parameters**

<i>move</i>	the move
-------------	----------

**Returns**

the move

**1.45.4.4 equals()** `boolean equals ( Object o )`

Equals boolean.

**Parameters**

<i>o</i>	the o
----------	-------

**Returns**

the boolean

Reimplemented from **BasicMove** (p. 18).

**1.45.4.5 flipMove()** `static Move flipMove ( Move move ) [static]`

Flip move move.

## Parameters

<i>move</i>	the move
-------------	----------

## Returns

the move

**1.45.4.6 getAnnotation()** `String getAnnotation ( )`

Gets annotation.

## Returns

the annotation

**1.45.4.7 getCapturingPieceType()** `PieceType getCapturingPieceType ( )`

Gets capturing piece type.

## Returns

the capturing piece type

**1.45.4.8 getCreatorList()** `MovesList getCreatorList ( )`

Gets creator list.

## Returns

the creator list

**1.45.4.9 getDisabledCastling()** `byte getDisabledCastling ( )`

Gets disabled castling.

## Returns

the disabled castling

**1.45.4.10 getEnPassantLoc()**   **Location** getEnPassantLoc ( )

Gets en passant loc.

**Returns**

the en passant loc

**1.45.4.11 getIntermediateMove()**   **BasicMove** getIntermediateMove ( )

Gets intermediate move.

**Returns**

the intermediate move

**1.45.4.12 getMoveEvaluation()**   **Evaluation** getMoveEvaluation ( )

Gets move evaluation.

**Returns**

the move evaluation

**1.45.4.13 getMoveFlag()**   **MoveFlag** getMoveFlag ( )

Gets move flag.

**Returns**

the move flag

**1.45.4.14 getMovingColor()**   **PlayerColor** getMovingColor ( )

Gets moving color.

**Returns**

the moving color



**1.45.4.15 getPrevFullMoveClock()** `int getPrevFullMoveClock ( )`

Gets prev full move clock.

Returns

the prev full move clock

**1.45.4.16 getPrevHalfMoveClock()** `int getPrevHalfMoveClock ( )`

Gets prev half move clock.

Returns

the prev half move clock

**1.45.4.17 getPromotingTo()** `PieceType getPromotingTo ( )`

Gets promoting to.

Returns

the promoting to

**1.45.4.18 isCapturing()** `boolean isCapturing ( )`

Is capturing boolean.

Returns

the boolean

**1.45.4.19 isCheck()** `boolean isCheck ( )`

Is check boolean.

Returns

the boolean

**1.45.4.20 isReversible()** `boolean isReversible ( )`

Is reversible boolean.

Returns

the boolean

**1.45.4.21 setCapturing()** `void setCapturing (   
 PieceType pieceType )`

Sets capturing.

**Parameters**

<i>pieceType</i>	the piece type
------------------	----------------

**1.45.4.22 setCreatorList()** `void setCreatorList ( MovesList creatorList )`

Sets creator list.

**Parameters**

<i>creatorList</i>	the creator list
--------------------	------------------

**1.45.4.23 setDisabledCastling()** `void setDisabledCastling ( byte disabledCastling )`

Sets disabled castling.

**Parameters**

<i>disabledCastling</i>	the disabled castling
-------------------------	-----------------------

**1.45.4.24 setEnPassantLoc()** `void setEnPassantLoc ( Location epsnLoc )`

Sets en passant loc.

**Parameters**

<i>epsnLoc</i>	the epsn loc
----------------	--------------

**1.45.4.25 setIntermediateMove()** `void setIntermediateMove ( BasicMove intermediateMove )`

Sets intermediate move.

**Parameters**

<i>intermediateMove</i>	the intermediate move
-------------------------	-----------------------

**1.45.4.26 setMoveAnnotation()** `void setMoveAnnotation (`  
`String moveAnnotation )`

Sets move annotation.

Parameters

<i>moveAnnotation</i>	the move annotation
-----------------------	---------------------

**1.45.4.27 setMoveEvaluation()** `void setMoveEvaluation (`  
`Evaluation moveEvaluation )`

Sets move evaluation.

Parameters

<i>moveEvaluation</i>	the move evaluation
-----------------------	---------------------

**1.45.4.28 setMoveFlag()** `void setMoveFlag (`  
`MoveFlag moveFlag )`

Sets move flag.

Parameters

<i>moveFlag</i>	the move flag
-----------------	---------------

**1.45.4.29 setMovingColor()** `void setMovingColor (`  
`PlayerColor movingPlayerColor )`

Sets moving color.

Parameters

<i>movingPlayerColor</i>	the moving player color
--------------------------	-------------------------

**1.45.4.30 setPrevFullMoveClock()** `void setPrevFullMoveClock (`  
`int prevFullMoveClock )`

Sets prev full move clock.

Parameters

<i>prevFullMoveClock</i>	the prev full move clock
--------------------------	--------------------------

**1.45.4.31 setPrevHalfMoveClock()** `void setPrevHalfMoveClock (`  
`int prevHalfMoveClock )`

Sets prev half move clock.

Parameters

<i>prevHalfMoveClock</i>	the prev half move clock
--------------------------	--------------------------

**1.45.4.32 setPromotingTo()** `void setPromotingTo (`  
`PieceType promotingTo )`

Sets promoting to.

Parameters

<i>promotingTo</i>	the promoting to
--------------------	------------------

**1.45.4.33 setReversible()** `void setReversible (`  
`boolean reversible )`

Sets reversible.

Parameters

<i>reversible</i>	the reversible
-------------------	----------------

**1.45.4.34 setThreefoldOption()** `void setThreefoldOption ( )`

Sets threefold option.

**1.45.4.35 strictEquals()** `boolean strictEquals (`  
`Move move )`

checks source and destination equals and move flags equals.

## Parameters

<i>move</i>	the move
-------------	----------

## Returns

the boolean

**1.45.4.36 threefoldClaim()** `static Move threefoldClaim ( ) [static]`

Threefold claim move.

## Returns

the move

**1.45.4.37 toString()** `String toString ( )`

To string string.

## Returns

the string

Reimplemented from **BasicMove** (p. 20).

## 1.46 MoveAnnotation

### Static Public Member Functions

- static String **annotate** ( **Move** move, **Piece** movingPiece)
- static String **annotate** ( **Move** move, **Piece** movingPiece, String unique)
- static String **basicAnnotate** ( **BasicMove** move)

### 1.46.1 Detailed Description

**Move** (p. 146) annotation - utility class that annotates moves.

## Author

Bezalel Avrahami ( bezalel3250@gmail.com)

### 1.46.2 Member Function Documentation

**1.46.2.1 annotate()** [1/2] `static String annotate (`  
`Move move,`  
`Piece movingPiece ) [static]`

Annotate move.

**Parameters**

<i>move</i>	the move
<i>movingPiece</i>	the moving piece

**Returns**

the annotation

```
1.46.2.2 annotate() [2/2] static String annotate (
    Move move,
    Piece movingPiece,
    String unique ) [static]
```

Annotate move with a unique string.

**Parameters**

<i>move</i>	the move
<i>movingPiece</i>	the moving piece
<i>unique</i>	the unique string

**Returns**

the string

```
1.46.2.3 basicAnnotate() static String basicAnnotate (
    BasicMove move ) [static]
```

Basic annotate a move. just the source and destination.

**Parameters**

<i>move</i>	the move
-------------	----------

**Returns**

the string

## 1.47 Move.MoveFlag Enum Reference

**Public Member Functions**

- **MoveFlag** ()
- **MoveFlag** (CastlingRights.Side side)
- boolean **equals** ( **Move** myMove, **Move** otherMove)

**Public Attributes**

- **NormalMove**
- **EnPassant**
- **DoublePawnPush**
- **Promotion**
- **ShortCastle**
- **LongCastle**
- final boolean **isCastling**
- final CastlingRights.Side **castlingSide**

**Static Public Attributes**

- static final **MoveFlag[] CASTLING\_FLAGS**

**1.47.1 Detailed Description**

**Move** (p. 146) flag - which type of move this is.

Author

Bezalel Avrahami ( bezalel3250@gmail.com)

**1.47.2 Constructor & Destructor Documentation****1.47.2.1 MoveFlag() [1/2]    MoveFlag ( )**

Instantiates a new **Move** (p. 146) flag.

**1.47.2.2 MoveFlag() [2/2]    MoveFlag ( CastlingRights.Side side )**

Instantiates a new **Move** (p. 146) flag.

Parameters

<i>side</i>	the side
-------------	----------

**1.47.3 Member Function Documentation**

**1.47.3.1 equals()** `boolean equals (`  
    **Move** *myMove*,  
    **Move** *otherMove* )

checks if the flags are equals.

#### Parameters

<i>myMove</i>	the my move
<i>otherMove</i>	the other move

#### Returns

the boolean

### 1.47.4 Member Data Documentation

**1.47.4.1 CASTLING\_FLAGS** `final MoveFlag [ ] CASTLING_FLAGS [static]`

The Castling flags.

**1.47.4.2 castlingSide** `final CastlingRights.Side castlingSide`

if this is a castling flag, which side is it.

**1.47.4.3 DoublePawnPush** `DoublePawnPush`

Double pawn push move flag.

**1.47.4.4 EnPassant** `EnPassant`

En passant move flag.

**1.47.4.5 isCastling** `final boolean isCastling`

Is this move flag a castling flag.

**1.47.4.6 LongCastle** `LongCastle`

Long castle move flag.

**1.47.4.7 NormalMove** `NormalMove`

Normal move move flag.



**1.47.4.8 Promotion** `Promotion`

The Promotion.

**1.47.4.9 ShortCastle** `ShortCastle`

Short castle move flag.

**1.48 MyError****Public Member Functions**

- **MyError** ()
- **MyError** (Throwable throwable)
- **MyError** (String message)
- **MyError** (String message, Throwable cause)
- String **getHandledStr** ()
- String **toString** ()

**Static Public Member Functions**

- static String **errToString** (Throwable error)

**1.48.1 Detailed Description**

My error - .

Author

Bezalel Avrahami ( `bezalel3250@gmail.com` )

**1.48.2 Constructor & Destructor Documentation****1.48.2.1 MyError()** [1/4] `MyError ( )`

Instantiates a new My error.

**1.48.2.2 MyError()** [2/4] `MyError (   
 Throwable throwable )`

Instantiates a new My error.

Parameters

<i>throwable</i>	the throwable
------------------	---------------

### 1.48.2.3 **MyError()** [3/4]   **MyError** (           String *message* )

Instantiates a new My error.

#### Parameters

<i>message</i>	the message
----------------	-------------

### 1.48.2.4 **MyError()** [4/4]   **MyError** (           String *message*,           Throwable *cause* )

Instantiates a new My error.

#### Parameters

<i>message</i>	the message
<i>cause</i>	the cause

## 1.48.3 Member Function Documentation

### 1.48.3.1 **errToString()**   static String errToString (           Throwable *error* )   [static]

Err to string string.

#### Parameters

<i>error</i>	the error
--------------	-----------

#### Returns

the string

### 1.48.3.2 **getHandledStr()**   String getHandledStr ( )

Gets handled str.

## Returns

the handled str

### 1.48.3.3 toString() `String toString ( )`

To string string.

## Returns

the string

## 1.49 MyJButton

### Public Member Functions

- **MyJButton** (String text, **VoidCallback** callback)
- **MyJButton** (String text)
- void **setOnClick** ( **VoidCallback** onClick)
- void **setText** (String text)
- **MyJButton** ()
- **MyJButton** (String text, Font font, **VoidCallback** onClick, JPanel addTo)
- **MyJButton** (String text, Font font, **VoidCallback** onClick)
- **MyJButton** (String text, Font font)
- void **setFont** (Font font)
- int **getMinSize** ()
- void **replaceWithCancel** ( **VoidCallback** onCancelled)
- void **resetState** (boolean e)

### 1.49.1 Detailed Description

MyJButton- represents a button.

## Author

Bezalel Avrahami ( [bezalel3250@gmail.com](mailto:bezalel3250@gmail.com))

### 1.49.2 Constructor & Destructor Documentation

#### 1.49.2.1 MyJButton() [1/6] `MyJButton (` `String text,` `VoidCallback callback )`

Instantiates a new My j button.

## Parameters

<i>text</i>	the text
<i>callback</i>	the callback

**1.49.2.2 MyJButton()** [2/6]    **MyJButton** (   
          String *text* )

Instantiates a new My j button.

## Parameters

<i>text</i>	the text
-------------	----------

**1.49.2.3 MyJButton()** [3/6]    **MyJButton** ( )

Instantiates a new My j button.

**1.49.2.4 MyJButton()** [4/6]    **MyJButton** (   
          String *text*,  
          Font *font*,  
          VoidCallback *onClick*,  
          JPanel *addTo* )

Instantiates a new My j button.

## Parameters

<i>text</i>	the text
<i>font</i>	the font
<i>onClick</i>	the on click
<i>addTo</i>	the add to

**1.49.2.5 MyJButton()** [5/6]    **MyJButton** (   
          String *text*,  
          Font *font*,  
          VoidCallback *onClick* )

Instantiates a new My j button.

## Parameters

<i>text</i>	the text
<i>font</i>	the font
<i>onClick</i>	the on click

**1.49.2.6 MyJButton()** [6/6] `MyJButton (`  
`String text,`  
`Font font )`

Instantiates a new My j button.

Parameters

<i>text</i>	the text
<i>font</i>	the font

### 1.49.3 Member Function Documentation

**1.49.3.1 getMinSize()** `int getMinSize ( )`

Gets min size.

Returns

the min size

**1.49.3.2 replaceWithCancel()** `void replaceWithCancel (`  
`VoidCallback onCancelled )`

Replace with cancel.

Parameters

<i>onCancelled</i>	the on cancelled
--------------------	------------------

**1.49.3.3 resetState()** `void resetState (`  
`boolean e )`

Reset state.

Parameters

<i>e</i>	the e
----------	-------

**1.49.3.4 setFont()** `void setFont (`  
    `Font font )`

Sets font.

Parameters

<i>font</i>	the font
-------------	----------

**1.49.3.5 setOnClick()** `void setOnClick (`  
    `VoidCallback onClick )`

Sets on click.

Parameters

<i>onClick</i>	the on click
----------------	--------------

**1.49.3.6 setText()** `void setText (`  
    `String text )`

Sets text.

Parameters

<i>text</i>	the text
-------------	----------

## 1.50 MyJFrame

### Classes

- interface **BooleanClosing**
- interface **Closing**
- interface **StringClosing**

### Public Member Functions

- **MyJFrame** () throws HeadlessException
- MyAdapter **getMyAdapter** ()
- void **setOnExit** ( **Closing**<?> **onClose**)
- void **doXClick** ()
- void **setOnResize** ( **VoidCallback** onResize)

### Static Public Member Functions

- static MyAdapter **debugAdapter** (Window addTo)

### Protected Attributes

- **Closing**<?> **onClose**

#### 1.50.1 Detailed Description

The type My j frame.

#### 1.50.2 Constructor & Destructor Documentation

##### 1.50.2.1 MyJFrame() `MyJFrame ( ) throws HeadlessException`

Instantiates a new My j frame.

##### Exceptions

<i>HeadlessException</i>	the headless exception
--------------------------	------------------------

#### 1.50.3 Member Function Documentation

##### 1.50.3.1 debugAdapter() `static MyAdapter debugAdapter ( Window addTo ) [static]`

Debug adapter my adapter.

##### Parameters

<i>addTo</i>	the add to
--------------	------------

##### Returns

the my adapter

##### 1.50.3.2 doXClick() `void doXClick ( )`

Do x click.

**1.50.3.3 getMyAdapter()** `MyAdapter getMyAdapter ( )`

Gets my adapter.

**Returns**

the my adapter

**1.50.3.4 setOnExit()** `void setOnExit (   
 Closing<?> onClose )`

Sets on exit.

**Parameters**

<i>onClose</i>	the on close
----------------	--------------

**1.50.3.5 setOnResize()** `void setOnResize (   
 VoidCallback onResize )`

Sets on resize.

**Parameters**

<i>onResize</i>	the on resize
-----------------	---------------

**1.50.4 Member Data Documentation****1.50.4.1 onClose** `Closing<?> onClose [protected]`

The On close.

**1.51 MyThread****Public Member Functions**

- **MyThread** ()
- void **stopRun** ()
- void **ignoreErrs** ()
- void **reactivateErrs** ()



**Static Public Member Functions**

- static void **currentThread** ( **Callback**< **MyThread** > run)

**Protected Member Functions**

- abstract void **handledRun** () throws Throwable

**1.51.1 Detailed Description**

The type My thread.

**1.51.2 Constructor & Destructor Documentation****1.51.2.1 MyThread()** **MyThread** ( )

Instantiates a new My thread.

**1.51.3 Member Function Documentation****1.51.3.1 currentThread()** static void **currentThread** ( **Callback**< **MyThread** > run ) [static]

Current thread. will only execute code if inside a **MyThread** (p. 168)

**Parameters**

<i>run</i>	the run
------------	---------

**1.51.3.2 handledRun()** abstract void **handledRun** ( ) throws Throwable [abstract], [protected]

Handled run.

**Exceptions**

<i>Throwable</i>	the throwable
------------------	---------------

Reimplemented in **AppSocket** (p. 4), and **HandledThread** (p. 94).

### 1.51.3.3 ignoreErrs() `void ignoreErrs ( )`

Ignore errs.

### 1.51.3.4 reactivateErrs() `void reactivateErrs ( )`

Reactivate errs.

### 1.51.3.5 stopRun() `void stopRun ( )`

Stop run.

## 1.52 ObjCallback< T >

### 1.52.1 Detailed Description

Parameters

< T >	the type
-------	----------

## 1.53 Piece Enum Reference

### Public Member Functions

- String **getPiecelcon** ()
- String **getFen** ()
- boolean **isOnMyTeam** ( **Piece** otherPiece)
- boolean **isOnMyTeam** ( **PlayerColor** otherPlayerColor)

### Static Public Member Functions

- static **Piece** **getPieceFromFen** (char c)
- static **Piece** **getPiece** ( **PieceType** pieceType, **PlayerColor** playerColor)
- static String **getPiecelcon** ( **Piece** piece)

### Public Attributes

- **W\_P**
- **W\_R**
- **W\_B**
- **W\_N**
- **W\_Q**
- **W\_K**
- **B\_P**

- **B\_R**
- **B\_B**
- **B\_N**
- **B\_Q**
- **B\_K**
- final **PieceType** *pieceType*
- final **PlayerColor** *playerColor*

#### Static Public Attributes

- static final **Piece**[] **ALL\_PIECES**

#### 1.53.1 Detailed Description

**Piece** (p. 170) - represents a combination of a piece type, and a color.

#### Author

Bezalel Avrahami ( [bezalel13250@gmail.com](mailto:bezalel13250@gmail.com) )

#### See also

**PieceType** (p. 175)

**PlayerColor** (p. 181)

#### 1.53.2 Member Function Documentation

##### 1.53.2.1 **getFen()** `String getFen ( )`

Gets fen.

#### Returns

the fen

##### 1.53.2.2 **getPiece()** `static Piece getPiece ( PieceType pieceType, PlayerColor playerColor ) [static]`

Gets piece.

#### Parameters

<i>pieceType</i>	the piece type
<i>playerColor</i>	the player color

**Returns**

the piece

**1.53.2.3 getPieceFromFen()** `static Piece getPieceFromFen ( char c ) [static]`

Gets piece from fen.

**Parameters**

<i>c</i>	the c
----------	-------

**Returns**

the piece from fen

**1.53.2.4 getPiecelcon()** `[1/2] String getPiecelcon ( )`

Gets piece icon.

**Returns**

the piece icon

**1.53.2.5 getPiecelcon()** `[2/2] static String getPiecelcon ( Piece piece ) [static]`

Gets piece icon.

**Parameters**

<i>piece</i>	the piece
--------------	-----------

**Returns**

the piece icon

**1.53.2.6 isOnMyTeam()** `[1/2] boolean isOnMyTeam ( Piece otherPiece )`

Is on my team boolean.

## Parameters

<i>otherPiece</i>	the other piece
-------------------	-----------------

## Returns

the boolean

**1.53.2.7 isOnMyTeam()** [2/2] `boolean isOnMyTeam (   
 PlayerColor otherPlayerColor )`

Is on my team boolean.

## Parameters

<i>otherPlayerColor</i>	the other player color
-------------------------	------------------------

## Returns

the boolean

**1.53.3 Member Data Documentation**

**1.53.3.1 ALL\_PIECES** `final Piece [] ALL_PIECES [static]`

The constant ALL\_PIECES.

**1.53.3.2 B\_B** `B_B`

represents a Black Bishop.

**1.53.3.3 B\_K** `B_K`

represents a Black King.

**1.53.3.4 B\_N** `B_N`

represents a Black Knight.

**1.53.3.5 B\_P** `B_P`

represents a Black p piece.

**1.53.3.6 B\_Q** B\_Q

represents a Black Queen.

**1.53.3.7 B\_R** B\_R

represents a Black Rook.

**1.53.3.8 pieceType** final PieceType pieceType

The **Piece** (p. 170) type.

**1.53.3.9 playerColor** final PlayerColor playerColor

The Player color.

**1.53.3.10 W\_B** W\_B

represents a White Bishop.

**1.53.3.11 W\_K** W\_K

represents a White King.

**1.53.3.12 W\_N** W\_N

represents a White Knight.

**1.53.3.13 W\_P** W\_P

represents a White p piece.

**1.53.3.14 W\_Q** W\_Q

represents a White Queen.

**1.53.3.15 W\_R** W\_R

represents a White Rook.

## 1.54 PieceType Enum Reference

### Public Member Functions

- String **getWhitePieceFen** ()
- String **getPieceIcon** ( **PlayerColor** playerColor)
- boolean **compareMovementType** ( **PieceType** compareTo)
- String **getPieceName** ()
- boolean **isAttack** ( **Direction** direction, int maxDistance)
- **Direction[]** **getAttackingDirections** ()
- **Direction[]** **getWalkingDirections** ()

### Static Public Member Functions

- static **PieceType** **getPieceType** (int pieceType)
- static **Direction[]** **getAttackingDirections** ( **PieceType** pieceType)
- static boolean **compareMovementType** ( **PieceType** piece1Type, **PieceType** piece2Type)
- static boolean **isDiagonalPiece** ( **PieceType** pieceType)
- static boolean **isLinePiece** ( **PieceType** pieceType)

### Public Attributes

- **PAWN**
- **ROOK**
- **BISHOP**
- **KNIGHT**
- **QUEEN**
- **KING**
- final String **whitelcon**
- final String **blacklcon**
- final int **value**
- final boolean **isSliding**
- final int **asInt**

### Static Public Attributes

- static final int **NUM\_OF\_PIECE\_TYPES**
- static final **PieceType[]** **PIECE\_TYPES**
- static final **PieceType[]** **UNIQUE\_MOVES\_PIECE\_TYPES**
- static final **PieceType[]** **MINOR\_PIECES**
- static final **PieceType[]** **MAJOR\_PIECES**
- static final **PieceType[]** **CAN\_PROMOTE\_TO**
- static final **PieceType[]** **ATTACKING\_PIECE\_TYPES**
- static final **Direction[][]** **ATTACKING\_DIRECTIONS**

#### 1.54.1 Detailed Description

**Piece** (p. 170) type - represents the **Piece** (p. 170) Type .

#### Author

Bezalel Avrahami ( bezalel13250@gmail.com)

## 1.54.2 Member Function Documentation

**1.54.2.1 compareMovementType()** [1/2] `boolean compareMovementType (   
 PieceType compareTo )`

Compare movement type boolean.

### Parameters

<i>compareTo</i>	the compare to
------------------	----------------

### Returns

the boolean

**1.54.2.2 compareMovementType()** [2/2] `static boolean compareMovementType (   
 PieceType piece1Type,   
 PieceType piece2Type ) [static]`

Compare movement type boolean.

### Parameters

<i>piece1Type</i>	the piece 1 type
<i>piece2Type</i>	the piece 2 type

### Returns

the boolean

**1.54.2.3 getAttackingDirections()** [1/2] `Direction[] getAttackingDirections ( )`

Get attacking directions direction [ ].

### Returns

the direction [ ]

**1.54.2.4 getAttackingDirections()** [2/2] `static Direction[] getAttackingDirections (   
 PieceType pieceType ) [static]`

Get attacking directions direction [ ].



## Parameters

<i>pieceType</i>	the piece type
------------------	----------------

## Returns

the direction [ ]

**1.54.2.5 getPieceIcon()** `String getPieceIcon (   
           PlayerColor playerColor )`

Gets piece icon.

## Parameters

<i>playerColor</i>	the player color
--------------------	------------------

## Returns

the piece icon

**1.54.2.6 getPieceName()** `String getPieceName ( )`

Gets piece name.

## Returns

the piece name

**1.54.2.7 getPieceType()** `static PieceType getPieceType (   
           int pieceType ) [static]`

Gets piece type.

## Parameters

<i>pieceType</i>	the piece type
------------------	----------------

## Returns

the piece type

**1.54.2.8 getWalkingDirections()** `Direction[] getWalkingDirections ( )`

Get walking directions direction [ ].

**Returns**

the direction [ ]

**1.54.2.9 getWhitePieceFen()** `String getWhitePieceFen ( )`

Gets white piece fen.

**Returns**

the white piece fen

**1.54.2.10 isAttack()** `boolean isAttack (   
    Direction direction,   
    int maxDistance )`

Is attack boolean.

**Parameters**

<i>direction</i>	the direction
<i>maxDistance</i>	the max distance

**Returns**

the boolean

**1.54.2.11 isDiagonalPiece()** `static boolean isDiagonalPiece (   
    PieceType pieceType ) [static]`

Is diagonal piece boolean.

**Parameters**

<i>pieceType</i>	the piece type
------------------	----------------

**Returns**

the boolean

**1.54.2.12 isLinePiece()** `static boolean isLinePiece (   
 PieceType pieceType ) [static]`

Is line piece boolean.

**Parameters**

<i>pieceType</i>	the piece type
------------------	----------------

**Returns**

the boolean

**1.54.3 Member Data Documentation**

**1.54.3.1 asInt** `final int asInt`

The As int.

**1.54.3.2 ATTACKING\_DIRECTIONS** `final Direction [][] ATTACKING_DIRECTIONS [static]`

The Attacking directions. each piece type has its own set of attacking direction.

**1.54.3.3 ATTACKING\_PIECE\_TYPES** `final PieceType [] ATTACKING_PIECE_TYPES [static]`

The constant ATTACKING\_PIECE\_TYPES.

**1.54.3.4 BISHOP** `BISHOP`

Bishop piece type.

**1.54.3.5 blackIcon** `final String blackIcon`

The Black icon.

**1.54.3.6 CAN\_PROMOTE\_TO** `final PieceType [] CAN_PROMOTE_TO [static]`

The types of pieces a pawn Can promote to.

**1.54.3.7 isSliding** `final boolean isSliding`

The Is sliding.

**1.54.3.8 KING** `KING`

The King.

**1.54.3.9 KNIGHT** `KNIGHT`

The Knight.

**1.54.3.10 MAJOR\_PIECES** `final PieceType [] MAJOR_PIECES [static]`

The Major pieces.

**1.54.3.11 MINOR\_PIECES** `final PieceType [] MINOR_PIECES [static]`

The Minor pieces.

**1.54.3.12 NUM\_OF\_PIECE\_TYPES** `final int NUM_OF_PIECE_TYPES [static]`

The constant NUM\_OF\_PIECE\_TYPES.

**1.54.3.13 PAWN** `PAWN`

Pawn **Piece** (p. 170) Type.

**1.54.3.14 PIECE\_TYPES** `final PieceType [] PIECE_TYPES [static]`

The constant PIECE\_TYPES.

**1.54.3.15 QUEEN** `QUEEN`

Queen piece type.

**1.54.3.16 ROOK** `ROOK`

Rook piece type.

**1.54.3.17 UNIQUE\_MOVES\_PIECE\_TYPES** `final PieceType [] UNIQUE_MOVES_PIECE_TYPES [static]`

The Unique moves piece types.

**1.54.3.18 value** `final int value`

The Value.

**1.54.3.19 whitelcon** `final String whiteIcon`

The White icon.

**1.55 PlayerColor Enum Reference****Public Member Functions**

- **PlayerColor** (int **startingRow**, int **diff**)
- abstract **PlayerColor** **getOpponent** ()
- String **toString** ()
- String **getName** ()

**Static Public Member Functions**

- static **PlayerColor** **getColor** (int clr)
- static **PlayerColor** **getPlayerFromFen** (String playerToMove)

**Public Attributes**

- **WHITE**
- **BLACK**
- **NO\_PLAYER**
- final int **asInt**
- final int **indexOf2**
- final int **startingRow**
- final int **diff**

**Static Public Attributes**

- static final **PlayerColor[]** **PLAYER\_COLORS**
- static final int **NUM\_OF\_PLAYERS**

**1.55.1 Detailed Description**

Player color - represents a player color.

Author

Bezalel Avrahami ( bezalel3250@gmail.com)

**1.55.2 Constructor & Destructor Documentation****1.55.2.1 PlayerColor()** `PlayerColor (`  
`int startingRow,`  
`int diff )`

Instantiates a new Player color.

**Parameters**

<i>startingRow</i>	the starting row
<i>diff</i>	the diff

**1.55.3 Member Function Documentation**

**1.55.3.1 getColor()** `static PlayerColor getColor (`  
`int clr ) [static]`

Gets color.

**Parameters**

<i>clr</i>	the clr
------------	---------

**Returns**

the color

**1.55.3.2 getName()** `String getName ( )`

Gets name.

**Returns**

the name

**1.55.3.3 getOpponent()** `abstract PlayerColor getOpponent ( ) [abstract]`

Gets opponent.

**Returns**

the opponent

**1.55.3.4 getPlayerFromFen()** `static PlayerColor getPlayerFromFen (`  
`String playerToMove ) [static]`

Gets player from fen.

## Parameters

<i>playerToMove</i>	the player to move
---------------------	--------------------

## Returns

the player from fen

**1.55.3.5 toString()** `String toString ( )`

To string string.

## Returns

the string

**1.55.4 Member Data Documentation****1.55.4.1 asInt** `final int asInt`

The As int.

**1.55.4.2 BLACK** `BLACK`

Black.

**1.55.4.3 diff** `final int diff`

The moving up ratio.

**1.55.4.4 indexOf2** `final int indexOf2`

index of two. calculated at initialization for performance. calculation: **asInt** (p. 183) \*2

**1.55.4.5 NO\_PLAYER** `NO_PLAYER`

No player.

**1.55.4.6 NUM\_OF\_PLAYERS** `final int NUM_OF_PLAYERS [static]`

The constant NUM\_OF\_PLAYERS.

**1.55.4.7 PLAYER\_COLORS** `final PlayerColor [] PLAYER_COLORS [static]`

The constant PLAYER\_COLORS.

**1.55.4.8 startingRow** `final int startingRow`

The player's Starting row.

**1.55.4.9 WHITE** `WHITE`

White.

## 1.56 PreMadeRequest

### Classes

- interface **VariationCreator**

### Public Member Functions

- **PreMadeRequest[] getRequestVariations ()**
- **RequestBuilder createBuilder ()**

### Public Attributes

- final int **authSettings**

### Static Public Attributes

- static final **PreMadeRequest TopPlayers**
- static final **PreMadeRequest Games**
- static final **PreMadeRequest DeleteUnfGames**
- static final **PreMadeRequest StatsByTimeOfDay**
- static final **PreMadeRequest ChangeProfilePic**
- static final **PreMadeRequest[] statistics**

### 1.56.1 Detailed Description

Pre made request - .

#### Author

Bezalel Avrahami ( [bezalel3250@gmail.com](mailto:bezalel3250@gmail.com))

### 1.56.2 Member Function Documentation



**1.56.2.1 createBuilder()** `RequestBuilder createBuilder ( )`

Create builder request builder.

**Returns**

the request builder

**1.56.2.2 getRequestVariations()** `PreMadeRequest [] getRequestVariations ( )`

Get request variations pre made request [ ].

**Returns**

the pre made request [ ]

**1.56.3 Member Data Documentation****1.56.3.1 authSettings** `final int authSettings`

The Auth settings.

**1.56.3.2 ChangeProfilePic** `final PreMadeRequest ChangeProfilePic [static]`

The constant ChangeProfilePic.

**1.56.3.3 DeleteUnfGames** `final PreMadeRequest DeleteUnfGames [static]`

The constant DeleteUnfGames.

**1.56.3.4 Games** `final PreMadeRequest Games [static]`

The constant Games.

**1.56.3.5 statistics** `final PreMadeRequest [] statistics [static]`

The Statistics.

**1.56.3.6 StatsByTimeOfDay** `final PreMadeRequest StatsByTimeOfDay [static]`

The constant StatsByTimeOfDay.

### 1.56.3.7 TopPlayers `final PreMadeRequest TopPlayers [static]`

The constant TopPlayers.

## 1.57 Question

### Classes

- enum **QuestionType**

### Public Member Functions

- **Question** (String **questionStr**, Answer... possibleAnswers)
- **Question** (String **questionStr**, **QuestionType questionType**, Answer... possibleAnswers)
- **Answer** **getDefaultAnswer** ()
- void **setDefaultAnswer** ( **Answer** defaultAnswer)
- String **getQuestionStr** ()
- **Answer[]** **getPossibleAnswers** ()
- boolean **equals** (Object o)
- String **toString** ()
- record **Answer** (String answerStr) implements Serializable

### Static Public Member Functions

- static **Question** **drawOffer** (String offeringPlayer)

### Public Attributes

- final String **questionStr**
- final **QuestionType** **questionType**

### Static Public Attributes

- static final **Question** **Threefold**
- static final **Question** **Rematch**

### 1.57.1 Detailed Description

**Question** (p. 186).

#### Author

Bezalel Avrahami ( [bezalel13250@gmail.com](mailto:bezalel13250@gmail.com))

### 1.57.2 Class Documentation

**1.57.2.1** `enum ver14::SharedClasses::Misc::Question::QuestionType` **Question** (p. 186) type.

#### Author

Bezalel Avrahami ( [bezalel13250@gmail.com](mailto:bezalel13250@gmail.com))

## Enumerator

DRAW_OFFER	Draw offer question type.
NO_TYPE	No type question type.
REMATCH	Rematch question type.
THREEFOLD	Threefold question type.

## 1.57.3 Constructor &amp; Destructor Documentation

### 1.57.3.1 Question() [1/2] `Question (` `String questionStr,` `Answer... possibleAnswers )`

Instantiates a new **Question** (p. 186).

## Parameters

<i>questionStr</i>	the question str
<i>possibleAnswers</i>	the possible answers

### 1.57.3.2 Question() [2/2] `Question (` `String questionStr,` `QuestionType questionType,` `Answer... possibleAnswers )`

Instantiates a new **Question** (p. 186).

## Parameters

<i>questionStr</i>	the question str
<i>questionType</i>	the question type
<i>possibleAnswers</i>	the possible answers

## 1.57.4 Member Function Documentation

### 1.57.4.1 Answer() `record Answer (` `String answerStr )`

Answer.

**Author**

Bezalel Avrahami ( bezalel3250@gmail.com)

**1.57.4.2 drawOffer()** `static Question drawOffer (`  
`String offeringPlayer ) [static]`

Draw offer question.

**Parameters**

<i>offeringPlayer</i>	the offering player
-----------------------	---------------------

**Returns**

the question

**1.57.4.3 equals()** `boolean equals (`  
`Object o )`

Equals boolean.

**Parameters**

<i>o</i>	the o
----------	-------

**Returns**

the boolean

**1.57.4.4 getDefaultAnswer()** `Answer getDefaultAnswer ( )`

Gets default answer.

**Returns**

the default answer

**1.57.4.5** `getPossibleAnswers()` `Answer[] getPossibleAnswers ( )`

Get possible answers answer [ ].

**Returns**

the answer [ ]

**1.57.4.6** `getQuestionStr()` `String getQuestionStr ( )`

Gets question str.

**Returns**

the question str

**1.57.4.7** `setDefaultAnswer()` `void setDefaultAnswer (`  
`Answer defaultAnswer )`

Sets default answer.

**Parameters**

<i>defaultAnswer</i>	the default answer
----------------------	--------------------

**1.57.4.8** `toString()` `String toString ( )`

To string string.

**Returns**

the string

**1.57.5 Member Data Documentation****1.57.5.1** `questionStr` `final String questionStr`

The **Question** (p. 186) str.

### 1.57.5.2 `questionType` `final QuestionType questionType`

The `Question` (p. 186) type.

### 1.57.5.3 `Rematch` `final Question Rematch [static]`

The constant `Rematch`.

### 1.57.5.4 `Threefold` `final Question Threefold [static]`

The constant `Threefold`.

## 1.58 `Regex`

### Public Member Functions

- `Regex (@Language("RegExp") String regex, String details, String... dontMatch)`
- `Regex (@Language("RegExp") String regex, String details, boolean useDontMatch, String... dontMatch)`
- `Regex canBeEmpty` (boolean bool, String emptyDetails)
- `Regex get` (boolean useDontMatch)
- void `setUseDontMatch` (boolean useDontMatch)
- String `getDetails` ()
- boolean `check` (String str)
- String `getRegex` ()

### Static Public Member Functions

- static boolean `isSavedDate` (String str)

### Public Attributes

- final String[] `dontMatch`

### Static Public Attributes

- static final `Regex Fen`
- static final `Regex Username`
- static final `Regex Password`
- static final `Regex Icon`
- static final `Regex StrUtilSkip`
- static final `Regex Numbers`
- static final `Regex URL`
- static final `Regex DontSaveGame`
- static final `Regex IPPAddress`
- static final `Regex Any`

### 1.58.1 Detailed Description

Reg ex.

Author

Bezalel Avrahami ( bezalel3250@gmail.com)

### 1.58.2 Constructor & Destructor Documentation

**1.58.2.1 RegEx() [1/2]** **RegEx** (  
    @Language("RegExp") String *regEx*,  
    String *details*,  
    String... *dontMatch* )

Instantiates a new Reg ex.

Parameters

<i>regEx</i>	the reg ex
<i>details</i>	the details
<i>dontMatch</i>	the dont match

**1.58.2.2 RegEx() [2/2]** **RegEx** (  
    @Language("RegExp") String *regEx*,  
    String *details*,  
    boolean *useDontMatch*,  
    String... *dontMatch* )

Instantiates a new Reg ex.

Parameters

<i>regEx</i>	the reg ex
<i>details</i>	the details
<i>useDontMatch</i>	the use dont match
<i>dontMatch</i>	the dont match

### 1.58.3 Member Function Documentation

**1.58.3.1 canBeEmpty()** **RegEx** canBeEmpty (   
     boolean bool ,   
     String emptyDetails )

Can be empty reg ex.

**Parameters**

<i>bool</i>	the bool
<i>emptyDetails</i>	the empty details

**Returns**

the reg ex

**1.58.3.2 check()** boolean check (   
     String str )

Check boolean.

**Parameters**

<i>str</i>	the str
------------	---------

**Returns**

the boolean

**1.58.3.3 get()** **RegEx** get (   
     boolean useDontMatch )

Get reg ex.

**Parameters**

<i>useDontMatch</i>	the use dont match
---------------------	--------------------

**Returns**

the reg ex

**1.58.3.4 getDetails()** String getDetails ( )

Gets details.



**Returns**

the details

**1.58.3.5 getRegex()** `String getRegex ( )`

Gets regex.

**Returns**

the regex

**1.58.3.6 isSavedDate()** `static boolean isSavedDate ( String str ) [static]`

Is saved date boolean.

**Parameters**

<i>str</i>	the str
------------	---------

**Returns**

the boolean

**1.58.3.7 setUseDontMatch()** `void setUseDontMatch ( boolean useDontMatch )`

Sets use dont match.

**Parameters**

<i>useDontMatch</i>	the use dont match
---------------------	--------------------

**1.58.4 Member Data Documentation****1.58.4.1 Any** `final RegEx Any [static]`

The constant Any.

**1.58.4.2 dontMatch** `final String [] dontMatch`

The Dont match.

**1.58.4.3 DontSaveGame** `final RegEx DontSaveGame [static]`

The constant DontSaveGame.

**1.58.4.4 Fen** `final RegEx Fen [static]`

The constant Fen.

**1.58.4.5 Icon** `final RegEx Icon [static]`

The constant Icon.

**1.58.4.6 IPAddress** `final RegEx IPAddress [static]`

The constant IPAddress.

**1.58.4.7 Numbers** `final RegEx Numbers [static]`

The constant Numbers.

**1.58.4.8 Password** `final RegEx Password [static]`

The constant Password.

**1.58.4.9 StrUtilSkip** `final RegEx StrUtilSkip [static]`

The constant StrUtilSkip.

**1.58.4.10 URL** `final RegEx URL [static]`

The constant URL.

**1.58.4.11 Username** `final RegEx Username [static]`

The constant Username.

## 1.59 RequestBuilder

### Public Member Functions

- **RequestBuilder** ( **DBRequest** request, **PreMadeRequest.Variation** variation)
- **RequestBuilder** ( **SQLStatement** statement, **String** name, **Arg...** args)
- **RequestBuilder** ( **SQLStatement** statement, **String** name, **String** desc, **Arg...** args)
- **RequestBuilder** ( **SQLStatement** statement, **String** name, **String** postDescription, **String** preDescription, **Arg...** args)
- **void** **setSubBuilder** ( **RequestBuilder** subBuilder)
- **ArrayList**< **SyncedListType** > **getShouldSync** ()
- **String** **getArgVal** (**int** index)
- **DBResponse** **createResponse** (**ResultSet** rs, **DBRequest** request)
- **String** **getPreDescription** ()
- **String** **getPostDescription** ()
- **String** **getName** ()
- **Arg[]** **getArgs** ()
- **DBRequest** **build** (**Object...** argsVals)

### Static Public Member Functions

- **static RequestBuilder** **createVariation** ( **ObjCallback**< **RequestBuilder** > og, **PreMadeRequest.VariationCreator** variationCreator)
- **static RequestBuilder** **changePassword** ()
- **static RequestBuilder** **changeProfilePic** ()
- **static RequestBuilder** **deleteAllUnFinishedGames** ()
- **static RequestBuilder** **games** ()
- **static RequestBuilder** **statsByTimeOfDay** ()
- **static RequestBuilder** **top** ()

### Public Attributes

- **final Arg[]** args

### Static Public Attributes

- **static final String** TIE\_STR

### Protected Member Functions

- **void** **addShouldSync** (**SyncedListType** listType)

### Protected Attributes

- **final SQLStatement** statement
- **final String** name
- **String** postDescription
- **String** preDescription
- **RequestBuilder** subBuilder

### 1.59.1 Detailed Description

Request builder - creates builders capable of generating complete sql statements. after building with the required arguments .

Author

Bezalel Avrahami ( bezalel3250@gmail.com)

### 1.59.2 Constructor & Destructor Documentation

**1.59.2.1 RequestBuilder() [1/4]**    `RequestBuilder (`  
                  `DBRequest request,`  
                  `PreMadeRequest.Variation variation )`

Instantiates a new Request builder.

Parameters

<i>request</i>	the request
<i>variation</i>	the variation

**1.59.2.2 RequestBuilder() [2/4]**    `RequestBuilder (`  
                  `SQLStatement statement,`  
                  `String name,`  
                  `Arg... args )`

Instantiates a new Request builder.

Parameters

<i>statement</i>	the statement
<i>name</i>	the name
<i>args</i>	the args

**1.59.2.3 RequestBuilder() [3/4]**    `RequestBuilder (`  
                  `SQLStatement statement,`  
                  `String name,`  
                  `String desc,`  
                  `Arg... args )`

Instantiates a new Request builder.

## Parameters

<i>statement</i>	the statement
<i>name</i>	the name
<i>desc</i>	the desc
<i>args</i>	the args

**1.59.2.4 RequestBuilder()** [4/4] **RequestBuilder** (

```

    SQLStatement statement,
    String name,
    String postDescription,
    String preDescription,
    Arg... args )

```

Instantiates a new Request builder.

## Parameters

<i>statement</i>	the statement
<i>name</i>	the name
<i>postDescription</i>	the post description
<i>preDescription</i>	the pre description
<i>args</i>	the args

**1.59.3 Member Function Documentation****1.59.3.1 addShouldSync()** **void** addShouldSync (

```

    SynedListType listType ) [protected]

```

Add should sync.

## Parameters

<i>listType</i>	the list type
-----------------	---------------

**1.59.3.2 build()** **DBRequest** build (

```

    Object... argsVals )

```

Build db request.

**Parameters**

<i>argsVals</i>	the args vals
-----------------	---------------

**Returns**

the db request

**1.59.3.3 changePassword()** `static RequestBuilder changePassword ( ) [static]`

Change password request builder.

**Returns**

the request builder

**1.59.3.4 changeProfilePic()** `static RequestBuilder changeProfilePic ( ) [static]`

Change profile pic request builder.

**Returns**

the request builder

**1.59.3.5 createResponse()** `DBResponse createResponse (`  
    `ResultSet rs,`  
    `DBRequest request )`

Create response db response.

**Parameters**

<i>rs</i>	the rs
<i>request</i>	the request

**Returns**

the db response

**1.59.3.6 createVariation()** `static RequestBuilder createVariation (`  
     `ObjCallback< RequestBuilder > og,`  
     `PreMadeRequest.VariationCreator variationCreator ) [static]`

Create variation request builder.

#### Parameters

<i>og</i>	the og
<i>variationCreator</i>	the variation creator

#### Returns

the request builder

**1.59.3.7 deleteAllUnFinishedGames()** `static RequestBuilder deleteAllUnFinishedGames ( ) [static]`

Delete all un finished games request builder.

#### Returns

the request builder

**1.59.3.8 games()** `static RequestBuilder games ( ) [static]`

Games request builder.

#### Returns

the request builder

**1.59.3.9 getArgs()** `Arg[] getArgs ( )`

Get args arg [].

#### Returns

the arg []

**1.59.3.10 getArgVal()** `String getArgVal (`  
     `int index )`

Gets arg val.

**Parameters**

<i>index</i>	the index
--------------	-----------

**Returns**

the arg val

**1.59.3.11 getName()** `String getName ( )`

Gets name.

**Returns**

the name

**1.59.3.12 getPostDescription()** `String getPostDescription ( )`

Gets post description.

**Returns**

the post description

**1.59.3.13 getPreDescription()** `String getPreDescription ( )`

Gets pre description.

**Returns**

the pre description

**1.59.3.14 getShouldSync()** `ArrayList< SyncedListType > getShouldSync ( )`

Gets should sync.

**Returns**

the should sync

**1.59.3.15 setSubBuilder()** `void setSubBuilder (   
    RequestBuilder subBuilder )`

Sets sub builder.



## Parameters

<i>subBuilder</i>	the sub builder
-------------------	-----------------

**1.59.3.16 statsByTimeOfDay()** `static RequestBuilder statsByTimeOfDay ( ) [static]`

Stats by time of day request builder.

## Returns

the request builder

**1.59.3.17 top()** `static RequestBuilder top ( ) [static]`

Top request builder.

## Returns

the request builder

## 1.59.4 Member Data Documentation

**1.59.4.1 args** `final Arg [] args`

The request Arguments.

**1.59.4.2 name** `final String name [protected]`

The Name.

**1.59.4.3 postDescription** `String postDescription [protected]`

The Post description.

**1.59.4.4 preDescription** `String preDescription [protected]`

The Pre description.

**1.59.4.5 statement** `final SQLStatement statement [protected]`

The Statement.

**1.59.4.6 subBuilder** `RequestBuilder subBuilder [protected]`

The Sub builder.

**1.59.4.7 TIE\_STR** `final String TIE_STR [static]`

The constant TIE\_STR.

## 1.60 Selection

### Classes

- interface **Join**
- interface **Order**

### Public Member Functions

- **Selection** (Object selectFrom, Object[] select)
- **Selection** (Object selectFrom, **Condition** condition, Object[] select)
- **Selection nestMe** (Col... outerSelect)
- void **top** (Object top)
- void **join** ( **@ Join** String joinType, **Table** joinWith, **Condition** condition, Col... groupBy)
- void **orderBy** ( **Col** col, **@ Order** String order)

### Protected Member Functions

- String **createStatement** ()

### Additional Inherited Members

#### 1.60.1 Detailed Description

**Selection** (p. 202) - a selection sql statement.

#### Author

Bezalel Avrahami ( bezalel3250@gmail.com)

#### 1.60.2 Class Documentation

**1.60.2.1 interface ver14::SharedClasses::DBActions::Statements::Selection::Join** **Join** (p. ??) - selection join.

#### Author

Bezalel Avrahami ( bezalel3250@gmail.com)

## Class Members

String	LEFT	The constant LEFT.
--------	------	--------------------

**1.60.2.2 interface ver14::SharedClasses::DBActions::Statements::Selection::Order** **Order** (p. ??) - selection order by.

## Author

Bezalel Avrahami ( bezalel3250@gmail.com)

## Class Members

String	ASC	The constant ASC.
String	DESC	The constant DESC.

## 1.60.3 Constructor &amp; Destructor Documentation

**1.60.3.1 Selection()** [1/2] **Selection** (  
     Object *selectFrom*,  
     Object[] *select* )

Instantiates a new **Selection** (p. 202).

## Parameters

<i>selectFrom</i>	the select from
<i>select</i>	the select

**1.60.3.2 Selection()** [2/2] **Selection** (  
     Object *selectFrom*,  
     **Condition** *condition*,  
     Object[] *select* )

Instantiates a new **Selection** (p. 202).

## Parameters

<i>selectFrom</i>	the select from
<i>condition</i>	the condition
<i>select</i>	the select

## 1.60.4 Member Function Documentation

### 1.60.4.1 `createStatement()` `String createStatement ( ) [protected]`

Create statement string.

#### Returns

the string

Reimplemented from **SQLStatement** (p.211).

### 1.60.4.2 `join()` `void join (` `@ Join String joinType,` `Table joinWith,` `Condition condition,` `Col... groupBy )`

**Join** (p. ??).

#### Parameters

<i>joinType</i>	the join type
<i>joinWith</i>	the join with
<i>condition</i>	the condition
<i>groupBy</i>	the group by

### 1.60.4.3 `nestMe()` `Selection nestMe (` `Col... outerSelect )`

Nest me selection.

#### Parameters

<i>outerSelect</i>	the outer select
--------------------	------------------

#### Returns

the selection

**1.60.4.4 orderBy()** `void orderBy (`  
     `Col col,`  
     `@ Order String order )`

**Order** (p. ??) by.

Parameters

<i>col</i>	the col
<i>order</i>	the order

**1.60.4.5 top()** `void top (`  
     `Object top )`

Top.

Parameters

<i>top</i>	the top
------------	---------

## 1.61 CastlingRights.Side Enum Reference

### Public Member Functions

- **Side** (int **castledKingCol**, int **rookStartingCol**, int **castledRookCol**)
- String **toString** ()
- **Location** **kingFinalLoc** ( **Location** currentKingLoc)

### Public Attributes

- **KING**
- **QUEEN**
- final int **rookStartingCol**
- final int **castledRookCol**
- final int **castledKingCol**
- final int **kingTravelDistance**
- final String **castlingNotation**
- final int **asInt**
- final int **mult**

### Static Public Attributes

- static final **Side[] SIDES**

### 1.61.1 Detailed Description

**Side** (p. 205) - Castling side.

Author

Bezalel Avrahami ( bezalel3250@gmail.com)

### 1.61.2 Constructor & Destructor Documentation

**1.61.2.1 Side()** `Side (`  
    `int castledKingCol,`  
    `int rookStartingCol,`  
    `int castledRookCol )`

Instantiates a new **Side** (p. 205).

Parameters

<i>castledKingCol</i>	the castled king col
<i>rookStartingCol</i>	the rook starting col
<i>castledRookCol</i>	the castled rook col

### 1.61.3 Member Function Documentation

**1.61.3.1 kingFinalLoc()** `Location kingFinalLoc (`  
    `Location currentKingLoc )`

King final loc location.

Parameters

<i>currentKingLoc</i>	the current king loc
-----------------------	----------------------

Returns

the location

**1.61.3.2 toString()** `String toString ( )`

To string string.

## Returns

the string @hidden

### 1.61.4 Member Data Documentation

#### 1.61.4.1 **asInt** `final int asInt`

The As int.

#### 1.61.4.2 **castledKingCol** `final int castledKingCol`

The Castled king col.

#### 1.61.4.3 **castledRookCol** `final int castledRookCol`

The Castled rook col.

#### 1.61.4.4 **castlingNotation** `final String castlingNotation`

The Castling notation.

#### 1.61.4.5 **KING** `KING`

King side.

#### 1.61.4.6 **kingTravelDistance** `final int kingTravelDistance`

The King travel distance.

#### 1.61.4.7 **mult** `final int mult`

The King Movement **Direction** (p. 63) Mult

#### 1.61.4.8 **QUEEN** `QUEEN`

Queen side.

#### 1.61.4.9 **rookStartingCol** `final int rookStartingCol`

The Rook starting col.

#### 1.61.4.10 SIDES `final Side [] SIDES [static]`

The constant SIDES.

## 1.62 GameStatus.SpecificStatus Enum Reference

### Public Member Functions

- **SpecificStatus ()**
- **SpecificStatus ( GameStateType gameStatusType)**
- **String toString ()**

### Public Attributes

- **Checkmate**
- **TimedOut**
- **TimedOutVsInsufficientMaterial**
- **Resignation**
- **GameGoesOn**
- **ThreeFoldRepetition**
- **Stalemate**
- **InsufficientMaterial**
- **FiftyMoveRule**
- **TieByAgreement**
- **PlayerDisconnectedVsAi**
- **PlayerDisconnectedVsReal**
- **ServerStoppedGame**
- **final GameStateType gameStatusType**

### 1.62.1 Detailed Description

Specific status - specific game status .

#### Author

Bezalel Avrahami ( [bezalel3250@gmail.com](mailto:bezalel3250@gmail.com))

### 1.62.2 Constructor & Destructor Documentation

#### 1.62.2.1 SpecificStatus() [1/2] `SpecificStatus ( )`

Instantiates a new Specific status.

#### 1.62.2.2 SpecificStatus() [2/2] `SpecificStatus ( GameStateType gameStatusType )`

Instantiates a new Specific status.



## Parameters

<i>gameStatusType</i>	the game status type
-----------------------	----------------------

### 1.62.3 Member Function Documentation

#### 1.62.3.1 toString() `String toString ( )`

To string string.

## Returns

the string

### 1.62.4 Member Data Documentation

#### 1.62.4.1 Checkmate `Checkmate`

Checkmate.

#### 1.62.4.2 FiftyMoveRule `FiftyMoveRule`

Fifty move rule.

#### 1.62.4.3 GameGoesOn `GameGoesOn`

Game goes on .

#### 1.62.4.4 gameStatusType `final GameStateType gameStatusType`

The Game status type.

#### 1.62.4.5 InsufficientMaterial `InsufficientMaterial`

Insufficient material.

#### 1.62.4.6 PlayerDisconnectedVsAi `PlayerDisconnectedVsAi`

The Player disconnected vs ai.

**1.62.4.7 PlayerDisconnectedVsReal** `PlayerDisconnectedVsReal`

The Player disconnected vs real.

**1.62.4.8 Resignation** `Resignation`

Resignation.

**1.62.4.9 ServerStoppedGame** `ServerStoppedGame`

Server stopped game.

**1.62.4.10 Stalemate** `Stalemate`

Stalemate.

**1.62.4.11 ThreeFoldRepetition** `ThreeFoldRepetition`

Three fold repetition.

**1.62.4.12 TieByAgreement** `TieByAgreement`

The Tie by agreement.

**1.62.4.13 TimedOut** `TimedOut`

Timed out.

**1.62.4.14 TimedOutVsInsufficientMaterial** `TimedOutVsInsufficientMaterial`

Timed out vs insufficient material.

## 1.63 SQLStatement

### Public Member Functions

- **SQLStatement** (DBRequest.Type **type**)
- void **replace** (String replacing, String replaceWith)
- String **getStatement** ()
- String **toString** ()

### Public Attributes

- final DBRequest.Type **type**

**Protected Member Functions**

- abstract String **createStatement** ()

**1.63.1 Detailed Description**

Sql statement - represents an sql statement.

**Author**

Bezalel Avrahami ( bezalel3250@gmail.com)

**1.63.2 Constructor & Destructor Documentation****1.63.2.1 SQLStatement()** `SQLStatement ( DBRequest.Type type )`

Instantiates a new Sql statement.

**Parameters**

<i>type</i>	the type
-------------	----------

**1.63.3 Member Function Documentation****1.63.3.1 createStatement()** `abstract String createStatement ( ) [abstract], [protected]`

Create statement string.

**Returns**

the string

Reimplemented in **CustomStatement** (p. 56), **Delete** (p. 63), **Selection** (p. 204), and **Update** (p. 240).

**1.63.3.2 getStatement()** `String getStatement ( )`

Gets statement.

**Returns**

the statement

**1.63.3.3 replace()** `void replace (`  
    `String replacing,`  
    `String replaceWith )`

Replace.

Parameters

<i>replacing</i>	the replacing
<i>replaceWith</i>	the replace with

**1.63.3.4 toString()** `String toString ( )`

To string string.

Returns

the string

## 1.63.4 Member Data Documentation

**1.63.4.1 type** `final DBRequest.Type type`

The request Type.

## 1.64 Square

### Public Member Functions

- **Square** ( **Location** loc)
- void **setEmpty** ()
- **Square** ( **Piece** piece, **Location** loc)
- **Piece** **getPiece** ()
- void **setPiece** ( **Piece** piece)
- **Location** **getLoc** ()
- String **toString** ()
- String **getFen** ()
- boolean **isEmpty** ()
- String **getPiecelcon** ()

### Static Public Attributes

- static final **Piece** **EMPTY\_PIECE**

### 1.64.1 Detailed Description

**Square** (p. 212) represents a square on the logic board.

Author

Bezalel Avrahami ( [bezalel13250@gmail.com](mailto:bezalel13250@gmail.com) )

### 1.64.2 Constructor & Destructor Documentation

#### 1.64.2.1 Square() [1/2] Square ( Location loc )

Instantiates a new **Square** (p. 212).

Parameters

<i>loc</i>	the loc
------------	---------

#### 1.64.2.2 Square() [2/2] Square ( Piece piece, Location loc )

Instantiates a new **Square** (p. 212).

Parameters

<i>piece</i>	the piece
<i>loc</i>	the loc

### 1.64.3 Member Function Documentation

#### 1.64.3.1 getFen() String getFen ( )

Gets fen.

Returns

the fen

**1.64.3.2 getLoc()** `Location getLoc ( )`

Gets loc.

**Returns**

the loc

**1.64.3.3 getPiece()** `Piece getPiece ( )`

Gets piece.

**Returns**

the piece

**1.64.3.4 getPieceIcon()** `String getPieceIcon ( )`

Gets piece icon.

**Returns**

the piece icon

**1.64.3.5 isEmpty()** `boolean isEmpty ( )`

Is empty boolean.

**Returns**

the boolean

**1.64.3.6 setEmpty()** `void setEmpty ( )`

Sets empty.

**1.64.3.7 setPiece()** `void setPiece (   
 Piece piece )`

Sets piece.

## Parameters

<i>piece</i>	the piece
--------------	-----------

**1.64.3.8 toString()** `String toString ( )`

To string string.

## Returns

the string

**1.64.4 Member Data Documentation****1.64.4.1 EMPTY\_PIECE** `final Piece EMPTY_PIECE [static]`

The constant EMPTY\_PIECE.

**1.65 StatusResponse****Public Member Functions**

- **StatusResponse** ( **Status** status, **DBRequest** request, int updatedRows)
- **StatusResponse** ( **Status** status, String details, **DBRequest** request, int updatedRows)
- String **getDetails** ()
- boolean **isAnyData** ()
- **DBResponse** **clean** ()

**Additional Inherited Members****1.65.1 Detailed Description**

Status response - .

## Author

Bezalel Avrahami ( bezalel3250@gmail.com)

**1.65.2 Constructor & Destructor Documentation****1.65.2.1 StatusResponse() [1/2]** `StatusResponse (`  
`Status status,`  
`DBRequest request,`  
`int updatedRows )`

Instantiates a new Status response.

**Parameters**

<i>status</i>	the status
<i>request</i>	the request
<i>updatedRows</i>	the updated rows

**1.65.2.2 StatusResponse() [2/2]**    **StatusResponse** (

```
    Status status,  
    String details,  
    DBRequest request,  
    int updatedRows )
```

Instantiates a new Status response.

**Parameters**

<i>status</i>	the status
<i>details</i>	the details
<i>request</i>	the request
<i>updatedRows</i>	the updated rows

**1.65.3 Member Function Documentation****1.65.3.1 clean()**    **DBResponse** **clean** ( )

Clean db response.

**Returns**

the db response

Reimplemented from **DBResponse** (p. 60).

**1.65.3.2 getDetails()**    **String** **getDetails** ( )

Gets details.

**Returns**

the details



### 1.65.3.3 isAnyData() `boolean isAnyData ( )`

Is any data boolean.

#### Returns

the boolean

Reimplemented from **DBResponse** (p.61).

## 1.66 MyJFrame.StringClosing

### Public Member Functions

- default String **show** ()
- default boolean **checkClosingVal** (String val)

### Additional Inherited Members

#### 1.66.1 Detailed Description

The interface String closing.

#### 1.66.2 Member Function Documentation

##### 1.66.2.1 checkClosingVal() `default boolean checkClosingVal (String val )`

Check closing val boolean.

#### Parameters

<i>val</i>	the val
------------	---------

#### Returns

the boolean

##### 1.66.2.2 show() `default String show ( )`

Show string.

#### Returns

the string

## 1.67 StrUtils

### Classes

- interface **IterationThingy**

### Static Public Member Functions

- static String **dontCapFull** (String str)
- static String **htmlNewLines** (String str)
- static int **countMatches** (String str, @Language("RegExp") String match)
- static String[][] **format** (String[][] mat)
- static String[] **format** (String[] arr)
- static String **format** (String str)
- static boolean **isEmpty** (String str)
- static String **formatDate** (String longStr)
- static String **formatDate** (Date date)
- static String **formatDate** (Date date, String **format**)
- static String **parseURLS** (String str)
- static String **uppercase** (String str)
- static String **dontCapWord** (String str)
- static String **formatDateSQL** (Date date)
- static int **getPort** (Socket socket)
- static int **getPort** (SocketAddress socketAddress)
- static String **getUrl** (Socket socket)
- static String **getUrl** (SocketAddress socketAddress)
- static String **fitInside** (String str, JComponent comp)
- static String **fixHtml** (String str)
- static String **strINN** (Object... objs)
- static String **splitArr** (Object[] arr)
- static String **splitArr** (String divide, Object[] arr)
- static String **splitArr** (String divide, Object[] arr, boolean **format**)
- static String **clean** (String str)
- static String **createTimeGreeting** ()
- static void **main** (String[] args)
- static String **createTimeStr** (long millis)
- static String **awful** (String og)
- static String **dateTimePrefix** (String str)
- static String **repeat** ( **IterationThingy**< String > iterationThingy, int numOfIterations)

### 1.67.1 Detailed Description

Str utils.

#### Author

Bezalel Avrahami ( bezalel3250@gmail.com)

### 1.67.2 Member Function Documentation

**1.67.2.1 awful()** static String awful (  
String og ) [static]

Awful string.

## Parameters

<i>og</i>	the og
-----------	--------

## Returns

the string

**1.67.2.2 clean()** `static String clean (String str ) [static]`

Clean string.

## Parameters

<i>str</i>	the str
------------	---------

## Returns

the string

**1.67.2.3 countMatches()** `static int countMatches (String str, @Language("RegExp") String match ) [static]`

Count matches int.

## Parameters

<i>str</i>	the str
<i>match</i>	the match

## Returns

the int

**1.67.2.4 createTimeGreeting()** `static String createTimeGreeting ( ) [static]`

Create time greeting string.

## Returns

the string

**1.67.2.5 createTimeStr()** `static String createTimeStr (`  
`long millis ) [static]`

Create time str string.

**Parameters**

<i>millis</i>	the millis
---------------	------------

**Returns**

the string

**1.67.2.6 dateTimePrefix()** `static String dateTimePrefix (`  
`String str ) [static]`

Date time prefix string.

**Parameters**

<i>str</i>	the str
------------	---------

**Returns**

the string

**1.67.2.7 dontCapFull()** `static String dontCapFull (`  
`String str ) [static]`

Dont cap full string.

**Parameters**

<i>str</i>	the str
------------	---------

**Returns**

the string

**1.67.2.8 dontCapWord()** `static String dontCapWord (`  
`String str ) [static]`

Dont cap word string.

## Parameters

<i>str</i>	the str
------------	---------

## Returns

the string

**1.67.2.9 fitInside()** `static String fitInside (`  
    `String str,`  
    `JComponent comp ) [static]`

Fit inside string.

## Parameters

<i>str</i>	the str
<i>comp</i>	the comp

## Returns

the string

**1.67.2.10 fixHtml()** `static String fixHtml (`  
    `String str ) [static]`

Fix html string.

## Parameters

<i>str</i>	the str
------------	---------

## Returns

the string

**1.67.2.11 format()** `[1/3] static String format (`  
    `String str ) [static]`

Format string.

**Parameters**

<i>str</i>	the str
------------	---------

**Returns**

the string

**1.67.2.12 format()** [2/3] `static String[] format (String[] arr) [static]`

Format string [ ].

**Parameters**

<i>arr</i>	the arr
------------	---------

**Returns**

the string [ ]

**1.67.2.13 format()** [3/3] `static String[][] format (String mat[][]) [static]`

Format string [ ][ ].

**Parameters**

<i>mat</i>	the mat
------------	---------

**Returns**

the string [ ][ ]

**1.67.2.14 formatDate()** [1/3] `static String formatDate (Date date) [static]`

Format date string.

**Parameters**

<i>date</i>	the date
-------------	----------

**Returns**

the string

**1.67.2.15 formatDate()** [2/3] static String formatDate (  
    Date *date*,  
    String *format* ) [static]

Format date string.

**Parameters**

<i>date</i>	the date
<i>format</i>	the format

**Returns**

the string

**1.67.2.16 formatDate()** [3/3] static String formatDate (  
    String *longStr* ) [static]

Format date string.

**Parameters**

<i>longStr</i>	the long str
----------------	--------------

**Returns**

the string

**1.67.2.17 formatDateSQL()** static String formatDateSQL (  
    Date *date* ) [static]

Format date sql string.

**Parameters**

<i>date</i>	the date
-------------	----------

**Returns**

the string

**1.67.2.18** **getPort()** [1/2] `static int getPort (`  
`Socket socket ) [static]`

Gets port.

**Parameters**

<code>socket</code>	the socket
---------------------	------------

**Returns**

the port

**1.67.2.19** **getPort()** [2/2] `static int getPort (`  
`SocketAddress socketAddress ) [static]`

Gets port.

**Parameters**

<code>socketAddress</code>	the socket address
----------------------------	--------------------

**Returns**

the port

**1.67.2.20** **getUrl()** [1/2] `static String getUrl (`  
`Socket socket ) [static]`

Gets url.

**Parameters**

<code>socket</code>	the socket
---------------------	------------

**Returns**

the url



**1.67.2.21 getUrl()** [2/2] `static String getUrl (`  
`SocketAddress socketAddress ) [static]`

Gets url.

**Parameters**

<code>socketAddress</code>	the socket address
----------------------------	--------------------

**Returns**

the url

**1.67.2.22 htmlNewLines()** `static String htmlNewLines (`  
`String str ) [static]`

Html new lines string.

**Parameters**

<code>str</code>	the str
------------------	---------

**Returns**

the string

**1.67.2.23 isEmpty()** `static boolean isEmpty (`  
`String str ) [static]`

Is empty boolean.

**Parameters**

<code>str</code>	the str
------------------	---------

**Returns**

the boolean

**1.67.2.24 main()** `static void main (`  
`String[] args ) [static]`

The entry point of application.

**Parameters**

<i>args</i>	the input arguments
-------------	---------------------

**1.67.2.25 parseURLS()** `static String parseURLS (String str ) [static]`

Parse urls string.

**Parameters**

<i>str</i>	the str
------------	---------

**Returns**

the string

**1.67.2.26 repeat()** `static String repeat ( IterationThingy< String > iterationThingy, int numOfIterations ) [static]`

Repeat string.

**Parameters**

<i>iterationThingy</i>	the iteration thingy
<i>numOfIterations</i>	the num of iterations

**Returns**

the string

**1.67.2.27 splitArr()** `[1/3] static String splitArr ( Object[] arr ) [static]`

Split arr string.

**Parameters**

<i>arr</i>	the arr
------------	---------

**Returns**

the string

**1.67.2.28 splitArr()** [2/3] `static String splitArr (`  
    `String divide,`  
    `Object[] arr ) [static]`

Split arr string.

**Parameters**

<i>divide</i>	the divide
<i>arr</i>	the arr

**Returns**

the string

**1.67.2.29 splitArr()** [3/3] `static String splitArr (`  
    `String divide,`  
    `Object[] arr,`  
    `boolean format ) [static]`

Split arr string.

**Parameters**

<i>divide</i>	the divide
<i>arr</i>	the arr
<i>format</i>	the format

**Returns**

the string

**1.67.2.30 strINN()** `static String strINN (`  
    `Object... objs ) [static]`

Str inn string.

**Parameters**

<i>objs</i>	the objs
-------------	----------

**Returns**

the string

**1.67.2.31 uppercase()** `static String uppercase (String str) [static]`

Uppercase string.

**Parameters**

<i>str</i>	the str
------------	---------

**Returns**

the string

## 1.68 SwitchCase

**Public Member Functions**

- **SwitchCase** ( **Condition** condition, **Col** ifTrue)
- **Condition** condition ()
- **Col** ifTrue ()
- String **toString** ()

**Static Public Member Functions**

- static **SwitchCase** **equals** ( **Col** col, String value, **Col** ifTrue)
- static **SwitchCase** **defaultCase** ( **Col** ifTrue)

### 1.68.1 Detailed Description

Switch case - represents a case that is meant to be used inside a switch case col. if the **condition** (p.??) is true, the **ifTrue** (p.??) col will display in the switch case col

**Author**

Bezalel Avrahami ( bezalel3250@gmail.com)

### 1.68.2 Constructor & Destructor Documentation

**1.68.2.1 SwitchCase()** `SwitchCase (Condition condition, Col ifTrue )`

Instantiates a new Switch case.

## Parameters

<i>condition</i>	the condition
<i>ifTrue</i>	the if true

### 1.68.3 Member Function Documentation

#### 1.68.3.1 condition() `Condition condition ( )`

**Condition** (p. 43) condition.

## Returns

the condition

#### 1.68.3.2 defaultCase() `static SwitchCase defaultCase ( Col ifTrue ) [static]`

Default case switch case.

## Parameters

<i>ifTrue</i>	the if true
---------------	-------------

## Returns

the switch case

#### 1.68.3.3 equals() `static SwitchCase equals ( Col col, String value, Col ifTrue ) [static]`

Equals switch case.

## Parameters

<i>col</i>	the col
<i>value</i>	the value
<i>ifTrue</i>	the if true

**Returns**

the switch case

**1.68.3.4 ifTrue()** `Col ifTrue ( )`

If true col.

**Returns**

the col

**1.68.3.5 toString()** `String toString ( )`

To string string.

**Returns**

the string

## 1.69 Table Enum Reference

**Public Member Functions**

- **Table** (Col... cols)
- String **tableAndValues** ()

**Static Public Member Functions**

- static String **escapeValues** (Object[] values, boolean quotes, boolean parentheses)

**Public Attributes**

- **Games**
- **UnfinishedGames**
- **Users**
- final **Col[] cols**

### 1.69.1 Detailed Description

**Table** (p. 230) - represents the tables in the db.

**Author**

Bezalel Avrahami ( bezalel3250@gmail.com)

### 1.69.2 Constructor & Destructor Documentation

**1.69.2.1 Table()** `Table ( Col... cols )`

Instantiates a new **Table** (p. 230).

## Parameters

<i>cols</i>	the cols
-------------	----------

## 1.69.3 Member Function Documentation

**1.69.3.1 escapeValues()** `static String escapeValues (`  
     `Object[] values,`  
     `boolean quotes,`  
     `boolean parentheses ) [static]`

Escape values string.

## Parameters

<i>values</i>	the values
<i>quotes</i>	quotes
<i>parentheses</i>	parentheses

## Returns

the escaped string

**1.69.3.2 tableAndValues()** `String tableAndValues ( )`

**Table** (p. 230) and values string.

## Returns

the string

## 1.69.4 Member Data Documentation

**1.69.4.1 cols** `final Col [] cols`

The Cols in this table

**1.69.4.2 Games** `Games`

Games table.

#### 1.69.4.3 UnfinishedGames `UnfinishedGames`

Unfinished games table.

#### 1.69.4.4 Users `Users`

Users table.

### 1.70 TableDBResponse

#### Public Member Functions

- **TableDBResponse** (`String[] columns`, `String[][] rows`, `DBRequest request`)
- **TableDBResponse** (`String[] columns`, `String[][] rows`, `Status status`, `DBRequest request`)
- `int numOfRows ()`
- `String[] getFirstRow ()`
- `String getCell (int row, String col)`
- `String getCell (int row, Col col)`
- `boolean isAnyData ()`
- **TableDBResponse clean ()**
- `String toString ()`
- `String rowToString (String[] row)`
- `String[] getColumns ()`
- `String[][] getRows ()`

#### Protected Member Functions

- **TableDBResponse ()**
- `int getColumnIndex (String column)`

#### Protected Attributes

- `String[] columns`
- `String[][] rows`

#### 1.70.1 Detailed Description

Table db response - .

##### Author

Bezalel Avrahami ( `bezalel3250@gmail.com`)

#### 1.70.2 Constructor & Destructor Documentation

**1.70.2.1 TableDBResponse() [1/3]** `TableDBResponse (`  
    `String[] columns,`  
    `String rows[][],`  
    `DBRequest request )`

Instantiates a new Table db response.



## Parameters

<i>columns</i>	the columns
<i>rows</i>	the rows
<i>request</i>	the request

**1.70.2.2 TableDBResponse()** [2/3] **TableDBResponse** (

```
String[] columns,
String rows[][],
    Status status,
    DBRequest request )
```

Instantiates a new Table db response.

## Parameters

<i>columns</i>	the columns
<i>rows</i>	the rows
<i>status</i>	the status
<i>request</i>	the request

**1.70.2.3 TableDBResponse()** [3/3] **TableDBResponse** ( ) [protected]

Instantiates a new Table db response.

**1.70.3 Member Function Documentation****1.70.3.1 clean()** **TableDBResponse** clean ( )

Clean table db response.

## Returns

the table db response

Reimplemented from **DBResponse** (p. 60).

**1.70.3.2 getCell()** [1/2] **String** getCell (

```
int row,
    Col col )
```

Gets cell.

**Parameters**

<i>row</i>	the row
<i>col</i>	the col

**Returns**

the cell

```
1.70.3.3 getCell() [2/2] String getCell (  
    int row,  
    String col )
```

Gets cell.

**Parameters**

<i>row</i>	the row
<i>col</i>	the col

**Returns**

the cell

```
1.70.3.4 getColumnIndex() int getColumnIndex (  
    String column ) [protected]
```

Gets column index.

**Parameters**

<i>column</i>	the column
---------------	------------

**Returns**

the column index

```
1.70.3.5 getColumns() String[] getColumns ( )
```

Get columns string [ ].

**Returns**

the string [ ]

**1.70.3.6 getFirstRow()** `String[] getFirstRow ( )`

Get first row string [ ].

**Returns**

the string [ ]

**1.70.3.7 getRows()** `String[][] getRows ( )`

Get rows string [ ] [ ].

**Returns**

the string [ ] [ ]

**1.70.3.8 isAnyData()** `boolean isAnyData ( )`

Is any data boolean.

**Returns**

the boolean

Reimplemented from **DBResponse** (p. 61).

**1.70.3.9 numOfRows()** `int numOfRows ( )`

Num of rows int.

**Returns**

the int

**1.70.3.10 rowToString()** `String rowToString ( String[] row )`

Row to string string.

**Parameters**

<i>row</i>	the row
------------	---------

**Returns**

the string

**1.70.3.11 toString()** `String toString ( )`

To string string.

**Returns**

the string

Reimplemented from **DBResponse** (p. 62).

**1.70.4 Member Data Documentation****1.70.4.1 columns** `String [] columns [protected]`

The Columns.

**1.70.4.2 rows** `String [][] rows [protected]`

The Rows.

**1.71 ThreadsManager****Static Public Member Functions**

- static void **handleErrors** (ThrowingRunnable runnable)
- static **MyThread createThread** (ThrowingRunnable runnable, boolean start)

**1.71.1 Detailed Description**

The type Threads manager.

**1.71.2 Member Function Documentation****1.71.2.1 createThread()** `static MyThread createThread (`  
    `ThrowingRunnable runnable,`  
    `boolean start ) [static]`

Create thread my thread.

## Parameters

<i>runnable</i>	the runnable
<i>start</i>	the start

## Returns

the my thread

**1.71.2.2 handleError()** `static void handleError ( ThrowingRunnable runnable ) [static]`

Handle errors.

## Parameters

<i>runnable</i>	the runnable
-----------------	--------------

## 1.72 ThrowingCallback&lt; T &gt;

## Public Member Functions

- void **callback** (T obj) throws Exception

## 1.72.1 Detailed Description

Throwing callback - a callback that might throw an exception .

## Parameters

< T >	the callback type
-------	-------------------

## Author

Bezalel Avrahami ( `bezalel13250@gmail.com` )

## 1.72.2 Member Function Documentation

**1.72.2.1 callback()** `void callback ( T obj ) throws Exception`

**Callback** (p. 26).

**Parameters**

<i>obj</i>	the obj
------------	---------

**Exceptions**

<i>Exception</i>	the exception
------------------	---------------

## 1.73 UnfinishedGame

**Public Member Functions**

- **UnfinishedGame** (String **gameId**, String **creatorUsername**, GameSettings **gameSettings**, String **opponentUsername**, **PlayerColor** **playerColorToMove**, String **playerToMove**, Stack< **Move** > moveStack)
- boolean **isCreatorToMove** ()

**Public Attributes**

- final **PlayerColor** **playerColorToMove**
- final String **playerToMove**

**Additional Inherited Members**

### 1.73.1 Detailed Description

The type Unfinished game.

**Author**

Bezalel Avrahami ( bezalel3250@gmail.com)

### 1.73.2 Constructor & Destructor Documentation

**1.73.2.1 UnfinishedGame()**    **UnfinishedGame** (  
    String *gameId*,  
    String *creatorUsername*,  
    GameSettings *gameSettings*,  
    String *opponentUsername*,  
    **PlayerColor** *playerColorToMove*,  
    String *playerToMove*,  
    Stack< **Move** > *moveStack* )

Instantiates a new Unfinished game.

## Parameters

<i>gameId</i>	the game id
<i>creatorUsername</i>	the creator username
<i>gameSettings</i>	the game settings
<i>opponentUsername</i>	the opponent username
<i>playerColorToMove</i>	the player color to move
<i>playerToMove</i>	the player to move
<i>moveStack</i>	the move stack

## 1.73.3 Member Function Documentation

1.73.3.1 isCreatorToMove() `boolean isCreatorToMove ( )`

Is creator to move boolean.

## Returns

the boolean

## 1.73.4 Member Data Documentation

1.73.4.1 playerColorToMove `final PlayerColor playerColorToMove`

The Player color to move.

1.73.4.2 playerToMove `final String playerToMove`

The Player to move.

## 1.74 Update

## Public Member Functions

- **Update** ( **Table** updating, **Condition** condition, **NewValue...** newValues)

## Protected Member Functions

- String **createStatement** ()

## Additional Inherited Members

### 1.74.1 Detailed Description

**Update** (p. 239) - a sql update statement.

#### Author

Bezalel Avrahami ( `bezalel3250@gmail.com` )

### 1.74.2 Constructor & Destructor Documentation

**1.74.2.1 Update()** `Update (`  
    **Table** *updating*,  
    **Condition** *condition*,  
    NewValue... *newValues* )

Instantiates a new **Update** (p. 239).

#### Parameters

<i>updating</i>	the updating
<i>condition</i>	the condition
<i>newValues</i>	the new values

### 1.74.3 Member Function Documentation

**1.74.3.1 createStatement()** `String createStatement ( )` [protected]

Create statement string.

#### Returns

the string

Reimplemented from **SQLStatement** (p. 211).

## 1.75 ValueBtn< T >

### Public Member Functions

- **ValueBtn** (String text, Font font, T value, **Callback**< T > onClick)

### 1.75.1 Detailed Description

Value btn - a value holding button.



## Parameters

< T >	the value's type
-------	------------------

## Author

Bezalel Avrahami ( bezalel13250@gmail.com)

## 1.75.2 Constructor &amp; Destructor Documentation

**1.75.2.1 ValueBtn()** `ValueBtn (`  
     *String text*,  
     *Font font*,  
     *T value*,  
     **Callback**< T > *onClick* )

Instantiates a new Value btn.

## Parameters

<i>text</i>	the text
<i>font</i>	the font
<i>value</i>	the value
<i>onClick</i>	the on click

## 1.76 PreMadeRequest.VariationCreator

## Public Member Functions

- Variation **create** ( **RequestBuilder** *actualBuilder*)

## 1.76.1 Detailed Description

Variation creator - .

## Author

Bezalel Avrahami ( bezalel13250@gmail.com)

## 1.76.2 Member Function Documentation

**1.76.2.1 create()** `Variation create (`  
     **RequestBuilder** *actualBuilder* )

Create variation.

**Parameters**

<i>actualBuilder</i>	the actual builder
----------------------	--------------------

**Returns**

the variation

## 1.77 VoidCallback

**Public Member Functions**

- void **callback** ()

### 1.77.1 Detailed Description

Void callback - .

**Author**

Bezalel Avrahami ( `bezalel13250@gmail.com`)

### 1.77.2 Member Function Documentation

#### 1.77.2.1 **callback()** `void callback ( )`

**Callback** (p. 26).

## Index

- add
  - Condition, 44
- addDetail
  - Evaluation, 75
- addedRes
  - DBResponse, 62
- addShouldSync
  - RequestBuilder, 197
- addSuggestion
  - Config< V >, 51
- ALL\_DIRECTIONS
  - Direction, 66
- ALL\_PIECES
  - Piece, 173
- ALL\_USED\_DIRECTIONS
  - Direction, 66
- and
  - Condition, 45
- andWith
  - Direction, 66
- annotate
  - MoveAnnotation, 157, 158
- annotation
  - GameStatus.GameStatusType, 92
- Answer
  - Question, 187
- AnswerCallback, 1
- answerQuestion
  - Message, 116
- Any
  - RegEx, 193
- apply
  - Math, 110
- AppSocket, 2
  - AppSocket, 2, 3
  - close, 3
  - getLocalAddress, 3
  - getMessagesHandler, 4
  - getRemoteAddress, 4
  - handledRun, 4
  - interruptListener, 4
  - isClosed, 4
  - isConnected, 5
  - msgSocket, 6
  - requestMessage, 5
  - respond, 5
  - setMessagesHandler, 6
  - stopReading, 6
  - writeMessage, 6
- ArchivedGameInfo, 6
  - ArchivedGameInfo, 7
  - getWinner, 7
  - toString, 7
- Arg, 8
  - Arg, 8, 9
  - argType, 11
  - config, 11
  - createVal, 9
  - equals, 10
  - escape, 11
  - isUserInput, 10
  - replnStr, 11
  - setUserInput, 10
  - toString, 10
- args
  - RequestBuilder, 201
- ArgsUtil, 11
  - create, 11
  - equalsSign, 12
  - OptionalArg, 12
  - plainTextIgnoreCase, 12
- ArgType, 13
  - ArgType, 13
  - Date, 14
  - DateRange, 14
  - isUserInput, 14
  - Number, 14
  - Password, 14
  - PictureUrl, 14
  - ServerAddress, 14
  - Text, 14
  - Url, 14
  - Username, 14
- argType
  - Arg, 11
- ArrUtils, 15
  - concat, 15
  - createList, 15
  - exists, 16
- as
  - Col, 36
- asFloat
  - Math, 110
- asInt
  - CastlingRights.Side, 207
  - Direction, 66
  - Location, 104
  - PieceType, 179
  - PlayerColor, 183
- askForGameSettings
  - Message, 117
- askForLogin
  - Message, 117
- askQuestion
  - Message, 117
- asLong
  - Location, 104
- assertNotGameOver
  - Evaluation, 75
- asUser

- LoginInfo, 105
- LoginType, 108
- ATTACKING\_DIRECTIONS
  - PieceType, 179
- ATTACKING\_PIECE\_TYPES
  - PieceType, 179
- authSettings
  - PreMadeRequest, 185
- awful
  - StrUtils, 218
- B\_B
  - Piece, 173
- B\_K
  - Piece, 173
- B\_N
  - Piece, 173
- B\_P
  - Piece, 173
- B\_Q
  - Piece, 173
- B\_R
  - Piece, 174
- basicAnnotate
  - MoveAnnotation, 158
- BasicMove, 16
  - BasicMove, 17
  - cp, 17
  - createBatch, 17
  - equals, 18
  - flip, 18
  - getBasicMoveAnnotation, 18
  - getFlipped, 18
  - getMovingFrom, 19
  - getMovingTo, 19
  - hashCode, 19
  - setMovingFrom, 19
  - setMovingTo, 20
  - toString, 20
- between
  - Condition, 45
- BISHOP
  - PieceType, 179
- BitData, 20
  - everything, 20
  - notAFile, 20
  - notHFile, 21
- BLACK
  - PlayerColor, 183
- blackIcon
  - PieceType, 179
- blockTilRes
  - MessagesHandler, 136
- Board, 21
  - Board, 21, 22
  - example, 25
  - fenSetup, 22
  - getPiece, 22, 23
  - getRow, 23
  - getSquare, 23
  - isEmpty, 23
  - iterator, 24
  - print, 24
  - setPiece, 24
  - setSquareEmpty, 24
  - startingFen, 25
  - startingPos, 25
  - toString, 25
- book
  - Evaluation, 75
- build
  - RequestBuilder, 197
- bye
  - Message, 118
- callback
  - Callback< T >, 26
  - ThrowingCallback< T >, 237
  - VoidCallback, 242
- Callback< T >, 26
  - callback, 26
- CAN\_PROMOTE\_TO
  - PieceType, 179
- canBeEmpty
  - RegEx, 191
- CANCEL
  - LoginType, 108
- canUseDefault
  - Config< V >, 52
- canUseId
  - IDsGenerator, 95
- castledKingCol
  - CastlingRights.Side, 207
- castledRookCol
  - CastlingRights.Side, 207
- castling
  - Move, 149
- CASTLING\_FLAGS
  - Move.MoveFlag, 160
- castlingNotation
  - CastlingRights.Side, 207
- CastlingRights, 27
  - CastlingRights, 27, 28
  - createFromStr, 28
  - disableCastling, 28
  - enable, 29
  - enableCastling, 29
  - getPlayersCastling, 29
  - getRights, 29
  - hasAny, 30
  - isEnabled, 30
  - main, 30
  - NO\_CASTLING\_ABILITY, 31
  - toString, 31
  - whosCastling, 31
- CastlingRights.Side, 205
  - asInt, 207
  - castledKingCol, 207

- castledRookCol, 207
- castlingNotation, 207
- KING, 207
- kingFinalLoc, 206
- kingTravelDistance, 207
- mult, 207
- QUEEN, 207
- rookStartingCol, 207
- Side, 206
- SIDES, 207
- toString, 206
- castlingSide
  - Move.MoveFlag, 160
- changePassword
  - RequestBuilder, 198
- ChangeProfilePic
  - PreMadeRequest, 185
- changeProfilePic
  - RequestBuilder, 198
- CHECK
  - GameStatus.GameStatusType, 92
- check
  - RegEx, 192
- checkClosingVal
  - MyJFrame.BooleanClosing, 25
  - MyJFrame.Closing< T >, 32
  - MyJFrame.StringClosing, 217
- Checkmate
  - GameStatus.SpecificStatus, 209
- checkmate
  - GameStatus, 85
- checkUsernameAvailability
  - Message, 118
- clean
  - DBResponse, 60
  - StatusResponse, 216
  - StrUtils, 219
  - TableDBResponse, 233
- close
  - AppSocket, 3
- closing
  - MyJFrame.BooleanClosing, 26
  - MyJFrame.Closing< T >, 32
- Col, 33
  - as, 36
  - Col, 34, 36
  - colName, 36
  - count, 37
  - countIf, 37
  - CreatedDateTime, 42
  - date, 38
  - GameID, 42
  - label, 38
  - math, 38
  - nested, 39
  - of, 39
  - Password, 42
  - Player1, 42
  - Player2, 42
  - PlayerToMove, 42
  - ProfilePic, 42
  - replace, 40
  - SavedGame, 42
  - setColName, 40
  - setWrapped, 40
  - sum, 40
  - switchCase, 41
  - time, 41
  - toString, 41
  - Username, 42
  - Winner, 42
  - wrap, 41
- col
  - Location, 104
  - Math, 113
- colName
  - Col, 36
- cols
  - Table, 231
- columns
  - TableDBResponse, 236
- combination
  - Direction, 66
- compareMovementType
  - PieceType, 176
- compareTo
  - MinimaxMove, 144
  - Move, 149
- concat
  - ArrUtils, 15
- Condition, 43
  - add, 44
  - and, 45
  - between, 45
  - Condition, 44
  - equals, 45
  - getStr, 47
  - math, 47
  - noNulls, 47
  - notEquals, 47
  - setStr, 48
  - toString, 48
  - wrap, 48
- condition
  - SwitchCase, 229
- Config
  - Config< V >, 49, 50
- config
  - Arg, 11
- Config< V >, 49
  - addSuggestion, 51
  - canUseDefault, 52
  - Config, 49, 50
  - description, 52
  - getDefault, 51
  - getDefaultDesc, 51

- getDescribedDefault, 51
  - getValuesSuggestion, 52
  - toString, 52
- confirm
  - ConfirmDialogs, 53
- ConfirmDialogs, 52
  - confirm, 53
  - main, 54
- convertFromCentipawns
  - Evaluation, 76
- copyMove
  - Move, 150
- count
  - Col, 37
- countIf
  - Col, 37
- countMatches
  - StrUtils, 219
- cp
  - BasicMove, 17
- create
  - ArgsUtil, 11
  - PreMadeRequest.VariationCreator, 241
- createBatch
  - BasicMove, 17
- createBuilder
  - PreMadeRequest, 184
- createdAt
  - EstablishedGameInfo, 73
- CreatedDateTime
  - Col, 42
- CreatedGame, 54
  - CreatedGame, 54
  - getGameDesc, 55
- createDisconnectedError
  - MessagesHandler, 136
- createFromStr
  - CastlingRights, 28
- createList
  - ArrUtils, 15
- createMatIndicesStr
  - Location, 98
- createResponse
  - RequestBuilder, 198
- createStatement
  - CustomStatement, 56
  - Delete, 63
  - Selection, 204
  - SQLStatement, 211
  - Update, 240
- createThread
  - ThreadsManager, 236
- createTimeGreeting
  - StrUtils, 219
- createTimeStr
  - StrUtils, 219
- createVal
  - Arg, 9
- createVariation
  - RequestBuilder, 198
- creatorUsername
  - GameInfo, 84
- criticalErr
  - EnvManager, 69
- currentThread
  - MyThread, 169
- CustomStatement, 55
  - createStatement, 56
  - CustomStatement, 55
- D
  - Direction, 67
- D\_D
  - Direction, 67
- D\_D\_L
  - Direction, 67
- D\_D\_R
  - Direction, 67
- D\_L
  - Direction, 67
- D\_L\_L
  - Direction, 67
- D\_R
  - Direction, 67
- D\_R\_R
  - Direction, 67
- Date
  - ArgType, 14
- date
  - Col, 38
- DateRange
  - ArgType, 14
- dateTimePrefix
  - StrUtils, 220
- DBRequest, 56
  - DBRequest, 57
  - getBuilder, 58
  - getRequest, 58
  - getSubRequest, 58
  - setSubRequest, 58
  - toString, 58
  - type, 59
- dbRequest
  - Message, 118
- DBResponse, 59
  - addedRes, 62
  - clean, 60
  - DBResponse, 60
  - getAddedRes, 60
  - getRequest, 60
  - getStatus, 61
  - isAnyData, 61
  - isSuccess, 61
  - print, 61
  - request, 62
  - setAddedRes, 61
  - status, 62

- toString, 62
- debugAdapter
  - MyJFrame, 167
- defaultCase
  - SwitchCase, 229
- Delete, 62
  - createStatement, 63
  - Delete, 63
- deleteAllUnFinishedGames
  - RequestBuilder, 199
- DeleteUnfGames
  - PreMadeRequest, 185
- description
  - Config< V >, 52
- diff
  - PlayerColor, 183
- Direction, 63
  - ALL\_DIRECTIONS, 66
  - ALL\_USED\_DIRECTIONS, 66
  - andWith, 66
  - asInt, 66
  - combination, 66
  - D, 67
  - D\_D, 67
  - D\_D\_L, 67
  - D\_D\_R, 67
  - D\_L, 67
  - D\_L\_L, 67
  - D\_R, 67
  - D\_R\_R, 67
  - getCombination, 65
  - getDirectionByOffset, 65
  - getRelative, 65
  - L, 67
  - normalPerspective, 67
  - NUM\_OF\_DIRECTIONS, 67
  - NUM\_OF\_DIRECTIONS\_WO\_KNIGHT, 68
  - NUM\_OF\_KNIGHT\_DIRECTIONS, 68
  - offset, 68
  - opposite, 65
  - perspective, 66
  - R, 68
  - U, 68
  - U\_L, 68
  - U\_L\_L, 68
  - U\_R, 68
  - U\_R\_R, 68
  - U\_U, 68
  - U\_U\_L, 68
  - U\_U\_R, 69
- disableCastling
  - CastlingRights, 28
- Div
  - Math, 113
- dontCapFull
  - StrUtils, 220
- dontCapWord
  - StrUtils, 220
- dontMatch
  - RegEx, 193
- DontSaveGame
  - RegEx, 194
- DoublePawnPush
  - Move.MoveFlag, 160
- doXClick
  - MyJFrame, 167
- drawOffer
  - Question, 188
- EG\_WEIGHT
  - EvaluationParameters, 80
- EMPTY\_PIECE
  - Square, 215
- enable
  - CastlingRights, 29
- enableCastling
  - CastlingRights, 29
- endPress
  - MyJFrame.MyAdapter.HeldDown, 95
- EnPassant
  - Move.MoveFlag, 160
- EnvManager, 69
  - criticalErr, 69
  - handledErr, 69
- equals
  - Arg, 10
  - BasicMove, 18
  - Condition, 45
  - Evaluation, 76
  - GameInfo, 82
  - MinimaxMove, 144
  - Move, 150
  - Move.MoveFlag, 159
  - Question, 188
  - SwitchCase, 229
- equalsSign
  - ArgsUtil, 12
- error
  - Message, 119
- ErrorHandler< E extends MyError >, 70
  - handle, 70
  - ignore, 70
- errToString
  - MyError, 162
- escape
  - Arg, 11
- escapeValues
  - Table, 231
- EstablishedGameInfo, 71
  - createdAt, 73
  - EstablishedGameInfo, 71
  - getCreatedAt, 72
  - getGameDesc, 72
  - getMoveStack, 72
  - opponentUsername, 73
  - setCreatedAt, 72
  - toString, 73

- Evaluation, 73
  - addDetail, 75
  - assertNotGameOver, 75
  - book, 75
  - convertFromCentipawns, 76
  - equals, 76
  - Evaluation, 74, 75
  - EvaluationDetail, 76
  - flipEval, 76
  - getEval, 76
  - getEvaluationDepth, 77
  - getEvaluationFor, 77
  - getGameStatus, 77
  - isCheck, 77
  - isGameOver, 77
  - isGreaterThan, 77
  - LOSS\_EVAL, 79
  - print, 78
  - setEval, 78
  - setEvaluationDepth, 78
  - setPerspective, 78
  - TIE\_EVAL, 79
  - toString, 79
  - WIN\_EVAL, 79
- EvaluationDetail
  - Evaluation, 76
- EvaluationParameters, 79
  - EG\_WEIGHT, 80
  - FORCE\_KING\_TO\_CORNER, 80
  - HANGING\_PIECES, 80
  - KING\_SAFETY, 80
  - MATERIAL, 80
  - MOVEMENT\_ABILITY, 80
  - PIECE\_TABLES, 80
  - SQUARE\_CONTROL, 80
  - STOCKFISH\_SAYS, 80
  - weight, 81
- everything
  - BitData, 20
- example
  - Board, 25
  - GameInfo, 82
- execute
  - Math, 110, 111
- exists
  - ArrUtils, 16
- Fen
  - RegEx, 194
- fenSetup
  - Board, 22
- FiftyMoveRule
  - GameStatus.SpecificStatus, 209
- fiftyMoveRule
  - GameStatus, 85
- fitInside
  - StrUtils, 221
- fixHtml
  - StrUtils, 221
- flip
  - BasicMove, 18
  - Location, 98
- flipEval
  - Evaluation, 76
- flipMove
  - Move, 150
- FORCE\_KING\_TO\_CORNER
  - EvaluationParameters, 80
- format
  - StrUtils, 221, 222
- formatDate
  - StrUtils, 222, 223
- formatDateSQL
  - StrUtils, 223
- formatNum
  - Math, 111
- GAME\_GOES\_ON
  - GameStatus.GameStatusType, 92
- GameGoesOn
  - GameStatus.SpecificStatus, 209
- gameGoesOn
  - GameStatus, 85
- GameID
  - Col, 42
- gameld
  - GameInfo, 84
- GameInfo, 81
  - creatorUsername, 84
  - equals, 82
  - example, 82
  - gameld, 84
  - GameInfo, 81
  - gameSettings, 84
  - getGameDesc, 82
  - getJoiningPlayerColor, 82
  - getStartingColor, 83
  - ID, 83
  - isCreator, 83
  - toString, 83
- gameOver
  - Message, 119
- gameOverStr
  - GameStatus.GameStatusType, 92
- Games
  - PreMadeRequest, 185
  - Table, 231
- games
  - RequestBuilder, 199
- gameSettings
  - GameInfo, 84
- GameStatus, 84
  - checkmate, 85
  - fiftyMoveRule, 85
  - gameGoesOn, 85
  - getCheckedKingLoc, 86
  - getDetailedStr, 86
  - getGameStatusType, 86



- getSpecificStatus, 87
- getWinningColor, 87
- insufficientMaterial, 87
- isCheck, 87
- isDisconnected, 87
- isGameOver, 88
- playerDisconnected, 88
- playerResigned, 88
- serverStoppedGame, 89
- setCustomStr, 89
- setDepth, 89
- setInCheck, 89
- stalemate, 90
- threeFoldRepetition, 90
- tieByAgreement, 90
- timedOut, 90
- toString, 91
- GameStatus.GameStatusType, 91
  - annotation, 92
  - CHECK, 92
  - GAME\_GOES\_ON, 92
  - gameOverStr, 92
  - GameStatusType, 91, 92
  - isGameOver, 92
  - TIE, 93
  - UNFINISHED, 93
  - WIN\_OR\_LOSS, 93
- GameStatus.SpecificStatus, 208
  - Checkmate, 209
  - FiftyMoveRule, 209
  - GameGoesOn, 209
  - gameStatusType, 209
  - InsufficientMaterial, 209
  - PlayerDisconnectedVsAi, 209
  - PlayerDisconnectedVsReal, 209
  - Resignation, 210
  - ServerStoppedGame, 210
  - SpecificStatus, 208
  - Stalemate, 210
  - ThreeFoldRepetition, 210
  - TieByAgreement, 210
  - TimedOut, 210
  - TimedOutVsInsufficientMaterial, 210
  - toString, 209
- GameStatusType
  - GameStatus.GameStatusType, 91, 92
- gameStatusType
  - GameStatus.SpecificStatus, 209
- generate
  - IDsGenerator, 96
- get
  - Regex, 192
- getAddedRes
  - DBResponse, 60
- getAnnotation
  - Move, 151
- getArgs
  - RequestBuilder, 199
- getArgVal
  - RequestBuilder, 199
- getAttackingDirections
  - PieceType, 176
- getAvailable
  - Message, 119
- getBasicMoveAnnotation
  - BasicMove, 18
- getBoard
  - Message, 119
- getBuilder
  - DBRequest, 58
- getCapturingPieceType
  - Move, 151
- getCell
  - TableDBResponse, 233, 234
- getCheckedKingLoc
  - GameStatus, 86
- getColor
  - PlayerColor, 182
- getColString
  - Location, 98
- getColumnIndex
  - TableDBResponse, 234
- getColumns
  - TableDBResponse, 234
- getCombination
  - Direction, 65
- getCreatedAt
  - EstablishedGameInfo, 72
- getCreatorList
  - Move, 151
- getDBRequest
  - Message, 120
- getDBResponse
  - Message, 120
- getDefault
  - Config< V >, 51
- getDefaultAnswer
  - Question, 188
- getDefaultDesc
  - Config< V >, 51
- getDescribedDefault
  - Config< V >, 51
- getDetailedStr
  - GameStatus, 86
- getDetails
  - Regex, 192
  - StatusResponse, 216
- getDirectionByOffset
  - Direction, 65
- getDisabledCastling
  - Move, 151
- getEnPassantLoc
  - Move, 151
- getError
  - Message, 120
- getEval

- Evaluation, 76
- getEvaluationDepth
  - Evaluation, 77
- getEvaluationFor
  - Evaluation, 77
- getFen
  - Piece, 171
  - Square, 213
- getFirstRow
  - TableDBResponse, 234
- getFlipped
  - BasicMove, 18
- getGameDesc
  - CreatedGame, 55
  - EstablishedGameInfo, 72
  - GameInfo, 82
- getGameSettings
  - Message, 120
- getGameStatus
  - Evaluation, 77
  - Message, 120
- getGameStatusType
  - GameStatus, 86
- getGameTime
  - Message, 121
- getHandledStr
  - MyError, 162
- getIntermediateMove
  - Move, 152
- getJoiningPlayerColor
  - GameInfo, 82
- getLoc
  - Location, 99–101
  - Square, 213
- getLocalAddress
  - AppSocket, 3
- getLoginInfo
  - Message, 121
- getLoginType
  - LoginInfo, 106
- getMaxDistance
  - Location, 101
- getMessagesHandler
  - AppSocket, 4
- getMessageType
  - Message, 121
- getMinSize
  - MyJButton, 165
- getMove
  - Message, 121
  - MinimaxMove, 145
- getMoveDepth
  - MinimaxMove, 145
- getMoveEvaluation
  - MinimaxMove, 145
  - Move, 152
- getMoveFlag
  - Move, 152
- getMoveStack
  - EstablishedGameInfo, 72
  - Message, 121
- getMovingColor
  - Move, 152
- getMovingFrom
  - BasicMove, 19
- getMovingTo
  - BasicMove, 19
- getMyAdapter
  - MyJFrame, 167
- getName
  - PlayerColor, 182
  - RequestBuilder, 200
- getOpponent
  - PlayerColor, 182
- getOtherPlayer
  - Message, 122
- getPassword
  - LoginInfo, 106
- getPiece
  - Board, 22, 23
  - Piece, 171
  - Square, 214
- getPieceFromFen
  - Piece, 172
- getPiecelcon
  - Piece, 172
  - PieceType, 177
  - Square, 214
- getPieceName
  - PieceType, 177
- getPieceType
  - PieceType, 177
- getPlayerColor
  - Message, 122
- getPlayerFromFen
  - PlayerColor, 182
- getPlayersCastling
  - CastlingRights, 29
- getPort
  - StrUtils, 224
- getPossibleAnswers
  - Question, 188
- getPossibleMoves
  - Message, 122
- getPostDescription
  - RequestBuilder, 200
- getPreDescription
  - RequestBuilder, 200
- getPreMoves
  - Message, 122
- getPrevFullMoveClock
  - Move, 152
- getPrevHalfMoveClock
  - Move, 153
- getProfilePic
  - LoginInfo, 106

- getPromotingTo
  - Move, 153
- getQuestion
  - Message, 122
- getQuestionStr
  - Question, 189
- getRegex
  - RegEx, 193
- getRelative
  - Direction, 65
- getRemoteAddress
  - AppSocket, 4
- getRequest
  - DBRequest, 58
  - DBResponse, 60
- getRequestVariations
  - PreMadeRequest, 185
- getRespondingToMsgId
  - Message, 123
- getRights
  - CastlingRights, 29
- getRow
  - Board, 23
- getRows
  - TableDBResponse, 235
- getRowString
  - Location, 102
- getShortPrintingStr
  - MinimaxMove, 145
- getShouldSync
  - RequestBuilder, 200
- getSpecificStatus
  - GameStatus, 87
- getSquare
  - Board, 23
- getStartingColor
  - GameInfo, 83
- getStatement
  - SQLStatement, 211
- getStatus
  - DBResponse, 61
- getStr
  - Condition, 47
- getSubject
  - Message, 123
- getSubRequest
  - DBRequest, 58
- getSyncedLists
  - Message, 123
- getUrl
  - StrUtils, 224
- getUsername
  - LoginInfo, 106
  - Message, 123
- getUsernameSuggestions
  - Message, 123
- getValuesSuggestion
  - Config< V >, 52
- getWalkingDirections
  - PieceType, 177
- getWhitePieceFen
  - PieceType, 178
- getWinner
  - ArchivedGameInfo, 7
- getWinningColor
  - GameStatus, 87
- GUEST
  - LoginType, 109
- handle
  - ErrorHandler< E extends MyError >, 70
- handledErr
  - EnvManager, 69
- handledRun
  - AppSocket, 4
  - HandledThread, 94
  - MyThread, 169
- HandledThread, 93
  - handledRun, 94
  - HandledThread, 93
  - runInHandledThread, 94
  - setRunnable, 94
- handleErrors
  - ThreadsManager, 237
- HANGING\_PIECES
  - EvaluationParameters, 80
- hasAny
  - CastlingRights, 30
- hash
  - Location, 102
- hashCode
  - BasicMove, 19
  - MinimaxMove, 145
- header
  - MyJFrame.Closing< T >, 33
- htmlNewLines
  - StrUtils, 225
- Icon
  - RegEx, 194
- icon
  - MyJFrame.Closing< T >, 33
- ID
  - GameInfo, 83
- IDsGenerator, 95
  - canUseId, 95
  - generate, 96
- ifTrue
  - SwitchCase, 230
- ignore
  - ErrorHandler< E extends MyError >, 70
- ignoreErrs
  - MyThread, 170
- indexOf2
  - PlayerColor, 183
- initDebugLoginValues
  - LoginInfo, 106

- initGame
  - Message, 124
- InsufficientMaterial
  - GameStatus.SpecificStatus, 209
- insufficientMaterial
  - GameStatus, 87
- interrupt
  - Message, 124
- interruptBlocking
  - MessagesHandler, 136
- interruptListener
  - AppSocket, 4
- IPAddress
  - RegEx, 194
- isAnyData
  - DBResponse, 61
  - StatusResponse, 216
  - TableDBResponse, 235
- isAttack
  - PieceType, 178
- isBlackSquare
  - Location, 102
- isCapturing
  - Move, 153
- isCastling
  - Move.MoveFlag, 160
- isCheck
  - Evaluation, 77
  - GameStatus, 87
  - Move, 153
- isClosed
  - AppSocket, 4
- isConnected
  - AppSocket, 5
- isCreator
  - GameInfo, 83
- isCreatorToMove
  - UnfinishedGame, 239
- isDeeperAndBetterThan
  - MinimaxMove, 145
- isDiagonalPiece
  - PieceType, 178
- isDisconnected
  - GameStatus, 87
- isEmpty
  - Square, 214
  - StrUtils, 225
- isEnabled
  - CastlingRights, 30
- isGameOver
  - Evaluation, 77
  - GameStatus, 88
  - GameStatus.GameStatusType, 92
- isGreaterThan
  - Evaluation, 77
- isGuest
  - LoginInfo, 106
- isInBounds
  - Location, 102
- isLinePiece
  - PieceType, 179
- isOnMyTeam
  - Piece, 172, 173
- isResponse
  - Message, 124
- isReversible
  - Move, 153
- isSavedDate
  - RegEx, 193
- isSliding
  - PieceType, 179
- isSquareEmpty
  - Board, 23
- isSubject
  - Message, 124
- isSuccess
  - DBResponse, 61
- isUserInput
  - Arg, 10
  - ArgType, 14
- isWhiteSquare
  - Location, 103
- iteration
  - StrUtils.IterationThingy< T >, 96
- iterator
  - Board, 24
- join
  - Selection, 204
- key
  - MyJFrame.MyAdapter.HeldDown, 95
- KING
  - CastlingRights.Side, 207
  - PieceType, 180
- KING\_SAFETY
  - EvaluationParameters, 80
- kingFinalLoc
  - CastlingRights.Side, 206
- kingTravelDistance
  - CastlingRights.Side, 207
- KNIGHT
  - PieceType, 180
- L
  - Direction, 67
- label
  - Col, 38
- Location, 97
  - asInt, 104
  - asLong, 104
  - col, 104
  - createMatIndicesStr, 98
  - flip, 98
  - getColString, 98
  - getLoc, 99–101
  - getMaxDistance, 101

- getRowString, 102
- hash, 102
- isBlackSquare, 102
- isInBounds, 102
- isWhiteSquare, 103
- Location, 98
- locsMat, 103
- locsMatStr, 103
- matrixStr, 103
- row, 104
- toString, 103
- valueOf, 103
- locsMat
  - Location, 103
- locsMatStr
  - Location, 103
- LOGIN
  - LoginType, 109
- LoginInfo, 104
  - asUser, 105
  - getLoginType, 106
  - getPassword, 106
  - getProfilePic, 106
  - getUsername, 106
  - initDebugLoginValues, 106
  - isGuest, 106
  - LoginInfo, 105
  - setLoginType, 106
  - setPassword, 107
  - setProfilePic, 107
  - setUsername, 107
  - toString, 107
- LoginType, 108
  - asUser, 108
  - CANCEL, 108
  - GUEST, 109
  - LOGIN, 109
  - NOT\_SET\_YET, 109
  - REGISTER, 109
  - toString, 108
- LongCastle
  - Move.MoveFlag, 160
- LOSS\_EVAL
  - Evaluation, 79
- main
  - CastlingRights, 30
  - ConfirmDialogs, 54
  - StrUtils, 225
- MAJOR\_PIECES
  - PieceType, 180
- MATERIAL
  - EvaluationParameters, 80
- Math, 109
  - apply, 110
  - asFloat, 110
  - col, 113
  - Div, 113
  - execute, 110, 111
  - formatNum, 111
  - Mult, 113
  - nullIf0, 112
  - Plus, 113
  - simpleOperation, 112
  - str, 112
  - strSource, 112
  - strVal, 113
  - zeroIfNull, 113
- math
  - Col, 38
  - Condition, 47
- matrixStr
  - Location, 103
- Message, 114
  - answerQuestion, 116
  - askForGameSettings, 117
  - askForLogin, 117
  - askQuestion, 117
  - bye, 118
  - checkUsernameAvailability, 118
  - dbRequest, 118
  - error, 119
  - gameOver, 119
  - getAvailable, 119
  - getBoard, 119
  - getDBRequest, 120
  - getDBResponse, 120
  - getError, 120
  - getGameSettings, 120
  - getGameStatus, 120
  - getGameTime, 121
  - getLoginInfo, 121
  - getMessageType, 121
  - getMove, 121
  - getMoveStack, 121
  - getOtherPlayer, 122
  - getPlayerColor, 122
  - getPossibleMoves, 122
  - getPreMoves, 122
  - getQuestion, 122
  - getRespondingToMsgId, 123
  - getSubject, 123
  - getSyncedLists, 123
  - getUsername, 123
  - getUsernameSuggestions, 123
  - initGame, 124
  - interrupt, 124
  - isResponse, 124
  - isSubject, 124
  - Message, 115, 116
  - messageID, 133
  - returnDBResponse, 125
  - returnGameSettings, 125
  - returnLogin, 125
  - returnMove, 126
  - returnUsernameAvailable, 126
  - returnUsernameNotAvailable, 126

- setAvailable, 127
- setBoard, 127
- setDbRequest, 127
- setDBResponse, 127
- setError, 128
- setGameSettings, 128
- setGameStatus, 128
- setGameTime, 128
- setLoginInfo, 129
- setMove, 129
- setMoveStack, 129
- setOtherPlayer, 129
- setPlayerColor, 129
- setPossibleMoves, 130
- setPreMoves, 130
- setQuestion, 130
- setRespondingTo, 130
- setRespondingToMsgId, 131
- setSyncedList, 131
- setUsername, 131
- setUsernameSuggestions, 131
- syncLists, 132
- throwError, 132
- updateByMove, 132
- waitForMatch, 132
- waitForYourTurn, 133
- welcomeMessage, 133
- MessageCallback, 134
  - onMsg, 134
- messageID
  - Message, 133
- MessagesHandler, 134
  - blockTilRes, 136
  - createDisconnectedError, 136
  - interruptBlocking, 136
  - MessagesHandler, 135
  - noBlockRequest, 136
  - onAddTime, 137
  - onAlive, 137
  - onAnyDisconnection, 137
  - onAnyMsg, 137
  - onBye, 137
  - onCancelQuestion, 138
  - onDBRequest, 138
  - onDBResponse, 138
  - onDisconnected, 138
  - onError, 138
  - onGameOver, 138
  - onGetGameSettings, 139
  - onGetMove, 139
  - onInitGame, 139
  - onInterrupt, 139
  - onIsAlive, 139
  - onLogin, 140
  - onOfferDraw, 140
  - onPlannedDisconnect, 140
  - onQuestion, 140
  - onResign, 140
  - onUnplannedDisconnect, 141
  - onUpdateByMove, 141
  - onUpdateSyncedList, 141
  - onUsernameAvailability, 141
  - onWaitForMatch, 141
  - onWaitTurn, 142
  - onWelcomeMessage, 142
  - prepareForDisconnect, 142
  - receivedMessage, 142
  - socket, 142
- MinimaxMove, 143
  - compareTo, 144
  - equals, 144
  - getMove, 145
  - getMoveDepth, 145
  - getMoveEvaluation, 145
  - getShortPrintingStr, 145
  - hashCode, 145
  - isDeeperAndBetterThan, 145
  - MinimaxMove, 143, 144
  - setMove, 146
  - setMoveEvaluation, 146
  - toString, 146
- MINOR\_PIECES
  - PieceType, 180
- Move, 146
  - castling, 149
  - compareTo, 149
  - copyMove, 150
  - equals, 150
  - flipMove, 150
  - getAnnotation, 151
  - getCapturingPieceType, 151
  - getCreatorList, 151
  - getDisabledCastling, 151
  - getEnPassantLoc, 151
  - getIntermediateMove, 152
  - getMoveEvaluation, 152
  - getMoveFlag, 152
  - getMovingColor, 152
  - getPrevFullMoveClock, 152
  - getPrevHalfMoveClock, 153
  - getPromotingTo, 153
  - isCapturing, 153
  - isCheck, 153
  - isReversible, 153
  - Move, 148, 149
  - setCapturing, 153
  - setCreatorList, 154
  - setDisabledCastling, 154
  - setEnPassantLoc, 154
  - setIntermediateMove, 154
  - setMoveAnnotation, 155
  - setMoveEvaluation, 155
  - setMoveFlag, 155
  - setMovingColor, 155
  - setPrevFullMoveClock, 155
  - setPrevHalfMoveClock, 156

- setPromotingTo, 156
- setReversible, 156
- setThreefoldOption, 156
- strictEquals, 156
- threefoldClaim, 157
- toString, 157
- Move.MoveFlag, 158
  - CASTLING\_FLAGS, 160
  - castlingSide, 160
  - DoublePawnPush, 160
  - EnPassant, 160
  - equals, 159
  - isCastling, 160
  - LongCastle, 160
  - MoveFlag, 159
  - NormalMove, 160
  - Promotion, 160
  - ShortCastle, 161
- MoveAnnotation, 157
  - annotate, 157, 158
  - basicAnnotate, 158
- MoveFlag
  - Move.MoveFlag, 159
- MOVEMENT\_ABILITY
  - EvaluationParameters, 80
- msgSocket
  - AppSocket, 6
- Mult
  - Math, 113
- mult
  - CastlingRights.Side, 207
- MyError, 161
  - errToString, 162
  - getHandledStr, 162
  - MyError, 161, 162
  - toString, 163
- MyJButton, 163
  - getMinSize, 165
  - MyJButton, 163–165
  - replaceWithCancel, 165
  - resetState, 165
  - setFont, 166
  - setOnClick, 166
  - setText, 166
- MyJFrame, 166
  - debugAdapter, 167
  - doXClick, 167
  - getMyAdapter, 167
  - MyJFrame, 167
  - onClose, 168
  - setOnExit, 168
  - setOnResize, 168
- MyJFrame.BooleanClosing, 25
  - checkClosingVal, 25
  - closing, 26
  - show, 26
- MyJFrame.Closing< T >, 31
  - checkClosingVal, 32
  - closing, 32
  - header, 33
  - icon, 33
  - show, 32
  - title, 33
  - tryClose, 33
- MyJFrame.MyAdapter.HeldDown, 95
  - endPress, 95
  - key, 95
  - startPress, 95
- MyJFrame.StringClosing, 217
  - checkClosingVal, 217
  - show, 217
- MyThread, 168
  - currentThread, 169
  - handledRun, 169
  - ignoreErrs, 170
  - MyThread, 169
  - reactivateErrs, 170
  - stopRun, 170
- name
  - RequestBuilder, 201
- nested
  - Col, 39
- nestMe
  - Selection, 204
- NO\_CASTLING\_ABILITY
  - CastlingRights, 31
- NO\_PLAYER
  - PlayerColor, 183
- noBlockRequest
  - MessagesHandler, 136
- noNulls
  - Condition, 47
- NormalMove
  - Move.MoveFlag, 160
- normalPerspective
  - Direction, 67
- NOT\_SET\_YET
  - LoginType, 109
- notAFile
  - BitData, 20
- notEquals
  - Condition, 47
- notHFile
  - BitData, 21
- nullIf0
  - Math, 112
- NUM\_OF\_DIRECTIONS
  - Direction, 67
- NUM\_OF\_DIRECTIONS\_WO\_KNIGHT
  - Direction, 68
- NUM\_OF\_KNIGHT\_DIRECTIONS
  - Direction, 68
- NUM\_OF\_PIECE\_TYPES
  - PieceType, 180
- NUM\_OF\_PLAYERS
  - PlayerColor, 183

- Number
  - ArgType, 14
- Numbers
  - RegEx, 194
- numOfRows
  - TableDBResponse, 235
- ObjCallback< T >, 170
- of
  - Col, 39
- offset
  - Direction, 68
- onAddTime
  - MessagesHandler, 137
- onAlive
  - MessagesHandler, 137
- onAnyDisconnection
  - MessagesHandler, 137
- onAnyMsg
  - MessagesHandler, 137
- onBye
  - MessagesHandler, 137
- onCancelQuestion
  - MessagesHandler, 138
- onClose
  - MyJFrame, 168
- onDBRequest
  - MessagesHandler, 138
- onDBResponse
  - MessagesHandler, 138
- onDisconnected
  - MessagesHandler, 138
- onError
  - MessagesHandler, 138
- onGameOver
  - MessagesHandler, 138
- onGetGameSettings
  - MessagesHandler, 139
- onGetMove
  - MessagesHandler, 139
- onInitGame
  - MessagesHandler, 139
- onInterrupt
  - MessagesHandler, 139
- onIsAlive
  - MessagesHandler, 139
- onLogin
  - MessagesHandler, 140
- onMsg
  - MessageCallback, 134
- onOfferDraw
  - MessagesHandler, 140
- onPlannedDisconnect
  - MessagesHandler, 140
- onQuestion
  - MessagesHandler, 140
- onResign
  - MessagesHandler, 140
- onUnplannedDisconnect
  - MessagesHandler, 141
- onUpdateByMove
  - MessagesHandler, 141
- onUpdateSyncedList
  - MessagesHandler, 141
- onUsernameAvailability
  - MessagesHandler, 141
- onWaitForMatch
  - MessagesHandler, 141
- onWaitTurn
  - MessagesHandler, 142
- onWelcomeMessage
  - MessagesHandler, 142
- opponentUsername
  - EstablishedGameInfo, 73
- opposite
  - Direction, 65
- OptionalArg
  - ArgsUtil, 12
- orderBy
  - Selection, 204
- parseURLS
  - StrUtils, 226
- Password
  - ArgType, 14
  - Col, 42
  - RegEx, 194
- PAWN
  - PieceType, 180
- perspective
  - Direction, 66
- PictureUrl
  - ArgType, 14
- Piece, 170
  - ALL\_PIECES, 173
  - B\_B, 173
  - B\_K, 173
  - B\_N, 173
  - B\_P, 173
  - B\_Q, 173
  - B\_R, 174
  - getFen, 171
  - getPiece, 171
  - getPieceFromFen, 172
  - getPieceIcon, 172
  - isOnMyTeam, 172, 173
  - pieceType, 174
  - playerColor, 174
  - W\_B, 174
  - W\_K, 174
  - W\_N, 174
  - W\_P, 174
  - W\_Q, 174
  - W\_R, 174
- PIECE\_TABLES
  - EvaluationParameters, 80
- PIECE\_TYPES
  - PieceType, 180



- PieceType, 175
  - asInt, 179
  - ATTACKING\_DIRECTIONS, 179
  - ATTACKING\_PIECE\_TYPES, 179
  - BISHOP, 179
  - blackIcon, 179
  - CAN\_PROMOTE\_TO, 179
  - compareMovementType, 176
  - getAttackingDirections, 176
  - getPiecelcon, 177
  - getPieceName, 177
  - getPieceType, 177
  - getWalkingDirections, 177
  - getWhitePieceFen, 178
  - isAttack, 178
  - isDiagonalPiece, 178
  - isLinePiece, 179
  - isSliding, 179
  - KING, 180
  - KNIGHT, 180
  - MAJOR\_PIECES, 180
  - MINOR\_PIECES, 180
  - NUM\_OF\_PIECE\_TYPES, 180
  - PAWN, 180
  - PIECE\_TYPES, 180
  - QUEEN, 180
  - ROOK, 180
  - UNIQUE\_MOVES\_PIECE\_TYPES, 180
  - value, 180
  - whitelcon, 181
- pieceType
  - Piece, 174
- plainTextIgnoreCase
  - ArgsUtil, 12
- Player1
  - Col, 42
- Player2
  - Col, 42
- PLAYER\_COLORS
  - PlayerColor, 183
- PlayerColor, 181
  - asInt, 183
  - BLACK, 183
  - diff, 183
  - getColor, 182
  - getName, 182
  - getOpponent, 182
  - getPlayerFromFen, 182
  - indexOf2, 183
  - NO\_PLAYER, 183
  - NUM\_OF\_PLAYERS, 183
  - PLAYER\_COLORS, 183
  - PlayerColor, 181
  - startingRow, 184
  - toString, 183
  - WHITE, 184
- playerColor
  - Piece, 174
- playerColorToMove
  - UnfinishedGame, 239
- playerDisconnected
  - GameStatus, 88
- PlayerDisconnectedVsAi
  - GameStatus.SpecificStatus, 209
- PlayerDisconnectedVsReal
  - GameStatus.SpecificStatus, 209
- playerResigned
  - GameStatus, 88
- PlayerToMove
  - Col, 42
- playerToMove
  - UnfinishedGame, 239
- Plus
  - Math, 113
- postDescription
  - RequestBuilder, 201
- preDescription
  - RequestBuilder, 201
- PreMadeRequest, 184
  - authSettings, 185
  - ChangeProfilePic, 185
  - createBuilder, 184
  - DeleteUnfGames, 185
  - Games, 185
  - getRequestVariations, 185
  - statistics, 185
  - StatsByTimeOfDay, 185
  - TopPlayers, 185
- PreMadeRequest.VariationCreator, 241
  - create, 241
- prepareForDisconnect
  - MessagesHandler, 142
- print
  - Board, 24
  - DBResponse, 61
  - Evaluation, 78
- ProfilePic
  - Col, 42
- Promotion
  - Move.MoveFlag, 160
- QUEEN
  - CastlingRights.Side, 207
  - PieceType, 180
- Question, 186
  - Answer, 187
  - drawOffer, 188
  - equals, 188
  - getDefaultAnswer, 188
  - getPossibleAnswers, 188
  - getQuestionStr, 189
  - Question, 187
  - questionStr, 189
  - questionType, 189
  - Rematch, 190
  - setDefaultAnswer, 189
  - Threefold, 190

- toString, 189
- questionStr
  - Question, 189
- questionType
  - Question, 189
- R
  - Direction, 68
- reactivateErrs
  - MyThread, 170
- receivedMessage
  - MessagesHandler, 142
- Regex, 190
  - Any, 193
  - canBeEmpty, 191
  - check, 192
  - dontMatch, 193
  - DontSaveGame, 194
  - Fen, 194
  - get, 192
  - getDetails, 192
  - getRegex, 193
  - Icon, 194
  - IPPAAddress, 194
  - isSavedDate, 193
  - Numbers, 194
  - Password, 194
  - Regex, 191
  - setUseDontMatch, 193
  - StrUtilSkip, 194
  - URL, 194
  - Username, 194
- REGISTER
  - LoginType, 109
- Rematch
  - Question, 190
- repeat
  - StrUtils, 226
- replnStr
  - Arg, 11
- replace
  - Col, 40
  - SQLStatement, 211
- replaceWithCancel
  - MyJButton, 165
- request
  - DBResponse, 62
- RequestBuilder, 195
  - addShouldSync, 197
  - args, 201
  - build, 197
  - changePassword, 198
  - changeProfilePic, 198
  - createResponse, 198
  - createVariation, 198
  - deleteAllUnFinishedGames, 199
  - games, 199
  - getArgs, 199
  - getArgVal, 199
  - getName, 200
  - getPostDescription, 200
  - getPreDescription, 200
  - getShouldSync, 200
  - name, 201
  - postDescription, 201
  - preDescription, 201
  - RequestBuilder, 196, 197
  - setSubBuilder, 200
  - statement, 201
  - statsByTimeOfDay, 201
  - subBuilder, 202
  - TIE\_STR, 202
  - top, 201
- requestMessage
  - AppSocket, 5
- resetState
  - MyJButton, 165
- Resignation
  - GameStatus.SpecificStatus, 210
- respond
  - AppSocket, 5
- returnDBResponse
  - Message, 125
- returnGameSettings
  - Message, 125
- returnLogin
  - Message, 125
- returnMove
  - Message, 126
- returnUsernameAvailable
  - Message, 126
- returnUsernameNotAvailable
  - Message, 126
- ROOK
  - PieceType, 180
- rookStartingCol
  - CastlingRights.Side, 207
- row
  - Location, 104
- rows
  - TableDBResponse, 236
- rowToString
  - TableDBResponse, 235
- runInHandledThread
  - HandledThread, 94
- SavedGame
  - Col, 42
- Selection, 202
  - createStatement, 204
  - join, 204
  - nestMe, 204
  - orderBy, 204
  - Selection, 203
  - top, 205
- ServerAddress
  - ArgType, 14
- ServerStoppedGame

- GameStatus.SpecificStatus, 210
- serverStoppedGame
  - GameStatus, 89
- setAddedRes
  - DBResponse, 61
- setAvailable
  - Message, 127
- setBoard
  - Message, 127
- setCapturing
  - Move, 153
- setColName
  - Col, 40
- setCreatedAt
  - EstablishedGameInfo, 72
- setCreatorList
  - Move, 154
- setCustomStr
  - GameStatus, 89
- setDbRequest
  - Message, 127
- setDBResponse
  - Message, 127
- setDefaultAnswer
  - Question, 189
- setDepth
  - GameStatus, 89
- setDisabledCastling
  - Move, 154
- setEmpty
  - Square, 214
- setEnPassantLoc
  - Move, 154
- setError
  - Message, 128
- setEval
  - Evaluation, 78
- setEvaluationDepth
  - Evaluation, 78
- setFont
  - MyJButton, 166
- setGameSettings
  - Message, 128
- setGameStatus
  - Message, 128
- setGameTime
  - Message, 128
- setInCheck
  - GameStatus, 89
- setIntermediateMove
  - Move, 154
- setLoginInfo
  - Message, 129
- setLoginType
  - LoginInfo, 106
- setMessageHandler
  - AppSocket, 6
- setMove
  - Message, 129
  - MinimaxMove, 146
- setMoveAnnotation
  - Move, 155
- setMoveEvaluation
  - MinimaxMove, 146
  - Move, 155
- setMoveFlag
  - Move, 155
- setMoveStack
  - Message, 129
- setMovingColor
  - Move, 155
- setMovingFrom
  - BasicMove, 19
- setMovingTo
  - BasicMove, 20
- setOnClick
  - MyJButton, 166
- setOnExit
  - MyJFrame, 168
- setOnResize
  - MyJFrame, 168
- setOtherPlayer
  - Message, 129
- setPassword
  - LoginInfo, 107
- setPerspective
  - Evaluation, 78
- setPiece
  - Board, 24
  - Square, 214
- setPlayerColor
  - Message, 129
- setPossibleMoves
  - Message, 130
- setPreMoves
  - Message, 130
- setPrevFullMoveClock
  - Move, 155
- setPrevHalfMoveClock
  - Move, 156
- setProfilePic
  - LoginInfo, 107
- setPromotingTo
  - Move, 156
- setQuestion
  - Message, 130
- setRespondingTo
  - Message, 130
- setRespondingToMsgId
  - Message, 131
- setReversible
  - Move, 156
- setRunnable
  - HandledThread, 94
- setSquareEmpty
  - Board, 24

- setStr
  - Condition, 48
- setSubBuilder
  - RequestBuilder, 200
- setSubRequest
  - DBRequest, 58
- setSyncedList
  - Message, 131
- setText
  - MyJButton, 166
- setThreefoldOption
  - Move, 156
- setUseDontMatch
  - RegEx, 193
- setUserInput
  - Arg, 10
- setUsername
  - LoginInfo, 107
  - Message, 131
- setUsernameSuggestions
  - Message, 131
- setWrapped
  - Col, 40
- ShortCastle
  - Move.MoveFlag, 161
- show
  - MyJFrame.BooleanClosing, 26
  - MyJFrame.Closing< T >, 32
  - MyJFrame.StringClosing, 217
- Side
  - CastlingRights.Side, 206
- SIDES
  - CastlingRights.Side, 207
- simpleOperation
  - Math, 112
- socket
  - MessagesHandler, 142
- SpecificStatus
  - GameStatus.SpecificStatus, 208
- splitArr
  - StrUtils, 226, 227
- SQLStatement, 210
  - createStatement, 211
  - getStatement, 211
  - replace, 211
  - SQLStatement, 211
  - toString, 212
  - type, 212
- Square, 212
  - EMPTY\_PIECE, 215
  - getFen, 213
  - getLoc, 213
  - getPiece, 214
  - getPieceIcon, 214
  - isEmpty, 214
  - setEmpty, 214
  - setPiece, 214
  - Square, 213
  - toString, 215
- SQUARE\_CONTROL
  - EvaluationParameters, 80
- Stalemate
  - GameStatus.SpecificStatus, 210
- stalemate
  - GameStatus, 90
- startingFen
  - Board, 25
- startingPos
  - Board, 25
- startingRow
  - PlayerColor, 184
- startPress
  - MyJFrame.MyAdapter.HeldDown, 95
- statement
  - RequestBuilder, 201
- statistics
  - PreMadeRequest, 185
- StatsByTimeOfDay
  - PreMadeRequest, 185
- statsByTimeOfDay
  - RequestBuilder, 201
- status
  - DBResponse, 62
- StatusResponse, 215
  - clean, 216
  - getDetails, 216
  - isAnyData, 216
  - StatusResponse, 215, 216
- STOCKFISH\_SAYS
  - EvaluationParameters, 80
- stopReading
  - AppSocket, 6
- stopRun
  - MyThread, 170
- str
  - Math, 112
- strictEquals
  - Move, 156
- strlNN
  - StrUtils, 227
- strSource
  - Math, 112
- StrUtils, 218
  - awful, 218
  - clean, 219
  - countMatches, 219
  - createTimeGreeting, 219
  - createTimeStr, 219
  - dateTimePrefix, 220
  - dontCapFull, 220
  - dontCapWord, 220
  - fitInside, 221
  - fixHtml, 221
  - format, 221, 222
  - formatDate, 222, 223
  - formatDateSQL, 223

- getPort, 224
- getUrl, 224
- htmlNewLines, 225
- isEmpty, 225
- main, 225
- parseURLS, 226
- repeat, 226
- splitArr, 226, 227
- strlNN, 227
- uppercase, 228
- StrUtils.IterationThingy< T >, 96
  - iteration, 96
- StrUtilSkip
  - RegEx, 194
- strVal
  - Math, 113
- subBuilder
  - RequestBuilder, 202
- sum
  - Col, 40
- SwitchCase, 228
  - condition, 229
  - defaultCase, 229
  - equals, 229
  - ifTrue, 230
  - SwitchCase, 228
  - toString, 230
- switchCase
  - Col, 41
- syncLists
  - Message, 132
- Table, 230
  - cols, 231
  - escapeValues, 231
  - Games, 231
  - Table, 230
  - tableAndValues, 231
  - UnfinishedGames, 231
  - Users, 232
- tableAndValues
  - Table, 231
- TableDBResponse, 232
  - clean, 233
  - columns, 236
  - getCell, 233, 234
  - getColumnIndex, 234
  - getColumns, 234
  - getFirstRow, 234
  - getRows, 235
  - isAnyData, 235
  - numOfRows, 235
  - rows, 236
  - rowToString, 235
  - TableDBResponse, 232, 233
  - toString, 236
- Text
  - ArgType, 14
- ThreadsManager, 236
  - createThread, 236
  - handleErrors, 237
- Threefold
  - Question, 190
- threefoldClaim
  - Move, 157
- ThreeFoldRepetition
  - GameStatus.SpecificStatus, 210
- threeFoldRepetition
  - GameStatus, 90
- throwError
  - Message, 132
- ThrowingCallback< T >, 237
  - callback, 237
- TIE
  - GameStatus.GameStatusType, 93
- TIE\_EVAL
  - Evaluation, 79
- TIE\_STR
  - RequestBuilder, 202
- TieByAgreement
  - GameStatus.SpecificStatus, 210
- tieByAgreement
  - GameStatus, 90
- time
  - Col, 41
- TimedOut
  - GameStatus.SpecificStatus, 210
- timedOut
  - GameStatus, 90
- TimedOutVsInsufficientMaterial
  - GameStatus.SpecificStatus, 210
- title
  - MyJFrame.Closing< T >, 33
- top
  - RequestBuilder, 201
  - Selection, 205
- TopPlayers
  - PreMadeRequest, 185
- toString
  - ArchivedGameInfo, 7
  - Arg, 10
  - BasicMove, 20
  - Board, 25
  - CastlingRights, 31
  - CastlingRights.Side, 206
  - Col, 41
  - Condition, 48
  - Config< V >, 52
  - DBRequest, 58
  - DBResponse, 62
  - EstablishedGameInfo, 73
  - Evaluation, 79
  - GameInfo, 83
  - GameStatus, 91
  - GameStatus.SpecificStatus, 209
  - Location, 103
  - LoginInfo, 107

- LoginType, 108
- MinimaxMove, 146
- Move, 157
- MyError, 163
- PlayerColor, 183
- Question, 189
- SQLStatement, 212
- Square, 215
- SwitchCase, 230
- TableDBResponse, 236
- tryClose
  - MyJFrame.Closing< T >, 33
- type
  - DBRequest, 59
  - SQLStatement, 212
- U
  - Direction, 68
- U\_L
  - Direction, 68
- U\_L\_L
  - Direction, 68
- U\_R
  - Direction, 68
- U\_R\_R
  - Direction, 68
- U\_U
  - Direction, 68
- U\_U\_L
  - Direction, 68
- U\_U\_R
  - Direction, 69
- UNFINISHED
  - GameStatus.GameStatusType, 93
- UnfinishedGame, 238
  - isCreatorToMove, 239
  - playerColorToMove, 239
  - playerToMove, 239
  - UnfinishedGame, 238
- UnfinishedGames
  - Table, 231
- UNIQUE\_MOVES\_PIECE\_TYPES
  - PieceType, 180
- Update, 239
  - createStatement, 240
  - Update, 240
- updateByMove
  - Message, 132
- uppercase
  - StrUtils, 228
- URL
  - RegEx, 194
- Url
  - ArgType, 14
- Username
  - ArgType, 14
  - Col, 42
  - RegEx, 194
- Users
  - Table, 232
- value
  - PieceType, 180
- ValueBtn
  - ValueBtn< T >, 241
- ValueBtn< T >, 240
  - ValueBtn, 241
- valueOf
  - Location, 103
- ver14::SharedClasses::DBActions::Condition::Relation, 43
- ver14::SharedClasses::DBActions::DBRequest::DBRequest::Type, 56
- ver14::SharedClasses::DBActions::DBResponse::DBResponse::Status, 59
- ver14::SharedClasses::DBActions::Statements::Selection::Join, 202
- ver14::SharedClasses::DBActions::Statements::Selection::Order, 203
- ver14::SharedClasses::Game::Moves::Move::ThreefoldStatus, 148
- ver14::SharedClasses::Misc::Question::QuestionType, 186
- VoidCallback, 242
  - callback, 242
- W\_B
  - Piece, 174
- W\_K
  - Piece, 174
- W\_N
  - Piece, 174
- W\_P
  - Piece, 174
- W\_Q
  - Piece, 174
- W\_R
  - Piece, 174
- waitForMatch
  - Message, 132
- waitForYourTurn
  - Message, 133
- weight
  - EvaluationParameters, 81
- welcomeMessage
  - Message, 133
- WHITE
  - PlayerColor, 184
- whitelcon
  - PieceType, 181
- whosCastling
  - CastlingRights, 31
- WIN\_EVAL
  - Evaluation, 79
- WIN\_OR\_LOSS
  - GameStatus.GameStatusType, 93
- Winner
  - Col, 42

wrap  
    Col, 41  
    Condition, 48  
writeMessage  
    AppSocket, 6  
  
zeroIfNull  
    Math, 113