

## My Project

Generated by Doxygen 1.9.3



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>5</b>
2.1 Class List	5
<b>3 Class Documentation</b>	<b>9</b>
3.1 AiParameters	9
3.1.1 Class Documentation	9
3.1.1.1 enum ver14::SharedClasses::Game::GameSetup::AiParameters::AiType	9
3.1.2 Member Function Documentation	10
3.1.2.1 setTimeFormat()	10
3.2 AnswerCallback	10
3.2.1 Detailed Description	10
3.3 AppSocket	10
3.3.1 Detailed Description	11
3.3.2 Constructor & Destructor Documentation	11
3.3.2.1 AppSocket() [1/2]	11
3.3.2.2 AppSocket() [2/2]	11
3.3.3 Member Function Documentation	12
3.3.3.1 close() [1/2]	12
3.3.3.2 close() [2/2]	12
3.3.3.3 interruptListener()	12
3.3.3.4 requestMessage() [1/2]	13
3.3.3.5 handledRun()	13
3.3.3.6 getMessagesHandler()	13
3.3.3.7 setMessagesHandler()	13
3.3.3.8 respond()	14
3.3.3.9 writeMessage()	14
3.3.3.10 isConnected()	14
3.3.3.11 getLocalAddress()	14
3.3.3.12 getRemoteAddress()	15
3.3.3.13 isClosed()	15
3.3.3.14 requestMessage() [2/2]	15
3.3.3.15 stopReading()	15
3.3.4 Member Data Documentation	16
3.3.4.1 msgSocket	16
3.4 ArchivedGameInfo	16
3.4.1 Detailed Description	16
3.4.2 Constructor & Destructor Documentation	16
3.4.2.1 ArchivedGameInfo()	16
3.4.3 Member Function Documentation	17
3.4.3.1 getWinner()	17

3.4.3.2 toString()	17
3.5 Arg	17
3.5.1 Detailed Description	18
3.5.2 Constructor & Destructor Documentation	18
3.5.2.1 Arg() [1/3]	18
3.5.2.2 Arg() [2/3]	18
3.5.2.3 Arg() [3/3]	19
3.5.3 Member Function Documentation	19
3.5.3.1 isUserInput()	19
3.5.3.2 setUserInput()	19
3.5.3.3 equals()	19
3.5.3.4 toString()	20
3.5.3.5 createVal()	20
3.5.4 Member Data Documentation	20
3.5.4.1 replnStr	20
3.5.4.2 escape	21
3.5.4.3 argType	21
3.5.4.4 config	21
3.6 ArgsUtil	21
3.6.1 Detailed Description	21
3.6.2 Member Function Documentation	21
3.6.2.1 create()	21
3.6.2.2 equalsSign()	22
3.6.2.3 plainTextIgnoreCase()	22
3.6.2.4 OptionalArg()	22
3.7 ArgType Enum Reference	23
3.7.1 Detailed Description	23
3.7.2 Constructor & Destructor Documentation	23
3.7.2.1 ArgType() [1/2]	23
3.7.2.2 ArgType() [2/2]	23
3.7.3 Member Data Documentation	24
3.7.3.1 Date	24
3.7.3.2 DateRange	24
3.7.3.3 Text	24
3.7.3.4 ServerAddress	24
3.7.3.5 Number	24
3.7.3.6 Username	24
3.7.3.7 Password	25
3.7.3.8 Url	25
3.7.3.9 PictureUrl	25
3.7.3.10 isUserInput	25
3.8 ArrUtils	25

3.8.1 Detailed Description	25
3.8.2 Member Function Documentation	25
3.8.2.1 concat()	25
3.8.2.2 createList()	26
3.8.2.3 exists()	26
3.9 AuthSettings	27
3.9.1 Detailed Description	27
3.9.2 Member Data Documentation	27
3.9.2.1 GUEST	27
3.9.2.2 USER	27
3.9.2.3 ANY_LOGIN	27
3.9.2.4 NEVER_AUTH	28
3.9.2.5 NO_AUTH	28
3.10 BasicMove	28
3.10.1 Detailed Description	28
3.10.2 Constructor & Destructor Documentation	28
3.10.2.1 BasicMove() [1/3]	28
3.10.2.2 BasicMove() [2/3]	29
3.10.2.3 BasicMove() [3/3]	29
3.10.3 Member Function Documentation	29
3.10.3.1 createBatch()	29
3.10.3.2 getFlipped() [1/2]	30
3.10.3.3 getFlipped() [2/2]	30
3.10.3.4 flip()	30
3.10.3.5 getMovingFrom()	30
3.10.3.6 setMovingFrom()	31
3.10.3.7 getMovingTo()	31
3.10.3.8 setMovingTo()	31
3.10.3.9 hashCode()	31
3.10.3.10 equals()	32
3.10.3.11 toString()	32
3.10.3.12 getBasicMoveAnnotation()	32
3.10.3.13 cp()	33
3.11 BitData	33
3.11.1 Detailed Description	33
3.11.2 Member Data Documentation	33
3.11.2.1 notAFile	33
3.11.2.2 notHFile	33
3.11.2.3 everything	34
3.12 Board	34
3.12.1 Detailed Description	34
3.12.2 Constructor & Destructor Documentation	34

3.12.2.1 Board() [1/3]	34
3.12.2.2 Board() [2/3]	35
3.12.2.3 Board() [3/3]	35
3.12.3 Member Function Documentation	35
3.12.3.1 setPiece()	35
3.12.3.2 getSquare()	35
3.12.3.3 fenSetup()	36
3.12.3.4 startingPos()	36
3.12.3.5 getRow()	36
3.12.3.6 getPiece() [1/2]	37
3.12.3.7 getPiece() [2/2]	37
3.12.3.8 print()	37
3.12.3.9 iterator()	38
3.12.3.10 setSquareEmpty()	38
3.12.3.11 isSquareEmpty()	38
3.12.3.12 toString()	38
3.12.4 Member Data Documentation	39
3.12.4.1 startingFen	39
3.12.4.2 example	39
3.13 MyJFrame.BooleanClosing	39
3.13.1 Detailed Description	39
3.13.2 Member Function Documentation	39
3.13.2.1 show()	40
3.13.2.2 checkClosingVal()	40
3.13.2.3 closing()	40
3.14 Callback< T >	40
3.14.1 Detailed Description	40
3.14.2 Member Function Documentation	41
3.14.2.1 callback()	41
3.15 CastlingRights	41
3.15.1 Detailed Description	42
3.15.2 Constructor & Destructor Documentation	42
3.15.2.1 CastlingRights() [1/3]	42
3.15.2.2 CastlingRights() [2/3]	42
3.15.2.3 CastlingRights() [3/3]	42
3.15.3 Member Function Documentation	42
3.15.3.1 createFromStr()	43
3.15.3.2 enableCastling()	43
3.15.3.3 main()	43
3.15.3.4 whosCastling()	43
3.15.3.5 toString()	44
3.15.3.6 isEnabled()	44

3.15.3.7 disableCastling()	44
3.15.3.8 getRights()	45
3.15.3.9 enable()	45
3.15.3.10 hasAny()	45
3.15.3.11 getPlayersCastling()	46
3.15.4 Member Data Documentation	46
3.15.4.1 NO_CASTLING_ABILITY	46
3.16 MyJFrame.Closing< T >	46
3.16.1 Detailed Description	46
3.16.2 Member Function Documentation	47
3.16.2.1 tryClose()	47
3.16.2.2 show()	47
3.16.2.3 checkClosingVal()	47
3.16.2.4 closing()	47
3.16.3 Member Data Documentation	48
3.16.3.1 title	48
3.16.3.2 header	48
3.16.3.3 icon	48
3.17 Col	48
3.17.1 Detailed Description	49
3.17.2 Constructor & Destructor Documentation	49
3.17.2.1 Col() [1/3]	49
3.17.2.2 Col() [2/3]	49
3.17.2.3 Col() [3/3]	50
3.17.3 Member Function Documentation	50
3.17.3.1 count() [1/2]	50
3.17.3.2 count() [2/2]	50
3.17.3.3 countIf()	51
3.17.3.4 sum()	51
3.17.3.5 label()	52
3.17.3.6 switchCase()	52
3.17.3.7 wrap()	52
3.17.3.8 time()	52
3.17.3.9 date()	53
3.17.3.10 setWrapped()	53
3.17.3.11 as() [1/2]	53
3.17.3.12 as() [2/2]	53
3.17.3.13 setColName()	54
3.17.3.14 colName()	54
3.17.3.15 of() [1/2]	54
3.17.3.16 of() [2/2]	55
3.17.3.17 replace()	55

3.17.3.18 nested()	55
3.17.3.19 math() [1/2]	56
3.17.3.20 math() [2/2]	57
3.17.3.21 toString()	57
3.17.4 Member Data Documentation	57
3.17.4.1 gameId	58
3.17.4.2 SavedGame	58
3.17.4.3 Password	58
3.17.4.4 Player1	58
3.17.4.5 Player2	58
3.17.4.6 Username	58
3.17.4.7 Winner	58
3.17.4.8 PlayerToMove	58
3.17.4.9 CreatedDateTime	59
3.17.4.10 ProfilePic	59
3.18 MovesList.CompareMoves	59
3.19 Condition	59
3.19.1 Detailed Description	59
3.19.2 Class Documentation	59
3.19.2.1 enum ver14::SharedClasses::DBActions::Condition::Relation	59
3.19.3 Constructor & Destructor Documentation	60
3.19.3.1 Condition()	60
3.19.4 Member Function Documentation	60
3.19.4.1 equals()	60
3.19.4.2 noNulls()	61
3.19.4.3 add() [1/2]	61
3.19.4.4 add() [2/2]	61
3.19.4.5 wrap()	62
3.19.4.6 math()	62
3.19.4.7 between()	62
3.19.4.8 notEquals()	63
3.19.4.9 and()	63
3.19.4.10 getStr()	63
3.19.4.11 setStr()	63
3.19.4.12 toString()	64
3.20 Config< V >	64
3.20.1 Detailed Description	64
3.20.2 Constructor & Destructor Documentation	65
3.20.2.1 Config() [1/6]	65
3.20.2.2 Config() [2/6]	65
3.20.2.3 Config() [3/6]	65
3.20.2.4 Config() [4/6]	65



3.20.2.5 Config() [5/6]	66
3.20.2.6 Config() [6/6]	66
3.20.3 Member Function Documentation	66
3.20.3.1 addSuggestion()	67
3.20.3.2 getValuesSuggestion()	67
3.20.3.3 getDescribedDefault()	67
3.20.3.4 getDefault()	67
3.20.3.5 toString()	68
3.20.3.6 getDefaultDesc()	68
3.20.4 Member Data Documentation	68
3.20.4.1 canUseDefault	68
3.20.4.2 description	68
3.21 ConfirmDialogs	68
3.21.1 Detailed Description	68
3.21.2 Member Function Documentation	69
3.21.2.1 confirm() [1/2]	69
3.21.2.2 main()	69
3.21.2.3 confirm() [2/2]	69
3.22 CreatedGame	71
3.22.1 Detailed Description	71
3.22.2 Constructor & Destructor Documentation	71
3.22.2.1 CreatedGame()	71
3.22.3 Member Function Documentation	72
3.22.3.1 getGameDesc()	72
3.23 CustomStatement	72
3.23.1 Detailed Description	72
3.23.2 Constructor & Destructor Documentation	73
3.23.2.1 CustomStatement()	73
3.23.3 Member Function Documentation	73
3.23.3.1 createStatement()	73
3.24 DBRequest	73
3.24.1 Detailed Description	74
3.24.2 Class Documentation	74
3.24.2.1 enum ver14::SharedClasses::DBActions::DBRequest::DBRequest::Type	74
3.24.3 Constructor & Destructor Documentation	74
3.24.3.1 DBRequest() [1/3]	74
3.24.3.2 DBRequest() [2/3]	75
3.24.3.3 DBRequest() [3/3]	75
3.24.4 Member Function Documentation	75
3.24.4.1 getBuilder()	75
3.24.4.2 getSubRequest()	76
3.24.4.3 setSubRequest()	76

3.24.4.4 toString()	76
3.24.4.5 getRequest()	76
3.24.5 Member Data Documentation	77
3.24.5.1 type	77
3.25 DBResponse	77
3.25.1 Detailed Description	77
3.25.2 Class Documentation	77
3.25.2.1 enum ver14::SharedClasses::DBActions::DBResponse::DBResponse::Status	77
3.25.3 Constructor & Destructor Documentation	78
3.25.3.1 DBResponse()	78
3.25.4 Member Function Documentation	78
3.25.4.1 getRequest()	78
3.25.4.2 getStatus()	78
3.25.4.3 isSuccess()	79
3.25.4.4 isAnyData()	79
3.25.4.5 getAddedRes()	79
3.25.4.6 setAddedRes()	79
3.25.4.7 clean()	80
3.25.4.8 print()	80
3.25.4.9 toString()	80
3.25.5 Member Data Documentation	80
3.25.5.1 status	80
3.25.5.2 request	81
3.25.5.3 addedRes	81
3.26 Delete	81
3.26.1 Detailed Description	81
3.26.2 Constructor & Destructor Documentation	81
3.26.2.1 Delete()	81
3.26.3 Member Function Documentation	82
3.26.3.1 createStatement()	82
3.27 Direction Enum Reference	82
3.27.1 Detailed Description	83
3.27.2 Member Function Documentation	83
3.27.2.1 getRelative()	83
3.27.2.2 getDirectionByOffset()	84
3.27.2.3 getCombination()	84
3.27.2.4 perspective()	84
3.27.2.5 opposite()	85
3.27.3 Member Data Documentation	85
3.27.3.1 U	85
3.27.3.2 D	85
3.27.3.3 L	85

3.27.3.4 R	86
3.27.3.5 U_U	86
3.27.3.6 D_D	86
3.27.3.7 U_R	86
3.27.3.8 U_L	87
3.27.3.9 D_R	87
3.27.3.10 D_L	87
3.27.3.11 U_U_R	87
3.27.3.12 U_U_L	88
3.27.3.13 U_R_R	88
3.27.3.14 U_L_L	88
3.27.3.15 D_D_R	88
3.27.3.16 D_D_L	89
3.27.3.17 D_R_R	89
3.27.3.18 D_L_L	89
3.27.3.19 NUM_OF_DIRECTIONS	89
3.27.3.20 NUM_OF_KNIGHT_DIRECTIONS	89
3.27.3.21 NUM_OF_DIRECTIONS_WO_KNIGHT	90
3.27.3.22 ALL_DIRECTIONS	90
3.27.3.23 ALL_USED_DIRECTIONS	90
3.27.3.24 normalPerspective	90
3.27.3.25 andWith	90
3.27.3.26 offset	90
3.27.3.27 asInt	90
3.27.3.28 combination	91
3.28 EfficientGen< K, V >	91
3.28.1 Detailed Description	91
3.28.2 Constructor & Destructor Documentation	91
3.28.2.1 EfficientGen()	91
3.28.3 Member Function Documentation	91
3.28.3.1 get()	91
3.29 Environment	93
3.29.1 Member Data Documentation	93
3.29.1.1 IS_JAR	93
3.30 EnvManager	93
3.30.1 Detailed Description	93
3.30.2 Member Function Documentation	93
3.30.2.1 handledErr()	93
3.30.2.2 criticalErr()	94
3.31 ErrorHandler< E extends MyError >	94
3.31.1 Detailed Description	94
3.31.2 Member Function Documentation	94

3.31.2.1 ignore()	94
3.31.2.2 handle()	95
3.32 EstablishedGameInfo	95
3.32.1 Detailed Description	95
3.32.2 Constructor & Destructor Documentation	96
3.32.2.1 EstablishedGameInfo()	96
3.32.3 Member Function Documentation	96
3.32.3.1 getCreatedAt()	96
3.32.3.2 setCreatedAt()	96
3.32.3.3 getMoveStack()	97
3.32.3.4 toString()	97
3.32.3.5 getGameDesc()	97
3.32.4 Member Data Documentation	98
3.32.4.1 opponentUsername	98
3.32.4.2 createdAt	98
3.33 Evaluation	98
3.33.1 Detailed Description	98
3.33.2 Constructor & Destructor Documentation	98
3.33.2.1 Evaluation() [ 1 / 4 ]	98
3.33.2.2 Evaluation() [ 2 / 4 ]	99
3.33.2.3 Evaluation() [ 3 / 4 ]	99
3.33.2.4 Evaluation() [ 4 / 4 ]	99
3.33.3 Member Function Documentation	100
3.33.3.1 book()	100
3.33.3.2 addDetail()	100
3.33.3.3 assertNotGameOver()	100
3.33.3.4 getEvaluationDepth()	100
3.33.3.5 setEvaluationDepth()	101
3.33.3.6 isGameOver()	101
3.33.3.7 isCheck()	101
3.33.3.8 isGreaterThan()	101
3.33.3.9 getEval()	102
3.33.3.10 setEval()	102
3.33.3.11 getGameStatus()	102
3.33.3.12 equals()	102
3.33.3.13 toString()	103
3.33.3.14 convertFromCentipawns()	103
3.33.3.15 getEvaluationFor()	103
3.33.3.16 setPerspective()	103
3.33.3.17 flipEval()	104
3.33.3.18 print()	104
3.33.3.19 EvaluationDetail()	104

3.33.4 Member Data Documentation	104
3.33.4.1 TIE_EVAL	104
3.33.4.2 WIN_EVAL	105
3.33.4.3 LOSS_EVAL	105
3.34 EvaluationParameters Enum Reference	105
3.34.1 Detailed Description	105
3.34.2 Member Data Documentation	105
3.34.2.1 MATERIAL	105
3.34.2.2 PIECE_TABLES	106
3.34.2.3 KING_SAFETY	106
3.34.2.4 HANGING_PIECES	106
3.34.2.5 SQUARE_CONTROL	106
3.34.2.6 MOVEMENT_ABILITY	106
3.34.2.7 FORCE_KING_TO_CORNER	106
3.34.2.8 EG_WEIGHT	106
3.34.2.9 STOCKFISH_SAYS	106
3.34.2.10 weight	107
3.35 FontManager	107
3.36 GameInfo	107
3.36.1 Detailed Description	108
3.36.2 Constructor & Destructor Documentation	108
3.36.2.1 GameInfo()	108
3.36.3 Member Function Documentation	108
3.36.3.1 example()	108
3.36.3.2 isCreator()	109
3.36.3.3 ID()	109
3.36.3.4 equals()	109
3.36.3.5 toString()	110
3.36.3.6 getGameDesc()	110
3.36.3.7 getJoiningPlayerColor()	110
3.36.3.8 getStartingColor()	110
3.36.4 Member Data Documentation	111
3.36.4.1 gameId	111
3.36.4.2 creatorUsername	111
3.36.4.3 gameSettings	111
3.37 GameSettings	111
3.37.1 Class Documentation	111
3.37.1.1 enum ver14::SharedClasses::Game::GameSetup::GameSettings::GameType	111
3.37.2 Member Function Documentation	112
3.37.2.1 setTimeFormat()	112
3.38 GameStatus	112
3.38.1 Detailed Description	112

3.38.2 Member Function Documentation	113
3.38.2.1 checkmate()	113
3.38.2.2 gameGoesOn()	114
3.38.2.3 tieByAgreement()	114
3.38.2.4 stalemate()	114
3.38.2.5 fiftyMoveRule()	115
3.38.2.6 serverStoppedGame()	115
3.38.2.7 setCustomStr()	115
3.38.2.8 threeFoldRepetition()	115
3.38.2.9 insufficientMaterial()	116
3.38.2.10 playerDisconnected()	116
3.38.2.11 timedOut()	116
3.38.2.12 playerResigned()	117
3.38.2.13 isDisconnected()	117
3.38.2.14 getCheckedKingLoc()	117
3.38.2.15 getWinningColor()	118
3.38.2.16 getGameStatusType()	118
3.38.2.17 setInCheck()	118
3.38.2.18 isCheck()	118
3.38.2.19 getDetailedStr() [1/2]	119
3.38.2.20 getDetailedStr() [2/2]	119
3.38.2.21 isGameOver()	119
3.38.2.22 toString()	119
3.38.2.23 setDepth()	120
3.38.2.24 getSpecificStatus()	120
3.39 GameStatus.GameStatusType Enum Reference	120
3.39.1 Detailed Description	121
3.39.2 Constructor & Destructor Documentation	121
3.39.2.1 GameStatusType() [1/2]	121
3.39.2.2 GameStatusType() [2/2]	121
3.39.3 Member Function Documentation	121
3.39.3.1 isGameOver()	121
3.39.4 Member Data Documentation	122
3.39.4.1 TIE	122
3.39.4.2 CHECK	122
3.39.4.3 GAME_GOES_ON	122
3.39.4.4 WIN_OR_LOSS	122
3.39.4.5 UNFINISHED	122
3.39.4.6 annotation	122
3.39.4.7 gameOverStr	123
3.40 GameTime	123
3.41 GameView	123

3.42 Graphable	124
3.43 GraphableDBResponse	124
3.43.1 Member Function Documentation	124
3.43.1.1 toString()	124
3.44 GraphElement	125
3.45 GraphElementType Enum Reference	125
3.46 HandledThread	125
3.46.1 Detailed Description	126
3.46.2 Constructor & Destructor Documentation	126
3.46.2.1 HandledThread() [1/2]	126
3.46.2.2 HandledThread() [2/2]	126
3.46.3 Member Function Documentation	126
3.46.3.1 runInHandledThread()	126
3.46.3.2 setRunnable()	127
3.46.3.3 handledRun()	127
3.47 MyJFrame.MyAdapter.HeldDown	127
3.47.1 Detailed Description	127
3.47.2 Member Function Documentation	127
3.47.2.1 startPress()	127
3.47.2.2 endPress()	128
3.47.2.3 key()	128
3.48 IDsGenerator	128
3.48.1 Detailed Description	128
3.48.2 Member Function Documentation	128
3.48.2.1 generate()	128
3.48.2.2 canUseId()	128
3.49 StrUtils.IterationThingy< T >	129
3.49.1 Detailed Description	129
3.49.2 Member Function Documentation	129
3.49.2.1 iteration()	129
3.50 LazyHashSupplier< T >	130
3.50.1 Detailed Description	130
3.51 LinkLabel	130
3.51.1 Member Function Documentation	130
3.51.1.1 setText()	130
3.52 Location Enum Reference	131
3.52.1 Detailed Description	132
3.52.2 Constructor & Destructor Documentation	132
3.52.2.1 Location()	133
3.52.3 Member Function Documentation	133
3.52.3.1 locsMatStr()	133
3.52.3.2 locsMat()	133

3.52.3.3 matrixStr()	133
3.52.3.4 createMatIndicesStr()	134
3.52.3.5 getLoc() [1/7]	134
3.52.3.6 getLoc() [2/7]	134
3.52.3.7 getLoc() [3/7]	135
3.52.3.8 getLoc() [4/7]	135
3.52.3.9 valueOf()	135
3.52.3.10 getLoc() [5/7]	136
3.52.3.11 getLoc() [6/7]	136
3.52.3.12 getLoc() [7/7]	137
3.52.3.13 flip() [1/2]	137
3.52.3.14 isInBounds()	137
3.52.3.15 getColString()	138
3.52.3.16 flip() [2/2]	138
3.52.3.17 isBlackSquare()	138
3.52.3.18 isWhiteSquare()	138
3.52.3.19 toString()	139
3.52.3.20 getMaxDistance()	139
3.52.3.21 getRowString()	139
3.52.3.22 hash()	139
3.52.4 Member Data Documentation	140
3.52.4.1 asLong	140
3.52.4.2 asInt	140
3.52.4.3 row	140
3.52.4.4 col	140
3.53 LoginInfo	140
3.53.1 Detailed Description	141
3.53.2 Constructor & Destructor Documentation	141
3.53.2.1 LoginInfo() [1/3]	141
3.53.2.2 LoginInfo() [2/3]	141
3.53.2.3 LoginInfo() [3/3]	141
3.53.3 Member Function Documentation	142
3.53.3.1 getProfilePic()	142
3.53.3.2 setProfilePic()	142
3.53.3.3 isGuest()	142
3.53.3.4 getLoginType()	142
3.53.3.5 setLoginType()	143
3.53.3.6 getUsername()	143
3.53.3.7 setUsername()	143
3.53.3.8 getPassword()	143
3.53.3.9 setPassword()	144
3.53.3.10 toString()	144



3.53.3.11 asUser()	144
3.53.3.12 initDebugLoginValues()	144
3.54 LoginType Enum Reference	145
3.54.1 Detailed Description	145
3.54.2 Member Function Documentation	145
3.54.2.1 asUser()	145
3.54.2.2 toString()	145
3.54.3 Member Data Documentation	145
3.54.3.1 LOGIN	146
3.54.3.2 REGISTER	146
3.54.3.3 GUEST	146
3.54.3.4 CANCEL	146
3.54.3.5 NOT_SET_YET	146
3.55 Math Enum Reference	146
3.55.1 Detailed Description	147
3.55.2 Member Function Documentation	147
3.55.2.1 nullIf0()	147
3.55.2.2 formatNum() [1/2]	147
3.55.2.3 strVal()	148
3.55.2.4 formatNum() [2/2]	148
3.55.2.5 str()	148
3.55.2.6 asFloat()	149
3.55.2.7 zeroIfNull()	149
3.55.2.8 execute() [1/2]	149
3.55.2.9 execute() [2/2]	150
3.55.2.10 apply()	150
3.55.2.11 strSource()	150
3.55.2.12 simpleOperation()	151
3.55.3 Member Data Documentation	152
3.55.3.1 Plus	152
3.55.3.2 Mult	152
3.55.3.3 Div	152
3.55.3.4 col	153
3.56 MathUtils	153
3.56.1 Detailed Description	153
3.56.2 Member Function Documentation	153
3.56.2.1 log()	153
3.57 Message	154
3.57.1 Detailed Description	154
3.57.1.1 Message.	154
3.57.2 Constructor & Destructor Documentation	154
3.57.2.1 Message() [1/4]	154

3.57.2.2 Message() [ 2 / 4 ]	154
3.57.2.3 Message() [ 3 / 4 ]	155
3.57.2.4 Message() [ 4 / 4 ]	155
3.57.3 Member Function Documentation	155
3.57.3.1 askForLogin()	155
3.57.3.2 returnLogin()	156
3.57.3.3 welcomeMessage()	156
3.57.3.4 initGame()	156
3.57.3.5 throwError()	157
3.57.3.6 updateByMove()	157
3.57.3.7 waitYourTurn()	158
3.57.3.8 waitMatch()	158
3.57.3.9 gameOver()	158
3.57.3.10 askForGameSettings()	159
3.57.3.11 returnGameSettings()	159
3.57.3.12 interrupt()	159
3.57.3.13 bye()	160
3.57.3.14 error()	160
3.57.3.15 returnMove()	160
3.57.3.16 answerQuestion()	161
3.57.3.17 askQuestion()	161
3.57.3.18 checkUsernameAvailability()	162
3.57.3.19 returnUsernameNotAvailable()	162
3.57.3.20 returnUsernameAvailable()	162
3.57.3.21 dbRequest()	163
3.57.3.22 setDbRequest()	163
3.57.3.23 returnDBResponse()	163
3.57.3.24 syncLists()	164
3.57.3.25 getError()	164
3.57.3.26 setError()	164
3.57.3.27 getDBRequest()	165
3.57.3.28 getUsernameSuggestions()	165
3.57.3.29 setUsernameSuggestions()	165
3.57.3.30 getDBResponse()	165
3.57.3.31 setDBResponse()	166
3.57.3.32 getMoveStack()	166
3.57.3.33 setMoveStack()	166
3.57.3.34 getSyncedLists()	166
3.57.3.35 setSyncedLists()	167
3.57.3.36 getRespondingToMsgId()	167
3.57.3.37 setRespondingToMsgId()	167
3.57.3.38 setRespondingTo()	167

3.57.3.39 isResponse()	168
3.57.3.40 getOtherPlayer()	168
3.57.3.41 setOtherPlayer()	168
3.57.3.42 getQuestion()	168
3.57.3.43 setQuestion()	169
3.57.3.44 getAvailable()	169
3.57.3.45 setAvailable()	169
3.57.3.46 getUsername()	169
3.57.3.47 setUsername()	170
3.57.3.48 getGameSettings()	170
3.57.3.49 setGameSettings()	170
3.57.3.50 getPreMoves()	170
3.57.3.51 setPreMoves()	171
3.57.3.52 getSubject()	171
3.57.3.53 getPlayerColor()	171
3.57.3.54 setPlayerColor()	171
3.57.3.55 getMove()	172
3.57.3.56 setMove()	172
3.57.3.57 getGameTime()	172
3.57.3.58 setGameTime()	172
3.57.3.59 getGameStatus()	173
3.57.3.60 setGameStatus()	173
3.57.3.61 getPossibleMoves()	173
3.57.3.62 setPossibleMoves()	173
3.57.3.63 getBoard()	174
3.57.3.64 setBoard()	174
3.57.3.65 getLoginInfo()	174
3.57.3.66 setLoginInfo()	174
3.57.3.67 getMessageType()	175
3.57.3.68 isSubject()	175
3.57.4 Member Data Documentation	175
3.57.4.1 messageId	175
3.58 MessageCallback	176
3.58.1 Detailed Description	176
3.59 MessagesHandler	176
3.59.1 Detailed Description	176
3.59.2 Constructor & Destructor Documentation	176
3.59.2.1 MessagesHandler()	176
3.59.3 Member Function Documentation	177
3.59.3.1 onCancelQuestion()	177
3.59.3.2 interruptBlocking()	177
3.59.3.3 blockTilRes()	177

3.59.3.4 noBlockRequest()	178
3.59.3.5 receivedMessage()	178
3.59.3.6 prepareForDisconnect()	178
3.59.3.7 onAnyMsg()	178
3.59.3.8 onDisconnected()	179
3.59.3.9 onAnyDisconnection()	179
3.59.3.10 onPlannedDisconnect()	179
3.59.3.11 onUnplannedDisconnect()	179
3.59.3.12 createDisconnectedError()	179
3.59.3.13 onLogin()	179
3.59.3.14 onResign()	180
3.59.3.15 onAddTime()	180
3.59.3.16 onOfferDraw()	180
3.59.3.17 onWelcomeMessage()	180
3.59.3.18 onGetGameSettings()	181
3.59.3.19 onWaitForMatch()	181
3.59.3.20 onInitGame()	181
3.59.3.21 onWaitTurn()	181
3.59.3.22 onGetMove()	182
3.59.3.23 onUpdateByMove()	182
3.59.3.24 onGameOver()	182
3.59.3.25 onError()	182
3.59.3.26 onQuestion()	183
3.59.3.27 onBye()	183
3.59.3.28 onUsernameAvailability()	183
3.59.3.29 onDBRequest()	183
3.59.3.30 onDBResponse()	184
3.59.3.31 onUpdateSyncedList()	184
3.59.3.32 onInterrupt()	184
3.59.3.33 onIsAlive()	184
3.59.3.34 onAlive()	185
3.59.4 Member Data Documentation	185
3.59.4.1 socket	185
3.60 MessageType Enum Reference	185
3.61 MinimaxMove	186
3.61.1 Detailed Description	186
3.61.2 Constructor & Destructor Documentation	186
3.61.2.1 MinimaxMove() [1/3]	186
3.61.2.2 MinimaxMove() [2/3]	186
3.61.2.3 MinimaxMove() [3/3]	187
3.61.3 Member Function Documentation	187
3.61.3.1 isDeeperAndBetterThan()	187

3.61.3.2	<a href="#">getMoveDepth()</a>	187
3.61.3.3	<a href="#">getMove()</a>	188
3.61.3.4	<a href="#">setMove()</a>	188
3.61.3.5	<a href="#">getMoveEvaluation()</a>	188
3.61.3.6	<a href="#">setMoveEvaluation()</a>	188
3.61.3.7	<a href="#">hashCode()</a>	189
3.61.3.8	<a href="#">equals()</a>	189
3.61.3.9	<a href="#">toString()</a>	189
3.61.3.10	<a href="#">getShortPrintingStr()</a>	189
3.61.3.11	<a href="#">compareTo()</a>	190
3.62	<a href="#">Move</a>	190
3.62.1	<a href="#">Detailed Description</a>	190
3.62.2	<a href="#">Class Documentation</a>	190
3.62.2.1	<a href="#">enum ver14::SharedClasses::Game::Moves::Move::ThreefoldStatus</a>	190
3.62.3	<a href="#">Constructor &amp; Destructor Documentation</a>	191
3.62.3.1	<a href="#">Move() [1/4]</a>	191
3.62.3.2	<a href="#">Move() [2/4]</a>	191
3.62.3.3	<a href="#">Move() [3/4]</a>	191
3.62.3.4	<a href="#">Move() [4/4]</a>	192
3.62.4	<a href="#">Member Function Documentation</a>	192
3.62.4.1	<a href="#">castling()</a>	192
3.62.4.2	<a href="#">threefoldClaim()</a>	192
3.62.4.3	<a href="#">flipMove()</a>	193
3.62.4.4	<a href="#">copyMove()</a>	193
3.62.4.5	<a href="#">setMoveAnnotation()</a>	193
3.62.4.6	<a href="#">getDisabledCastling()</a>	194
3.62.4.7	<a href="#">setDisabledCastling()</a>	194
3.62.4.8	<a href="#">setThreefoldOption()</a>	194
3.62.4.9	<a href="#">getMovingColor()</a>	194
3.62.4.10	<a href="#">setMovingColor()</a>	195
3.62.4.11	<a href="#">getPrevFullMoveClock()</a>	195
3.62.4.12	<a href="#">setPrevFullMoveClock()</a>	195
3.62.4.13	<a href="#">getPrevHalfMoveClock()</a>	195
3.62.4.14	<a href="#">setPrevHalfMoveClock()</a>	196
3.62.4.15	<a href="#">getPromotingTo()</a>	196
3.62.4.16	<a href="#">setPromotingTo()</a>	196
3.62.4.17	<a href="#">getIntermediateMove()</a>	196
3.62.4.18	<a href="#">setIntermediateMove()</a>	197
3.62.4.19	<a href="#">isCheck()</a>	197
3.62.4.20	<a href="#">getMoveEvaluation()</a>	197
3.62.4.21	<a href="#">setMoveEvaluation()</a>	197
3.62.4.22	<a href="#">getCapturingPieceType()</a>	198

3.62.4.23 isReversible()	198
3.62.4.24 setReversible()	198
3.62.4.25 isCapturing()	198
3.62.4.26 setCapturing()	199
3.62.4.27 equals()	199
3.62.4.28 toString()	199
3.62.4.29 getAnnotation()	200
3.62.4.30 strictEquals()	200
3.62.4.31 getMoveFlag()	200
3.62.4.32 setMoveFlag()	200
3.62.4.33 compareTo()	201
3.62.4.34 getEnPassantLoc()	201
3.62.4.35 setEnPassantLoc()	201
3.63 MoveAnnotation	202
3.63.1 Detailed Description	202
3.63.2 Member Function Documentation	202
3.63.2.1 annotate() [1/2]	202
3.63.2.2 annotate() [2/2]	202
3.63.2.3 basicAnnotate()	203
3.64 Move.MoveFlag Enum Reference	203
3.64.1 Detailed Description	203
3.64.2 Constructor & Destructor Documentation	204
3.64.2.1 MoveFlag() [1/2]	204
3.64.2.2 MoveFlag() [2/2]	204
3.64.3 Member Function Documentation	204
3.64.3.1 equals()	204
3.64.4 Member Data Documentation	204
3.64.4.1 NormalMove	205
3.64.4.2 EnPassant	205
3.64.4.3 DoublePawnPush	205
3.64.4.4 Promotion	205
3.64.4.5 ShortCastle	205
3.64.4.6 LongCastle	205
3.64.4.7 CASTLING_FLAGS	205
3.64.4.8 isCastling	206
3.64.4.9 castlingSide	206
3.65 MoveFlags Enum Reference	206
3.66 MovesList	206
3.67 MyError	207
3.67.1 Detailed Description	207
3.67.2 Constructor & Destructor Documentation	207
3.67.2.1 MyError() [1/4]	207

3.67.2.2 MyError() [2/4]	207
3.67.2.3 MyError() [3/4]	208
3.67.2.4 MyError() [4/4]	208
3.67.3 Member Function Documentation	208
3.67.3.1 getHandledStr()	208
3.67.3.2 toString()	208
3.67.3.3 errToString()	209
3.68 MyJButton	209
3.68.1 Detailed Description	209
3.68.2 Constructor & Destructor Documentation	209
3.68.2.1 MyJButton() [1/6]	209
3.68.2.2 MyJButton() [2/6]	210
3.68.2.3 MyJButton() [3/6]	210
3.68.2.4 MyJButton() [4/6]	210
3.68.2.5 MyJButton() [5/6]	210
3.68.2.6 MyJButton() [6/6]	211
3.68.3 Member Function Documentation	211
3.68.3.1 setOnClick()	211
3.68.3.2 setText()	211
3.68.3.3 setFont()	212
3.68.3.4 getMinSize()	212
3.68.3.5 replaceWithCancel()	212
3.68.3.6 resetState()	213
3.69 MyJFrame	213
3.69.1 Detailed Description	213
3.69.2 Constructor & Destructor Documentation	213
3.69.2.1 MyJFrame()	213
3.69.3 Member Function Documentation	214
3.69.3.1 debugAdapter()	214
3.69.3.2 getMyAdapter()	214
3.69.3.3 setOnExit()	214
3.69.3.4 doXClick()	215
3.69.3.5 setOnResize()	215
3.69.4 Member Data Documentation	215
3.69.4.1 onClose	215
3.70 MyLbl	215
3.71 MyThread	216
3.71.1 Detailed Description	216
3.71.2 Constructor & Destructor Documentation	216
3.71.2.1 MyThread()	216
3.71.3 Member Function Documentation	216
3.71.3.1 currentThread()	216

3.71.3.2 stopRun()	217
3.71.3.3 ignoreErrs()	217
3.71.3.4 reactivateErrs()	217
3.71.3.5 handledRun()	217
3.72 Piece Enum Reference	217
3.72.1 Detailed Description	218
3.72.2 Member Function Documentation	218
3.72.2.1 getPieceFromFen()	218
3.72.2.2 getPiece()	219
3.72.2.3 getPieceIcon() [1/2]	219
3.72.2.4 getPieceIcon() [2/2]	219
3.72.2.5 getFen()	220
3.72.2.6 isOnMyTeam() [1/2]	220
3.72.2.7 isOnMyTeam() [2/2]	220
3.72.3 Member Data Documentation	221
3.72.3.1 W_P	221
3.72.3.2 W_R	221
3.72.3.3 W_B	221
3.72.3.4 W_N	221
3.72.3.5 W_Q	221
3.72.3.6 W_K	221
3.72.3.7 B_P	222
3.72.3.8 B_R	222
3.72.3.9 B_B	222
3.72.3.10 B_N	222
3.72.3.11 B_Q	222
3.72.3.12 B_K	222
3.72.3.13 ALL_PIECES	222
3.72.3.14 pieceType	222
3.72.3.15 playerColor	223
3.73 PieceType Enum Reference	223
3.73.1 Detailed Description	224
3.73.2 Member Function Documentation	224
3.73.2.1 getPieceType()	224
3.73.2.2 getAttackingDirections() [1/2]	224
3.73.2.3 getWhitePieceFen()	225
3.73.2.4 getPieceIcon()	225
3.73.2.5 compareMovementType() [1/2]	225
3.73.2.6 compareMovementType() [2/2]	226
3.73.2.7 isDiagonalPiece()	226
3.73.2.8 isLinePiece()	226
3.73.2.9 getPieceName()	227



3.73.2.10 isAttack()	227
3.73.2.11 getAttackingDirections() [2/2]	227
3.73.2.12 getWalkingDirections()	228
3.73.3 Member Data Documentation	228
3.73.3.1 PAWN	228
3.73.3.2 ROOK	228
3.73.3.3 BISHOP	228
3.73.3.4 KNIGHT	228
3.73.3.5 QUEEN	229
3.73.3.6 KING	229
3.73.3.7 NUM_OF_PIECE_TYPES	229
3.73.3.8 PIECE_TYPES	229
3.73.3.9 UNIQUE_MOVES_PIECE_TYPES	229
3.73.3.10 MINOR_PIECES	229
3.73.3.11 MAJOR_PIECES	229
3.73.3.12 CAN_PROMOTE_TO	230
3.73.3.13 ATTACKING_PIECE_TYPES	230
3.73.3.14 ATTACKING_DIRECTIONS	230
3.73.3.15 whitelcon	230
3.73.3.16 blacklcon	230
3.73.3.17 value	230
3.73.3.18 isSliding	230
3.73.3.19 asInt	231
3.74 PlayerColor Enum Reference	231
3.74.1 Detailed Description	231
3.74.2 Constructor & Destructor Documentation	231
3.74.2.1 PlayerColor()	231
3.74.3 Member Function Documentation	232
3.74.3.1 getColor()	232
3.74.3.2 getPlayerFromFen()	232
3.74.3.3 getOpponent()	232
3.74.3.4 toString()	233
3.74.3.5 getName()	233
3.74.4 Member Data Documentation	233
3.74.4.1 WHITE	233
3.74.4.2 BLACK	234
3.74.4.3 NO_PLAYER	234
3.74.4.4 PLAYER_COLORS	234
3.74.4.5 NUM_OF_PLAYERS	234
3.74.4.6 asInt	234
3.74.4.7 indexOf2	234
3.74.4.8 startingRow	235

3.74.4.9 diff	235
3.75 PreMadeRequest	235
3.75.1 Detailed Description	235
3.75.2 Member Function Documentation	236
3.75.2.1 getRequestVariations()	236
3.75.2.2 createBuilder()	236
3.75.3 Member Data Documentation	236
3.75.3.1 TopPlayers	236
3.75.3.2 Games	236
3.75.3.3 DeleteUnfGames	237
3.75.3.4 StatsByTimeOfDay	237
3.75.3.5 ChangeProfilePic	237
3.75.3.6 statistics	237
3.75.3.7 authSettings	237
3.76 PremovesGenerator	237
3.77 Question	238
3.77.1 Detailed Description	238
3.77.2 Class Documentation	238
3.77.2.1 enum ver14::SharedClasses::Misc::Question::QuestionType	238
3.77.3 Constructor & Destructor Documentation	239
3.77.3.1 Question() [1/2]	239
3.77.3.2 Question() [2/2]	239
3.77.4 Member Function Documentation	239
3.77.4.1 drawOffer()	240
3.77.4.2 getDefaultAnswer()	240
3.77.4.3 setDefaultAnswer()	240
3.77.4.4 getQuestionStr()	240
3.77.4.5 getPossibleAnswers()	241
3.77.4.6 equals()	241
3.77.4.7 toString()	241
3.77.4.8 Answer()	242
3.77.5 Member Data Documentation	242
3.77.5.1 Threefold	242
3.77.5.2 Rematch	242
3.77.5.3 questionStr	242
3.77.5.4 questionType	242
3.78 RegEx	243
3.78.1 Detailed Description	243
3.78.2 Constructor & Destructor Documentation	243
3.78.2.1 RegEx() [1/2]	243
3.78.2.2 RegEx() [2/2]	244
3.78.3 Member Function Documentation	244

3.78.3.1 isSavedDate()	244
3.78.3.2 canBeEmpty()	245
3.78.3.3 get()	245
3.78.3.4 setUseDontMatch()	245
3.78.3.5 getDetails()	246
3.78.3.6 check()	246
3.78.3.7 getRegex()	246
3.78.4 Member Data Documentation	246
3.78.4.1 Fen	246
3.78.4.2 Username	247
3.78.4.3 Password	247
3.78.4.4 Icon	247
3.78.4.5 StrUtilSkip	247
3.78.4.6 Numbers	247
3.78.4.7 URL	247
3.78.4.8 DontSaveGame	247
3.78.4.9 IPPortAddress	248
3.78.4.10 Any	248
3.78.4.11 dontMatch	248
3.79 SyncedItems< E extends SyncableItem >.Remover< L >	248
3.80 RequestBuilder	248
3.80.1 Detailed Description	249
3.80.2 Constructor & Destructor Documentation	249
3.80.2.1 RequestBuilder() [1/4]	249
3.80.2.2 RequestBuilder() [2/4]	249
3.80.2.3 RequestBuilder() [3/4]	250
3.80.2.4 RequestBuilder() [4/4]	250
3.80.3 Member Function Documentation	251
3.80.3.1 createVariation()	251
3.80.3.2 changePassword()	251
3.80.3.3 changeProfilePic()	251
3.80.3.4 addShouldSync()	251
3.80.3.5 deleteAllUnFinishedGames()	252
3.80.3.6 games()	252
3.80.3.7 setSubBuilder()	252
3.80.3.8 statsByTimeOfDay()	252
3.80.3.9 top()	253
3.80.3.10 getShouldSync()	253
3.80.3.11 getArgVal()	253
3.80.3.12 createResponse()	254
3.80.3.13 getPreDescription()	254
3.80.3.14 getPostDescription()	254

3.80.3.15 getName()	254
3.80.3.16 getArgs()	255
3.80.3.17 build()	255
3.80.4 Member Data Documentation	255
3.80.4.1 TIE_STR	255
3.80.4.2 args	255
3.80.4.3 statement	255
3.80.4.4 name	256
3.80.4.5 postDescription	256
3.80.4.6 preDescription	256
3.80.4.7 subBuilder	256
3.81 Selection	256
3.81.1 Detailed Description	257
3.81.2 Class Documentation	257
3.81.2.1 interface ver14::SharedClasses::DBActions::Statements::Selection::Join	257
3.81.2.2 interface ver14::SharedClasses::DBActions::Statements::Selection::Order	257
3.81.3 Constructor & Destructor Documentation	257
3.81.3.1 Selection() [1/2]	257
3.81.3.2 Selection() [2/2]	258
3.81.4 Member Function Documentation	258
3.81.4.1 nestMe()	258
3.81.4.2 createStatement()	258
3.81.4.3 top()	259
3.81.4.4 join()	259
3.81.4.5 orderBy()	259
3.82 CastlingRights.Side Enum Reference	260
3.82.1 Detailed Description	260
3.82.2 Constructor & Destructor Documentation	260
3.82.2.1 Side()	260
3.82.3 Member Function Documentation	261
3.82.3.1 toString()	261
3.82.3.2 kingFinalLoc()	261
3.82.4 Member Data Documentation	261
3.82.4.1 KING	261
3.82.4.2 QUEEN	262
3.82.4.3 SIDES	262
3.82.4.4 rookStartingCol	262
3.82.4.5 castledRookCol	262
3.82.4.6 castledKingCol	262
3.82.4.7 kingTravelDistance	262
3.82.4.8 castlingNotation	262
3.82.4.9 asInt	262

3.82.4.10 mult . . . . .	263
3.83 GameState.SpecificStatus Enum Reference . . . . .	263
3.83.1 Detailed Description . . . . .	263
3.83.2 Constructor & Destructor Documentation . . . . .	263
3.83.2.1 SpecificStatus() [1/2] . . . . .	263
3.83.2.2 SpecificStatus() [2/2] . . . . .	263
3.83.3 Member Function Documentation . . . . .	264
3.83.3.1 toString() . . . . .	264
3.83.4 Member Data Documentation . . . . .	264
3.83.4.1 Checkmate . . . . .	264
3.83.4.2 TimedOut . . . . .	264
3.83.4.3 TimedOutVsInsufficientMaterial . . . . .	264
3.83.4.4 Resignation . . . . .	265
3.83.4.5 GameGoesOn . . . . .	265
3.83.4.6 ThreeFoldRepetition . . . . .	265
3.83.4.7 Stalemate . . . . .	265
3.83.4.8 InsufficientMaterial . . . . .	265
3.83.4.9 FiftyMoveRule . . . . .	265
3.83.4.10 TieByAgreement . . . . .	265
3.83.4.11 PlayerDisconnectedVsAi . . . . .	266
3.83.4.12 PlayerDisconnectedVsReal . . . . .	266
3.83.4.13 ServerStoppedGame . . . . .	266
3.83.4.14 gameStateType . . . . .	266
3.84 SQLStatement . . . . .	266
3.84.1 Detailed Description . . . . .	267
3.84.2 Constructor & Destructor Documentation . . . . .	267
3.84.2.1 SQLStatement() . . . . .	267
3.84.3 Member Function Documentation . . . . .	267
3.84.3.1 replace() . . . . .	267
3.84.3.2 createStatement() . . . . .	268
3.84.3.3 getStatement() . . . . .	268
3.84.3.4 toString() . . . . .	268
3.84.4 Member Data Documentation . . . . .	268
3.84.4.1 type . . . . .	268
3.85 Square . . . . .	269
3.85.1 Detailed Description . . . . .	269
3.85.2 Constructor & Destructor Documentation . . . . .	269
3.85.2.1 Square() [1/2] . . . . .	269
3.85.2.2 Square() [2/2] . . . . .	269
3.85.3 Member Function Documentation . . . . .	270
3.85.3.1 setEmpty() . . . . .	270
3.85.3.2 getPiece() . . . . .	270

3.85.3.3 setPiece()	270
3.85.3.4 getLoc()	270
3.85.3.5 toString()	271
3.85.3.6 getFen()	271
3.85.3.7 isEmpty()	271
3.85.3.8 getPiecelcon()	271
3.85.4 Member Data Documentation	272
3.85.4.1 EMPTY_PIECE	272
3.86 StatusResponse	272
3.86.1 Detailed Description	272
3.86.2 Constructor & Destructor Documentation	272
3.86.2.1 StatusResponse() [1/2]	272
3.86.2.2 StatusResponse() [2/2]	273
3.86.3 Member Function Documentation	273
3.86.3.1 getDetails()	273
3.86.3.2 isAnyData()	273
3.86.3.3 clean()	274
3.87 MyJFrame.StringClosing	274
3.87.1 Detailed Description	274
3.87.2 Member Function Documentation	274
3.87.2.1 show()	274
3.87.2.2 checkClosingVal()	274
3.88 MyLbl.StringModifier	275
3.89 StrUtils	275
3.89.1 Detailed Description	275
3.89.2 Member Function Documentation	275
3.89.2.1 isAbsoluteUrl()	275
3.89.2.2 dontCapFull()	276
3.89.2.3 htmlNewLines()	276
3.89.2.4 countMatches()	276
3.89.2.5 format() [1/3]	277
3.89.2.6 format() [2/3]	277
3.89.2.7 format() [3/3]	278
3.89.2.8 isEmpty()	278
3.89.2.9 clean()	278
3.89.2.10 formatDate() [1/3]	279
3.89.2.11 formatDate() [2/3]	279
3.89.2.12 formatDate() [3/3]	279
3.89.2.13 parseURLS()	280
3.89.2.14 uppercase()	280
3.89.2.15 dontCapWord()	280
3.89.2.16 formatDateSQL()	281

3.89.2.17	<a href="#">getPort()</a> [1/2]	281
3.89.2.18	<a href="#">getPort()</a> [2/2]	281
3.89.2.19	<a href="#">getUrl()</a> [1/2]	283
3.89.2.20	<a href="#">getUrl()</a> [2/2]	283
3.89.2.21	<a href="#">fitInside()</a>	283
3.89.2.22	<a href="#">fixHtml()</a>	284
3.89.2.23	<a href="#">strlNN()</a>	284
3.89.2.24	<a href="#">splitArr()</a> [1/3]	285
3.89.2.25	<a href="#">splitArr()</a> [2/3]	285
3.89.2.26	<a href="#">splitArr()</a> [3/3]	285
3.89.2.27	<a href="#">createTimeGreeting()</a>	286
3.89.2.28	<a href="#">main()</a>	286
3.89.2.29	<a href="#">createTimeStr()</a>	286
3.89.2.30	<a href="#">awful()</a>	287
3.89.2.31	<a href="#">dateTimePrefix()</a>	287
3.89.2.32	<a href="#">repeat()</a>	287
3.90	<a href="#">SwitchCase</a>	288
3.90.1	<a href="#">Detailed Description</a>	288
3.90.2	<a href="#">Constructor &amp; Destructor Documentation</a>	288
3.90.2.1	<a href="#">SwitchCase()</a>	288
3.90.3	<a href="#">Member Function Documentation</a>	288
3.90.3.1	<a href="#">equals()</a>	288
3.90.3.2	<a href="#">defaultCase()</a>	290
3.90.3.3	<a href="#">condition()</a>	290
3.90.3.4	<a href="#">ifTrue()</a>	290
3.90.3.5	<a href="#">toString()</a>	291
3.91	<a href="#">SyncableItem</a>	291
3.91.1	<a href="#">Member Function Documentation</a>	291
3.91.1.1	<a href="#">ID()</a>	291
3.92	<a href="#">SyncedItems&lt; E extends SyncableItem &gt;</a>	292
3.93	<a href="#">SyncedListType Enum Reference</a>	292
3.93.1	<a href="#">Member Data Documentation</a>	292
3.93.1.1	<a href="#">RESUMABLE_GAMES</a>	292
3.93.1.2	<a href="#">JOINABLE_GAMES</a>	292
3.93.1.3	<a href="#">CONNECTED_USERS</a>	293
3.93.1.4	<a href="#">ONGOING_GAMES</a>	293
3.94	<a href="#">Table Enum Reference</a>	293
3.94.1	<a href="#">Detailed Description</a>	293
3.94.2	<a href="#">Constructor &amp; Destructor Documentation</a>	293
3.94.2.1	<a href="#">Table()</a>	293
3.94.3	<a href="#">Member Function Documentation</a>	294
3.94.3.1	<a href="#">tableAndValues()</a>	294

3.94.3.2 escapeValues()	294
3.94.4 Member Data Documentation	294
3.94.4.1 Games	294
3.94.4.2 UnfinishedGames	295
3.94.4.3 Users	295
3.94.4.4 cols	295
3.95 TableDBResponse	295
3.95.1 Detailed Description	295
3.95.2 Constructor & Destructor Documentation	296
3.95.2.1 TableDBResponse() [1/3]	296
3.95.2.2 TableDBResponse() [2/3]	296
3.95.2.3 TableDBResponse() [3/3]	296
3.95.3 Member Function Documentation	296
3.95.3.1 numRows()	297
3.95.3.2 getFirstRow()	297
3.95.3.3 getCell() [1/2]	297
3.95.3.4 getColumnIndex()	297
3.95.3.5 getCell() [2/2]	298
3.95.3.6 isAnyData()	298
3.95.3.7 clean()	298
3.95.3.8 toString()	299
3.95.3.9 rowToString()	299
3.95.3.10 getColumns()	299
3.95.3.11 getRows()	299
3.95.4 Member Data Documentation	300
3.95.4.1 columns	300
3.95.4.2 rows	300
3.96 ThreadsManager	300
3.96.1 Detailed Description	300
3.96.2 Member Function Documentation	300
3.96.2.1 handleErrors()	300
3.96.2.2 createThread()	301
3.97 MyThread.ThreadStatus Enum Reference	301
3.97.1 Member Data Documentation	301
3.97.1.1 NOT_STARTED	301
3.97.1.2 RUNNING	302
3.97.1.3 DONE	302
3.98 ThrowingCallback< T >	302
3.98.1 Detailed Description	302
3.98.2 Member Function Documentation	302
3.98.2.1 callback()	302
3.99 ThrowingRunnable	303



3.100 TimeFormat . . . . .	303
3.101 TimeFormatComponent . . . . .	304
3.102 UnfinishedGame . . . . .	304
3.102.1 Detailed Description . . . . .	304
3.102.2 Constructor & Destructor Documentation . . . . .	304
3.102.2.1 UnfinishedGame() . . . . .	305
3.102.3 Member Function Documentation . . . . .	305
3.102.3.1 isCreatorToMove() . . . . .	305
3.102.4 Member Data Documentation . . . . .	305
3.102.4.1 playerColorToMove . . . . .	305
3.102.4.2 playerToMove . . . . .	306
3.103 Update . . . . .	306
3.103.1 Detailed Description . . . . .	306
3.103.2 Constructor & Destructor Documentation . . . . .	306
3.103.2.1 Update() . . . . .	306
3.103.3 Member Function Documentation . . . . .	307
3.103.3.1 createState() . . . . .	307
3.104 UserInfo . . . . .	307
3.104.1 Member Function Documentation . . . . .	307
3.104.1.1 ID() . . . . .	308
3.105 ValueBtn< T > . . . . .	308
3.105.1 Detailed Description . . . . .	308
3.105.2 Constructor & Destructor Documentation . . . . .	308
3.105.2.1 ValueBtn() . . . . .	308
3.106 PreMadeRequest.VariationCreator . . . . .	309
3.106.1 Detailed Description . . . . .	309
3.106.2 Member Function Documentation . . . . .	309
3.106.2.1 create() . . . . .	309
3.107 VoidCallback . . . . .	310
3.107.1 Detailed Description . . . . .	310
3.107.2 Member Function Documentation . . . . .	310
3.107.2.1 callback() . . . . .	310
<b>Index</b>	<b>311</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

AiParameters.AiType . . . . .	9
ArgsUtil . . . . .	21
ArgType . . . . .	23
ArrUtils . . . . .	25
AuthSettings . . . . .	27
BitData . . . . .	33
Callback< T > . . . . .	40
Callback< Message > . . . . .	40
MessageCallback . . . . .	176
Callback< Question.Answer > . . . . .	40
AnswerCallback . . . . .	10
Callback< ver14.SharedClasses.Sync.SyncedItems< E > > . . . . .	40
MyJFrame.Closing< T > . . . . .	46
MyJFrame.Closing< Boolean > . . . . .	46
MyJFrame.BooleanClosing . . . . .	39
MyJFrame.Closing< String > . . . . .	46
MyJFrame.StringClosing . . . . .	274
MyJFrame.Closing<?> . . . . .	46
Comparable . . . . .	
MinimaxMove . . . . .	186
Move . . . . .	190
MovesList.CompareMoves . . . . .	59
Config<?> . . . . .	64
ConfirmDialogs . . . . .	68
Direction . . . . .	82
EfficientGen< K, V > . . . . .	91
Environment . . . . .	93
EnvManager . . . . .	93
Error . . . . .	
MyError . . . . .	207
ErrorHandler< E extends MyError > . . . . .	94
EvaluationParameters . . . . .	105
FontManager . . . . .	107
GameSettings.GameType . . . . .	111

Graphable	124
GraphableDBResponse	124
GraphElementType	125
MyJFrame.MyAdapter.HeldDown	127
IDsGenerator	128
Iterable	
Board	34
StrUtils.IterationThingy< T >	129
JButton	
MyJButton	209
ValueBtn< T >	308
JFrame	
GameView	123
MyJFrame	213
JLabel	
MyLbl	215
LinkLabel	130
Selection.Join	256
LazyHashSupplier< Long >	130
Location	131
LoginType	145
Math	146
MathUtils	153
MessagesHandler	176
MessageType	185
MoveAnnotation	202
Move.MoveFlag	203
MoveFlags	206
Selection.Order	256
Piece	217
PlayerColor	231
PreMadeRequest	235
PremovesGenerator	237
Question.QuestionType	238
Condition.Relation	59
SyncedItems< E extends SyncableItem >.Remover< L >	248
CastlingRights.Side	260
GameStatus.SpecificStatus	263
DBResponse.Status	77
MyLbl.StringModifier	275
StrUtils	275
SwitchCase	288
SyncableItem	291
GameInfo	107
CreatedGame	71
EstablishedGameInfo	95
ArchivedGameInfo	16
UnfinishedGame	304
UserInfo	307
SyncedItems< ? >	292
SyncedListType	292
Table	293
Thread	
MyThread	216
AppSocket	10
HandledThread	125
ThreadsManager	300

MyThread.ThreadStatus . . . . .	301
Move.ThreefoldStatus . . . . .	190
ThrowingCallback< T > . . . . .	302
ThrowingRunnable . . . . .	303
TimeFormatComponent . . . . .	304
AiParameters . . . . .	9
GameSettings . . . . .	111
DBRequest.Type . . . . .	73
PreMadeRequest.VariationCreator . . . . .	309
VoidCallback . . . . .	310
ArrayList	
MovesList . . . . .	206
ConcurrentHashMap	
SyncedItems< E extends SyncableItem > . . . . .	292
Serializable	
LazyHashSupplier< T > . . . . .	130
Arg . . . . .	17
Config< V > . . . . .	64
Condition . . . . .	59
DBRequest . . . . .	73
DBResponse . . . . .	77
GraphableDBResponse . . . . .	124
StatusResponse . . . . .	272
TableDBResponse . . . . .	295
GraphElement . . . . .	125
RequestBuilder . . . . .	248
SQLStatement . . . . .	266
CustomStatement . . . . .	72
Delete . . . . .	81
Selection . . . . .	256
Update . . . . .	306
Col . . . . .	48
Evaluation . . . . .	98
GameStatus . . . . .	112
GameStatus.GameStatusType . . . . .	120
AiParameters . . . . .	9
Board . . . . .	34
PieceType . . . . .	223
Square . . . . .	269
GameSettings . . . . .	111
GameTime . . . . .	123
TimeFormat . . . . .	303
BasicMove . . . . .	28
Move . . . . .	190
CastlingRights . . . . .	41
MinimaxMove . . . . .	186
MovesList . . . . .	206
GameInfo . . . . .	107
LoginInfo . . . . .	140
Question . . . . .	238
Message . . . . .	154
UserInfo . . . . .	307
RegEx . . . . .	243
Supplier	
LazyHashSupplier< T > . . . . .	130



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">AiParameters</a>	9
<a href="#">AnswerCallback</a>	10
<a href="#">AppSocket</a>	10
<a href="#">ArchivedGameInfo</a>	16
<a href="#">Arg</a>	17
<a href="#">ArgsUtil</a>	21
<a href="#">ArgType</a>	23
<a href="#">ArrUtils</a>	25
<a href="#">AuthSettings</a>	27
<a href="#">BasicMove</a>	28
<a href="#">BitData</a>	33
<a href="#">Board</a>	34
<a href="#">MyJFrame.BooleanClosing</a>	39
<a href="#">Callback&lt; T &gt;</a>	40
<a href="#">CastlingRights</a>	41
<a href="#">MyJFrame.Closing&lt; T &gt;</a>	46
<a href="#">Col</a>	48
<a href="#">MovesList.CompareMoves</a>	59
<a href="#">Condition</a>	59
<a href="#">Config&lt; V &gt;</a>	64
<a href="#">ConfirmDialogs</a>	68
<a href="#">CreatedGame</a>	71
<a href="#">CustomStatement</a>	72
<a href="#">DBRequest</a>	73
<a href="#">DBResponse</a>	77
<a href="#">Delete</a>	81
<a href="#">Direction</a>	82
<a href="#">EfficientGen&lt; K, V &gt;</a>	91
<a href="#">Environment</a>	93
<a href="#">EnvManager</a>	93
<a href="#">ErrorHandler&lt; E extends MyError &gt;</a>	94
<a href="#">EstablishedGameInfo</a>	95
<a href="#">Evaluation</a>	98
<a href="#">EvaluationParameters</a>	105
<a href="#">FontManager</a>	107

GameInfo	107
GameSettings	111
GameStatus	112
GameStatus.GameStatusType	120
GameTime	123
GameView	123
Graphable	124
GraphableDBResponse	124
GraphElement	125
GraphElementType	125
HandledThread	125
MyJFrame.MyAdapter.HeldDown	127
IDsGenerator	128
StrUtils.IterationThingy< T >	129
LazyHashSupplier< T >	130
LinkLabel	130
Location	131
LoginInfo	140
LoginType	145
Math	146
MathUtils	153
Message	154
MessageCallback	176
MessagesHandler	176
MessageType	185
MinimaxMove	186
Move	190
MoveAnnotation	202
Move.MoveFlag	203
MoveFlags	206
MovesList	206
MyError	207
MyJButton	209
MyJFrame	213
MyLbl	215
MyThread	216
Piece	217
PieceType	223
PlayerColor	231
PreMadeRequest	235
PremovesGenerator	237
Question	238
RegEx	243
SyncedItems< E extends SyncableItem >.Remover< L >	248
RequestBuilder	248
Selection	256
CastlingRights.Side	260
GameStatus.SpecificStatus	263
SQLStatement	266
Square	269
StatusResponse	272
MyJFrame.StringClosing	274
MyLbl.StringModifier	275
StrUtils	275
SwitchCase	288
SyncableItem	291
SyncedItems< E extends SyncableItem >	292
SyncedListType	292



Table	293
TableDBResponse	295
ThreadsManager	300
MyThread.ThreadStatus	301
ThrowingCallback< T >	302
ThrowingRunnable	303
TimeFormat	303
TimeFormatComponent	304
UnfinishedGame	304
Update	306
UserInfo	307
ValueBtn< T >	308
PreMadeRequest.VariationCreator	309
VoidCallback	310

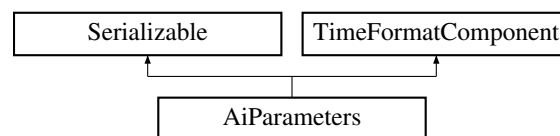


## Chapter 3

# Class Documentation

### 3.1 AiParameters

Inheritance diagram for AiParameters:



#### Classes

- enum [AiType](#)

#### Static Public Attributes

- static final [AiParameters](#) `EZ_MY_AI` = new [AiParameters](#)(`AiType.MyAi`, `TimeFormat.ULTRA_BULLET`)
- static final [AiParameters](#) `EZ_STOCKFISH` = new [AiParameters](#)(`AiType.Stockfish`, `TimeFormat.ULTRA_BULLET`)
- static final int `numOfFields` = 1 + `TimeFormat.numOfFields`

#### 3.1.1 Class Documentation

##### 3.1.1.1 enum ver14::SharedClasses::Game::GameSetup::AiParameters::AiType

Enumerator

Stockfish	
MyAi	

## 3.1.2 Member Function Documentation

### 3.1.2.1 setTimeFormat()

```
void setTimeFormat (
    TimeFormat timeFormat )
```

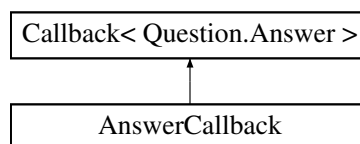
Implements [TimeFormatComponent](#).

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/GameSetup/AiParameters.java

## 3.2 AnswerCallback

Inheritance diagram for AnswerCallback:



### 3.2.1 Detailed Description

represents an answer callback.

Author

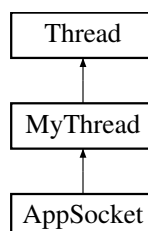
Bezalel Avrahami ( [bezalel3250@gmail.com](mailto:bezalel3250@gmail.com) )

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/Callbacks/AnswerCallback.java

## 3.3 AppSocket

Inheritance diagram for AppSocket:



## Classes

- class **AppSocketError**

## Protected Attributes

- final Socket [msgSocket](#)

### 3.3.1 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) )

### 3.3.2 Constructor & Destructor Documentation

#### 3.3.2.1 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
--------------------	------------------

#### 3.3.2.2 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
--------------------	------------------

### 3.3.3 Member Function Documentation

#### 3.3.3.1 close() [1/2]

```
void close ( )
```

Close.

#### 3.3.3.2 close() [2/2]

```
void close (
    MyError err )
```

Close.

**Parameters**

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

#### 3.3.3.3 interruptListener()

```
void interruptListener (
    MyError err )
```

Interrupt listener.

**Parameters**

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

#### 3.3.3.4 requestMessage() [1/2]

```
void requestMessage (
    Message requestMsg,
    MessageCallback onRes )
```

Request message.

##### Parameters

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

#### 3.3.3.5 handledRun()

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

Handled run.

Reimplemented from [MyThread](#).

#### 3.3.3.6 getMessagesHandler()

```
MessagesHandler getMessagesHandler ( )
```

Gets messages handler.

##### Returns

the messages handler

#### 3.3.3.7 setMessagesHandler()

```
void setMessagesHandler (
    MessagesHandler messagesHandler )
```

Sets messages handler.

##### Parameters

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

### 3.3.3.8 respond()

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

Respond.

#### Parameters

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

### 3.3.3.9 writeMessage()

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

Write message.

#### Parameters

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

### 3.3.3.10 isConnected()

```
boolean isConnected ( )
```

Is connected boolean.

#### Returns

the boolean

### 3.3.3.11 getLocalAddress()

```
String getLocalAddress ( )
```

Gets local address.

#### Returns

the local address



### 3.3.3.12 getRemoteAddress()

```
String getRemoteAddress ( )
```

Gets remote address.

#### Returns

the remote address

### 3.3.3.13 isClosed()

```
boolean isClosed ( )
```

Is closed boolean.

#### Returns

the boolean

### 3.3.3.14 requestMessage() [2/2]

```
Message requestMessage (
    Message requestMsg )
```

sending request and blocking til res

#### Parameters

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

#### Returns

response

### 3.3.3.15 stopReading()

```
void stopReading ( )
```

Stop reading.

### 3.3.4 Member Data Documentation

#### 3.3.4.1 msgSocket

```
final Socket msgSocket [protected]
```

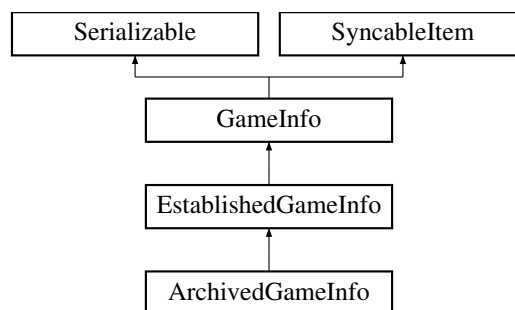
The Msg socket.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Networking/AppSocket.java

## 3.4 ArchivedGameInfo

Inheritance diagram for ArchivedGameInfo:



### Additional Inherited Members

#### 3.4.1 Detailed Description

Archived game info.

Author

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

#### 3.4.2 Constructor & Destructor Documentation

##### 3.4.2.1 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

### 3.4.3 Member Function Documentation

#### 3.4.3.1 getWinner()

```
String getWinner ( )
```

Gets winner.

## Returns

the winner

#### 3.4.3.2 toString()

```
String toString ( )
```

To string string.

## Returns

the string

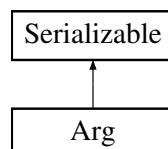
Reimplemented from [EstablishedGameInfo](#).

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/SavedGames/ArchivedGameInfo.java

## 3.5 Arg

Inheritance diagram for Arg:



## Public Attributes

- final String [replnStr](#)
- final boolean [escape](#)
- final [ArgType](#) [argType](#)
- final [Config](#)<?> [config](#)

### 3.5.1 Detailed Description

represents an argument that will later be replaced with a value. used for creating db requests.

#### Author

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

### 3.5.2 Constructor & Destructor Documentation

#### 3.5.2.1 [Arg\(\)](#) [1/3]

```
Arg (
    ArgType argType )
```

Instantiates a new [Arg](#).

#### Parameters

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

#### 3.5.2.2 [Arg\(\)](#) [2/3]

```
Arg (
    ArgType argType,
    Config<?> config )
```

Instantiates a new [Arg](#) with its configuration.

#### Parameters

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

### 3.5.2.3 Arg() [3/3]

```
Arg (
    ArgType argType,
    boolean escape,
    Config<?> config )
```

Instantiates a new [Arg](#).

#### Parameters

<i>argType</i>	the arg type
<i>escape</i>	should this argument's value be escaped
<i>config</i>	the config

## 3.5.3 Member Function Documentation

### 3.5.3.1 isUserInput()

```
boolean isUserInput ( )
```

Is user input boolean.

#### Returns

the boolean

### 3.5.3.2 setUserInput()

```
void setUserInput (
    boolean userInput )
```

Sets user input.

#### Parameters

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

### 3.5.3.3 equals()

```
boolean equals (
```

```
Object o )
```

Equals boolean.

#### Parameters

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

#### Returns

the boolean

### 3.5.3.4 toString()

```
String toString ( )
```

To string string.

#### Returns

the string

### 3.5.3.5 createVal()

```
String createVal (
    Object val )
```

Creates a string value to replace this argument from a provided value.

#### Parameters

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

#### Returns

the string

## 3.5.4 Member Data Documentation

### 3.5.4.1 replnStr

```
final String replnStr
```

The Representation in the string. when the argument is provided, it will replace the temporary replnStr

### 3.5.4.2 escape

```
final boolean escape
```

The Escape.

### 3.5.4.3 argType

```
final ArgType argType
```

The Arg type.

### 3.5.4.4 config

```
final Config<?> config
```

The Configuration of this argument .

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/Arg/Arg.java

## 3.6 ArgsUtil

### 3.6.1 Detailed Description

The type Args utils.

Author

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

### 3.6.2 Member Function Documentation

#### 3.6.2.1 create()

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

Create args util.

**Parameters**

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

**Returns**

the args util

**3.6.2.2 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

**3.6.2.3 plainTextIgnoreCase()**

```
OptionalArg plainTextIgnoreCase (
    String str )
```

Plain text ignore case optional arg.

**Parameters**

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

**Returns**

the optional arg

**3.6.2.4 OptionalArg()**

```
record OptionalArg (
    String str )
```

Optional arg.



## Author

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

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Utils/ArgsUtil.java

## 3.7 ArgType Enum Reference

### Public Attributes

- [Date](#)
- [DateRange](#)
- [Text](#)
- [ServerAddress](#)
- [Number](#)
- [Username](#) =(false)
- [Password](#)
- [Url](#)
- [PictureUrl](#)
- final boolean [isUserInput](#)

### 3.7.1 Detailed Description

[Arg](#) type - argument type.

## Author

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

### 3.7.2 Constructor & Destructor Documentation

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

[ArgType](#) ( )

Instantiates a new [Arg](#) type.

#### 3.7.2.2 ArgType() [2/2]

[ArgType](#) (  
    boolean *isUserInput* )

Instantiates a new [Arg](#) type.

**Parameters**

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

### 3.7.3 Member Data Documentation

#### 3.7.3.1 Date

`Date`

Date arg type.

#### 3.7.3.2 DateRange

`DateRange`

Date range arg type.

#### 3.7.3.3 Text

`Text`

Text arg type.

#### 3.7.3.4 ServerAddress

`ServerAddress`

Server address arg type.

#### 3.7.3.5 Number

`Number`

Number arg type.

#### 3.7.3.6 Username

`Username = (false)`

Username arg type.

### 3.7.3.7 Password

Password

Password arg type.

### 3.7.3.8 Url

Url

Url arg type.

### 3.7.3.9 PictureUrl

PictureUrl

Picture url arg type.

### 3.7.3.10 isUserInput

```
final boolean isUserInput
```

The Is user input.

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/DBActions/Arg/ArgType.java

## 3.8 ArrUtils

### 3.8.1 Detailed Description

Array Utility Class.

Author

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

### 3.8.2 Member Function Documentation

#### 3.8.2.1 concat()

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

[credit](#)

**Parameters**

<i>&lt; T &gt;</i>	the type parameter
<i>array1</i>	the array 1
<i>array2</i>	the array 2

**Returns**

`t []`

**3.8.2.2 createList()**

```
static< T > ArrayList< T > createList (
    Supplier< T > objCreator,
    int size ) [static]
```

Create list array list.

**Parameters**

<i>&lt; T &gt;</i>	the type parameter
<i>objCreator</i>	the obj creator
<i>size</i>	the size

**Returns**

the array list

**3.8.2.3 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

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Utils/ArrUtils.java

## 3.9 AuthSettings

### Public Attributes

- int `GUEST` = 1
- int `USER` = 2
- int `ANY_LOGIN` = `GUEST` | `USER`
- int `NEVER_AUTH` = 4
- int `NO_AUTH` = 5

### 3.9.1 Detailed Description

Auth settings. represents all types of login authentication.

#### Author

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

### 3.9.2 Member Data Documentation

#### 3.9.2.1 GUEST

```
int GUEST = 1
```

logged in as GUEST.

#### 3.9.2.2 USER

```
int USER = 2
```

logged in as USER.

#### 3.9.2.3 ANY\_LOGIN

```
int ANY_LOGIN = GUEST | USER
```

logged in as guest / user.

### 3.9.2.4 NEVER\_AUTH

```
int NEVER_AUTH = 4
```

will never authorize.

### 3.9.2.5 NO\_AUTH

```
int NO_AUTH = 5
```

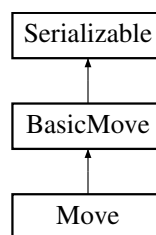
no authentication is required (not used at the moment, because nothing is accessible without some sort of login).

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/Login/AuthSettings.java

## 3.10 BasicMove

Inheritance diagram for BasicMove:



### 3.10.1 Detailed Description

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

Author

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

### 3.10.2 Constructor & Destructor Documentation

#### 3.10.2.1 BasicMove() [1/3]

```
BasicMove (
    BasicMove other )
```

Copy constructor.

## Parameters

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

**3.10.2.2 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

**3.10.2.3 BasicMove()** [3/3]

```
BasicMove (
    String move )
```

Instantiates a new Basic move.

## Parameters

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

**3.10.3 Member Function Documentation****3.10.3.1 createBatch()**

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

Create batch basic move [ ].

## Parameters

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

**Returns**

the basic move [ ]

**3.10.3.2 getFlipped() [1/2]**

```
BasicMove getFlipped ( )
```

Gets flipped.

**Returns**

the flipped

**3.10.3.3 getFlipped() [2/2]**

```
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

**3.10.3.4 flip()**

```
void flip ( )
```

Flips the source and destination.

**3.10.3.5 getMovingFrom()**

```
Location getMovingFrom ( )
```

Gets moving from. aka source

**Returns**

the moving from



### 3.10.3.6 setMovingFrom()

```
void setMovingFrom (
    Location movingFrom )
```

Sets moving from.

#### Parameters

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

### 3.10.3.7 getMovingTo()

```
Location getMovingTo ( )
```

Gets moving to.

#### Returns

the moving to

### 3.10.3.8 setMovingTo()

```
void setMovingTo (
    Location movingTo )
```

Sets moving to.

#### Parameters

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

### 3.10.3.9 hashCode()

```
int hashCode ( )
```

Hash code int.

#### Returns

the int

### 3.10.3.10 equals()

```
boolean equals (
    Object o )
```

Equals boolean.

#### Parameters

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

#### Returns

the boolean

Reimplemented in [Move](#).

### 3.10.3.11 toString()

```
String toString ( )
```

To string string.

#### Returns

the string

Reimplemented in [Move](#).

### 3.10.3.12 getBasicMoveAnnotation()

```
String getBasicMoveAnnotation ( )
```

Gets basic move annotation.

#### Returns

the basic move annotation

### 3.10.3.13 cp()

```
BasicMove cp ( )
```

copies this move.

#### Returns

the new copy

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/Moves/BasicMove.java

## 3.11 BitData

### Static Public Attributes

- static final long `notAFile` = 0xfefefefefefefefeL
- static final long `notHFile` = 0x7f7f7f7f7f7f7fL
- static final long `everything` = 0xffffffffffffL

### 3.11.1 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))

### 3.11.2 Member Data Documentation

#### 3.11.2.1 notAFile

```
final long notAFile = 0xfefefefefefefefeL [static]
```

The constant notAFile. the whole board but the A file

#### 3.11.2.2 notHFile

```
final long notHFile = 0x7f7f7f7f7f7f7fL [static]
```

The constant notHFile. the whole board but the H file

### 3.11.2.3 everything

```
final long everything = 0xffffffffffffffffL [static]
```

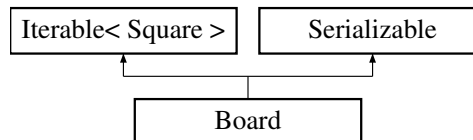
The constant everything. the whole board

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/Moves/BitData.java

## 3.12 Board

Inheritance diagram for Board:



### Static Public Attributes

- static final String [startingFen](#) = "rnbqkbnr/pppppppp/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1"
- static final [Board example](#)

### 3.12.1 Detailed Description

[Board](#) - represents the logic board of the model

Author

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

### 3.12.2 Constructor & Destructor Documentation

#### 3.12.2.1 Board() [1/3]

```
Board (
    Board other )
```

Instantiates a new [Board](#).

## Parameters

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

**3.12.2.2 Board()** [2/3]

```
Board ( )
```

Instantiates a new [Board](#).

**3.12.2.3 Board()** [3/3]

```
Board (
    String fen )
```

Instantiates a new [Board](#).

## Parameters

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

**3.12.3 Member Function Documentation****3.12.3.1 setPiece()**

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

Sets piece.

## Parameters

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

**3.12.3.2 getSquare()**

```
Square getSquare (
    Location loc )
```

Gets square.

#### Parameters

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

#### Returns

the square

### 3.12.3.3 fenSetup()

```
void fenSetup (
    String fen )
```

Fen setup.

#### Parameters

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

### 3.12.3.4 startingPos()

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

Starting pos board.

#### Returns

the board

### 3.12.3.5 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 [ ]

**3.12.3.6 getPiece() [1/2]**

```
Piece getPiece (
    Location loc,
    boolean notNull )
```

Gets piece.

**Parameters**

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

**Returns**

the piece

**3.12.3.7 getPiece() [2/2]**

```
Piece getPiece (
    Location loc )
```

Gets piece.

**Parameters**

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

**Returns**

the piece

**3.12.3.8 print()**

```
void print ( )
```

Print.

### 3.12.3.9 iterator()

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

Iterator iterator.

#### Returns

the iterator

### 3.12.3.10 setSquareEmpty()

```
void setSquareEmpty (
    Location loc )
```

Sets square empty.

#### Parameters

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

### 3.12.3.11 isSquareEmpty()

```
boolean isSquareEmpty (
    Location loc )
```

Is square empty boolean.

#### Parameters

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

#### Returns

the boolean

### 3.12.3.12 toString()

```
String toString ( )
```

To string string.

#### Returns

the string



### 3.12.4 Member Data Documentation

#### 3.12.4.1 startingFen

```
final String startingFen = "rnbqkbnr/pppppppp/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1" [static]
```

The constant startingFen.

#### 3.12.4.2 example

```
final Board example [static]
```

**Initial value:**

```
= new Board() {{  
    fenSetup("rnbqkbnr/pppppppp/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1");  
}}
```

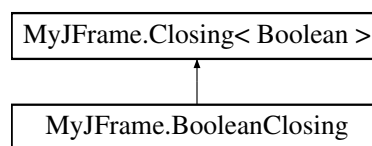
The constant example.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/GameSetup/BoardSetup/Board.java

## 3.13 MyJFrame.BooleanClosing

Inheritance diagram for MyJFrame.BooleanClosing:



### Additional Inherited Members

#### 3.13.1 Detailed Description

The interface Boolean closing.

#### 3.13.2 Member Function Documentation

### 3.13.2.1 show()

```
default Boolean show ( )
```

Show boolean.

#### Returns

the boolean

### 3.13.2.2 checkClosingVal()

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

Check closing val boolean.

#### Parameters

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

#### Returns

the boolean

### 3.13.2.3 closing()

```
void closing ( )
```

[Closing](#).

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/UI/MyJFrame.java

## 3.14 Callback< T >

### 3.14.1 Detailed Description

[Callback](#) - represents an asynchronous callback with an object. some actions to execute at an unknown point in the future. used for things like button clicks.

## Parameters

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

## Author

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

## 3.14.2 Member Function Documentation

### 3.14.2.1 callback()

```
void callback (  
    T obj )
```

[Callback.](#)

## Parameters

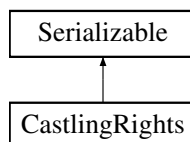
<code>obj</code>	the obj
------------------	---------

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/Callbacks/Callback.java

## 3.15 CastlingRights

Inheritance diagram for CastlingRights:



### Classes

- enum [Side](#)

### Static Public Attributes

- static final String [NO\\_CASTLING\\_ABILITY](#) = "-"

### 3.15.1 Detailed Description

Castling rights - .

Author

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

### 3.15.2 Constructor & Destructor Documentation

#### 3.15.2.1 CastlingRights() [1/3]

```
CastlingRights (
    byte rights )
```

Instantiates a new Castling rights.

Parameters

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

#### 3.15.2.2 CastlingRights() [2/3]

```
CastlingRights ( )
```

Instantiates a new Castling rights.

#### 3.15.2.3 CastlingRights() [3/3]

```
CastlingRights (
    CastlingRights other )
```

Instantiates a new Castling rights.

Parameters

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

### 3.15.3 Member Function Documentation

### 3.15.3.1 createFromStr()

```
static CastlingRights createFromStr (
    String castlingAbilityStr ) [static]
```

Create from str castling rights.

#### Parameters

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

#### Returns

the castling rights

### 3.15.3.2 enableCastling()

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

Enable castling.

#### Parameters

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

### 3.15.3.3 main()

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

The entry point of application.

#### Parameters

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

### 3.15.3.4 whosCastling()

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

Whos castling player color.

#### Parameters

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

#### Returns

the player color

### 3.15.3.5 toString()

```
String toString ( )
```

To string string.

#### Returns

the string

### 3.15.3.6 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

### 3.15.3.7 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

**3.15.3.8 getRights()**

```
byte getRights ( )
```

Gets rights.

## Returns

the rights

**3.15.3.9 enable()**

```
void enable (
    byte b )
```

Enable.

## Parameters

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

**3.15.3.10 hasAny()**

```
boolean hasAny (
    PlayerColor playerColor )
```

Has any boolean.

## Parameters

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

**Returns**

the boolean

**3.15.3.11 getPlayersCastling()**

```
byte getPlayersCastling (
    PlayerColor playerColor )
```

Gets players castling.

**Parameters**

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

**Returns**

the players castling

**3.15.4 Member Data Documentation****3.15.4.1 NO\_CASTLING\_ABILITY**

```
final String NO_CASTLING_ABILITY = "-" [static]
```

The constant NO\_CASTLING\_ABILITY.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/Moves/CastlingRights.java

**3.16 MyJFrame.Closing< T >****Public Attributes**

- String [title](#) = "Exit Confirmation"
- String [header](#) = "Are You Sure You Want To Exit?"
- ImageIcon [icon](#) = null

**3.16.1 Detailed Description**

The interface [Closing](#).



## Parameters

< T >	the type parameter
-------	--------------------

## 3.16.2 Member Function Documentation

### 3.16.2.1 tryClose()

```
default void tryClose ( )
```

Try close.

### 3.16.2.2 show()

```
T show ( )
```

Show t.

## Returns

the t

### 3.16.2.3 checkClosingVal()

```
boolean checkClosingVal (
    T val )
```

Check closing val boolean.

## Parameters

val	the val
-----	---------

## Returns

the boolean

### 3.16.2.4 closing()

```
void closing (
    T val )
```

[Closing.](#)

#### Parameters

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

### 3.16.3 Member Data Documentation

#### 3.16.3.1 title

```
String title = "Exit Confirmation"
```

The constant title.

#### 3.16.3.2 header

```
String header = "Are You Sure You Want To Exit?"
```

The constant header.

#### 3.16.3.3 icon

```
ImageIcon icon = null
```

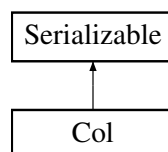
The constant icon.

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/UI/MyJFrame.java

## 3.17 Col

Inheritance diagram for Col:



### Classes

- class **CustomCol**

## Static Public Attributes

- static final [Col](#) [GameID](#) = new [Col](#)("GameID")
- static final [Col](#) [SavedGame](#) = new [Col](#)("SavedGame")
- static final [Col](#) [Password](#) = new [Col](#)("password")
- static final [Col](#) [Player1](#) = new [Col](#)("Player1")
- static final [Col](#) [Player2](#) = new [Col](#)("Player2")
- static final [Col](#) [Username](#) = new [Col](#)("username")
- static final [Col](#) [Winner](#) = new [Col](#)("Winner")
- static final [Col](#) [PlayerToMove](#) = new [Col](#)("PlayerToMove")
- static final [Col](#) [CreatedDateTime](#) = new [Col](#)("CreatedDateTime")
- static final [Col](#) [ProfilePic](#) = new [Col](#)("ProfilePic")

### 3.17.1 Detailed Description

[Col](#) - represents a column. either existing column in the db (the constant columns [GameID](#), [SavedGame](#) ...) or created columns.

#### Author

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

### 3.17.2 Constructor & Destructor Documentation

#### 3.17.2.1 [Col\(\)](#) [1/3]

```
Col (
    Col col )
```

Instantiates a new [Col](#).

#### Parameters

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

#### 3.17.2.2 [Col\(\)](#) [2/3]

```
Col (
    String colName,
    String alias )
```

Instantiates a new [Col](#).

**Parameters**

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

**3.17.2.3 Col()** [3/3]

```
Col (  
    String colName )
```

Instantiates a new Col.

**Parameters**

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

**3.17.3 Member Function Documentation****3.17.3.1 count()** [1/2]

```
static Col count (  
    String as ) [static]
```

Count col. counts every row (\*)

**Parameters**

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

**Returns**

the col

**3.17.3.2 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

**3.17.3.3 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

**3.17.3.4 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

### 3.17.3.5 label()

```
String label ( )
```

Label string.

#### Returns

the string

### 3.17.3.6 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

### 3.17.3.7 wrap()

```
void wrap ( )
```

Wrap.

### 3.17.3.8 time()

```
Col time ( )
```

Time col.

#### Returns

a new col representing time

### 3.17.3.9 date()

```
Col date ( )
```

Date col.

#### Returns

a new col representing datetime

### 3.17.3.10 setWrapped()

```
void setWrapped (
    boolean wrapped )
```

Sets wrapped.

#### Parameters

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

### 3.17.3.11 as() [1/2]

```
Col as ( )
```

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

#### Returns

the col

### 3.17.3.12 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

**3.17.3.13 setColName()**

```
void setColName (
    String colName )
```

Sets col name.

**Parameters**

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

**3.17.3.14 colName()**

```
String colName ( )
```

Col name string.

**Returns**

the string

**3.17.3.15 of() [1/2]**

```
Col of (
    Table table )
```

new col like this that belongs to the given table

**Parameters**

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

**Returns**

the col



**3.17.3.16 of()** [2/2]

```
Col of (
    String ofWhom )
```

new col like this that belongs to the given owner

**Parameters**

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

**Returns**

the col

**3.17.3.17 replace()**

```
Col replace (
    String replacing,
    String replaceWith )
```

Replace col.

**Parameters**

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

**Returns**

the col

**3.17.3.18 nested()**

```
String nested ( )
```

Nested string.

**Returns**

the string

**3.17.3.19 math() [1/2]**

```
Col math (
    Math operation,
    Object value )
```

**Math** col.

## Parameters

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

## Returns

the col

**3.17.3.20 math() [2/2]**

```
Col math (
    Math operation,
    Object value,
    boolean changeSelf )
```

Math col.

## Parameters

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

## Returns

the col

**3.17.3.21 toString()**

```
String toString ( )
```

To string string.

## Returns

the string

**3.17.4 Member Data Documentation**

#### 3.17.4.1 GameID

```
final Col GameID = new Col("GameID") [static]
```

The constant GameID.

#### 3.17.4.2 SavedGame

```
final Col SavedGame = new Col("SavedGame") [static]
```

The constant SavedGame.

#### 3.17.4.3 Password

```
final Col Password = new Col("password") [static]
```

The constant Password.

#### 3.17.4.4 Player1

```
final Col Player1 = new Col("Player1") [static]
```

The constant Player1.

#### 3.17.4.5 Player2

```
final Col Player2 = new Col("Player2") [static]
```

The constant Player2.

#### 3.17.4.6 Username

```
final Col Username = new Col("username") [static]
```

The constant Username.

#### 3.17.4.7 Winner

```
final Col Winner = new Col("Winner") [static]
```

The constant Winner.

#### 3.17.4.8 PlayerToMove

```
final Col PlayerToMove = new Col("PlayerToMove") [static]
```

The constant PlayerToMove.

#### 3.17.4.9 CreatedDateTime

```
final Col CreatedDateTime = new Col("CreatedDateTime") [static]
```

The constant CreatedDateTime.

#### 3.17.4.10 ProfilePic

```
final Col ProfilePic = new Col("ProfilePic") [static]
```

The constant ProfilePic.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/Table/Col.java

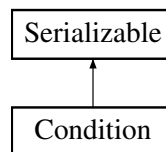
## 3.18 MovesList.CompareMoves

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/Game/Moves/MovesList.java

## 3.19 Condition

Inheritance diagram for Condition:



### Classes

- enum [Relation](#)

#### 3.19.1 Detailed Description

[Condition](#) - represents a condition.

Author

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

#### 3.19.2 Class Documentation

##### 3.19.2.1 enum ver14::SharedClasses::DBActions::Condition::Relation

[Relation](#) - relations between conditions.

Author

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

#### Enumerator

AND	And relation.
OR	Or relation.

### 3.19.3 Constructor & Destructor Documentation

#### 3.19.3.1 Condition()

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

Instantiates a new [Condition](#).

#### Parameters

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

### 3.19.4 Member Function Documentation

#### 3.19.4.1 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

### 3.19.4.2 noNulls()

```
Condition noNulls ( )
```

No nulls condition.

#### Returns

the condition

### 3.19.4.3 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

### 3.19.4.4 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

#### 3.19.4.5 wrap()

```
void wrap ( )
```

Wrap.

#### 3.19.4.6 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

#### 3.19.4.7 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



#### 3.19.4.8 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

#### 3.19.4.9 and()

```
Condition and (
    Condition condition )
```

wraps

##### Parameters

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

##### Returns

condition

#### 3.19.4.10 getStr()

```
String getStr ( )
```

Gets str.

##### Returns

the str

#### 3.19.4.11 setStr()

```
void setStr (
    String str )
```

Sets str.

**Parameters**

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

**3.19.4.12 toString()**

```
String toString ( )
```

To string string.

**Returns**

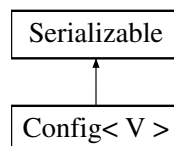
the string

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/Condition.java

**3.20 Config< V >**

Inheritance diagram for Config< V >:

**Public Attributes**

- final boolean [canUseDefault](#)
- final String [description](#)

**3.20.1 Detailed Description**

[Config](#) - an argument's configuration.

**Parameters**

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

## Author

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

## 3.20.2 Constructor & Destructor Documentation

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

```
Config ( )
```

Instantiates a new [Config](#).

### 3.20.2.2 Config() [2/6]

```
Config (
    String description )
```

Instantiates a new [Config](#).

## Parameters

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

### 3.20.2.3 Config() [3/6]

```
Config (
    String description,
    boolean canUseDefault,
    Described< V > defaultValue )
```

Instantiates a new [Config](#).

## Parameters

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

### 3.20.2.4 Config() [4/6]

```
Config (
```

```
String description,
V defVal )
```

Instantiates a new [Config](#).

#### Parameters

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

### 3.20.2.5 Config() [5/6]

```
Config (
    String description,
    V defVal,
    String defDesc )
```

Instantiates a new [Config](#).

#### Parameters

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

### 3.20.2.6 Config() [6/6]

```
Config (
    String description,
    Described< V > defaultValue )
```

Instantiates a new [Config](#).

#### Parameters

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

## 3.20.3 Member Function Documentation

### 3.20.3.1 addSuggestion()

```
void addSuggestion (
    Described< V > suggestion )
```

Add suggestion.

#### Parameters

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

### 3.20.3.2 getValuesSuggestion()

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

Gets values suggestion.

#### Returns

the values suggestion

### 3.20.3.3 getDescribedDefault()

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

Gets described default.

#### Returns

the described default

### 3.20.3.4 getDefault()

```
V getDefault ( )
```

Gets default.

#### Returns

the default

### 3.20.3.5 toString()

```
String toString ( )
```

To string string.

#### Returns

the string

### 3.20.3.6 getDefaultDesc()

```
String getDefaultDesc ( )
```

Gets default desc.

#### Returns

the default desc

## 3.20.4 Member Data Documentation

### 3.20.4.1 canUseDefault

```
final boolean canUseDefault
```

The Can use default.

### 3.20.4.2 description

```
final String description
```

The Description.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/Arg/Config.java

## 3.21 ConfirmDialogs

### 3.21.1 Detailed Description

The utility class Confirm dialogs.

## 3.21.2 Member Function Documentation

### 3.21.2.1 confirm() [1/2]

```
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

### 3.21.2.2 main()

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

The entry point of application.

#### Parameters

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

### 3.21.2.3 confirm() [2/2]

```
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

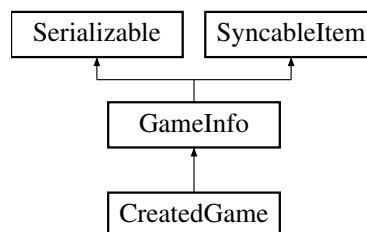
the string

The documentation for this class was generated from the following file:

- ver14/SharedClasses/UI/dialogs/ConfirmDialogs.java

## 3.22 CreatedGame

Inheritance diagram for CreatedGame:



### Additional Inherited Members

#### 3.22.1 Detailed Description

Created game.

## Author

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

#### 3.22.2 Constructor & Destructor Documentation

##### 3.22.2.1 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

### 3.22.3 Member Function Documentation

#### 3.22.3.1 `getGameDesc()`

```
String getGameDesc ( )
```

Gets game desc.

**Returns**

the game desc

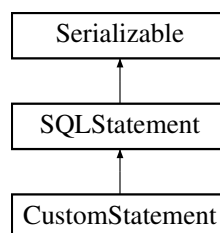
Reimplemented from [GameInfo](#).

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/SavedGames/CreatedGame.java

## 3.23 CustomStatement

Inheritance diagram for CustomStatement:



### Additional Inherited Members

#### 3.23.1 Detailed Description

Custom statement - a custom sql statement.

**Author**

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

### 3.23.2 Constructor & Destructor Documentation

#### 3.23.2.1 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

### 3.23.3 Member Function Documentation

#### 3.23.3.1 createStatement()

```
String createStatement ( ) [protected]
```

Create statement string.

##### Returns

the string

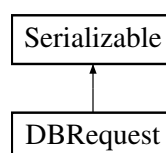
Reimplemented from [SQLStatement](#).

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/Statements/CustomStatement.java

## 3.24 DBRequest

Inheritance diagram for DBRequest:



## Classes

- enum [Type](#)

## Public Attributes

- final [Type](#) `type`

### 3.24.1 Detailed Description

Db request - a database request.

#### Author

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

### 3.24.2 Class Documentation

#### 3.24.2.1 enum `ver14::SharedClasses::DBActions::DBRequest::DBRequest::Type`

[Type](#) - .

#### Author

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

#### Enumerator

Query	Query type.
Update	Update type.

### 3.24.3 Constructor & Destructor Documentation

#### 3.24.3.1 `DBRequest()` [1/3]

```
DBRequest (  
    SQLStatement sqlStatement )
```

Instantiates a new Db request.

#### Parameters

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

### 3.24.3.2 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

### 3.24.3.3 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

## 3.24.4 Member Function Documentation

### 3.24.4.1 getBuilder()

```
RequestBuilder getBuilder ( )
```

Gets builder.

#### Returns

the builder

#### 3.24.4.2 `getSubRequest()`

```
DBRequest getSubRequest ( )
```

Gets sub request.

##### Returns

the sub request

#### 3.24.4.3 `setSubRequest()`

```
void setSubRequest (
    DBRequest subRequest )
```

Sets sub request.

##### Parameters

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

#### 3.24.4.4 `toString()`

```
String toString ( )
```

To string string.

##### Returns

the string

#### 3.24.4.5 `getRequest()`

```
String getRequest ( )
```

Gets request.

##### Returns

the request

### 3.24.5 Member Data Documentation

#### 3.24.5.1 type

final [Type](#) type

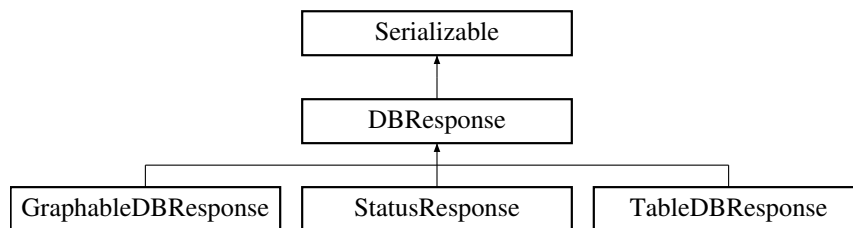
The [Type](#).

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/DBRequest/DBRequest.java

## 3.25 DBResponse

Inheritance diagram for DBResponse:



### Classes

- enum [Status](#)

### Protected Attributes

- final [Status](#) status
- final [DBRequest](#) request
- [DBResponse](#) addedRes = null

### 3.25.1 Detailed Description

Db response - .

Author

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

### 3.25.2 Class Documentation

#### 3.25.2.1 enum ver14::SharedClasses::DBActions::DBResponse::DBResponse::Status

[Status](#) - .

Author

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

**Enumerator**

SUCCESS	Success status.
ERROR	Error status.

### 3.25.3 Constructor & Destructor Documentation

#### 3.25.3.1 DBResponse()

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

Instantiates a new Db response.

**Parameters**

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

### 3.25.4 Member Function Documentation

#### 3.25.4.1 getRequest()

```
DBRequest getRequest ( )
```

Gets request.

**Returns**

the request

#### 3.25.4.2 getStatus()

```
Status getStatus ( )
```

Gets status.

**Returns**

the status



#### 3.25.4.3 isSuccess()

```
boolean isSuccess ( )
```

Is success boolean.

##### Returns

the boolean

#### 3.25.4.4 isAnyData()

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

Is any data boolean.

##### Returns

the boolean

Reimplemented in [StatusResponse](#), and [TableDBResponse](#).

#### 3.25.4.5 getAddedRes()

```
DBResponse getAddedRes ( )
```

Gets added res.

##### Returns

the added res

#### 3.25.4.6 setAddedRes()

```
void setAddedRes (
    DBResponse addedRes )
```

Sets added res.

##### Parameters

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

#### 3.25.4.7 clean()

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

Clean db response.

##### Returns

the db response

Reimplemented in [StatusResponse](#), and [TableDBResponse](#).

#### 3.25.4.8 print()

```
void print ( )
```

Print.

#### 3.25.4.9 toString()

```
String toString ( )
```

To string string.

##### Returns

the string

Reimplemented in [GraphableDBResponse](#), and [TableDBResponse](#).

### 3.25.5 Member Data Documentation

#### 3.25.5.1 status

```
final Status status [protected]
```

The [Status](#).

### 3.25.5.2 request

```
final DBRequest request [protected]
```

The Request.

### 3.25.5.3 addedRes

```
DBResponse addedRes = null [protected]
```

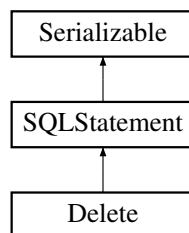
The Added res.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/DBResponse/DBResponse.java

## 3.26 Delete

Inheritance diagram for Delete:



### Additional Inherited Members

#### 3.26.1 Detailed Description

**Delete** - deletion statement.

Author

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

#### 3.26.2 Constructor & Destructor Documentation

##### 3.26.2.1 Delete()

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

Instantiates a new **Delete**.

**Parameters**

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

### 3.26.3 Member Function Documentation

#### 3.26.3.1 createStatement()

```
String createStatement ( ) [protected]
```

Create statement string.

**Returns**

the string

Reimplemented from [SQLStatement](#).

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/Statements/Delete.java

## 3.27 Direction Enum Reference

**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` = `values()`
- static final `List< Direction > ALL_USED_DIRECTIONS` = `Arrays.stream(values()).filter(d -> d != U_U && d != D_D).collect(Collectors.toList())`
- static final `PlayerColor normalPerspective` = `PlayerColor.WHITE`

### 3.27.1 Detailed Description

`Direction` - represents a moving direction on a board. sort of like a vector. has a `offset` 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](mailto:bezalel3250@gmail.com) )

### 3.27.2 Member Function Documentation

#### 3.27.2.1 `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

### 3.27.2.2 `getDirectionByOffset()`

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

Gets direction by offset.

#### Parameters

<i>offset</i>	the offset
---------------	------------

#### Returns

the direction by offset

### 3.27.2.3 `getCombination()`

```
Direction[] getCombination ( )
```

Get combination direction [ ].

#### Returns

the direction [ ]

### 3.27.2.4 `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](#)) is the exact opposite of a black pawn push ([D](#)). so the perspective needs to be in relation to the moving color.

#### Parameters

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

#### Returns

the direction

### 3.27.2.5 opposite()

```
abstract Direction opposite ( ) [abstract]
```

the Opposite direction to this one.

#### Returns

the direction

## 3.27.3 Member Data Documentation

### 3.27.3.1 U

U

#### Initial value:

```
=(8) {  
    @Override  
    public Direction opposite() {  
        return D;  
    }  
}
```

one square up the board.

### 3.27.3.2 D

D

#### Initial value:

```
=(-8) {  
    @Override  
    public Direction opposite() {  
        return U;  
    }  
}
```

one square down the board.

### 3.27.3.3 L

L

#### Initial value:

```
=(BitData.notAFile, -1) {  
    @Override  
    public Direction opposite() {  
        return R;  
    }  
}
```

one square left.

### 3.27.3.4 R

R

**Initial value:**

```
=(BitData.notHFile, 1) {  
    @Override  
    public Direction opposite() {  
        return L;  
    }  
}
```

one square right.

### 3.27.3.5 U\_U

U\_U

**Initial value:**

```
=(U, U) {  
    @Override  
    public Direction opposite() {  
        return D_D;  
    }  
}
```

two squares up.

### 3.27.3.6 D\_D

D\_D

**Initial value:**

```
=(D, D) {  
    @Override  
    public Direction opposite() {  
        return U_U;  
    }  
}
```

two squares down.

### 3.27.3.7 U\_R

U\_R

**Initial value:**

```
=(U, R) {  
    @Override  
    public Direction opposite() {  
        return D_L;  
    }  
}
```

one square up and one square right.



### 3.27.3.8 U\_L

U\_L

**Initial value:**

```
=(U, L) {  
    @Override  
    public Direction opposite() {  
        return D_R;  
    }  
}
```

one square up and one square left.

### 3.27.3.9 D\_R

D\_R

**Initial value:**

```
=(D, R) {  
    @Override  
    public Direction opposite() {  
        return U_L;  
    }  
}
```

one square down and one square right.

### 3.27.3.10 D\_L

D\_L

**Initial value:**

```
=(D, L) {  
    @Override  
    public Direction opposite() {  
        return U_R;  
    }  
}
```

one square down and one square left.

### 3.27.3.11 U\_U\_R

U\_U\_R

**Initial value:**

```
=(U, U, R) {  
    @Override  
    public Direction opposite() {  
        return D_D_R;  
    }  
}
```

two squares up and one square right.

### 3.27.3.12 U\_U\_L

U\_U\_L

#### Initial value:

```
=(U, U, L) {  
    @Override  
    public Direction opposite() {  
        return D_D_L;  
    }  
}
```

two squares up and one square left.

### 3.27.3.13 U\_R\_R

U\_R\_R

#### Initial value:

```
=(U, R, R) {  
    @Override  
    public Direction opposite() {  
        return D_L_L;  
    }  
}
```

two squares right and one square up.

### 3.27.3.14 U\_L\_L

U\_L\_L

#### Initial value:

```
=(U, L, L) {  
    @Override  
    public Direction opposite() {  
        return D_R_R;  
    }  
}
```

two squares left and one square up.

### 3.27.3.15 D\_D\_R

D\_D\_R

#### Initial value:

```
=(D, D, R) {  
    @Override  
    public Direction opposite() {  
        return U_U_L;  
    }  
}
```

two squares down and one square right.

### 3.27.3.16 D\_D\_L

D\_D\_L

**Initial value:**

```
=(D, D, L) {  
    @Override  
    public Direction opposite() {  
        return U_U_R;  
    }  
}
```

two squares down and one square left.

### 3.27.3.17 D\_R\_R

D\_R\_R

**Initial value:**

```
=(D, R, R) {  
    @Override  
    public Direction opposite() {  
        return U_L_L;  
    }  
}
```

two squares right and one square down.

### 3.27.3.18 D\_L\_L

D\_L\_L

**Initial value:**

```
=(D, L, L) {  
    @Override  
    public Direction opposite() {  
        return U_R_R;  
    }  
}
```

two squares left and one square down.

### 3.27.3.19 NUM\_OF\_DIRECTIONS

```
final int NUM_OF_DIRECTIONS [static]
```

The constant NUM\_OF\_DIRECTIONS.

### 3.27.3.20 NUM\_OF\_KNIGHT\_DIRECTIONS

```
final int NUM_OF_KNIGHT_DIRECTIONS [static]
```

The constant NUM\_OF\_KNIGHT\_DIRECTIONS.

### 3.27.3.21 NUM\_OF\_DIRECTIONS\_WO\_KNIGHT

```
final int NUM_OF_DIRECTIONS_WO_KNIGHT [static]
```

The constant NUM\_OF\_DIRECTIONS\_WO\_KNIGHT.

### 3.27.3.22 ALL\_DIRECTIONS

```
final Direction [] ALL_DIRECTIONS = values() [static]
```

The All directions.

### 3.27.3.23 ALL\_USED\_DIRECTIONS

```
final List<Direction> ALL_USED_DIRECTIONS = Arrays.stream(values()).filter(d -> d != U_U && d != D_D).collect(Collectors.toList()) [static]
```

The All used directions.

### 3.27.3.24 normalPerspective

```
final PlayerColor normalPerspective = PlayerColor.WHITE [static]
```

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

### 3.27.3.25 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

### 3.27.3.26 offset

```
final int offset
```

The actual offset.

### 3.27.3.27 asInt

```
final int asInt
```

an int for quick access by index.

### 3.27.3.28 combination

```
final Direction [] combination
```

some directions are combinations of other directions. the knight movement directions for example, are some combination of up,down,left and right.

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Game/Moves/Direction.java

## 3.28 EfficientGen< K, V >

### 3.28.1 Detailed Description

Efficient gen - represents an object that will only generate a new value when a key changes its value. used to save performance on calculating data that hasn't changed

#### Author

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

### 3.28.2 Constructor & Destructor Documentation

#### 3.28.2.1 EfficientGen()

```
EfficientGen (
    Supplier< V > generator )
```

Instantiates a new Efficient gen.

#### Parameters

<i>generator</i>	the generator
------------------	---------------

### 3.28.3 Member Function Documentation

#### 3.28.3.1 get()

```
V get (
    K currentKey )
```

Get v.

## Parameters

<i>currentKey</i>	the current key
-------------------	-----------------

## Returns

the v

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Misc/EfficientGen.java

## 3.29 Environment

### Static Public Attributes

- static final boolean [IS\\_JAR](#) = (Environment.class.getResource("Environment.class") + "").startsWith("jar")

### 3.29.1 Member Data Documentation

#### 3.29.1.1 IS\_JAR

```
final boolean IS_JAR = (Environment.class.getResource("Environment.class") + "").startsWith(
With("jar")    [static]
```

is this running from a jar or an ide. used for calculating paths.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Misc/Environment.java

## 3.30 EnvManager

### 3.30.1 Detailed Description

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

#### Author

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

### 3.30.2 Member Function Documentation

#### 3.30.2.1 handledErr()

```
void handledErr (
    MyError err )
```

notifies manager of a managed error

## Parameters

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

**3.30.2.2 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
------------	------------------

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/Threads/ErrorHandling/EnvManager.java

**3.31 ErrorHandler< E extends MyError >****3.31.1 Detailed Description**

The interface Error handler.

## Parameters

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

**3.31.2 Member Function Documentation****3.31.2.1 ignore()**

```
static boolean ignore (
    ThrowingRunnable runnable ) [static]
```

Ignore boolean.

## Parameters

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



**Returns**

true if the runnable threw, false otherwise

**3.31.2.2 handle()**

```
void handle (
    MyError err )
```

Handle.

**Parameters**

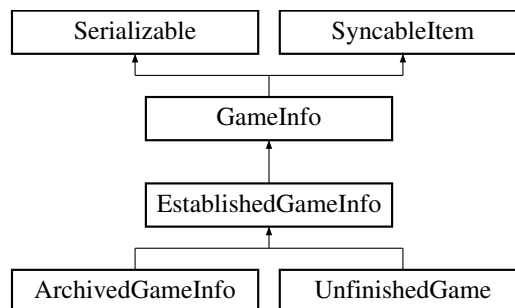
<i>err</i>	the err
------------	---------

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/Threads/ErrorHandling/ErrorHandler.java

**3.32 EstablishedGameInfo**

Inheritance diagram for EstablishedGameInfo:

**Public Attributes**

- final String [opponentUsername](#)

**Protected Attributes**

- Date [createdAt](#)

**3.32.1 Detailed Description**

Established game info.

**Author**

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

## 3.32.2 Constructor & Destructor Documentation

### 3.32.2.1 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

## 3.32.3 Member Function Documentation

### 3.32.3.1 getCreatedAt()

```
Date getCreatedAt ( )
```

Gets created at.

#### Returns

the created at

### 3.32.3.2 setCreatedAt()

```
void setCreatedAt (
    Date createdAt )
```

Sets created at.

## Parameters

<i>created</i> ↔ <i>At</i>	the created at
-------------------------------	----------------

**3.32.3.3 getMoveStack()**

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

Gets move stack.

## Returns

the move stack

**3.32.3.4 toString()**

```
String toString ( )
```

To string string.

## Returns

the string

Reimplemented from [GameInfo](#).

Reimplemented in [ArchivedGameInfo](#).

**3.32.3.5 getGameDesc()**

```
String getGameDesc ( )
```

Gets game desc.

## Returns

the game desc

Reimplemented from [GameInfo](#).

### 3.32.4 Member Data Documentation

#### 3.32.4.1 opponentUsername

```
final String opponentUsername
```

The Opponent username.

#### 3.32.4.2 createdAt

```
Date createdAt [protected]
```

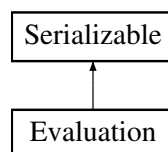
The Created at.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/SavedGames/EstablishedGameInfo.java

## 3.33 Evaluation

Inheritance diagram for Evaluation:



### Static Public Attributes

- static final int [TIE\\_EVAL](#) = 0
- static final int [WIN\\_EVAL](#) = Integer.MAX\_VALUE
- static final int [LOSS\\_EVAL](#) = -[WIN\\_EVAL](#)

#### 3.33.1 Detailed Description

[Evaluation](#) - represents a position's evaluation relative to a player color.

Author

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

#### 3.33.2 Constructor & Destructor Documentation

##### 3.33.2.1 Evaluation() [1/4]

```
Evaluation (
    GameState gameStatus,
    PlayerColor evaluationFor )
```

Instantiates a new [Evaluation](#).

## Parameters

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

**3.33.2.2 Evaluation()** [2/4]

```
Evaluation (
    int eval,
    GameStatus gameStatus,
    PlayerColor evaluationFor )
```

Instantiates a new [Evaluation](#).

## Parameters

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

**3.33.2.3 Evaluation()** [3/4]

```
Evaluation (
    PlayerColor evaluationFor )
```

Instantiates a new [Evaluation](#).

## Parameters

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

**3.33.2.4 Evaluation()** [4/4]

```
Evaluation (
    Evaluation other )
```

Instantiates a new [Evaluation](#).

## Parameters

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

### 3.33.3 Member Function Documentation

#### 3.33.3.1 book()

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

Book evaluation.

##### Returns

the evaluation

#### 3.33.3.2 addDetail()

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

Add detail.

##### Parameters

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

#### 3.33.3.3 assertNotGameOver()

```
void assertNotGameOver ( )
```

Assert not game over.

#### 3.33.3.4 getEvaluationDepth()

```
Integer getEvaluationDepth ( )
```

Gets evaluation depth.

##### Returns

the evaluation depth

### 3.33.3.5 setEvaluationDepth()

```
void setEvaluationDepth (
    Integer evaluationDepth )
```

Sets evaluation depth.

#### Parameters

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

### 3.33.3.6 isGameOver()

```
boolean isGameOver ( )
```

Is game over boolean.

#### Returns

the boolean

### 3.33.3.7 isCheck()

```
boolean isCheck ( )
```

Is check boolean.

#### Returns

the boolean

### 3.33.3.8 isGreaterThan()

```
boolean isGreaterThan (
    Evaluation other )
```

Is greater than boolean.

#### Parameters

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

**Returns**

the boolean

**3.33.3.9 getEval()**

```
int getEval ( )
```

Gets eval.

**Returns**

the eval

**3.33.3.10 setEval()**

```
void setEval (
    int eval )
```

Sets eval.

**Parameters**

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

**3.33.3.11 getGameStatus()**

```
GameStatus getGameStatus ( )
```

Gets game status.

**Returns**

the game status

**3.33.3.12 equals()**

```
boolean equals (
    Object o )
```

Equals boolean.



**Parameters**

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

**Returns**

the boolean

**3.33.3.13 toString()**

```
String toString ( )
```

To string string.

**Returns**

the string

**3.33.3.14 convertFromCentipawns()**

```
float convertFromCentipawns ( )
```

Convert from centipawns float.

**Returns**

the float

**3.33.3.15 getEvaluationFor()**

```
PlayerColor getEvaluationFor ( )
```

Gets evaluation for.

**Returns**

the evaluation for

**3.33.3.16 setPerspective()**

```
Evaluation setPerspective (
    PlayerColor playerColor )
```

Sets perspective.

**Parameters**

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

**Returns**

the perspective

**3.33.3.17 flipEval()**

```
void flipEval ( )
```

Flip eval.

**3.33.3.18 print()**

```
void print ( )
```

Print.

**3.33.3.19 EvaluationDetail()**

```
record EvaluationDetail (
    EvaluationParameters parm,
    int eval )
```

[Evaluation](#) detail.

**Author**

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

**3.33.4 Member Data Documentation****3.33.4.1 TIE\_EVAL**

```
final int TIE_EVAL = 0 [static]
```

The constant TIE\_EVAL.

#### 3.33.4.2 WIN\_EVAL

```
final int WIN_EVAL = Integer.MAX_VALUE [static]
```

The constant WIN\_EVAL.

#### 3.33.4.3 LOSS\_EVAL

```
final int LOSS_EVAL = -WIN_EVAL [static]
```

The constant LOSS\_EVAL.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/Evaluation/Evaluation.java

### 3.34 EvaluationParameters Enum Reference

#### Public Attributes

- MATERIAL =(1.5)
- PIECE\_TABLES =(1.3)
- KING\_SAFETY =(0.1)
- HANGING\_PIECES
- SQUARE\_CONTROL
- MOVEMENT\_ABILITY
- FORCE\_KING\_TO\_CORNER =(0.8)
- EG\_WEIGHT
- STOCKFISH\_SAYS
- final double weight

#### 3.34.1 Detailed Description

[Evaluation](#) parameters - all evaluation parameters. (some are unused).

Author

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

#### 3.34.2 Member Data Documentation

##### 3.34.2.1 MATERIAL

```
MATERIAL =(1.5)
```

Material evaluation parameters.

### 3.34.2.2 PIECE\_TABLES

PIECE\_TABLES = (1.3)

Piece tables evaluation parameters.

### 3.34.2.3 KING\_SAFETY

KING\_SAFETY = (.1)

King safety evaluation parameters.

### 3.34.2.4 HANGING\_PIECES

HANGING\_PIECES

Hanging pieces evaluation parameters.

### 3.34.2.5 SQUARE\_CONTROL

SQUARE\_CONTROL

Square control evaluation parameters.

### 3.34.2.6 MOVEMENT\_ABILITY

MOVEMENT\_ABILITY

Movement ability evaluation parameters.

### 3.34.2.7 FORCE\_KING\_TO\_CORNER

FORCE\_KING\_TO\_CORNER = (.8)

Force king to corner evaluation parameters.

### 3.34.2.8 EG\_WEIGHT

EG\_WEIGHT

Eg weight evaluation parameters.

### 3.34.2.9 STOCKFISH\_SAYS

STOCKFISH\_SAYS

Stockfish says evaluation parameters.

### 3.34.2.10 weight

```
final double weight
```

The parameter's weight

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Game/Evaluation/EvaluationParameters.java

## 3.35 FontManager

### Classes

- class **Base**
- class **Dialogs**
- class **JMenus**

### Static Public Attributes

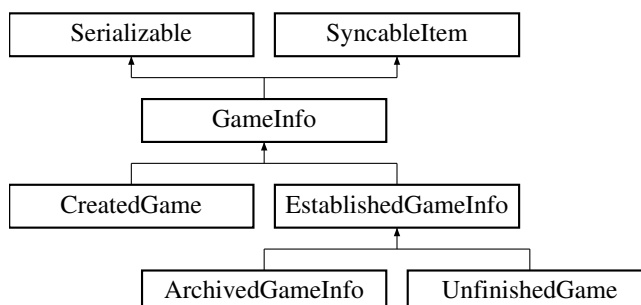
- static final Font **statusLbl** = Base.normal
- static final Font **dbResponseTable** = Base.normal
- static final Font **dbResponseTableHeader** = Base.normal
- static final Font **sidePanel** = Base.normal
- static final Font **coordinates** = Base.normal
- static final Font **boardButtons** = Base.normal
- static final Font **statistics** = Base.normal
- static final Font **backOk** = Base.normal
- static final Font **xLarge** = Base.xLarge
- static final Font **normal** = Base.normal
- static final Font **large** = Base.large
- static final Font **small** = Base.small
- static final Font **error** = Base.error
- static final Font **defaultLinkLbl** = Base.normal

The documentation for this class was generated from the following file:

- ver14/SharedClasses/UI/FontManager.java

## 3.36 GameInfo

Inheritance diagram for GameInfo:



## Public Attributes

- final String [gameId](#)
- final String [creatorUsername](#)
- final [GameSettings](#) [gameSettings](#)

### 3.36.1 Detailed Description

Game info.

#### Author

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

### 3.36.2 Constructor & Destructor Documentation

#### 3.36.2.1 GameInfo()

```
GameInfo (
    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

### 3.36.3 Member Function Documentation

#### 3.36.3.1 example()

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

Example game info.

#### Returns

the game info

### 3.36.3.2 isCreator()

```
boolean isCreator (
    String username )
```

Is creator boolean.

#### Parameters

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

#### Returns

the boolean

### 3.36.3.3 ID()

```
String ID ( )
```

Id string.

#### Returns

the string

Implements [SyncableItem](#).

### 3.36.3.4 equals()

```
boolean equals (
    Object o )
```

Equals boolean.

#### Parameters

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

#### Returns

the boolean

### 3.36.3.5 toString()

```
String toString ( )
```

To string string.

#### Returns

the string

Reimplemented in [ArchivedGameInfo](#), and [EstablishedGameInfo](#).

### 3.36.3.6 getGameDesc()

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

Gets game desc.

#### Returns

the game desc

Reimplemented in [CreatedGame](#), and [EstablishedGameInfo](#).

### 3.36.3.7 getJoiningPlayerColor()

```
PlayerColor getJoiningPlayerColor ( )
```

Gets joining player color.

#### Returns

the joining player color

### 3.36.3.8 getStartingColor()

```
PlayerColor getStartingColor ( )
```

Gets starting color.

#### Returns

the starting color



### 3.36.4 Member Data Documentation

#### 3.36.4.1 gameId

```
final String gameId
```

The Game id.

#### 3.36.4.2 creatorUsername

```
final String creatorUsername
```

The Creator username.

#### 3.36.4.3 gameSettings

```
final GameSettings gameSettings
```

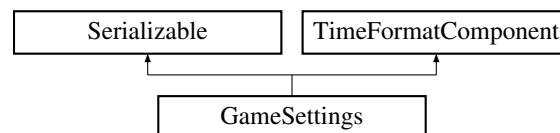
The Game settings.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/SavedGames/GameInfo.java

## 3.37 GameSettings

Inheritance diagram for GameSettings:



### Classes

- enum [GameType](#)

### Static Public Attributes

- static final [GameSettings](#) **EXAMPLE** = new [GameSettings](#)([PlayerColor.WHITE](#), [TimeFormat.ULTRA\\_BULLET](#), "rnbqkbnr/pppppppp/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1", [AiParameters.EZ\\_MY\\_AI](#), [GameType.RESUME](#))

### 3.37.1 Class Documentation

#### 3.37.1.1 enum ver14::SharedClasses::Game::GameSetup::GameSettings::GameType

## Enumerator

JOIN_EXISTING	
RESUME	
CREATE_NEW	
QUICK_MATCH	

### 3.37.2 Member Function Documentation

#### 3.37.2.1 setTimeFormat()

```
void setTimeFormat (
    TimeFormat timeFormat )
```

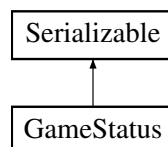
Implements [TimeFormatComponent](#).

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/GameSetup/GameSettings.java

## 3.38 GameStatus

Inheritance diagram for GameStatus:



### Classes

- enum [GameStatusType](#)
- enum [SpecificStatus](#)

#### 3.38.1 Detailed Description

Game status - represents a game status .

#### Author

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

## 3.38.2 Member Function Documentation

### 3.38.2.1 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

**3.38.2.2 gameGoesOn()**

```
static GameStatus gameGoesOn ( ) [static]
```

Game goes on game status.

**Returns**

the game status

**3.38.2.3 tieByAgreement()**

```
static GameStatus tieByAgreement ( ) [static]
```

Tie by agreement game status.

**Returns**

the game status

**3.38.2.4 stalemate()**

```
static GameStatus stalemate ( ) [static]
```

Stalemate game status.

**Returns**

the game status

### 3.38.2.5 fiftyMoveRule()

```
static GameStatus fiftyMoveRule ( ) [static]
```

Fifty move rule game status.

#### Returns

the game status

### 3.38.2.6 serverStoppedGame()

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

Server stopped game game status.

#### Parameters

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

#### Returns

the game status

### 3.38.2.7 setCustomStr()

```
void setCustomStr (
    String customStr )
```

Sets custom str.

#### Parameters

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

### 3.38.2.8 threeFoldRepetition()

```
static GameStatus threeFoldRepetition ( ) [static]
```

Three fold repetition game status.

**Returns**

the game status

**3.38.2.9 insufficientMaterial()**

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

Insufficient material game status.

**Returns**

the game status

**3.38.2.10 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

**3.38.2.11 timedOut()**

```
static GameStatus 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

**3.38.2.12 playerResigned()**

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

Player resigned game status.

**Parameters**

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

**Returns**

the game status

**3.38.2.13 isDisconnected()**

```
boolean isDisconnected ( )
```

Is disconnected boolean.

**Returns**

the boolean

**3.38.2.14 getCheckedKingLoc()**

```
Location getCheckedKingLoc ( )
```

Gets checked king loc.

**Returns**

the checked king loc

### 3.38.2.15 getWinningColor()

```
PlayerColor getWinningColor ( )
```

Gets winning color.

#### Returns

the winning color

### 3.38.2.16 getGameStatusType()

```
GameStatusType getGameStatusType ( )
```

Gets game status type.

#### Returns

the game status type

### 3.38.2.17 setInCheck()

```
void setInCheck (
    Location checkedKingLoc )
```

Sets in check.

#### Parameters

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

### 3.38.2.18 isCheck()

```
boolean isCheck ( )
```

Is check boolean.

#### Returns

the boolean



**3.38.2.19** `getDetailedStr()` [1/2]

```
String getDetailedStr ( )
```

Gets detailed str.

**Returns**

the detailed str

**3.38.2.20** `getDetailedStr()` [2/2]

```
String getDetailedStr (
    Map< PlayerColor, String > playerUsernamesMap )
```

Gets detailed str.

**Parameters**

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

**Returns**

the detailed str

**3.38.2.21** `isGameOver()`

```
boolean isGameOver ( )
```

Is game over boolean.

**Returns**

the boolean

**3.38.2.22** `toString()`

```
String toString ( )
```

To string string.

**Returns**

the string

### 3.38.2.23 setDepth()

```
void setDepth (
    int depth )
```

Sets depth.

#### Parameters

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

### 3.38.2.24 getSpecificStatus()

```
SpecificStatus getSpecificStatus ( )
```

Gets specific status.

#### Returns

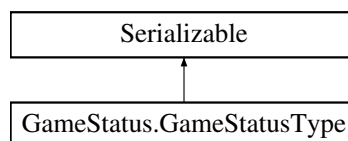
the specific status

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/Evaluation/GameState.java

## 3.39 GameState.GameStatusType Enum Reference

Inheritance diagram for GameState.GameStatusType:



### Public Attributes

- `TIE` = ("1½/2", "Tie")
- `CHECK` = ("+", " ")
- `GAME_GOES_ON` = ("")
- `WIN_OR_LOSS` = ("#", "Won")
- `UNFINISHED` = ("...", " ")
- final String `annotation`
- final String `gameOverStr`

### 3.39.1 Detailed Description

Game status type .

Author

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

### 3.39.2 Constructor & Destructor Documentation

#### 3.39.2.1 GameStateType() [1/2]

```
GameStateType (
    String annotation )
```

Instantiates a new Game status type.

Parameters

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

#### 3.39.2.2 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

### 3.39.3 Member Function Documentation

#### 3.39.3.1 isGameOver()

```
boolean isGameOver ( )
```

Is game over.

**Returns**

true if is game over. false otherwise

### 3.39.4 Member Data Documentation

#### 3.39.4.1 TIE

```
TIE = ("½", "Tie")
```

Tie game status type.

#### 3.39.4.2 CHECK

```
CHECK = ("+",)
```

Check game status type.

#### 3.39.4.3 GAME\_GOES\_ON

```
GAME_GOES_ON = ("")
```

Game goes on game status type.

#### 3.39.4.4 WIN\_OR\_LOSS

```
WIN_OR_LOSS = ("#", "Won")
```

Win or loss game status type.

#### 3.39.4.5 UNFINISHED

```
UNFINISHED = ("...")
```

Unfinished game status type.

#### 3.39.4.6 annotation

```
final String annotation
```

The game status annotation.

### 3.39.4.7 gameOverStr

```
final String gameOverStr
```

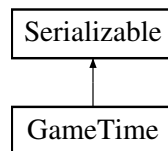
game over str

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Game/Evaluation/GameStatus.java

## 3.40 GameTime

Inheritance diagram for GameTime:



### Classes

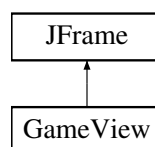
- class **RunningTime**

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/GameSetup/GameTime.java

## 3.41 GameView

Inheritance diagram for GameView:

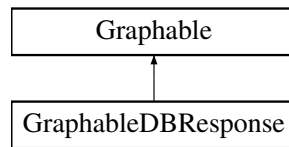


The documentation for this class was generated from the following file:

- ver14/SharedClasses/UI/GameView.java

## 3.42 Graphable

Inheritance diagram for Graphable:

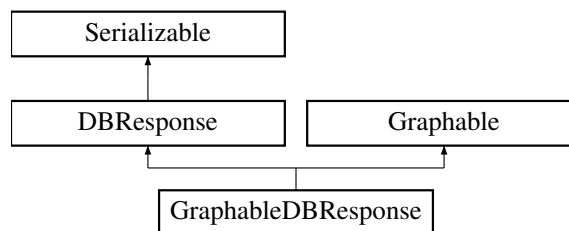


The documentation for this interface was generated from the following file:

- ver14/SharedClasses/DBActions/DBResponse/Graphable/Graphable.java

## 3.43 GraphableDBResponse

Inheritance diagram for GraphableDBResponse:



### Additional Inherited Members

#### 3.43.1 Member Function Documentation

##### 3.43.1.1 toString()

```
String toString ( )
```

To string string.

##### Returns

the string

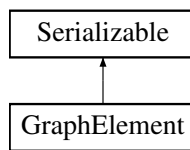
Reimplemented from [DBResponse](#).

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/DBResponse/Graphable/GraphableDBResponse.java

## 3.44 GraphElement

Inheritance diagram for GraphElement:



### Protected Attributes

- final double **num**
- final [GraphElementType](#) **graphElementType**
- final String **name**

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/DBResponse/Graphable/GraphElement.java

## 3.45 GraphElementType Enum Reference

### Public Attributes

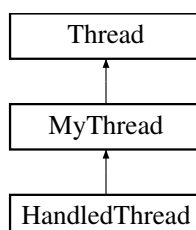
- **GREEN**
- **RED**
- **YELLOW**
- **NORMAL**

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/DBActions/DBResponse/Graphable/GraphElementType.java

## 3.46 HandledThread

Inheritance diagram for HandledThread:



### 3.46.1 Detailed Description

The type Handled thread.

### 3.46.2 Constructor & Destructor Documentation

#### 3.46.2.1 HandledThread() [1/2]

```
HandledThread ( )
```

Instantiates a new Handled thread.

#### 3.46.2.2 HandledThread() [2/2]

```
HandledThread (  
    ThrowingRunnable runnable )
```

Instantiates a new Handled thread.

##### Parameters

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

### 3.46.3 Member Function Documentation

#### 3.46.3.1 runInHandledThread()

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

Run in handled thread handled thread.

##### Parameters

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

##### Returns

the handled thread



### 3.46.3.2 setRunnable()

```
void setRunnable (
    ThrowingRunnable runnable )
```

Sets runnable.

#### Parameters

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

### 3.46.3.3 handledRun()

```
void handledRun ( ) throws Throwable [protected]
```

Handled run.

#### Exceptions

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

Reimplemented from [MyThread](#).

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Threads/HandledThread.java

## 3.47 MyJFrame.MyAdapter.HeldDown

### 3.47.1 Detailed Description

The interface Held down.

### 3.47.2 Member Function Documentation

#### 3.47.2.1 startPress()

```
void startPress ( )
```

Start press.

### 3.47.2.2 endPress()

```
void endPress ( )
```

End press.

### 3.47.2.3 key()

```
int key ( )
```

Key int.

#### Returns

the int

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/UI/MyJFrame.java

## 3.48 IDsGenerator

### 3.48.1 Detailed Description

IDs generator.

#### Author

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

### 3.48.2 Member Function Documentation

#### 3.48.2.1 generate()

```
synchronized String generate ( )
```

Generate id.

#### Returns

the string

#### 3.48.2.2 canUseId()

```
boolean canUseId (
    String id )
```

Can use id boolean.

## Parameters

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

## Returns

the boolean

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Misc/IDsGenerator.java

## 3.49 StrUtils.IterationThingy< T >

### 3.49.1 Detailed Description

Iteration thingy.

## Author

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

### 3.49.2 Member Function Documentation

#### 3.49.2.1 iteration()

```
T iteration (
    int i,
    boolean isLast )
```

Iteration t.

## Parameters

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

## Returns

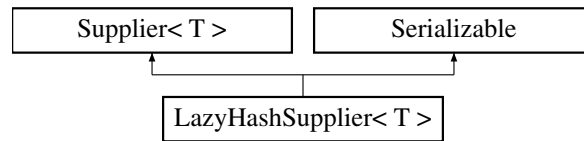
the t

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/Utils/StrUtils.java

## 3.50 LazyHashSupplier< T >

Inheritance diagram for LazyHashSupplier< T >:



### 3.50.1 Detailed Description

represents a supplier for a hash that will later be fully calculated.

#### Author

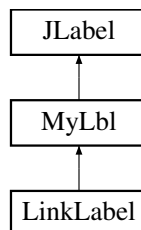
Bezalel Avrahami ( [bezalel13250@gmail.com](mailto:bezalel13250@gmail.com) )

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/Callbacks/LazyHashSupplier.java

## 3.51 LinkLabel

Inheritance diagram for LinkLabel:



### 3.51.1 Member Function Documentation

#### 3.51.1.1 setText()

```
void setText (
    String text )
```

Reimplemented from [MyLbl](#).

The documentation for this class was generated from the following file:

- ver14/SharedClasses/UI/LinkLabel.java

## 3.52 Location Enum Reference

### Public Attributes

- A8
- B8
- C8
- D8
- E8
- F8
- G8
- H8
- A7
- B7
- C7
- D7
- E7
- F7
- G7
- H7
- A6
- B6
- C6
- D6
- E6
- F6
- G6
- H6
- A5
- B5
- C5
- D5
- E5
- F5
- G5
- H5
- A4
- B4
- C4
- D4
- E4
- F4
- G4
- H4
- A3
- B3
- C3
- D3
- E3
- F3
- G3
- H3
- A2
- B2

- **C2**
- **D2**
- **E2**
- **F2**
- **G2**
- **H2**
- **A1**
- **B1**
- **C1**
- **D1**
- **E1**
- **F1**
- **G1**
- **H1**
- final long [asLong](#)
- final int [asInt](#)
- final int [row](#)
- final int [col](#)

### Static Public Attributes

- static final int **A** = A1.col
- static final int **B** = B1.col
- static final int **C** = C1.col
- static final int **D** = D1.col
- static final int **E** = E1.col
- static final int **F** = F1.col
- static final int **G** = G1.col
- static final int **H** = H1.col
- static final ArrayList< [Location](#) > **ALL\_LOCS**
- static final int **NUM\_OF\_SQUARES** = 64
- static final int **KING\_STARTING\_COL** = E1.col
- static final int **WHITE\_STARTING\_ROW** = E1.row
- static final int **BLACK\_STARTING\_ROW** = [flip](#)(WHITE\_STARTING\_ROW)
- static final int **WHITE\_DIFF** = WHITE\_STARTING\_ROW > BLACK\_STARTING\_ROW ? -1 : 1
- static final int **BLACK\_DIFF** = -WHITE\_DIFF
- static final [PlayerColor](#) **normalPerspective** = [PlayerColor.WHITE](#)

### 3.52.1 Detailed Description

[Location](#) - 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](mailto:bezalel13250@gmail.com) )

### 3.52.2 Constructor & Destructor Documentation

### 3.52.2.1 Location()

`Location ( )`

Instantiates a new [Location](#).

## 3.52.3 Member Function Documentation

### 3.52.3.1 locsMatStr()

`static String locsMatStr ( ) [static]`

Locs mat str string.

#### Returns

the string

### 3.52.3.2 locsMat()

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

Locs mat location [ ] [ ].

#### Returns

the location [ ] [ ]

### 3.52.3.3 matrixStr()

`String matrixStr ( )`

Matrix str string.

#### Returns

the string

### 3.52.3.4 createMatIndicesStr()

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

Create mat indices str string.

#### Returns

the string

### 3.52.3.5 getLoc() [1/7]

```
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

### 3.52.3.6 getLoc() [2/7]

```
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



### 3.52.3.7 getLoc() [3/7]

```
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

### 3.52.3.8 getLoc() [4/7]

```
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

### 3.52.3.9 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

**3.52.3.10 getLoc() [5/7]**

```
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**

...

**3.52.3.11 getLoc() [6/7]**

```
static Location getLoc (  
    int row,  
    int col ) [static]
```

Gets loc.

**Parameters**

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

**Returns**

the loc

**3.52.3.12 getLoc()** [7/7]

```
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

**3.52.3.13 flip()** [1/2]

```
static int flip (
    int num ) [static]
```

Flip int.

**Parameters**

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

**Returns**

the int

**3.52.3.14 isInBounds()**

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

Is in bounds boolean.

**Parameters**

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

**Returns**

the boolean

**3.52.3.15 getColString()**

```
String getColString ( )
```

Gets col string.

**Returns**

the col string

**3.52.3.16 flip() [2/2]**

```
Location flip ( )
```

Flip location.

**Returns**

the location

**3.52.3.17 isBlackSquare()**

```
boolean isBlackSquare ( )
```

Is black square boolean.

**Returns**

the boolean

**3.52.3.18 isWhiteSquare()**

```
boolean isWhiteSquare ( )
```

Is white square boolean.

**Returns**

the boolean

### 3.52.3.19 toString()

```
String toString ( )
```

To string string.

#### Returns

the string

### 3.52.3.20 getMaxDistance()

```
int getMaxDistance (
    Location other )
```

Gets max distance.

#### Parameters

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

#### Returns

the max distance

### 3.52.3.21 getRowString()

```
String getRowString ( )
```

Gets row string.

#### Returns

the row string

### 3.52.3.22 hash()

```
int hash (
    PieceType pieceType )
```

Hash int.

**Parameters**

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

**Returns**

the int

### 3.52.4 Member Data Documentation

#### 3.52.4.1 asLong

```
final long asLong
```

a long value with a bit set on this location

#### 3.52.4.2 asInt

```
final int asInt
```

the calculated index

#### 3.52.4.3 row

```
final int row
```

this location's row

#### 3.52.4.4 col

```
final int col
```

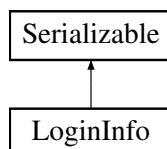
this location's column

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Game/Location.java

## 3.53 LoginInfo

Inheritance diagram for LoginInfo:



### 3.53.1 Detailed Description

Login info.

Author

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

### 3.53.2 Constructor & Destructor Documentation

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

```
LoginInfo ( )
```

Instantiates a new Login info.

#### 3.53.2.2 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

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

```
LoginInfo (
    LoginType loginType )
```

Instantiates a new Login info.

Parameters

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

### 3.53.3 Member Function Documentation

#### 3.53.3.1 `getProfilePic()`

```
String getProfilePic ( )
```

Gets profile pic.

##### Returns

the profile pic

#### 3.53.3.2 `setProfilePic()`

```
void setProfilePic (
    String profilePic )
```

Sets profile pic.

##### Parameters

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

#### 3.53.3.3 `isGuest()`

```
boolean isGuest ( )
```

Is guest boolean.

##### Returns

the boolean

#### 3.53.3.4 `getLoginType()`

```
LoginType getLoginType ( )
```

Gets login type.

##### Returns

the login type



### 3.53.3.5 setLoginType()

```
void setLoginType (
    LoginType loginType )
```

Sets login type.

#### Parameters

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

### 3.53.3.6 getUsername()

```
String getUsername ( )
```

Gets username.

#### Returns

the username

### 3.53.3.7 setUsername()

```
void setUsername (
    String username )
```

Sets username.

#### Parameters

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

### 3.53.3.8 getPassword()

```
String getPassword ( )
```

Gets password.

#### Returns

the password

### 3.53.3.9 setPassword()

```
void setPassword (
    String password )
```

Sets password.

#### Parameters

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

### 3.53.3.10 toString()

```
String toString ( )
```

To string string.

#### Returns

the string

### 3.53.3.11 asUser()

```
boolean asUser ( )
```

As user boolean.

#### Returns

the boolean

### 3.53.3.12 initDebugLoginValues()

```
void initDebugLoginValues ( )
```

Init debug login values.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Login/LoginInfo.java

## 3.54 LoginType Enum Reference

### Public Attributes

- [LOGIN](#)
- [REGISTER](#)
- [GUEST](#)
- [CANCEL](#)
- [NOT\\_SET\\_YET](#)

### 3.54.1 Detailed Description

Login Type.

Author

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

### 3.54.2 Member Function Documentation

#### 3.54.2.1 asUser()

```
boolean asUser ( )
```

As user boolean.

Returns

the boolean

#### 3.54.2.2 toString()

```
String toString ( )
```

To string string.

Returns

the string

### 3.54.3 Member Data Documentation

#### 3.54.3.1 LOGIN

LOGIN

Login.

#### 3.54.3.2 REGISTER

REGISTER

Register.

#### 3.54.3.3 GUEST

GUEST

Guest.

#### 3.54.3.4 CANCEL

CANCEL

Cancel.

#### 3.54.3.5 NOT\_SET\_YET

NOT\_SET\_YET

Not set yet.

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Login/LoginType.java

## 3.55 Math Enum Reference

### Public Attributes

- [Plus](#)
- [Mult](#)
- [Div](#)
- [Col col](#)

### 3.55.1 Detailed Description

[Math](#) - allows for math actions on columns and some math-related utilities for columns.

Author

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

### 3.55.2 Member Function Documentation

#### 3.55.2.1 nullIf0()

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

Null if 0 string.

Parameters

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

Returns

the string

#### 3.55.2.2 formatNum() [1/2]

```
static String formatNum (  
    Object num ) [static]
```

Format num string.

Parameters

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

Returns

the string

### 3.55.2.3 strVal()

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

Str val string.

#### Parameters

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

#### Returns

the string

### 3.55.2.4 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

### 3.55.2.5 str()

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

Str string.

#### Parameters

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

**Returns**

the string

**3.55.2.6 asFloat()**

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

As float string.

**Parameters**

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

**Returns**

the string

**3.55.2.7 zeroIfNull()**

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

Zero if null.

**3.55.2.8 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

### 3.55.2.9 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

### 3.55.2.10 apply()

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

Apply.

#### Parameters

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

### 3.55.2.11 strSource()

```
String strSource ( )
```

Str source string.

#### Returns

the string



**3.55.2.12 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

### 3.55.3 Member Data Documentation

#### 3.55.3.1 Plus

Plus

**Initial value:**

```
={
    @Override
    protected void apply(Object value) {
        simpleOperation(strSource(), "+", strVal(value));
    }
}
```

The Plus.

#### 3.55.3.2 Mult

Mult

**Initial value:**

```
={
    @Override
    protected void apply(Object value) {
        simpleOperation(formatNum(strSource()), "*", formatNum(strVal(value)));
    }
}
```

The Mult.

#### 3.55.3.3 Div

Div

**Initial value:**

```
={
    @Override
    protected void apply(Object value) {
        simpleOperation(formatNum(strSource()), "/", nullIf0(value));
        zeroIfNull();
    }
}
```

The Div.

#### 3.55.3.4 col

`Col col`

The `Col`.

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/DBActions/Table/Math.java

## 3.56 MathUtils

### 3.56.1 Detailed Description

Math utils.

Author

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

### 3.56.2 Member Function Documentation

#### 3.56.2.1 log()

```
static double log (  
    double num,  
    int base ) [static]
```

Log.

Parameters

<i>num</i>	the num
<i>base</i>	the base

Returns

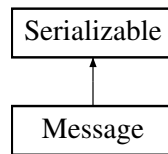
the double

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Utils/MathUtils.java

## 3.57 Message

Inheritance diagram for Message:



### Public Attributes

- final String [messageID](#)

### 3.57.1 Detailed Description

#### 3.57.1.1 Message.

by Ilan Peretz ( [ilanperets@gmail.com](mailto:ilanperets@gmail.com)) 10/11/2021

### 3.57.2 Constructor & Destructor Documentation

#### 3.57.2.1 Message() [1/4]

```
Message (
    MessageType messageType )
```

Instantiates a new [Message](#).

Parameters

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

#### 3.57.2.2 Message() [2/4]

```
Message (
    MessageType messageType,
    String subject,
    String respondingToMsgId )
```

Instantiates a new [Message](#).

## Parameters

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

**3.57.2.3 Message() [3/4]**

```
Message (
    MessageType messageType,
    Message respondingTo )
```

Instantiates a new [Message](#).

## Parameters

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

**3.57.2.4 Message() [4/4]**

```
Message (
    MessageType messageType,
    String subject )
```

Instantiates a new [Message](#).

## Parameters

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

**3.57.3 Member Function Documentation****3.57.3.1 askForLogin()**

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

Ask for login message.

**Returns**

the message

**3.57.3.2 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

**3.57.3.3 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

**3.57.3.4 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

### 3.57.3.5 throwError()

```
static Message throwError (  
    MyError error ) [static]
```

Throw error message.

#### Parameters

<i>error</i>	if null interrupts
--------------	--------------------

#### Returns

the message

### 3.57.3.6 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

**3.57.3.7 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

**3.57.3.8 waitForMatch()**

```
static Message waitForMatch ( ) [static]
```

Wait for match message.

**Returns**

the message

**3.57.3.9 gameOver()**

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

Game over message.

**Parameters**

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



**Returns**

the message

**3.57.3.10 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

**3.57.3.11 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

**3.57.3.12 interrupt()**

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

Interrupt message.

**Returns**

the message

**3.57.3.13 bye()**

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

Bye message.

**Parameters**

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

**Returns**

the message

**3.57.3.14 error()**

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

Error message.

**Parameters**

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

**Returns**

the message

**3.57.3.15 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

**3.57.3.16 answerQuestion()**

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

Answer question message.

## Parameters

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

## Returns

the message

**3.57.3.17 askQuestion()**

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

Ask question message.

## Parameters

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

## Returns

the message

### 3.57.3.18 checkUsernameAvailability()

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

Check username availability message.

#### Parameters

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

#### Returns

the message

### 3.57.3.19 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

### 3.57.3.20 returnUsernameAvailable()

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

Return username available message.

#### Parameters

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

**Returns**

the message

**3.57.3.21 dbRequest()**

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

Db request message.

**Parameters**

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

**Returns**

the message

**3.57.3.22 setDbRequest()**

```
void setDbRequest (
    DBRequest dbRequest )
```

Sets db request.

**Parameters**

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

**3.57.3.23 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

**3.57.3.24 syncLists()**

```
static Message syncLists (
    SyncedItems<?>... syncedLists ) [static]
```

Sync lists message.

**Parameters**

<i>syncedLists</i>	the synced lists
--------------------	------------------

**Returns**

the message

**3.57.3.25 getError()**

```
MyError getError ( )
```

Gets error.

**Returns**

the error

**3.57.3.26 setError()**

```
void setError (
    MyError error )
```

Sets error.

**Parameters**

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

**3.57.3.27 getDBRequest()**

```
DBRequest getDBRequest ( )
```

Gets db request.

**Returns**

the db request

**3.57.3.28 getUsernameSuggestions()**

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

Gets username suggestions.

**Returns**

the username suggestions

**3.57.3.29 setUsernameSuggestions()**

```
void setUsernameSuggestions (
    ArrayList< String > usernameSuggestions )
```

Sets username suggestions.

**Parameters**

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

**3.57.3.30 getDBResponse()**

```
DBResponse getDBResponse ( )
```

Gets db response.

**Returns**

the db response

### 3.57.3.31 setDBResponse()

```
void setDBResponse (
    DBResponse requestedStats )
```

Sets db response.

#### Parameters

<i>requestedStats</i>	the requested stats
-----------------------	---------------------

### 3.57.3.32 getMoveStack()

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

Gets move stack.

#### Returns

the move stack

### 3.57.3.33 setMoveStack()

```
void setMoveStack (
    Stack< Move > moveStack )
```

Sets move stack.

#### Parameters

<i>moveStack</i>	the move stack
------------------	----------------

### 3.57.3.34 getSyncedList()

```
SyncedList<?>[] getSyncedList ( )
```

Get synced lists synced items [ ].

#### Returns

the synced items [ ]



**3.57.3.35 setSyncedList()**

```
void setSyncedList (
    SyncedList<?>... syncedList )
```

Sets synced lists.

**Parameters**

<i>syncedList</i>	the synced lists
-------------------	------------------

**3.57.3.36 getRespondingToMsgId()**

```
String getRespondingToMsgId ( )
```

Gets responding to msg id.

**Returns**

the responding to msg id

**3.57.3.37 setRespondingToMsgId()**

```
void setRespondingToMsgId (
    String respondingToMsgId )
```

Sets responding to msg id.

**Parameters**

<i>respondingToMsgId</i>	the responding to msg id
--------------------------	--------------------------

**3.57.3.38 setRespondingTo()**

```
void setRespondingTo (
    Message msg )
```

Sets responding to.

**Parameters**

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

**3.57.3.39 isResponse()**

```
boolean isResponse ( )
```

Is response boolean.

**Returns**

the boolean

**3.57.3.40 getOtherPlayer()**

```
String getOtherPlayer ( )
```

Gets other player.

**Returns**

the other player

**3.57.3.41 setOtherPlayer()**

```
void setOtherPlayer (
    String otherPlayer )
```

Sets other player.

**Parameters**

<i>otherPlayer</i>	the other player
--------------------	------------------

**3.57.3.42 getQuestion()**

```
Question getQuestion ( )
```

Gets question.

**Returns**

the question

**3.57.3.43 setQuestion()**

```
void setQuestion (
    Question question )
```

Sets question.

**Parameters**

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

**3.57.3.44 getAvailable()**

```
Boolean getAvailable ( )
```

Gets available.

**Returns**

the available

**3.57.3.45 setAvailable()**

```
void setAvailable (
    Boolean available )
```

Sets available.

**Parameters**

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

**3.57.3.46 getUsername()**

```
String getUsername ( )
```

Gets username.

**Returns**

the username

**3.57.3.47 setUsername()**

```
void setUsername (
    String username )
```

Sets username.

**Parameters**

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

**3.57.3.48 getGameSettings()**

```
GameSettings getGameSettings ( )
```

Gets game settings.

**Returns**

the game settings

**3.57.3.49 setGameSettings()**

```
void setGameSettings (
    GameSettings gameSettings )
```

Sets game settings.

**Parameters**

<i>gameSettings</i>	the game settings
---------------------	-------------------

**3.57.3.50 getPreMoves()**

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

Gets pre moves.

**Returns**

the pre moves

**3.57.3.51 setPreMoves()**

```
void setPreMoves (
    ArrayList< Move > preMoves )
```

Sets pre moves.

**Parameters**

<i>preMoves</i>	the pre moves
-----------------	---------------

**3.57.3.52 getSubject()**

```
String getSubject ( )
```

Gets subject.

**Returns**

the subject

**3.57.3.53 getPlayerColor()**

```
PlayerColor getPlayerColor ( )
```

Gets player color.

**Returns**

the player color

**3.57.3.54 setPlayerColor()**

```
void setPlayerColor (
    PlayerColor playerColor )
```

Sets player color.

**Parameters**

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

**3.57.3.55 getMove()**

```
Move getMove ( )
```

Gets move.

**Returns**

the move

**3.57.3.56 setMove()**

```
void setMove (
    Move move )
```

Sets move.

**Parameters**

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

**3.57.3.57 getGameTime()**

```
GameTime getGameTime ( )
```

Gets game time.

**Returns**

the game time

**3.57.3.58 setGameTime()**

```
void setGameTime (
    GameTime gameTime )
```

Sets game time.

## Parameters

<i>gameTime</i>	the game time
-----------------	---------------

**3.57.3.59   getGameStatus()**

```
GameStatus getGameStatus ( )
```

Gets game status.

## Returns

the game status

**3.57.3.60   setGameStatus()**

```
void setGameStatus (
    GameStatus gameStatus )
```

Sets game status.

## Parameters

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

**3.57.3.61   getPossibleMoves()**

```
MovesList getPossibleMoves ( )
```

Gets possible moves.

## Returns

the possible moves

**3.57.3.62   setPossibleMoves()**

```
void setPossibleMoves (
    MovesList possibleMoves )
```

Sets possible moves.

**Parameters**

<i>possibleMoves</i>	the possible moves
----------------------	--------------------

**3.57.3.63 getBoard()**

```
Board getBoard ( )
```

Gets board.

**Returns**

the board

**3.57.3.64 setBoard()**

```
void setBoard (  
    Board board )
```

Sets board.

**Parameters**

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

**3.57.3.65 getLoginInfo()**

```
LoginInfo getLoginInfo ( )
```

Gets login info.

**Returns**

the login info

**3.57.3.66 setLoginInfo()**

```
void setLoginInfo (  
    LoginInfo loginInfo )
```

Sets login info.



## Parameters

<i>loginInfo</i>	the login info
------------------	----------------

**3.57.3.67 getMessageType()**

```
MessageType getMessageType ( )
```

Gets message type.

## Returns

the message type

**3.57.3.68 isSubject()**

```
boolean isSubject ( )
```

Is subject boolean.

## Returns

the boolean

**3.57.4 Member Data Documentation****3.57.4.1 messageId**

```
final String messageId
```

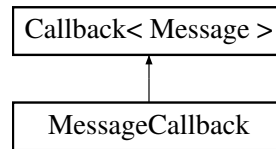
The [Message](#) id.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Networking/Messages/Message.java

## 3.58 MessageCallback

Inheritance diagram for MessageCallback:



### 3.58.1 Detailed Description

represents a message callback.

Author

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

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/Callbacks/MessageCallback.java

## 3.59 MessagesHandler

### Protected Attributes

- final [AppSocket](#) socket

### 3.59.1 Detailed Description

Messages handler - handles all types of messages by using a hash map to make all routing as fast as possible. when a request is sent to the server, a callback is passed with it. when a response is received, that callback is called with the response message.

if a message that isn't a response is received, the handling is deferred to the default callbacks map. which is where most of the implementation of this abstract class comes in. the individual message type handling

Author

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

### 3.59.2 Constructor & Destructor Documentation

#### 3.59.2.1 MessagesHandler()

```
MessagesHandler (
    AppSocket socket )
```

Instantiates a new Messages handler.

## Parameters

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

### 3.59.3 Member Function Documentation

#### 3.59.3.1 onCancelQuestion()

`MessageCallback` onCancelQuestion ( )

On cancel question message callback.

## Returns

the message callback

#### 3.59.3.2 interruptBlocking()

```
void interruptBlocking (
    MyError err )
```

## Parameters

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

#### 3.59.3.3 blockTilRes()

```
Message blockTilRes (
    Message request )
```

Blocking this thread until a response is received

## Parameters

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

## Returns

the message

### 3.59.3.4 noBlockRequest()

```
void noBlockRequest (
    Message request,
    MessageCallback onRes )
```

No block request.

#### Parameters

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

### 3.59.3.5 receivedMessage()

```
void receivedMessage (
    Message message )
```

Received message.

#### Parameters

<i>message</i>	the message
----------------	-------------

### 3.59.3.6 prepareForDisconnect()

```
void prepareForDisconnect ( )
```

Prepare for disconnect.

### 3.59.3.7 onAnyMsg()

```
void onAnyMsg (
    Message message )
```

On any msg.

#### Parameters

<i>message</i>	the message
----------------	-------------

**3.59.3.8 onDisconnected()**

```
final void onDisconnected ( )
```

On disconnected.

**3.59.3.9 onAnyDisconnection()**

```
void onAnyDisconnection ( ) [protected]
```

On any disconnection.

**3.59.3.10 onPlannedDisconnect()**

```
void onPlannedDisconnect ( ) [protected]
```

On planned disconnect.

**3.59.3.11 onUnplannedDisconnect()**

```
void onUnplannedDisconnect ( ) [protected]
```

On unplanned disconnect.

**3.59.3.12 createDisconnectedError()**

```
MyError.DisconnectedError createDisconnectedError ( ) [protected]
```

Create disconnected error my error . disconnected error.

**Returns**

the my error . disconnected error

**3.59.3.13 onLogin()**

```
MessageCallback onLogin ( )
```

On login message callback.

**Returns**

the message callback

**3.59.3.14 onResign()**

`MessageCallback onResign ( )`

On resign message callback.

**Returns**

the message callback

**3.59.3.15 onAddTime()**

`MessageCallback onAddTime ( )`

On add time message callback.

**Returns**

the message callback

**3.59.3.16 onOfferDraw()**

`MessageCallback onOfferDraw ( )`

On offer draw message callback.

**Returns**

the message callback

**3.59.3.17 onWelcomeMessage()**

`MessageCallback onWelcomeMessage ( )`

On welcome message message callback.

**Returns**

the message callback

**3.59.3.18 onGetGameSettings()**

`MessageCallback onGetGameSettings ( )`

On get game settings message callback.

**Returns**

the message callback

**3.59.3.19 onWaitForMatch()**

`MessageCallback onWaitForMatch ( )`

On wait for match message callback.

**Returns**

the message callback

**3.59.3.20 onInitGame()**

`MessageCallback onInitGame ( )`

On init game message callback.

**Returns**

the message callback

**3.59.3.21 onWaitTurn()**

`MessageCallback onWaitTurn ( )`

On wait turn message callback.

**Returns**

the message callback

### 3.59.3.22 onGetMove()

`MessageCallback onGetMove ( )`

On get move message callback.

#### Returns

the message callback

### 3.59.3.23 onUpdateByMove()

`MessageCallback onUpdateByMove ( )`

On update by move message callback.

#### Returns

the message callback

### 3.59.3.24 onGameOver()

`MessageCallback onGameOver ( )`

On game over message callback.

#### Returns

the message callback

### 3.59.3.25 onError()

`MessageCallback onError ( )`

On error message callback.

#### Returns

the message callback



**3.59.3.26 onQuestion()**

`MessageCallback onQuestion ( )`

On question message callback.

**Returns**

the message callback

**3.59.3.27 onBye()**

`MessageCallback onBye ( )`

On bye message callback.

**Returns**

the message callback

**3.59.3.28 onUsernameAvailability()**

`MessageCallback onUsernameAvailability ( )`

On username availability message callback.

**Returns**

the message callback

**3.59.3.29 onDBRequest()**

`MessageCallback onDBRequest ( )`

On db request message callback.

**Returns**

the message callback

### 3.59.3.30 onDBResponse()

`MessageCallback onDBResponse ( )`

On db response message callback.

#### Returns

the message callback

### 3.59.3.31 onUpdateSyncedList()

`MessageCallback onUpdateSyncedList ( )`

On update synced list message callback.

#### Returns

the message callback

### 3.59.3.32 onInterrupt()

`MessageCallback onInterrupt ( )`

On interrupt message callback.

#### Returns

the message callback

### 3.59.3.33 onIsAlive()

`MessageCallback onIsAlive ( )`

On is alive message callback.

#### Returns

the message callback

### 3.59.3.34 onAlive()

`MessageCallback onAlive ( )`

On alive message callback.

#### Returns

the message callback

## 3.59.4 Member Data Documentation

### 3.59.4.1 socket

`final AppSocket socket [protected]`

The Socket.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Networking/MessagesHandler.java

## 3.60 MessageType Enum Reference

### Public Attributes

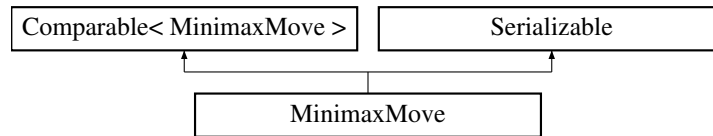
- LOGIN
- RESIGN
- ADD\_TIME
- WELCOME\_MESSAGE
- GET\_GAME\_SETTINGS
- WAIT\_FOR\_MATCH
- INIT\_GAME =(true)
- WAIT\_TURN =(true)
- GET\_MOVE =(true)
- THROW\_ERROR
- UPDATE\_BY\_MOVE =(true)
- GAME\_OVER
- ERROR
- QUESTION
- BYE =(true)
- USERNAME\_AVAILABILITY
- DB\_REQUEST
- DB\_RESPONSE
- UPDATE\_SYNCED\_LIST
- CANCEL\_QUESTION
- INTERRUPT
- final boolean **chronologicalImportance**

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Networking/Messages/MessageType.java

## 3.61 MinimaxMove

Inheritance diagram for MinimaxMove:



### 3.61.1 Detailed Description

Minimax move.

Author

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

### 3.61.2 Constructor & Destructor Documentation

#### 3.61.2.1 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

#### 3.61.2.2 MinimaxMove() [2/3]

```

MinimaxMove (
    Evaluation moveEvaluation )
  
```

Instantiates a new Minimax move.

Parameters

<i>moveEvaluation</i>	the move evaluation
-----------------------	---------------------

### 3.61.2.3 MinimaxMove() [3/3]

```
MinimaxMove (
    MinimaxMove other )
```

Instantiates a new Minimax move.

#### Parameters

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

## 3.61.3 Member Function Documentation

### 3.61.3.1 isDeeperAndBetterThan()

```
boolean isDeeperAndBetterThan (
    MinimaxMove other )
```

Is deeper and better than given minimax move.

#### Parameters

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

#### Returns

the boolean

### 3.61.3.2 getMoveDepth()

```
int getMoveDepth ( )
```

Gets move depth.

#### Returns

the move depth

### 3.61.3.3 `getMove()`

```
Move getMove ( )
```

Gets move.

#### Returns

the move

### 3.61.3.4 `setMove()`

```
void setMove (
    Move move )
```

Sets move.

#### Parameters

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

### 3.61.3.5 `getMoveEvaluation()`

```
Evaluation getMoveEvaluation ( )
```

Gets move evaluation.

#### Returns

the move evaluation

### 3.61.3.6 `setMoveEvaluation()`

```
void setMoveEvaluation (
    Evaluation moveEvaluation )
```

Sets move evaluation.

#### Parameters

<i>moveEvaluation</i>	the move evaluation
-----------------------	---------------------

### 3.61.3.7 hashCode()

```
int hashCode ( )
```

Hash code int.

#### Returns

the int

### 3.61.3.8 equals()

```
boolean equals (
    Object o )
```

Equals boolean.

#### Parameters

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

#### Returns

the boolean

### 3.61.3.9 toString()

```
String toString ( )
```

To string string.

#### Returns

the string

### 3.61.3.10 getShortPrintingStr()

```
String getShortPrintingStr ( )
```

Gets short printing str.

#### Returns

the short printing str

### 3.61.3.11 compareTo()

```
int compareTo (
    MinimaxMove o )
```

Compare to int.

#### Parameters

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

#### Returns

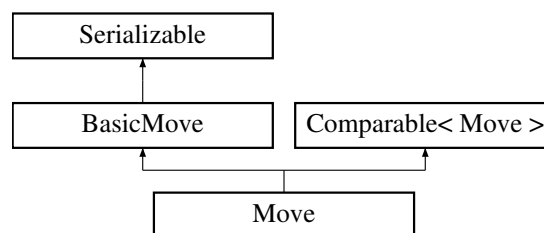
the int

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/Moves/MinimaxMove.java

## 3.62 Move

Inheritance diagram for Move:



### Classes

- enum [MoveFlag](#)
- enum [ThreefoldStatus](#)

### 3.62.1 Detailed Description

[Move](#) - represents a "heavy" move. with a lot of info.

#### Author

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

### 3.62.2 Class Documentation

#### 3.62.2.1 enum ver14::SharedClasses::Game::Moves::Move::ThreefoldStatus

Threefold status - represents a threefold draw status.

#### Author

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



## Enumerator

NONE	None threefold status.
CAN_CLAIM	Can claim threefold status.
CLAIMED	Claimed threefold status.

## 3.62.3 Constructor &amp; Destructor Documentation

## 3.62.3.1 Move() [1/4]

```
Move (
    Location movingFrom,
    Location movingTo,
    PieceType capturingPieceType )
```

Instantiates a new [Move](#).

## Parameters

<i>movingFrom</i>	the moving from
<i>movingTo</i>	the moving to
<i>capturingPieceType</i>	the capturing piece type

## 3.62.3.2 Move() [2/4]

```
Move (
    Location movingFrom,
    Location movingTo )
```

Instantiates a new [Move](#).

## Parameters

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

## 3.62.3.3 Move() [3/4]

```
Move (
    ThreefoldStatus threefoldStatus )
```

Instantiates a new [Move](#).

## Parameters

<i>threefoldStatus</i>	the threefold status
------------------------	----------------------

**3.62.3.4 Move()** [4/4]

```
Move (
    Move other )
```

Instantiates a new [Move](#).

## Parameters

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

**3.62.4 Member Function Documentation****3.62.4.1 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

**3.62.4.2 threefoldClaim()**

```
static Move threefoldClaim ( ) [static]
```

Threefold claim move.

**Returns**

the move

**3.62.4.3 flipMove()**

```
static Move flipMove (
    Move move ) [static]
```

Flip move move.

**Parameters**

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

**Returns**

the move

**3.62.4.4 copyMove()**

```
static Move copyMove (
    Move move ) [static]
```

Copy move.

**Parameters**

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

**Returns**

the move

**3.62.4.5 setMoveAnnotation()**

```
void setMoveAnnotation (
    String moveAnnotation )
```

Sets move annotation.

**Parameters**

<i>moveAnnotation</i>	the move annotation
-----------------------	---------------------

**3.62.4.6 getDisabledCastling()**

```
byte getDisabledCastling ( )
```

Gets disabled castling.

**Returns**

the disabled castling

**3.62.4.7 setDisabledCastling()**

```
void setDisabledCastling (
    byte disabledCastling )
```

Sets disabled castling.

**Parameters**

<i>disabledCastling</i>	the disabled castling
-------------------------	-----------------------

**3.62.4.8 setThreefoldOption()**

```
void setThreefoldOption ( )
```

Sets threefold option.

**3.62.4.9 getMovingColor()**

```
PlayerColor getMovingColor ( )
```

Gets moving color.

**Returns**

the moving color

**3.62.4.10 setMovingColor()**

```
void setMovingColor (
    PlayerColor movingPlayerColor )
```

Sets moving color.

**Parameters**

<i>movingPlayerColor</i>	the moving player color
--------------------------	-------------------------

**3.62.4.11 getPrevFullMoveClock()**

```
int getPrevFullMoveClock ( )
```

Gets prev full move clock.

**Returns**

the prev full move clock

**3.62.4.12 setPrevFullMoveClock()**

```
void setPrevFullMoveClock (
    int prevFullMoveClock )
```

Sets prev full move clock.

**Parameters**

<i>prevFullMoveClock</i>	the prev full move clock
--------------------------	--------------------------

**3.62.4.13 getPrevHalfMoveClock()**

```
int getPrevHalfMoveClock ( )
```

Gets prev half move clock.

**Returns**

the prev half move clock

#### 3.62.4.14 setPrevHalfMoveClock()

```
void setPrevHalfMoveClock (
    int prevHalfMoveClock )
```

Sets prev half move clock.

##### Parameters

<i>prevHalfMoveClock</i>	the prev half move clock
--------------------------	--------------------------

#### 3.62.4.15 getPromotingTo()

```
PieceType getPromotingTo ( )
```

Gets promoting to.

##### Returns

the promoting to

#### 3.62.4.16 setPromotingTo()

```
void setPromotingTo (
    PieceType promotingTo )
```

Sets promoting to.

##### Parameters

<i>promotingTo</i>	the promoting to
--------------------	------------------

#### 3.62.4.17 getIntermediateMove()

```
BasicMove getIntermediateMove ( )
```

Gets intermediate move.

##### Returns

the intermediate move

#### 3.62.4.18 setIntermediateMove()

```
void setIntermediateMove (
    BasicMove intermediateMove )
```

Sets intermediate move.

##### Parameters

<i>intermediateMove</i>	the intermediate move
-------------------------	-----------------------

#### 3.62.4.19 isCheck()

```
boolean isCheck ( )
```

Is check boolean.

##### Returns

the boolean

#### 3.62.4.20 getMoveEvaluation()

```
Evaluation getMoveEvaluation ( )
```

Gets move evaluation.

##### Returns

the move evaluation

#### 3.62.4.21 setMoveEvaluation()

```
void setMoveEvaluation (
    Evaluation moveEvaluation )
```

Sets move evaluation.

##### Parameters

<i>moveEvaluation</i>	the move evaluation
-----------------------	---------------------

#### 3.62.4.22 `getCapturingPieceType()`

```
PieceType getCapturingPieceType ( )
```

Gets capturing piece type.

##### Returns

the capturing piece type

#### 3.62.4.23 `isReversible()`

```
boolean isReversible ( )
```

Is reversible boolean.

##### Returns

the boolean

#### 3.62.4.24 `setReversible()`

```
void setReversible (
    boolean reversible )
```

Sets reversible.

##### Parameters

<i>reversible</i>	the reversible
-------------------	----------------

#### 3.62.4.25 `isCapturing()`

```
boolean isCapturing ( )
```

Is capturing boolean.

##### Returns

the boolean



### 3.62.4.26 setCapturing()

```
void setCapturing (
    PieceType pieceType )
```

Sets capturing.

#### Parameters

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

### 3.62.4.27 equals()

```
boolean equals (
    Object o )
```

Equals boolean.

#### Parameters

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

#### Returns

the boolean

Reimplemented from [BasicMove](#).

### 3.62.4.28 toString()

```
String toString ( )
```

To string string.

#### Returns

the string

Reimplemented from [BasicMove](#).

#### 3.62.4.29 `getAnnotation()`

```
String getAnnotation ( )
```

Gets annotation.

##### Returns

the annotation

#### 3.62.4.30 `strictEquals()`

```
boolean strictEquals (
    Move move )
```

checks source and destination equals and move flags equals.

##### Parameters

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

##### Returns

the boolean

#### 3.62.4.31 `getMoveFlag()`

```
MoveFlag getMoveFlag ( )
```

Gets move flag.

##### Returns

the move flag

#### 3.62.4.32 `setMoveFlag()`

```
void setMoveFlag (
    MoveFlag moveFlag )
```

Sets move flag.

## Parameters

<i>moveFlag</i>	the move flag
-----------------	---------------

**3.62.4.33 compareTo()**

```
int compareTo (
    Move o )
```

Compare to another move.

## Parameters

<i>o</i>	the other move
----------	----------------

## Returns

the comparison result

**3.62.4.34 getEnPassantLoc()**

```
Location getEnPassantLoc ( )
```

Gets en passant loc.

## Returns

the en passant loc

**3.62.4.35 setEnPassantLoc()**

```
void setEnPassantLoc (
    Location epsnLoc )
```

Sets en passant loc.

## Parameters

<i>epsnLoc</i>	the epsn loc
----------------	--------------

The documentation for this class was generated from the following file:

- `ver14/SharedClasses/Game/Moves/Move.java`

## 3.63 MoveAnnotation

### 3.63.1 Detailed Description

`Move` annotation - utility class that annotates moves.

#### Author

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

### 3.63.2 Member Function Documentation

#### 3.63.2.1 `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

#### 3.63.2.2 `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

**3.63.2.3 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

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/Moves/MoveAnnotation.java

## 3.64 Move.MoveFlag Enum Reference

**Public Attributes**

- [NormalMove](#)
- [EnPassant](#)
- [DoublePawnPush](#)
- [Promotion](#)
- [ShortCastle](#) =(CastlingRights.Side.KING)
- [LongCastle](#) =(CastlingRights.Side.QUEEN)
- final boolean [isCastling](#)
- final CastlingRights.Side [castlingSide](#)

**Static Public Attributes**

- static final [MoveFlag](#)[] [CASTLING\\_FLAGS](#)

**3.64.1 Detailed Description**

[Move](#) flag - which type of move this is.

**Author**

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

## 3.64.2 Constructor & Destructor Documentation

### 3.64.2.1 MoveFlag() [1/2]

```
MoveFlag ( )
```

Instantiates a new [Move](#) flag.

### 3.64.2.2 MoveFlag() [2/2]

```
MoveFlag (
    CastlingRights.Side side )
```

Instantiates a new [Move](#) flag.

#### Parameters

<i>side</i>	the side
-------------	----------

## 3.64.3 Member Function Documentation

### 3.64.3.1 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

## 3.64.4 Member Data Documentation

#### 3.64.4.1 NormalMove

NormalMove

Normal move move flag.

#### 3.64.4.2 EnPassant

EnPassant

En passant move flag.

#### 3.64.4.3 DoublePawnPush

DoublePawnPush

Double pawn push move flag.

#### 3.64.4.4 Promotion

Promotion

Initial value:

```
={
    @Override
    public boolean equals(Move myMove, Move otherMove) {
        return super.equals(myMove, otherMove) && myMove.getPromotingTo() ==
            otherMove.getPromotingTo();
    }
}
```

The Promotion.

#### 3.64.4.5 ShortCastle

ShortCastle =(CastlingRights.Side.KING)

Short castle move flag.

#### 3.64.4.6 LongCastle

LongCastle =(CastlingRights.Side.QUEEN)

Long castle move flag.

#### 3.64.4.7 CASTLING\_FLAGS

```
final MoveFlag [] CASTLING_FLAGS [static]
```

The Castling flags.

#### 3.64.4.8 isCastling

```
final boolean isCastling
```

Is this move flag a castling flag.

#### 3.64.4.9 castlingSide

```
final CastlingRights.Side castlingSide
```

if this is a castling flag, which side is it.

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Game/Moves/Move.java

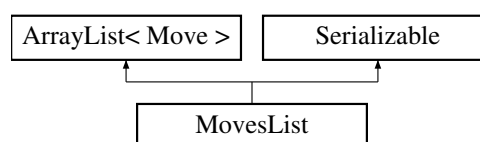
### 3.65 MoveFlags Enum Reference

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Game/Moves/MoveFlags.java

### 3.66 MovesList

Inheritance diagram for MovesList:



### Classes

- interface [CompareMoves](#)

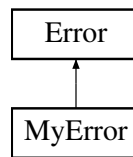
The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/Moves/MovesList.java



## 3.67 MyError

Inheritance diagram for MyError:



### Classes

- class **DBErr**
- class **DisconnectedError**

### 3.67.1 Detailed Description

My error - .

Author

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

### 3.67.2 Constructor & Destructor Documentation

#### 3.67.2.1 MyError() [1/4]

```
MyError ( )
```

Instantiates a new My error.

#### 3.67.2.2 MyError() [2/4]

```
MyError (
    Throwable throwable )
```

Instantiates a new My error.

Parameters

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

### 3.67.2.3 MyError() [3/4]

```
MyError (
    String message )
```

Instantiates a new My error.

#### Parameters

<i>message</i>	the message
----------------	-------------

### 3.67.2.4 MyError() [4/4]

```
MyError (
    String message,
    Throwable cause )
```

Instantiates a new My error.

#### Parameters

<i>message</i>	the message
<i>cause</i>	the cause

## 3.67.3 Member Function Documentation

### 3.67.3.1 getHandledStr()

```
String getHandledStr ( )
```

Gets handled str.

#### Returns

the handled str

### 3.67.3.2 toString()

```
String toString ( )
```

To string string.

#### Returns

the string

### 3.67.3.3 errToString()

```
static String errToString (
    Throwable error ) [static]
```

Err to string string.

#### Parameters

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

#### Returns

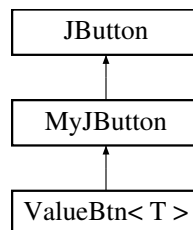
the string

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Threads/ErrorHandler/MyError.java

## 3.68 MyJButton

Inheritance diagram for MyJButton:



### 3.68.1 Detailed Description

MyJButton- represents a button.

#### Author

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

### 3.68.2 Constructor & Destructor Documentation

#### 3.68.2.1 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

**3.68.2.2 MyJButton()** [2/6]

```
MyJButton (
    String text )
```

Instantiates a new My j button.

## Parameters

<i>text</i>	the text
-------------	----------

**3.68.2.3 MyJButton()** [3/6]

```
MyJButton ( )
```

Instantiates a new My j button.

**3.68.2.4 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

**3.68.2.5 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

### 3.68.2.6 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

## 3.68.3 Member Function Documentation

### 3.68.3.1 setOnClick()

```
void setOnClick (  
    VoidCallback onClick )
```

Sets on click.

#### Parameters

<i>onClick</i>	the on click
----------------	--------------

### 3.68.3.2 setText()

```
void setText (  
    String text )
```

Sets text.

#### Parameters

<i>text</i>	the text
-------------	----------

### 3.68.3.3 setFont()

```
void setFont (
    Font font )
```

Sets font.

#### Parameters

<i>font</i>	the font
-------------	----------

### 3.68.3.4 getMinSize()

```
int getMinSize ( )
```

Gets min size.

#### Returns

the min size

### 3.68.3.5 replaceWithCancel()

```
void replaceWithCancel (
    VoidCallback onCancelled )
```

Replace with cancel.

#### Parameters

<i>onCancelled</i>	the on cancelled
--------------------	------------------

**3.68.3.6 resetState()**

```
void resetState (
    boolean e )
```

Reset state.

**Parameters**

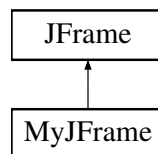
<i>e</i>	the e
----------	-------

The documentation for this class was generated from the following file:

- ver14/SharedClasses/UI/Buttons/MyJButton.java

**3.69 MyJFrame**

Inheritance diagram for MyJFrame:

**Classes**

- interface [BooleanClosing](#)
- interface [Closing](#)
- class **MyAdapter**
- interface [StringClosing](#)

**Protected Attributes**

- [Closing](#)<?> [onClose](#) = null

**3.69.1 Detailed Description**

The type My j frame.

**3.69.2 Constructor & Destructor Documentation****3.69.2.1 MyJFrame()**

```
MyJFrame ( ) throws HeadlessException
```

Instantiates a new My j frame.

### Exceptions

<i>HeadlessException</i>	the headless exception
--------------------------	------------------------

## 3.69.3 Member Function Documentation

### 3.69.3.1 debugAdapter()

```
static MyAdapter debugAdapter (
    Window addTo ) [static]
```

Debug adapter my adapter.

#### Parameters

<i>addTo</i>	the add to
--------------	------------

#### Returns

the my adapter

### 3.69.3.2 getMyAdapter()

```
MyAdapter getMyAdapter ( )
```

Gets my adapter.

#### Returns

the my adapter

### 3.69.3.3 setOnExit()

```
void setOnExit (
    Closing<?> onClose )
```

Sets on exit.



## Parameters

<i>onClose</i>	the on close
----------------	--------------

**3.69.3.4 doXClick()**

```
void doXClick ( )
```

Do x click.

**3.69.3.5 setOnResize()**

```
void setOnResize (
    VoidCallback onResize )
```

Sets on resize.

## Parameters

<i>onResize</i>	the on resize
-----------------	---------------

**3.69.4 Member Data Documentation****3.69.4.1 onClose**

```
Closing<?> onClose = null [protected]
```

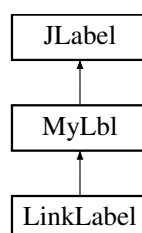
The On close.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/UI/MyJFrame.java

**3.70 MyLbl**

Inheritance diagram for MyLbl:



## Classes

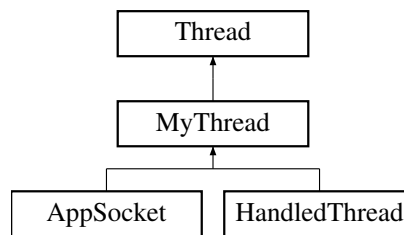
- interface [StringModifier](#)

The documentation for this class was generated from the following file:

- ver14/SharedClasses/UI/MyLbl.java

## 3.71 MyThread

Inheritance diagram for MyThread:



## Classes

- enum [ThreadStatus](#)

### 3.71.1 Detailed Description

The type My thread.

### 3.71.2 Constructor & Destructor Documentation

#### 3.71.2.1 MyThread()

`MyThread ( )`

Instantiates a new My thread.

### 3.71.3 Member Function Documentation

#### 3.71.3.1 currentThread()

```
static void currentThread (
    Callback< MyThread > run ) [static]
```

Current thread. will only execute code if inside a [MyThread](#)

**Parameters**

<i>run</i>	the run
------------	---------

**3.71.3.2 stopRun()**

```
void stopRun ( )
```

Stop run.

**3.71.3.3 ignoreErrs()**

```
void ignoreErrs ( )
```

Ignore errs.

**3.71.3.4 reactivateErrs()**

```
void reactivateErrs ( )
```

Reactivate errs.

**3.71.3.5 handledRun()**

```
abstract void handledRun ( ) throws Throwable [abstract], [protected]
```

Handled run.

**Exceptions**

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

Reimplemented in [AppSocket](#), and [HandledThread](#).

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Threads/MyThread.java

## 3.72 Piece Enum Reference

**Public Attributes**

- [W\\_P](#)=([PieceType.PAWN](#), [PlayerColor.WHITE](#))

- `W_R`=(`PieceType.ROOK`, `PlayerColor.WHITE`)
- `W_B`=(`PieceType.BISHOP`, `PlayerColor.WHITE`)
- `W_N`=(`PieceType.KNIGHT`, `PlayerColor.WHITE`)
- `W_Q`=(`PieceType.QUEEN`, `PlayerColor.WHITE`)
- `W_K`=(`PieceType.KING`, `PlayerColor.WHITE`)
- `B_P`=(`PieceType.PAWN`, `PlayerColor.BLACK`)
- `B_R`=(`PieceType.ROOK`, `PlayerColor.BLACK`)
- `B_B`=(`PieceType.BISHOP`, `PlayerColor.BLACK`)
- `B_N`=(`PieceType.KNIGHT`, `PlayerColor.BLACK`)
- `B_Q`=(`PieceType.QUEEN`, `PlayerColor.BLACK`)
- `B_K`=(`PieceType.KING`, `PlayerColor.BLACK`)
- final `PieceType` `pieceType`
- final `PlayerColor` `playerColor`

## Static Public Attributes

- static final `Piece[]` `ALL_PIECES` = `values()`

### 3.72.1 Detailed Description

`Piece` - represents a combination of a piece type, and a color.

#### Author

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

#### See also

[PieceType](#)

[PlayerColor](#)

### 3.72.2 Member Function Documentation

#### 3.72.2.1 `getPieceFromFen()`

```
static Piece getPieceFromFen (
    char c ) [static]
```

Gets piece from fen.

#### Parameters

<code>c</code>	the c
----------------	-------

**Returns**

the piece from fen

**3.72.2.2 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

**3.72.2.3 getPiecelcon() [1/2]**

```
String getPiecelcon ( )
```

Gets piece icon.

**Returns**

the piece icon

**3.72.2.4 getPiecelcon() [2/2]**

```
static String getPiecelcon (
    Piece piece ) [static]
```

Gets piece icon.

**Parameters**

<i>piece</i>	the piece
--------------	-----------

**Returns**

the piece icon

**3.72.2.5 getFen()**

```
String getFen ( )
```

Gets fen.

**Returns**

the fen

**3.72.2.6 isOnMyTeam() [1/2]**

```
boolean isOnMyTeam (
    Piece otherPiece )
```

Is on my team boolean.

**Parameters**

<i>otherPiece</i>	the other piece
-------------------	-----------------

**Returns**

the boolean

**3.72.2.7 isOnMyTeam() [2/2]**

```
boolean isOnMyTeam (
    PlayerColor otherPlayerColor )
```

Is on my team boolean.

**Parameters**

<i>otherPlayerColor</i>	the other player color
-------------------------	------------------------

**Returns**

the boolean

**3.72.3 Member Data Documentation****3.72.3.1 W\_P**

```
W_P = (PieceType.PAWN, PlayerColor.WHITE)
```

represents a White p piece.

**3.72.3.2 W\_R**

```
W_R = (PieceType.ROOK, PlayerColor.WHITE)
```

represents a White Rook.

**3.72.3.3 W\_B**

```
W_B = (PieceType.BISHOP, PlayerColor.WHITE)
```

represents a White Bishop.

**3.72.3.4 W\_N**

```
W_N = (PieceType.KNIGHT, PlayerColor.WHITE)
```

represents a White Knight.

**3.72.3.5 W\_Q**

```
W_Q = (PieceType.QUEEN, PlayerColor.WHITE)
```

represents a White Queen.

**3.72.3.6 W\_K**

```
W_K = (PieceType.KING, PlayerColor.WHITE)
```

represents a White King.

### 3.72.3.7 B\_P

```
B_P = (PieceType.PAWN, PlayerColor.BLACK)
```

represents a Black p piece.

### 3.72.3.8 B\_R

```
B_R = (PieceType.ROOK, PlayerColor.BLACK)
```

represents a Black Rook.

### 3.72.3.9 B\_B

```
B_B = (PieceType.BISHOP, PlayerColor.BLACK)
```

represents a Black Bishop.

### 3.72.3.10 B\_N

```
B_N = (PieceType.KNIGHT, PlayerColor.BLACK)
```

represents a Black Knight.

### 3.72.3.11 B\_Q

```
B_Q = (PieceType.QUEEN, PlayerColor.BLACK)
```

represents a Black Queen.

### 3.72.3.12 B\_K

```
B_K = (PieceType.KING, PlayerColor.BLACK)
```

represents a Black King.

### 3.72.3.13 ALL\_PIECES

```
final Piece [] ALL_PIECES = values() [static]
```

The constant ALL\_PIECES.

### 3.72.3.14 pieceType

```
final PieceType pieceType
```

The [Piece](#) type.



### 3.72.3.15 playerColor

```
final PlayerColor playerColor
```

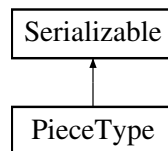
The Player color.

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Game/GameSetup/BoardSetup/Pieces/Piece.java

## 3.73 PieceType Enum Reference

Inheritance diagram for PieceType:



### Public Attributes

- PAWN
- ROOK =("", "", 500)
- BISHOP =("", "", 320)
- KNIGHT
- QUEEN =("", "", 900)
- 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 = 6
- static final PieceType[] PIECE\_TYPES = new PieceType[NUM\_OF\_PIECE\_TYPES]
- static final PieceType[] UNIQUE\_MOVES\_PIECE\_TYPES = {ROOK, KNIGHT, BISHOP, PAWN, KING}
- static final PieceType[] MINOR\_PIECES = {BISHOP, KNIGHT}
- static final PieceType[] MAJOR\_PIECES = {QUEEN, ROOK}
- static final PieceType[] CAN\_PROMOTE\_TO = {KNIGHT, ROOK, BISHOP, QUEEN}
- static final PieceType[] ATTACKING\_PIECE\_TYPES = {ROOK, BISHOP, KNIGHT, PAWN, QUEEN, KING}
- static final Direction[][] ATTACKING\_DIRECTIONS = new Direction[NUM\_OF\_PIECE\_TYPES][ ]

### 3.73.1 Detailed Description

`Piece` type - represents the `Piece` Type.

#### Author

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

### 3.73.2 Member Function Documentation

#### 3.73.2.1 `getPieceType()`

```
static PieceType getPieceType (
    int pieceType ) [static]
```

Gets piece type.

#### Parameters

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

#### Returns

the piece type

#### 3.73.2.2 `getAttackingDirections()` [1/2]

```
static Direction[] getAttackingDirections (
    PieceType pieceType ) [static]
```

Get attacking directions direction [ ].

#### Parameters

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

#### Returns

the direction [ ]

### 3.73.2.3 getWhitePieceFen()

```
String getWhitePieceFen ( )
```

Gets white piece fen.

#### Returns

the white piece fen

### 3.73.2.4 getPiecelcon()

```
String getPiecelcon (
    PlayerColor playerColor )
```

Gets piece icon.

#### Parameters

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

#### Returns

the piece icon

### 3.73.2.5 compareMovementType() [1/2]

```
boolean compareMovementType (
    PieceType compareTo )
```

Compare movement type boolean.

#### Parameters

<i>compareTo</i>	the compare to
------------------	----------------

#### Returns

the boolean

### 3.73.2.6 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

### 3.73.2.7 isDiagonalPiece()

```
static boolean isDiagonalPiece (  
    PieceType pieceType ) [static]
```

Is diagonal piece boolean.

#### Parameters

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

#### Returns

the boolean

### 3.73.2.8 isLinePiece()

```
static boolean isLinePiece (  
    PieceType pieceType ) [static]
```

Is line piece boolean.

#### Parameters

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

**Returns**

the boolean

**3.73.2.9 getPieceName()**

```
String getPieceName ( )
```

Gets piece name.

**Returns**

the piece name

**3.73.2.10 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

**3.73.2.11 getAttackingDirections() [2/2]**

```
Direction[] getAttackingDirections ( )
```

Get attacking directions direction [ ].

**Returns**

the direction [ ]

### 3.73.2.12 getWalkingDirections()

```
Direction[] getWalkingDirections ( )
```

Get walking directions direction [ ].

Returns

the direction [ ]

## 3.73.3 Member Data Documentation

### 3.73.3.1 PAWN

PAWN

**Initial value:**

```
=("", "", 100, false) {
    @Override
    public boolean isAttack(Direction direction, int maxDistance) {
        return maxDistance == 1 && super.isAttack(direction, maxDistance);
    }
    @Override
    public Direction[] getWalkingDirections() {
        return ArrUtils.concat(super.getWalkingDirections(), Direction.U, Direction.U_U);
    }
}
```

Pawn [Piece](#) Type.

### 3.73.3.2 ROOK

ROOK =("", "", 500)

Rook piece type.

### 3.73.3.3 BISHOP

BISHOP =("", "", 320)

Bishop piece type.

### 3.73.3.4 KNIGHT

KNIGHT

**Initial value:**

```
=("", "", 310, false) {
    @Override
    public boolean isAttack(Direction direction, int maxDistance) {
        return maxDistance == 2 && super.isAttack(direction, maxDistance);
    }
}
```

The Knight.

### 3.73.3.5 QUEEN

```
QUEEN = ("", "", 900)
```

Queen piece type.

### 3.73.3.6 KING

```
KING
```

**Initial value:**

```
=(" ", " ", 200000, false) {  
    @Override  
    public boolean isAttack(Direction direction, int maxDistance) {  
        return maxDistance == 1 && super.isAttack(direction, maxDistance);  
    }  
}
```

The King.

### 3.73.3.7 NUM\_OF\_PIECE\_TYPES

```
final int NUM_OF_PIECE_TYPES = 6 [static]
```

The constant NUM\_OF\_PIECE\_TYPES.

### 3.73.3.8 PIECE\_TYPES

```
final PieceType [ ] PIECE_TYPES = new PieceType[NUM_OF_PIECE_TYPES] [static]
```

The constant PIECE\_TYPES.

### 3.73.3.9 UNIQUE\_MOVES\_PIECE\_TYPES

```
final PieceType [ ] UNIQUE_MOVES_PIECE_TYPES = {ROOK, KNIGHT, BISHOP, PAWN, KING} [static]
```

The Unique moves piece types.

### 3.73.3.10 MINOR\_PIECES

```
final PieceType [ ] MINOR_PIECES = {BISHOP, KNIGHT} [static]
```

The Minor pieces.

### 3.73.3.11 MAJOR\_PIECES

```
final PieceType [ ] MAJOR_PIECES = {QUEEN, ROOK} [static]
```

The Major pieces.

### 3.73.3.12 CAN\_PROMOTE\_TO

```
final PieceType [] CAN_PROMOTE_TO = {KNIGHT, ROOK, BISHOP, QUEEN} [static]
```

The types of pieces a pawn Can promote to.

### 3.73.3.13 ATTACKING\_PIECE\_TYPES

```
final PieceType [] ATTACKING_PIECE_TYPES = {ROOK, BISHOP, KNIGHT, PAWN, QUEEN, KING} [static]
```

The constant ATTACKING\_PIECE\_TYPES.

### 3.73.3.14 ATTACKING\_DIRECTIONS

```
final Direction [][] ATTACKING_DIRECTIONS = new Direction[NUM_OF_PIECE_TYPES][] [static]
```

The Attacking directions. each piece type has its own set of attacking direction.

### 3.73.3.15 whiteIcon

```
final String whiteIcon
```

The White icon.

### 3.73.3.16 blackIcon

```
final String blackIcon
```

The Black icon.

### 3.73.3.17 value

```
final int value
```

The Value.

### 3.73.3.18 isSliding

```
final boolean isSliding
```

The Is sliding.



### 3.73.3.19 asInt

```
final int asInt
```

The As int.

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Game/GameSetup/BoardSetup/Pieces/PieceType.java

## 3.74 PlayerColor Enum Reference

### 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](#) = {[WHITE](#), [BLACK](#)}
- static final int [NUM\\_OF\\_PLAYERS](#) = 2

### 3.74.1 Detailed Description

Player color - represents a player color.

Author

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

### 3.74.2 Constructor & Destructor Documentation

#### 3.74.2.1 PlayerColor()

```
PlayerColor (
    int startingRow,
    int diff )
```

Instantiates a new Player color.

**Parameters**

<i>startingRow</i>	the starting row
<i>diff</i>	the diff

### 3.74.3 Member Function Documentation

#### 3.74.3.1 getColor()

```
static PlayerColor getColor (
    int clr ) [static]
```

Gets color.

**Parameters**

<i>clr</i>	the clr
------------	---------

**Returns**

the color

#### 3.74.3.2 getPlayerFromFen()

```
static PlayerColor getPlayerFromFen (
    String playerToMove ) [static]
```

Gets player from fen.

**Parameters**

<i>playerToMove</i>	the player to move
---------------------	--------------------

**Returns**

the player from fen

#### 3.74.3.3 getOpponent()

```
abstract PlayerColor getOpponent ( ) [abstract]
```

Gets opponent.

**Returns**

the opponent

**3.74.3.4 toString()**

```
String toString ( )
```

To string string.

**Returns**

the string

**3.74.3.5 getName()**

```
String getName ( )
```

Gets the players name.

**Returns**

the players name

**3.74.4 Member Data Documentation****3.74.4.1 WHITE**

```
WHITE
```

**Initial value:**

```
=(Location.WHITE_STARTING_ROW, Location.WHITE_DIFF) {  
    @Override  
    public PlayerColor getOpponent() {  
        return BLACK;  
    }  
}
```

White.

#### 3.74.4.2 BLACK

BLACK

##### Initial value:

```
=(Location.BLACK_STARTING_ROW, Location.BLACK_DIFF) {  
    @Override  
    public PlayerColor getOpponent() {  
        return WHITE;  
    }  
}
```

Black.

#### 3.74.4.3 NO\_PLAYER

NO\_PLAYER

##### Initial value:

```
=(-1, 0) {  
    @Override  
    public PlayerColor getOpponent() {  
        return NO_PLAYER;  
    }  
}
```

No player.

#### 3.74.4.4 PLAYER\_COLORS

```
final PlayerColor [] PLAYER_COLORS = {WHITE, BLACK} [static]
```

The constant PLAYER\_COLORS.

#### 3.74.4.5 NUM\_OF\_PLAYERS

```
final int NUM_OF_PLAYERS = 2 [static]
```

The constant NUM\_OF\_PLAYERS.

#### 3.74.4.6 asInt

```
final int asInt
```

The As int.

#### 3.74.4.7 indexOf2

```
final int indexOf2
```

index of two. calculated at initialization for performance. calculation: `asInt*2`

#### 3.74.4.8 startingRow

```
final int startingRow
```

The player's Starting row.

#### 3.74.4.9 diff

```
final int diff
```

The moving up ratio.

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Game/PlayerColor.java

## 3.75 PreMadeRequest

### Classes

- class **Variation**
- interface [VariationCreator](#)

### Public Attributes

- final int [authSettings](#)

### Static Public Attributes

- static final [PreMadeRequest](#) TopPlayers
- static final [PreMadeRequest](#) Games
- static final [PreMadeRequest](#) DeleteUnfGames = new [PreMadeRequest](#)([RequestBuilder::deleteAllUnFinishedGames](#), [AuthSettings.USER](#))
- static final [PreMadeRequest](#) StatsByTimeOfDay = new [PreMadeRequest](#)([RequestBuilder::statsByTimeOfDay](#), [AuthSettings.USER](#))
- static final [PreMadeRequest](#) ChangeProfilePic = new [PreMadeRequest](#)([RequestBuilder::changeProfilePic](#), [AuthSettings.USER](#))
- static final [PreMadeRequest](#)[] statistics = {TopPlayers, Games, StatsByTimeOfDay}

### 3.75.1 Detailed Description

Pre made request - .

#### Author

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

## 3.75.2 Member Function Documentation

### 3.75.2.1 getRequestVariations()

```
PreMadeRequest[] getRequestVariations ( )
```

Get request variations pre made request [ ].

Returns

the pre made request [ ]

### 3.75.2.2 createBuilder()

```
RequestBuilder createBuilder ( )
```

Create builder request builder.

Returns

the request builder

## 3.75.3 Member Data Documentation

### 3.75.3.1 TopPlayers

```
final PreMadeRequest TopPlayers [static]
```

Initial value:

```
= new PreMadeRequest(RequestBuilder::top,
    AuthSettings.ANY_LOGIN,
    builder ->
        new Variation("Top All Players", new Object[]{0}, new Arg[0]),
    builder -> new Variation("Top Five Players", new Object[]{5}, new Arg[0]))
```

The constant TopPlayers.

### 3.75.3.2 Games

```
final PreMadeRequest Games [static]
```

Initial value:

```
= new PreMadeRequest(RequestBuilder::games, AuthSettings.USER, builder -> {
    Arg un = builder.args[0];
    return new Variation("All Games", new Object[]{un.repInStr, new Date(0), (Supplier<Date>)
    Date::new}, new Arg[]{un});
}, builder -> {
    Arg un = builder.args[0];
    return new Variation("Games from last week", new Object[]{un.repInStr, (Supplier<Date>) () -> new
    Date(System.currentTimeMillis() - TimeUnit.DAYS.toMillis(7)), (Supplier<Date>) Date::new}, new
    Arg[]{un});
})
```

The constant Games.

### 3.75.3.3 DeleteUnfGames

```
final PreMadeRequest DeleteUnfGames = new PreMadeRequest(RequestBuilder::deleteAllUnFinishedGames,
AuthSettings.USER) [static]
```

The constant DeleteUnfGames.

### 3.75.3.4 StatsByTimeOfDay

```
final PreMadeRequest StatsByTimeOfDay = new PreMadeRequest(RequestBuilder::statsByTimeOfDay,
AuthSettings.USER) [static]
```

The constant StatsByTimeOfDay.

### 3.75.3.5 ChangeProfilePic

```
final PreMadeRequest ChangeProfilePic = new PreMadeRequest(RequestBuilder::changeProfilePic,
AuthSettings.USER) [static]
```

The constant ChangeProfilePic.

### 3.75.3.6 statistics

```
final PreMadeRequest [] statistics = {TopPlayers, Games, StatsByTimeOfDay} [static]
```

The Statistics.

### 3.75.3.7 authSettings

```
final int authSettings
```

The Auth settings.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/DBRequest/PreMadeRequest.java

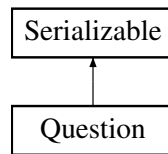
## 3.76 PremovesGenerator

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/Moves/PremovesGenerator.java

## 3.77 Question

Inheritance diagram for Question:



### Classes

- enum [QuestionType](#)

### Public Attributes

- final String [questionStr](#)
- final [QuestionType](#) [questionType](#)

### Static Public Attributes

- static final [Question Threefold](#) = new [Question](#)("Would you like to claim a Threefold repetition?", [QuestionType.THREEFOLD](#), [Answer.YES](#), [Answer.NO](#))
- static final [Question Rematch](#) = new [Question](#)("rematch", [QuestionType.REMATCH](#), [Answer.YES](#), [Answer.NO](#))

### 3.77.1 Detailed Description

[Question](#).

Author

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

### 3.77.2 Class Documentation

#### 3.77.2.1 enum ver14::SharedClasses::Misc::Question::QuestionType

[Question](#) type.

Author

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



## Enumerator

DRAW_OFFER	Draw offer question type.
THREEFOLD	Threefold question type.
REMATCH	Rematch question type.
NO_TYPE	No type question type.

### 3.77.3 Constructor & Destructor Documentation

#### 3.77.3.1 Question() [1/2]

```
Question (
    String questionStr,
    Answer... possibleAnswers )
```

Instantiates a new [Question](#).

## Parameters

<i>questionStr</i>	the question str
<i>possibleAnswers</i>	the possible answers

#### 3.77.3.2 Question() [2/2]

```
Question (
    String questionStr,
    QuestionType questionType,
    Answer... possibleAnswers )
```

Instantiates a new [Question](#).

## Parameters

<i>questionStr</i>	the question str
<i>questionType</i>	the question type
<i>possibleAnswers</i>	the possible answers

### 3.77.4 Member Function Documentation

### 3.77.4.1 drawOffer()

```
static Question drawOffer (
    String offeringPlayer ) [static]
```

Draw offer question.

#### Parameters

<i>offeringPlayer</i>	the offering player
-----------------------	---------------------

#### Returns

the question

### 3.77.4.2 getDefaultAnswer()

```
Answer getDefaultAnswer ( )
```

Gets default answer.

#### Returns

the default answer

### 3.77.4.3 setDefaultAnswer()

```
void setDefaultAnswer (
    Answer defaultAnswer )
```

Sets default answer.

#### Parameters

<i>defaultAnswer</i>	the default answer
----------------------	--------------------

### 3.77.4.4 getQuestionStr()

```
String getQuestionStr ( )
```

Gets question str.

**Returns**

the question str

**3.77.4.5 getPossibleAnswers()**

```
Answer[] getPossibleAnswers ( )
```

Get possible answers answer [ ].

**Returns**

the answer [ ]

**3.77.4.6 equals()**

```
boolean equals (
    Object o )
```

Equals boolean.

**Parameters**

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

**Returns**

the boolean

**3.77.4.7 toString()**

```
String toString ( )
```

To string string.

**Returns**

the string

### 3.77.4.8 Answer()

```
record Answer (
    String answerStr )
```

Answer.

Author

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

## 3.77.5 Member Data Documentation

### 3.77.5.1 Threefold

```
final Question Threefold = new Question("Would you like to claim a Threefold repetition?",
    QuestionType.THREEFOLD, Answer.YES, Answer.NO) [static]
```

The constant Threefold.

### 3.77.5.2 Rematch

```
final Question Rematch = new Question("rematch", QuestionType.REMATCH, Answer.YES, Answer.NO)
[static]
```

The constant Rematch.

### 3.77.5.3 questionStr

```
final String questionStr
```

The [Question](#) str.

### 3.77.5.4 questionType

```
final QuestionType questionType
```

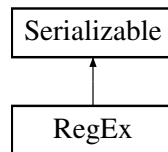
The [Question](#) type.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Misc/Question.java

## 3.78 RegEx

Inheritance diagram for RegEx:



### Classes

- class **Prefixes**

### Public Attributes

- final String[] [dontMatch](#)

### Static Public Attributes

- static final [RegEx Fen](#) = new [RegEx](#)("^\\$|\\s\*([rnbpkpRNBQKP1-8]+\\V){7}([rnbpkpRNBQKP1-8]+)\\s[bw-]\\\\s((([a-hkqA-HKQ]{1,4})|(-))\\\\s((([a-h][36])|(-))\\\\s\\\\d+\\\\s\\\\d+\\\\s\*)", "standard fen")
- static final [RegEx Username](#) = new [RegEx](#)("^[a-zA-Z0-9\_-]{5,10}\$", "5-10 characters a-z 0-9 \_.-", [Prefixes](#).← GUEST\_PREFIX, "User")
- static final [RegEx Password](#) = new [RegEx](#)("^[a-zA-Z0-9\_-]{5,10}\$", "5-10 characters a-z 0-9 \_.-", "password")
- static final [RegEx Icon](#) = new [RegEx](#)("\\. (png|gif)\$", "")
- static final [RegEx StrUtilSkip](#) = new [RegEx](#)("^[ \\n\\t\\r]|(<[^>]\*>)", "")
- static final [RegEx Numbers](#) = new [RegEx](#)("^[0-9]+\$", "enter number")
- static final [RegEx URL](#) = new [RegEx](#)("[-a-zA-Z0-9@:%.\_\\+~#={1,256}\\.[a-zA-Z0-9()]{1,6}\\b([-a-zA-Z0-9()@%\_\\+\\.~#?&//=]\*)", "enter valid url")
- static final [RegEx DontSaveGame](#) = new [RegEx](#)([Prefixes](#).GUEST\_PREFIX + "|" + [AiParameters](#).AiType.MyAi + "|" + [AiParameters](#).AiType.Stockfish, "")
- static final [RegEx IPPortAddress](#) = new [RegEx](#)("^[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3} \*: \*[0-9]{2,5}\$", "[IP : PORT]")
- static final [RegEx Any](#) = new [RegEx](#)("", "")

### 3.78.1 Detailed Description

Reg ex.

Author

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

### 3.78.2 Constructor & Destructor Documentation

#### 3.78.2.1 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

**3.78.2.2 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

**3.78.3 Member Function Documentation****3.78.3.1 isSavedDate()**

```
static boolean isSavedDate (
    String str ) [static]
```

Is saved date boolean.

## Parameters

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

## Returns

the boolean

### 3.78.3.2 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

### 3.78.3.3 get()

```
RegEx get (
    boolean useDontMatch )
```

Get reg ex.

#### Parameters

<i>useDontMatch</i>	the use dont match
---------------------	--------------------

#### Returns

the reg ex

### 3.78.3.4 setUseDontMatch()

```
void setUseDontMatch (
    boolean useDontMatch )
```

Sets use dont match.

#### Parameters

<i>useDontMatch</i>	the use dont match
---------------------	--------------------

### 3.78.3.5 `getDetails()`

```
String getDetails ( )
```

Gets details.

#### Returns

the details

### 3.78.3.6 `check()`

```
boolean check (
    String str )
```

Check boolean.

#### Parameters

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

#### Returns

the boolean

### 3.78.3.7 `getRegex()`

```
String getRegex ( )
```

Gets regex.

#### Returns

the regex

## 3.78.4 Member Data Documentation

### 3.78.4.1 `Fen`

```
final Regex Fen = new Regex("^$|\\s*([rnbqkpRNBQKP1-8]+\\/){7}([rnbqkpRNBQKP1-8]+)\\s[bw-]\\s(([a-hkqA-HKQ]{1,}
"standard fen") [static]
```

The constant Fen.



### 3.78.4.2 Username

```
final RegEx Username = new RegEx("^ [a-zA-Z0-9_-.]{5,10}$", "5-10 characters a-z 0-9 _.-",  
Prefixes.GUEST_PREFIX, "User") [static]
```

The constant Username.

### 3.78.4.3 Password

```
final RegEx Password = new RegEx("^ [a-zA-Z0-9_-.]{5,10}$", "5-10 characters a-z 0-9 _.-",  
"password") [static]
```

The constant Password.

### 3.78.4.4 Icon

```
final RegEx Icon = new RegEx("\\.(png|gif)$", "") [static]
```

The constant Icon.

### 3.78.4.5 StrUtilSkip

```
final RegEx StrUtilSkip = new RegEx("(^ [ \\n\\t\\[ ]|(<[^>]*>)", "") [static]
```

The constant StrUtilSkip.

### 3.78.4.6 Numbers

```
final RegEx Numbers = new RegEx("^ [0-9]+$", "enter number") [static]
```

The constant Numbers.

### 3.78.4.7 URL

```
final RegEx URL = new RegEx("[-a-zA-Z0-9@:%.\\_+~#={1,256}\\.[a-zA-Z0-9()]{1,6}\\b([-a-zA-Z0-9()@%<->_\\+\\.~#?&/=]*)", "enter valid url") [static]
```

The constant URL.

### 3.78.4.8 DontSaveGame

```
final RegEx DontSaveGame = new RegEx(Prefixes.GUEST_PREFIX + "|" + AiParameters.AiType.MyAi +  
"|" + AiParameters.AiType.Stockfish, "") [static]
```

The constant DontSaveGame.

### 3.78.4.9 IPPortAddress

```
final Regex IPPortAddress = new Regex("^[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3} *: *[0-9]{2,5}$",
"[IP : PORT]") [static]
```

The constant IPPAddress.

### 3.78.4.10 Any

```
final Regex Any = new Regex("", "") [static]
```

The constant Any.

### 3.78.4.11 dontMatch

```
final String [] dontMatch
```

The Dont match.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Utils/Regex.java

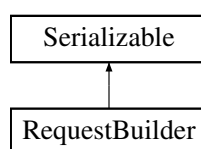
## 3.79 SyncedItems< E extends SyncableItem >.Remover< L >

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/Sync/SyncedItems.java

## 3.80 RequestBuilder

Inheritance diagram for RequestBuilder:



## Classes

- class **GraphableSelection**

## Public Attributes

- final [Arg\[\]](#) [args](#)

## Static Public Attributes

- static final String [TIE\\_STR](#) = "----tie----"

## Protected Attributes

- final [SQLStatement](#) [statement](#)
- final String [name](#)
- String [postDescription](#)
- String [preDescription](#)
- [RequestBuilder](#) [subBuilder](#) = null

### 3.80.1 Detailed Description

Request builder - creates builders capable of generating complete sql statements. after building with the required arguments .

#### Author

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

### 3.80.2 Constructor & Destructor Documentation

#### 3.80.2.1 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

#### 3.80.2.2 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

### 3.80.2.3 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

### 3.80.2.4 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

### 3.80.3 Member Function Documentation

#### 3.80.3.1 createVariation()

```
static RequestBuilder createVariation (
    Supplier< 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

#### 3.80.3.2 changePassword()

```
static RequestBuilder changePassword ( ) [static]
```

Change password request builder.

##### Returns

the request builder

#### 3.80.3.3 changeProfilePic()

```
static RequestBuilder changeProfilePic ( ) [static]
```

Change profile pic request builder.

##### Returns

the request builder

#### 3.80.3.4 addShouldSync()

```
void addShouldSync (
    SyncedListType listType ) [protected]
```

Add should sync.

**Parameters**

<i>listType</i>	the list type
-----------------	---------------

**3.80.3.5 deleteAllUnFinishedGames()**

```
static RequestBuilder deleteAllUnFinishedGames ( ) [static]
```

Delete all un finished games request builder.

**Returns**

the request builder

**3.80.3.6 games()**

```
static RequestBuilder games ( ) [static]
```

Games request builder.

**Returns**

the request builder

**3.80.3.7 setSubBuilder()**

```
void setSubBuilder (
    RequestBuilder subBuilder )
```

Sets sub builder.

**Parameters**

<i>subBuilder</i>	the sub builder
-------------------	-----------------

**3.80.3.8 statsByTimeOfDay()**

```
static RequestBuilder statsByTimeOfDay ( ) [static]
```

Stats by time of day request builder.

**Returns**

the request builder

**3.80.3.9 top()**

```
static RequestBuilder top ( ) [static]
```

Top request builder.

**Returns**

the request builder

**3.80.3.10 getShouldSync()**

```
ArrayList< SyncedListType > getShouldSync ( )
```

Gets should sync.

**Returns**

the should sync

**3.80.3.11 getArgVal()**

```
String getArgVal (
    int index )
```

Gets arg val.

**Parameters**

<i>index</i>	the index
--------------	-----------

**Returns**

the arg val

### 3.80.3.12 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

### 3.80.3.13 getPreDescription()

```
String getPreDescription ( )
```

Gets pre description.

#### Returns

the pre description

### 3.80.3.14 getPostDescription()

```
String getPostDescription ( )
```

Gets post description.

#### Returns

the post description

### 3.80.3.15 getName()

```
String getName ( )
```

Gets name.

#### Returns

the name



### 3.80.3.16 getArgs()

```
Arg[] getArgs ( )
```

Get args arg [].

Returns

the arg []

### 3.80.3.17 build()

```
DBRequest build (
    Object... argsVals )
```

Build db request.

Parameters

<i>argsVals</i>	the args vals
-----------------	---------------

Returns

the db request

## 3.80.4 Member Data Documentation

### 3.80.4.1 TIE\_STR

```
final String TIE_STR = "----tie----" [static]
```

The constant TIE\_STR.

### 3.80.4.2 args

```
final Arg [] args
```

The request Arguments.

### 3.80.4.3 statement

```
final SQLStatement statement [protected]
```

The Statement.

#### 3.80.4.4 name

```
final String name [protected]
```

The Name.

#### 3.80.4.5 postDescription

```
String postDescription [protected]
```

The Post description.

#### 3.80.4.6 preDescription

```
String preDescription [protected]
```

The Pre description.

#### 3.80.4.7 subBuilder

```
RequestBuilder subBuilder = null [protected]
```

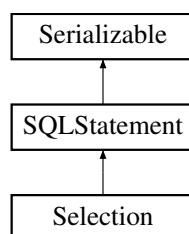
The Sub builder.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/RequestBuilder.java

## 3.81 Selection

Inheritance diagram for Selection:



### Classes

- interface [Join](#)
- interface [Order](#)

## Additional Inherited Members

### 3.81.1 Detailed Description

[Selection](#) - a selection sql statement.

Author

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

### 3.81.2 Class Documentation

#### 3.81.2.1 interface ver14::SharedClasses::DBActions::Statements::Selection::Join

[Join](#) - selection join.

Author

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

Class Members

String	LEFT	The constant LEFT.
--------	------	--------------------

#### 3.81.2.2 interface ver14::SharedClasses::DBActions::Statements::Selection::Order

[Order](#) - selection order by.

Author

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

Class Members

String	DESC	The constant DESC.
String	ASC	The constant ASC.

### 3.81.3 Constructor & Destructor Documentation

#### 3.81.3.1 Selection() [1/2]

```
Selection (
    Object selectFrom,
    Object[] select )
```

Instantiates a new [Selection](#).

#### Parameters

<i>selectFrom</i>	the select from
<i>select</i>	the select

### 3.81.3.2 Selection() [2/2]

```
Selection (
    Object selectFrom,
    Condition condition,
    Object[] select )
```

Instantiates a new [Selection](#).

#### Parameters

<i>selectFrom</i>	the select from
<i>condition</i>	the condition
<i>select</i>	the select

## 3.81.4 Member Function Documentation

### 3.81.4.1 nestMe()

```
Selection nestMe (
    Col... outerSelect )
```

Nest me selection.

#### Parameters

<i>outerSelect</i>	the outer select
--------------------	------------------

#### Returns

the selection

### 3.81.4.2 createStatement()

```
String createStatement ( ) [protected]
```

Create statement string.

**Returns**

the string

Reimplemented from [SQLStatement](#).

**3.81.4.3 top()**

```
void top (
    Object top )
```

Top.

**Parameters**

<i>top</i>	the top
------------	---------

**3.81.4.4 join()**

```
void join (
    @Join String joinType,
    Table joinWith,
    Condition condition,
    Col... groupBy )
```

[Join](#).

**Parameters**

<i>joinType</i>	the join type
<i>joinWith</i>	the join with
<i>condition</i>	the condition
<i>groupBy</i>	the group by

**3.81.4.5 orderBy()**

```
void orderBy (
    Col col,
    @Order String order )
```

[Order](#) by.

## Parameters

<i>col</i>	the col
<i>order</i>	the order

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/Statements/Selection.java

## 3.82 CastlingRights.Side Enum Reference

### Public Attributes

- [KING](#) =(Location.G, Location.H, Location.F)
- [QUEEN](#) =(Location.C, Location.A, Location.D)
- 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](#) = {[KING](#), [QUEEN](#)}

### 3.82.1 Detailed Description

[Side](#) - Castling side.

#### Author

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

### 3.82.2 Constructor & Destructor Documentation

#### 3.82.2.1 Side()

```
Side (
    int castledKingCol,
    int rookStartingCol,
    int castledRookCol )
```

Instantiates a new [Side](#).

## Parameters

<i>castledKingCol</i>	the castled king col
<i>rookStartingCol</i>	the rook starting col
<i>castledRookCol</i>	the castled rook col

### 3.82.3 Member Function Documentation

#### 3.82.3.1 toString()

```
String toString ( )
```

To string string.

## Returns

the string @hidden

#### 3.82.3.2 kingFinalLoc()

```
Location kingFinalLoc (
    Location currentKingLoc )
```

King final loc location.

## Parameters

<i>currentKingLoc</i>	the current king loc
-----------------------	----------------------

## Returns

the location

### 3.82.4 Member Data Documentation

#### 3.82.4.1 KING

```
KING =(Location.G, Location.H, Location.F)
```

King side.

#### 3.82.4.2 QUEEN

```
QUEEN =(Location.C, Location.A, Location.D)
```

Queen side.

#### 3.82.4.3 SIDES

```
final Side [] SIDES = {KING, QUEEN} [static]
```

The constant SIDES.

#### 3.82.4.4 rookStartingCol

```
final int rookStartingCol
```

The Rook starting col.

#### 3.82.4.5 castledRookCol

```
final int castledRookCol
```

The Castled rook col.

#### 3.82.4.6 castledKingCol

```
final int castledKingCol
```

The Castled king col.

#### 3.82.4.7 kingTravelDistance

```
final int kingTravelDistance
```

The King travel distance.

#### 3.82.4.8 castlingNotation

```
final String castlingNotation
```

The Castling notation.

#### 3.82.4.9 asInt

```
final int asInt
```

The As int.



**3.82.4.10 mult**

```
final int mult
```

The King Movement [Direction](#) Mult

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Game/Moves/CastlingRights.java

**3.83 GameStateSpecificStatus Enum Reference****Public Attributes**

- [Checkmate](#)
- [TimedOut](#)
- [TimedOutVsInsufficientMaterial](#)
- [Resignation](#)
- [GameGoesOn](#) =(GameStateType.GAME\_GOES\_ON)
- [ThreeFoldRepetition](#) =(GameStateType.TIE)
- [Stalemate](#) =(GameStateType.TIE)
- [InsufficientMaterial](#) =(GameStateType.TIE)
- [FiftyMoveRule](#) =(GameStateType.TIE)
- [TieByAgreement](#)
- [PlayerDisconnectedVsAi](#)
- [PlayerDisconnectedVsReal](#)
- [ServerStoppedGame](#) =(GameStateType.TIE)
- final [GameStateType](#) gameStateType

**3.83.1 Detailed Description**

Specific status - specific game status .

Author

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

**3.83.2 Constructor & Destructor Documentation****3.83.2.1 SpecificStatus() [1/2]**

```
SpecificStatus ( )
```

Instantiates a new Specific status.

**3.83.2.2 SpecificStatus() [2/2]**

```
SpecificStatus (
    GameStateType gameStateType )
```

Instantiates a new Specific status.

## Parameters

<i>gameStatusType</i>	the game status type
-----------------------	----------------------

### 3.83.3 Member Function Documentation

#### 3.83.3.1 toString()

```
String toString ( )
```

To string string.

## Returns

the string

### 3.83.4 Member Data Documentation

#### 3.83.4.1 Checkmate

```
Checkmate
```

Checkmate.

#### 3.83.4.2 TimedOut

```
TimedOut
```

## Initial value:

```
={  
    @Override  
    public String toString() {  
        return "Time Out";  
    }  
}
```

Timed out.

#### 3.83.4.3 TimedOutVsInsufficientMaterial

```
TimedOutVsInsufficientMaterial
```

## Initial value:

```
=(GameStatusType.TIE) {  
    @Override  
    public String toString() {  
        return "Time Out vs Insufficient Material";  
    }  
}
```

Timed out vs insufficient material.

#### 3.83.4.4 Resignation

Resignation

Resignation.

#### 3.83.4.5 GameGoesOn

GameGoesOn = (GameStateType.GAME\_GOES\_ON)

Game goes on .

#### 3.83.4.6 ThreeFoldRepetition

ThreeFoldRepetition = (GameStateType.TIE)

Three fold repetition.

#### 3.83.4.7 Stalemate

Stalemate = (GameStateType.TIE)

Stalemate.

#### 3.83.4.8 InsufficientMaterial

InsufficientMaterial = (GameStateType.TIE)

Insufficient material.

#### 3.83.4.9 FiftyMoveRule

FiftyMoveRule = (GameStateType.TIE)

Fifty move rule.

#### 3.83.4.10 TieByAgreement

TieByAgreement

##### Initial value:

```
=(GameStateType.TIE) {  
    @Override  
    public String toString() {  
        return "Agreement";  
    }  
}
```

The Tie by agreement.

### 3.83.4.11 PlayerDisconnectedVsAi

PlayerDisconnectedVsAi

#### Initial value:

```
=(GameStatusType.UNFINISHED) {  
    @Override  
    public String toString() {  
        return "Player Disconnected";  
    }  
}
```

The Player disconnected vs ai.

### 3.83.4.12 PlayerDisconnectedVsReal

PlayerDisconnectedVsReal

#### Initial value:

```
=({  
    @Override  
    public String toString() {  
        return "Other Player Disconnected";  
    }  
})
```

The Player disconnected vs real.

### 3.83.4.13 ServerStoppedGame

ServerStoppedGame =(GameStatusType.TIE)

Server stopped game.

### 3.83.4.14 gameStatusType

final GameStatusType gameStatusType

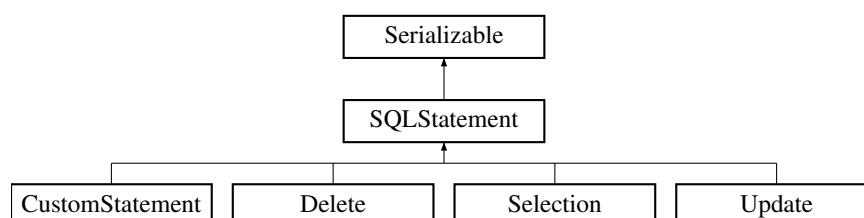
The Game status type.

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Game/Evaluation/GameStatus.java

## 3.84 SQLStatement

Inheritance diagram for SQLStatement:



## Public Attributes

- final DBRequest.Type [type](#)

### 3.84.1 Detailed Description

Sql statement - represents an sql statement.

#### Author

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

### 3.84.2 Constructor & Destructor Documentation

#### 3.84.2.1 SQLStatement()

```
SQLStatement (
    DBRequest.Type type )
```

Instantiates a new Sql statement.

#### Parameters

<i>type</i>	the type
-------------	----------

### 3.84.3 Member Function Documentation

#### 3.84.3.1 replace()

```
void replace (
    String replacing,
    String replaceWith )
```

Replace.

#### Parameters

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

### 3.84.3.2 createStatement()

```
abstract String createStatement ( ) [abstract], [protected]
```

Create statement string.

#### Returns

the string

Reimplemented in [CustomStatement](#), [Delete](#), [Selection](#), and [Update](#).

### 3.84.3.3 getStatement()

```
String getStatement ( )
```

Gets statement.

#### Returns

the statement

### 3.84.3.4 toString()

```
String toString ( )
```

To string string.

#### Returns

the string

## 3.84.4 Member Data Documentation

### 3.84.4.1 type

```
final DBRequest.Type type
```

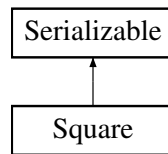
The request Type.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/Statements/SQLStatement.java

## 3.85 Square

Inheritance diagram for Square:



### Static Public Attributes

- static final `Piece EMPTY_PIECE` = null

### 3.85.1 Detailed Description

`Square` represents a square on the logic board.

Author

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

### 3.85.2 Constructor & Destructor Documentation

#### 3.85.2.1 `Square()` [1/2]

```
Square (
    Location loc )
```

Instantiates a new `Square`.

Parameters

<code>loc</code>	the loc
------------------	---------

#### 3.85.2.2 `Square()` [2/2]

```
Square (
    Piece piece,
    Location loc )
```

Instantiates a new `Square`.

**Parameters**

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

### 3.85.3 Member Function Documentation

#### 3.85.3.1 setEmpty()

```
void setEmpty ( )
```

Sets empty.

#### 3.85.3.2 getPiece()

```
Piece getPiece ( )
```

Gets piece.

**Returns**

the piece

#### 3.85.3.3 setPiece()

```
void setPiece (
    Piece piece )
```

Sets piece.

**Parameters**

<i>piece</i>	the piece
--------------	-----------

#### 3.85.3.4 getLoc()

```
Location getLoc ( )
```

Gets loc.



**Returns**

the loc

**3.85.3.5 toString()**

```
String toString ( )
```

To string string.

**Returns**

the string

**3.85.3.6 getFen()**

```
String getFen ( )
```

Gets fen.

**Returns**

the fen

**3.85.3.7 isEmpty()**

```
boolean isEmpty ( )
```

Is empty boolean.

**Returns**

the boolean

**3.85.3.8 getPieceIcon()**

```
String getPieceIcon ( )
```

Gets piece icon.

**Returns**

the piece icon

### 3.85.4 Member Data Documentation

#### 3.85.4.1 EMPTY\_PIECE

```
final Piece EMPTY_PIECE = null [static]
```

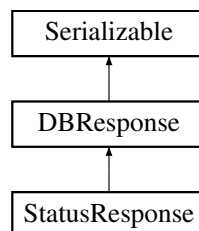
The constant EMPTY\_PIECE.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/GameSetup/BoardSetup/Square.java

## 3.86 StatusResponse

Inheritance diagram for StatusResponse:



### Additional Inherited Members

#### 3.86.1 Detailed Description

Status response - .

Author

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

#### 3.86.2 Constructor & Destructor Documentation

##### 3.86.2.1 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

**3.86.2.2 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

**3.86.3 Member Function Documentation****3.86.3.1 getDetails()**

```
String getDetails ( )
```

Gets details.

**Returns**

the details

**3.86.3.2 isAnyData()**

```
boolean isAnyData ( )
```

Is any data boolean.

**Returns**

the boolean

Reimplemented from [DBResponse](#).

### 3.86.3.3 clean()

```
DBResponse clean ( )
```

Clean db response.

#### Returns

the db response

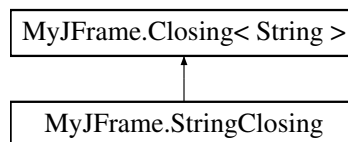
Reimplemented from [DBResponse](#).

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/DBResponse/StatusResponse.java

## 3.87 MyJFrame.StringClosing

Inheritance diagram for MyJFrame.StringClosing:



### Additional Inherited Members

#### 3.87.1 Detailed Description

The interface String closing.

#### 3.87.2 Member Function Documentation

##### 3.87.2.1 show()

```
default String show ( )
```

Show string.

#### Returns

the string

##### 3.87.2.2 checkClosingVal()

```
default boolean checkClosingVal (
    String val )
```

Check closing val boolean.

## Parameters

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

## Returns

the boolean

The documentation for this interface was generated from the following file:

- `ver14/SharedClasses/UI/MyJFrame.java`

## 3.88 MyLbl.StringModifier

The documentation for this interface was generated from the following file:

- `ver14/SharedClasses/UI/MyLbl.java`

## 3.89 StrUtils

## Classes

- interface [IterationThingy](#)
- enum **TimeRange**

### 3.89.1 Detailed Description

utility class for String related utilities

## Author

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

### 3.89.2 Member Function Documentation

#### 3.89.2.1 isAbsoluteUrl()

```
static boolean isAbsoluteUrl (
    String urlString ) [static]
```

Is absolute url boolean.

**Parameters**

<i>urlString</i>	the url string
------------------	----------------

**Returns**

the boolean

**3.89.2.2 dontCapFull()**

```
static String dontCapFull (  
    String str ) [static]
```

Dont cap full string.

**Parameters**

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

**Returns**

the string

**3.89.2.3 htmlNewLines()**

```
static String htmlNewLines (  
    String str ) [static]
```

Html new lines string.

**Parameters**

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

**Returns**

the string

**3.89.2.4 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

### 3.89.2.5 format() [1/3]

```
static String[][] format (  
    String mat[][] )    [static]
```

Format string [][].

#### Parameters

<i>mat</i>	the mat
------------	---------

#### Returns

the string [][ ]

### 3.89.2.6 format() [2/3]

```
static String[] format (  
    String[] arr )    [static]
```

Format string [ ].

#### Parameters

<i>arr</i>	the arr
------------	---------

#### Returns

the string [ ]

### 3.89.2.7 format() [3/3]

```
static String format (  
    String str ) [static]
```

Format string.

#### Parameters

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

#### Returns

the string

### 3.89.2.8 isEmpty()

```
static boolean isEmpty (  
    String str ) [static]
```

Is empty boolean.

#### Parameters

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

#### Returns

the boolean

### 3.89.2.9 clean()

```
static String clean (  
    String str ) [static]
```

Clean string.

#### Parameters

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

#### Returns

the string



**3.89.2.10 formatDate()** [1/3]

```
static String formatDate (  
    String longStr ) [static]
```

Format date string.

**Parameters**

<i>longStr</i>	the long str
----------------	--------------

**Returns**

the string

**3.89.2.11 formatDate()** [2/3]

```
static String formatDate (  
    Date date ) [static]
```

Format date string.

**Parameters**

<i>date</i>	the date
-------------	----------

**Returns**

the string

**3.89.2.12 formatDate()** [3/3]

```
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

**3.89.2.13 parseURLS()**

```
static String parseURLS (  
    String str ) [static]
```

Parse urls string.

**Parameters**

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

**Returns**

the string

**3.89.2.14 uppercase()**

```
static String uppercase (  
    String str ) [static]
```

Uppercase string.

**Parameters**

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

**Returns**

the string

**3.89.2.15 dontCapWord()**

```
static String dontCapWord (  
    String str ) [static]
```

Dont cap word string.

**Parameters**

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

**Returns**

the string

**3.89.2.16 formatDateSQL()**

```
static String formatDateSQL (  
    Date date ) [static]
```

Format date sql string.

**Parameters**

<i>date</i>	the date
-------------	----------

**Returns**

the string

**3.89.2.17 getPort() [1/2]**

```
static int getPort (  
    Socket socket ) [static]
```

Gets port.

**Parameters**

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

**Returns**

the port

**3.89.2.18 getPort() [2/2]**

```
static int getPort (  
    SocketAddress socketAddress ) [static]
```

Gets port.

## Parameters

<i>socketAddress</i>	the socket address
----------------------	--------------------

## Returns

the port

**3.89.2.19 getUrl() [1/2]**

```
static String getUrl (
    Socket socket ) [static]
```

Gets url.

## Parameters

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

## Returns

the url

**3.89.2.20 getUrl() [2/2]**

```
static String getUrl (
    SocketAddress socketAddress ) [static]
```

Gets url.

## Parameters

<i>socketAddress</i>	the socket address
----------------------	--------------------

## Returns

the url

**3.89.2.21 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

### 3.89.2.22 fixHtml()

```
static String fixHtml (  
    String str ) [static]
```

Fix html string.

#### Parameters

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

#### Returns

the string

### 3.89.2.23 strINN()

```
static String strINN (  
    Object... objs ) [static]
```

Str inn string.

#### Parameters

<i>objs</i>	the objs
-------------	----------

#### Returns

the string

**3.89.2.24 splitArr()** [1/3]

```
static String splitArr (  
    Object[] arr ) [static]
```

Split arr string.

**Parameters**

<i>arr</i>	the arr
------------	---------

**Returns**

the string

**3.89.2.25 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

**3.89.2.26 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

**3.89.2.27 createTimeGreeting()**

```
static String createTimeGreeting ( ) [static]
```

Create time greeting string.

**Returns**

the string

**3.89.2.28 main()**

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

The entry point of application.

**Parameters**

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

**3.89.2.29 createTimeStr()**

```
static String createTimeStr (
    long millis ) [static]
```

Create time str string.

**Parameters**

<i>millis</i>	the millis
---------------	------------

**Returns**

the string



### 3.89.2.30 awful()

```
static String awful (
    String og ) [static]
```

Awful string.

#### Parameters

<i>og</i>	the og
-----------	--------

#### Returns

the string

### 3.89.2.31 dateTimePrefix()

```
static String dateTimePrefix (
    String str ) [static]
```

Date time prefix string.

#### Parameters

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

#### Returns

the string

### 3.89.2.32 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

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Utils/StrUtils.java

## 3.90 SwitchCase

### 3.90.1 Detailed Description

Switch case - represents a case that is meant to be used inside a switch case col. if the `condition` is true, the `ifTrue` col will display in the switch case col

**Author**

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

### 3.90.2 Constructor & Destructor Documentation

#### 3.90.2.1 SwitchCase()

```
SwitchCase (
    Condition condition,
    Col ifTrue )
```

Instantiates a new Switch case.

**Parameters**

<i>condition</i>	the condition
<i>ifTrue</i>	the if true

### 3.90.3 Member Function Documentation

#### 3.90.3.1 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

**3.90.3.2 defaultCase()**

```
static SwitchCase defaultCase (  
    Col ifTrue ) [static]
```

Default case switch case.

**Parameters**

<i>ifTrue</i>	the if true
---------------	-------------

**Returns**

the switch case

**3.90.3.3 condition()**

```
Condition condition ( )
```

Condition condition.

**Returns**

the condition

**3.90.3.4 ifTrue()**

```
Col ifTrue ( )
```

If true col.

**Returns**

the col

### 3.90.3.5 toString()

```
String toString ( )
```

To string string.

#### Returns

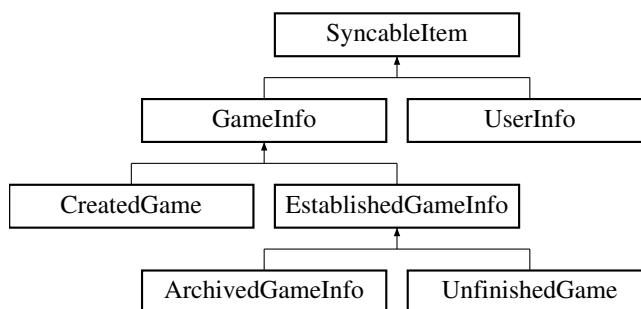
the string

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/Table/SwitchCase.java

## 3.91 SyncableItem

Inheritance diagram for SyncableItem:



### 3.91.1 Member Function Documentation

#### 3.91.1.1 ID()

```
String ID ( )
```

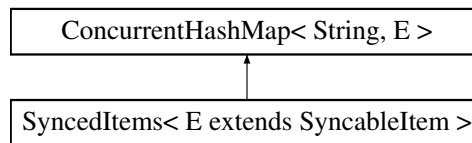
Implemented in [GameInfo](#).

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/Sync/SyncableItem.java

## 3.92 SyncedItems< E extends SyncableItem >

Inheritance diagram for SyncedItems< E extends SyncableItem >:



### Classes

- interface [Remover](#)

### Public Attributes

- final [SyncedListType](#) **syncedListType**

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Sync/SyncedItems.java

## 3.93 SyncedListType Enum Reference

### Public Attributes

- **RESUMABLE\_GAMES**
- **JOINABLE\_GAMES**
- **CONNECTED\_USERS**
- **ONGOING\_GAMES**

### 3.93.1 Member Data Documentation

#### 3.93.1.1 RESUMABLE\_GAMES

RESUMABLE\_GAMES

#### 3.93.1.2 JOINABLE\_GAMES

JOINABLE\_GAMES

### 3.93.1.3 CONNECTED\_USERS

CONNECTED\_USERS

### 3.93.1.4 ONGOING\_GAMES

ONGOING\_GAMES

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Sync/SyncedListType.java

## 3.94 Table Enum Reference

### Public Attributes

- `Games` =(Col.GameID, Col.Player1, Col.Player2, Col.SavedGame, Col.Winner)
- `UnfinishedGames` =(Col.GameID, Col.Player1, Col.Player2, Col.SavedGame, Col.PlayerToMove)
- `Users` =(Col.Username, Col.Password)
- final `Col[] cols`

### 3.94.1 Detailed Description

`Table` - represents the tables in the db.

Author

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

### 3.94.2 Constructor & Destructor Documentation

#### 3.94.2.1 Table()

```
Table (
    Col... cols )
```

Instantiates a new `Table`.

Parameters

<code>cols</code>	the cols
-------------------	----------

### 3.94.3 Member Function Documentation

#### 3.94.3.1 tableAndValues()

```
String tableAndValues ( )
```

Table and values string.

##### Returns

the string

#### 3.94.3.2 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

### 3.94.4 Member Data Documentation

#### 3.94.4.1 Games

```
Games =(Col.GameID, Col.Player1, Col.Player2, Col.SavedGame, Col.Winner)
```

Games table.



#### 3.94.4.2 UnfinishedGames

```
UnfinishedGames =(Col.GameID, Col.Player1, Col.Player2, Col.SavedGame, Col.PlayerToMove)
```

Unfinished games table.

#### 3.94.4.3 Users

```
Users =(Col.Username, Col.Password)
```

Users table.

#### 3.94.4.4 cols

```
final Col [] cols
```

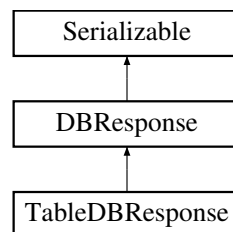
The Cols in this table

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/DBActions/Table/Table.java

## 3.95 TableDBResponse

Inheritance diagram for TableDBResponse:



### Protected Attributes

- String[] [columns](#)
- String[][] [rows](#)

### 3.95.1 Detailed Description

Table db response - .

Author

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

## 3.95.2 Constructor & Destructor Documentation

### 3.95.2.1 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

### 3.95.2.2 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

### 3.95.2.3 TableDBResponse() [3/3]

```
TableDBResponse ( ) [protected]
```

Instantiates a new Table db response.

## 3.95.3 Member Function Documentation

**3.95.3.1 numOfRows()**

```
int numOfRows ( )
```

Num of rows int.

**Returns**

the int

**3.95.3.2 getFirstRow()**

```
String[] getFirstRow ( )
```

Get first row string [ ].

**Returns**

the string [ ]

**3.95.3.3 getCell() [1/2]**

```
String getCell (
    int row,
    String col )
```

Gets cell.

**Parameters**

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

**Returns**

the cell

**3.95.3.4 getColumnIndex()**

```
int getColumnIndex (
    String column ) [protected]
```

Gets column index.

**Parameters**

<i>column</i>	the column
---------------	------------

**Returns**

the column index

**3.95.3.5   getCell()** [2/2]

```
String getCell (
    int row,
    Col col )
```

Gets cell.

**Parameters**

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

**Returns**

the cell

**3.95.3.6   isAnyData()**

```
boolean isAnyData ( )
```

Is any data boolean.

**Returns**

the boolean

Reimplemented from [DBResponse](#).

**3.95.3.7   clean()**

```
TableDBResponse clean ( )
```

Clean table db response.

**Returns**

the table db response

Reimplemented from [DBResponse](#).

### 3.95.3.8 toString()

```
String toString ( )
```

To string string.

#### Returns

the string

Reimplemented from [DBResponse](#).

### 3.95.3.9 rowToString()

```
String rowToString (
    String[] row )
```

Row to string string.

#### Parameters

<i>row</i>	the row
------------	---------

#### Returns

the string

### 3.95.3.10 getColumns()

```
String[] getColumns ( )
```

Get columns string [ ].

#### Returns

the string [ ]

### 3.95.3.11 getRows()

```
String[][] getRows ( )
```

Get rows string [ ] [ ].

#### Returns

the string [ ] [ ]

### 3.95.4 Member Data Documentation

#### 3.95.4.1 columns

```
String [] columns [protected]
```

The Columns.

#### 3.95.4.2 rows

```
String [][] rows [protected]
```

The Rows.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/DBResponse/TableDBResponse.java

## 3.96 ThreadsManager

### 3.96.1 Detailed Description

The type Threads manager.

### 3.96.2 Member Function Documentation

#### 3.96.2.1 handleErrors()

```
static void handleErrors (  
    ThrowingRunnable runnable ) [static]
```

Handle errors.

Parameters

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

### 3.96.2.2 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

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Threads/ThreadsManager.java

## 3.97 MyThread.ThreadStatus Enum Reference

### Public Attributes

- NOT\_STARTED
- RUNNING
- DONE

### 3.97.1 Member Data Documentation

#### 3.97.1.1 NOT\_STARTED

NOT\_STARTED

#### Initial value:

```
={
    @Override
    public ThreadStatus next() {
        return RUNNING;
    }
}
```

### 3.97.1.2 RUNNING

RUNNING

#### Initial value:

```
={
    @Override
    public ThreadStatus next() {
        return DONE;
    }
}
```

### 3.97.1.3 DONE

DONE

#### Initial value:

```
={
    @Override
    public ThreadStatus next() {
        return null;
    }
}
```

The documentation for this enum was generated from the following file:

- ver14/SharedClasses/Threads/MyThread.java

## 3.98 ThrowingCallback< T >

### 3.98.1 Detailed Description

represents a callback that might throw an exception .

#### Parameters

< T >	the callback type
-------	-------------------

#### Author

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

### 3.98.2 Member Function Documentation

#### 3.98.2.1 callback()

```
void callback (
    T obj ) throws Exception
```

[Callback.](#)



## Parameters

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

## Exceptions

<i>Exception</i>	the exception that might get thrown
------------------	-------------------------------------

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/Callbacks/ThrowingCallback.java

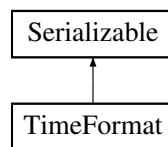
## 3.99 ThrowingRunnable

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/Threads/ErrorHandling/ThrowingRunnable.java

## 3.100 TimeFormat

Inheritance diagram for TimeFormat:



### Public Attributes

- final long **timeInMillis**

### Static Public Attributes

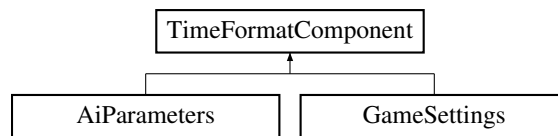
- static final **TimeFormat RAPID** = new **TimeFormat**(TimeUnit.MINUTES.toMillis(10))
- static final **TimeFormat ULTRA\_BULLET** = new **TimeFormat**(TimeUnit.SECONDS.toMillis(2))
- static final **TimeFormat BULLET** = new **TimeFormat**(TimeUnit.MINUTES.toMillis(1))
- static final **TimeFormat BULLET2** = new **TimeFormat**(TimeUnit.MINUTES.toMillis(2))
- static final **TimeFormat[] PRESETS** = {RAPID, ULTRA\_BULLET, BULLET, BULLET2}
- static final int **numOfFields** = 2

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/GameSetup/TimeFormat.java

### 3.101 TimeFormatComponent

Inheritance diagram for TimeFormatComponent:

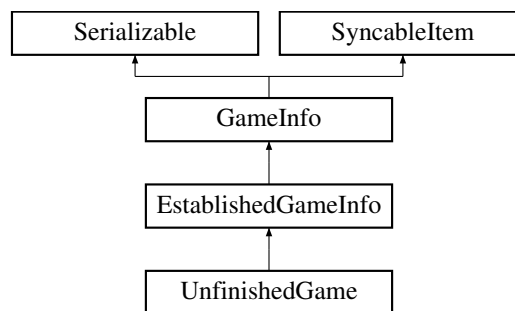


The documentation for this interface was generated from the following file:

- `ver14/SharedClasses/Game/GameSetup/TimeFormatComponent.java`

### 3.102 UnfinishedGame

Inheritance diagram for UnfinishedGame:



#### Public Attributes

- final `PlayerColor` `playerColorToMove`
- final `String` `playerToMove`

#### Additional Inherited Members

#### 3.102.1 Detailed Description

The type Unfinished game.

Author

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

#### 3.102.2 Constructor & Destructor Documentation

### 3.102.2.1 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

## 3.102.3 Member Function Documentation

### 3.102.3.1 isCreatorToMove()

```
boolean isCreatorToMove ( )
```

Is creator to move boolean.

#### Returns

the boolean

## 3.102.4 Member Data Documentation

### 3.102.4.1 playerColorToMove

```
final PlayerColor playerColorToMove
```

The Player color to move.

### 3.102.4.2 playerToMove

```
final String playerToMove
```

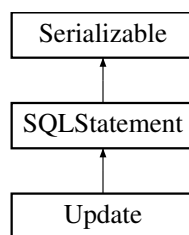
The Player to move.

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Game/SavedGames/UnfinishedGame.java

## 3.103 Update

Inheritance diagram for Update:



### Classes

- class **NewValue**

### Additional Inherited Members

#### 3.103.1 Detailed Description

[Update](#) - a sql update statement.

Author

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

#### 3.103.2 Constructor & Destructor Documentation

##### 3.103.2.1 Update()

```
Update (
    Table updating,
    Condition condition,
    NewValue... newValues )
```

Instantiates a new [Update](#).

## Parameters

<i>updating</i>	the updating
<i>condition</i>	the condition
<i>newValues</i>	the new values

### 3.103.3 Member Function Documentation

#### 3.103.3.1 createStatement()

```
String createStatement ( ) [protected]
```

Create statement string.

## Returns

the string

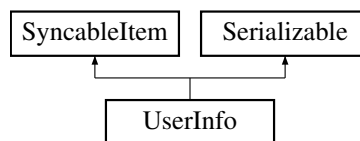
Reimplemented from [SQLStatement](#).

The documentation for this class was generated from the following file:

- ver14/SharedClasses/DBActions/Statements/Update.java

## 3.104 UserInfo

Inheritance diagram for UserInfo:



### Public Attributes

- final String **id**
- final String **profilePic**

### 3.104.1 Member Function Documentation

### 3.104.1.1 ID()

```
String ID ( )
```

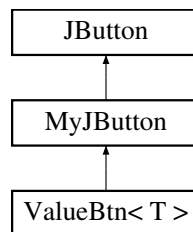
Implements [SyncableItem](#).

The documentation for this class was generated from the following file:

- ver14/SharedClasses/Sync/UserInfo.java

## 3.105 ValueBtn< T >

Inheritance diagram for ValueBtn< T >:



### 3.105.1 Detailed Description

Value btn - a value holding button.

#### Parameters

< T >	the value's type
-------	------------------

#### Author

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

### 3.105.2 Constructor & Destructor Documentation

#### 3.105.2.1 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

The documentation for this class was generated from the following file:

- ver14/SharedClasses/UI/Buttons/ValueBtn.java

## 3.106 PreMadeRequest.VariationCreator

### 3.106.1 Detailed Description

Variation creator - .

## Author

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

### 3.106.2 Member Function Documentation

#### 3.106.2.1 create()

```
Variation create (
    RequestBuilder actualBuilder )
```

Create variation.

## Parameters

<i>actualBuilder</i>	the actual builder
----------------------	--------------------

## Returns

the variation

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/DBActions/DBRequest/PreMadeRequest.java

## 3.107 VoidCallback

### 3.107.1 Detailed Description

represents a callback with no object attached to it.

#### Author

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

### 3.107.2 Member Function Documentation

#### 3.107.2.1 callback()

```
void callback ( )
```

[Callback](#).

The documentation for this interface was generated from the following file:

- ver14/SharedClasses/Callbacks/VoidCallback.java



# Index

- add
  - Condition, [61](#)
- addDetail
  - Evaluation, [100](#)
- addedRes
  - DBResponse, [81](#)
- addShouldSync
  - RequestBuilder, [251](#)
- addSuggestion
  - Config< V >, [66](#)
- AiParameters, [9](#)
  - setTimeFormat, [10](#)
- ALL\_DIRECTIONS
  - Direction, [90](#)
- ALL\_PIECES
  - Piece, [222](#)
- ALL\_USED\_DIRECTIONS
  - Direction, [90](#)
- and
  - Condition, [63](#)
- andWith
  - Direction, [90](#)
- annotate
  - MoveAnnotation, [202](#)
- annotation
  - GameStatus.GameStatusType, [122](#)
- Answer
  - Question, [241](#)
- AnswerCallback, [10](#)
- answerQuestion
  - Message, [161](#)
- Any
  - Regex, [248](#)
- ANY\_LOGIN
  - AuthSettings, [27](#)
- apply
  - Math, [150](#)
- AppSocket, [10](#)
  - AppSocket, [11](#)
  - close, [12](#)
  - getLocalAddress, [14](#)
  - getMessagesHandler, [13](#)
  - getRemoteAddress, [14](#)
  - handledRun, [13](#)
  - interruptListener, [12](#)
  - isClosed, [15](#)
  - isConnected, [14](#)
  - msgSocket, [16](#)
  - requestMessage, [12](#), [15](#)
  - respond, [14](#)
  - setMessagesHandler, [13](#)
  - stopReading, [15](#)
  - writeMessage, [14](#)
- ArchivedGameInfo, [16](#)
  - ArchivedGameInfo, [16](#)
  - getWinner, [17](#)
  - toString, [17](#)
- Arg, [17](#)
  - Arg, [18](#)
  - argType, [21](#)
  - config, [21](#)
  - createVal, [20](#)
  - equals, [19](#)
  - escape, [20](#)
  - isUserInput, [19](#)
  - replnStr, [20](#)
  - setUserInput, [19](#)
  - toString, [20](#)
- args
  - RequestBuilder, [255](#)
- ArgsUtil, [21](#)
  - create, [21](#)
  - equalsSign, [22](#)
  - OptionalArg, [22](#)
  - plainTextIgnoreCase, [22](#)
- ArgType, [23](#)
  - ArgType, [23](#)
  - Date, [24](#)
  - DateRange, [24](#)
  - isUserInput, [25](#)
  - Number, [24](#)
  - Password, [24](#)
  - PictureUrl, [25](#)
  - ServerAddress, [24](#)
  - Text, [24](#)
  - Url, [25](#)
  - Username, [24](#)
- argType
  - Arg, [21](#)
- ArrUtils, [25](#)
  - concat, [25](#)
  - createList, [26](#)
  - exists, [26](#)
- as
  - Col, [53](#)
- asFloat
  - Math, [149](#)
- asInt

- CastlingRights.Side, 262
  - Direction, 90
  - Location, 140
  - PieceType, 230
  - PlayerColor, 234
- askForGameSettings
  - Message, 159
- askForLogin
  - Message, 155
- askQuestion
  - Message, 161
- asLong
  - Location, 140
- assertNotGameOver
  - Evaluation, 100
- asUser
  - LoginInfo, 144
  - LoginType, 145
- ATTACKING\_DIRECTIONS
  - PieceType, 230
- ATTACKING\_PIECE\_TYPES
  - PieceType, 230
- AuthSettings, 27
  - ANY\_LOGIN, 27
  - GUEST, 27
  - NEVER\_AUTH, 27
  - NO\_AUTH, 28
  - USER, 27
- authSettings
  - PreMadeRequest, 237
- awful
  - StrUtils, 286
- B\_B
  - Piece, 222
- B\_K
  - Piece, 222
- B\_N
  - Piece, 222
- B\_P
  - Piece, 221
- B\_Q
  - Piece, 222
- B\_R
  - Piece, 222
- basicAnnotate
  - MoveAnnotation, 203
- BasicMove, 28
  - BasicMove, 28, 29
  - cp, 32
  - createBatch, 29
  - equals, 31
  - flip, 30
  - getBasicMoveAnnotation, 32
  - getFlipped, 30
  - getMovingFrom, 30
  - getMovingTo, 31
  - hashCode, 31
  - setMovingFrom, 30
  - setMovingTo, 31
  - toString, 32
- between
  - Condition, 62
- BISHOP
  - PieceType, 228
- BitData, 33
  - everything, 33
  - notAFile, 33
  - notHFile, 33
- BLACK
  - PlayerColor, 233
- blackIcon
  - PieceType, 230
- blockTilRes
  - MessagesHandler, 177
- Board, 34
  - Board, 34, 35
  - example, 39
  - fenSetup, 36
  - getPiece, 37
  - getRow, 36
  - getSquare, 35
  - isSquareEmpty, 38
  - iterator, 37
  - print, 37
  - setPiece, 35
  - setSquareEmpty, 38
  - startingFen, 39
  - startingPos, 36
  - toString, 38
- book
  - Evaluation, 100
- build
  - RequestBuilder, 255
- bye
  - Message, 160
- callback
  - Callback< T >, 41
  - ThrowingCallback< T >, 302
  - VoidCallback, 310
- Callback< T >, 40
  - callback, 41
- CAN\_PROMOTE\_TO
  - PieceType, 229
- canBeEmpty
  - Regex, 244
- CANCEL
  - LoginType, 146
- canUseDefault
  - Config< V >, 68
- canUseld
  - IDsGenerator, 128
- castledKingCol
  - CastlingRights.Side, 262
- castledRookCol
  - CastlingRights.Side, 262
- castling

- Move, [192](#)
- CASTLING\_FLAGS
  - Move.MoveFlag, [205](#)
- castlingNotation
  - CastlingRights.Side, [262](#)
- CastlingRights, [41](#)
  - CastlingRights, [42](#)
  - createFromStr, [42](#)
  - disableCastling, [44](#)
  - enable, [45](#)
  - enableCastling, [43](#)
  - getPlayersCastling, [46](#)
  - getRights, [45](#)
  - hasAny, [45](#)
  - isEnabled, [44](#)
  - main, [43](#)
  - NO\_CASTLING\_ABILITY, [46](#)
  - toString, [44](#)
  - whosCastling, [43](#)
- CastlingRights.Side, [260](#)
  - asInt, [262](#)
  - castledKingCol, [262](#)
  - castledRookCol, [262](#)
  - castlingNotation, [262](#)
  - KING, [261](#)
  - kingFinalLoc, [261](#)
  - kingTravelDistance, [262](#)
  - mult, [262](#)
  - QUEEN, [261](#)
  - rookStartingCol, [262](#)
  - Side, [260](#)
  - SIDES, [262](#)
  - toString, [261](#)
- castlingSide
  - Move.MoveFlag, [206](#)
- changePassword
  - RequestBuilder, [251](#)
- ChangeProfilePic
  - PreMadeRequest, [237](#)
- changeProfilePic
  - RequestBuilder, [251](#)
- CHECK
  - GameStatus.GameStatusType, [122](#)
- check
  - Regex, [246](#)
- checkClosingVal
  - MyJFrame.BooleanClosing, [40](#)
  - MyJFrame.Closing< T >, [47](#)
  - MyJFrame.StringClosing, [274](#)
- Checkmate
  - GameStatus.SpecificStatus, [264](#)
- checkmate
  - GameStatus, [113](#)
- checkUsernameAvailability
  - Message, [161](#)
- clean
  - DBResponse, [80](#)
  - StatusResponse, [273](#)
- StrUtils, [278](#)
- TableDBResponse, [298](#)
- close
  - AppSocket, [12](#)
- closing
  - MyJFrame.BooleanClosing, [40](#)
  - MyJFrame.Closing< T >, [47](#)
- Col, [48](#)
  - as, [53](#)
  - Col, [49](#), [50](#)
  - colName, [54](#)
  - count, [50](#)
  - countIf, [51](#)
  - CreatedDateTime, [58](#)
  - date, [52](#)
  - GameID, [57](#)
  - label, [51](#)
  - math, [55](#), [57](#)
  - nested, [55](#)
  - of, [54](#)
  - Password, [58](#)
  - Player1, [58](#)
  - Player2, [58](#)
  - PlayerToMove, [58](#)
  - ProfilePic, [59](#)
  - replace, [55](#)
  - SavedGame, [58](#)
  - setColName, [54](#)
  - setWrapped, [53](#)
  - sum, [51](#)
  - switchCase, [52](#)
  - time, [52](#)
  - toString, [57](#)
  - Username, [58](#)
  - Winner, [58](#)
  - wrap, [52](#)
- col
  - Location, [140](#)
  - Math, [152](#)
- colName
  - Col, [54](#)
- cols
  - Table, [295](#)
- columns
  - TableDBResponse, [300](#)
- combination
  - Direction, [90](#)
- compareMovementType
  - PieceType, [225](#)
- compareTo
  - MinimaxMove, [189](#)
  - Move, [201](#)
- concat
  - ArrUtils, [25](#)
- Condition, [59](#)
  - add, [61](#)
  - and, [63](#)
  - between, [62](#)

- Condition, 60
- equals, 60
- getStr, 63
- math, 62
- noNulls, 60
- notEquals, 62
- setStr, 63
- toString, 64
- wrap, 62
- condition
  - SwitchCase, 290
- Config
  - Config< V >, 65, 66
- config
  - Arg, 21
- Config< V >, 64
  - addSuggestion, 66
  - canUseDefault, 68
  - Config, 65, 66
  - description, 68
  - getDefault, 67
  - getDefaultDesc, 68
  - getDescribedDefault, 67
  - getValuesSuggestion, 67
  - toString, 67
- confirm
  - ConfirmDialogs, 69
- ConfirmDialogs, 68
  - confirm, 69
  - main, 69
- CONNECTED\_USERS
  - SyncedListType, 292
- convertFromCentipawns
  - Evaluation, 103
- copyMove
  - Move, 193
- count
  - Col, 50
- countIf
  - Col, 51
- countMatches
  - StrUtils, 276
- cp
  - BasicMove, 32
- create
  - ArgsUtil, 21
  - PreMadeRequest.VariationCreator, 309
- createBatch
  - BasicMove, 29
- createBuilder
  - PreMadeRequest, 236
- createdAt
  - EstablishedGameInfo, 98
- CreatedDateTime
  - Col, 58
- CreatedGame, 71
  - CreatedGame, 71
  - getGameDesc, 72
- createDisconnectedError
  - MessagesHandler, 179
- createFromStr
  - CastlingRights, 42
- createList
  - ArrUtils, 26
- createMatIndicesStr
  - Location, 133
- createResponse
  - RequestBuilder, 253
- createStatement
  - CustomStatement, 73
  - Delete, 82
  - Selection, 258
  - SQLStatement, 267
  - Update, 307
- createThread
  - ThreadsManager, 300
- createTimeGreeting
  - StrUtils, 286
- createTimeStr
  - StrUtils, 286
- createVal
  - Arg, 20
- createVariation
  - RequestBuilder, 251
- creatorUsername
  - GameInfo, 111
- criticalErr
  - EnvManager, 94
- currentThread
  - MyThread, 216
- CustomStatement, 72
  - createStatement, 73
  - CustomStatement, 73
- D
  - Direction, 85
- D\_D
  - Direction, 86
- D\_D\_L
  - Direction, 88
- D\_D\_R
  - Direction, 88
- D\_L
  - Direction, 87
- D\_L\_L
  - Direction, 89
- D\_R
  - Direction, 87
- D\_R\_R
  - Direction, 89
- Date
  - ArgType, 24
- date
  - Col, 52
- DateRange
  - ArgType, 24
- dateTimePrefix

- StrUtils, 287
- DBRequest, 73
  - DBRequest, 74, 75
  - getBuilder, 75
  - getRequest, 76
  - getSubRequest, 75
  - setSubRequest, 76
  - toString, 76
  - type, 77
- dbRequest
  - Message, 163
- DBResponse, 77
  - addedRes, 81
  - clean, 80
  - DBResponse, 78
  - getAddedRes, 79
  - getRequest, 78
  - getStatus, 78
  - isAnyData, 79
  - isSuccess, 78
  - print, 80
  - request, 80
  - setAddedRes, 79
  - status, 80
  - toString, 80
- debugAdapter
  - MyJFrame, 214
- defaultCase
  - SwitchCase, 290
- Delete, 81
  - createStatement, 82
  - Delete, 81
- deleteAllUnFinishedGames
  - RequestBuilder, 252
- DeleteUnfGames
  - PreMadeRequest, 236
- description
  - Config< V >, 68
- diff
  - PlayerColor, 235
- Direction, 82
  - ALL\_DIRECTIONS, 90
  - ALL\_USED\_DIRECTIONS, 90
  - andWith, 90
  - asInt, 90
  - combination, 90
  - D, 85
  - D\_D, 86
  - D\_D\_L, 88
  - D\_D\_R, 88
  - D\_L, 87
  - D\_L\_L, 89
  - D\_R, 87
  - D\_R\_R, 89
  - getCombination, 84
  - getDirectionByOffset, 83
  - getRelative, 83
  - L, 85
  - normalPerspective, 90
  - NUM\_OF\_DIRECTIONS, 89
  - NUM\_OF\_DIRECTIONS\_WO\_KNIGHT, 89
  - NUM\_OF\_KNIGHT\_DIRECTIONS, 89
  - offset, 90
  - opposite, 84
  - perspective, 84
  - R, 85
  - U, 85
  - U\_L, 86
  - U\_L\_L, 88
  - U\_R, 86
  - U\_R\_R, 88
  - U\_U, 86
  - U\_U\_L, 87
  - U\_U\_R, 87
- disableCastling
  - CastlingRights, 44
- Div
  - Math, 152
- DONE
  - MyThread.ThreadStatus, 302
- dontCapFull
  - StrUtils, 276
- dontCapWord
  - StrUtils, 280
- dontMatch
  - Regex, 248
- DontSaveGame
  - Regex, 247
- DoublePawnPush
  - Move.MoveFlag, 205
- doXClick
  - MyJFrame, 215
- drawOffer
  - Question, 239
- EfficientGen
  - EfficientGen< K, V >, 91
- EfficientGen< K, V >, 91
  - EfficientGen, 91
  - get, 91
- EG\_WEIGHT
  - EvaluationParameters, 106
- EMPTY\_PIECE
  - Square, 272
- enable
  - CastlingRights, 45
- enableCastling
  - CastlingRights, 43
- endPress
  - MyJFrame.MyAdapter.HeldDown, 127
- EnPassant
  - Move.MoveFlag, 205
- Environment, 93
  - IS\_JAR, 93
- EnvManager, 93
  - criticalErr, 94
  - handledErr, 93

- equals
  - Arg, 19
  - BasicMove, 31
  - Condition, 60
  - Evaluation, 102
  - GameInfo, 109
  - MinimaxMove, 189
  - Move, 199
  - Move.MoveFlag, 204
  - Question, 241
  - SwitchCase, 288
- equalsSign
  - ArgsUtil, 22
- error
  - Message, 160
- ErrorHandler< E extends MyError >, 94
  - handle, 95
  - ignore, 94
- errToString
  - MyError, 208
- escape
  - Arg, 20
- escapeValues
  - Table, 294
- EstablishedGameInfo, 95
  - createdAt, 98
  - EstablishedGameInfo, 96
  - getCreatedAt, 96
  - getGameDesc, 97
  - getMoveStack, 97
  - opponentUsername, 98
  - setCreatedAt, 96
  - toString, 97
- Evaluation, 98
  - addDetail, 100
  - assertNotGameOver, 100
  - book, 100
  - convertFromCentipawns, 103
  - equals, 102
  - Evaluation, 98, 99
  - EvaluationDetail, 104
  - flipEval, 104
  - getEval, 102
  - getEvaluationDepth, 100
  - getEvaluationFor, 103
  - getGameStatus, 102
  - isCheck, 101
  - isGameOver, 101
  - isGreaterThan, 101
  - LOSS\_EVAL, 105
  - print, 104
  - setEval, 102
  - setEvaluationDepth, 100
  - setPerspective, 103
  - TIE\_EVAL, 104
  - toString, 103
  - WIN\_EVAL, 104
- EvaluationDetail
  - Evaluation, 104
- EvaluationParameters, 105
  - EG\_WEIGHT, 106
  - FORCE\_KING\_TO\_CORNER, 106
  - HANGING\_PIECES, 106
  - KING\_SAFETY, 106
  - MATERIAL, 105
  - MOVEMENT\_ABILITY, 106
  - PIECE\_TABLES, 105
  - SQUARE\_CONTROL, 106
  - STOCKFISH\_SAYS, 106
  - weight, 106
- everything
  - BitData, 33
- example
  - Board, 39
  - GameInfo, 108
- execute
  - Math, 149
- exists
  - ArrUtils, 26
- Fen
  - RegEx, 246
- fenSetup
  - Board, 36
- FiftyMoveRule
  - GameStatus.SpecificStatus, 265
- fiftyMoveRule
  - GameStatus, 114
- fitInside
  - StrUtils, 283
- fixHtml
  - StrUtils, 284
- flip
  - BasicMove, 30
  - Location, 137, 138
- flipEval
  - Evaluation, 104
- flipMove
  - Move, 193
- FontManager, 107
- FORCE\_KING\_TO\_CORNER
  - EvaluationParameters, 106
- format
  - StrUtils, 277
- formatDate
  - StrUtils, 279
- formatDateSQL
  - StrUtils, 281
- formatNum
  - Math, 147, 148
- GAME\_GOES\_ON
  - GameStatus.GameStatusType, 122
- GameGoesOn
  - GameStatus.SpecificStatus, 265
- gameGoesOn
  - GameStatus, 114

- GameID
  - Col, [57](#)
- gameId
  - GameInfo, [111](#)
- GameInfo, [107](#)
  - creatorUsername, [111](#)
  - equals, [109](#)
  - example, [108](#)
  - gameId, [111](#)
  - GameInfo, [108](#)
  - gameSettings, [111](#)
  - getGameDesc, [110](#)
  - getJoiningPlayerColor, [110](#)
  - getStartingColor, [110](#)
  - ID, [109](#)
  - isCreator, [108](#)
  - toString, [109](#)
- gameOver
  - Message, [158](#)
- gameOverStr
  - GameStatus.GameStatusType, [122](#)
- Games
  - PreMadeRequest, [236](#)
  - Table, [294](#)
- games
  - RequestBuilder, [252](#)
- GameSettings, [111](#)
  - setTimeFormat, [112](#)
- gameSettings
  - GameInfo, [111](#)
- GameStatus, [112](#)
  - checkmate, [113](#)
  - fiftyMoveRule, [114](#)
  - gameGoesOn, [114](#)
  - getCheckedKingLoc, [117](#)
  - getDetailedStr, [118](#), [119](#)
  - getGameStatusType, [118](#)
  - getSpecificStatus, [120](#)
  - getWinningColor, [117](#)
  - insufficientMaterial, [116](#)
  - isCheck, [118](#)
  - isDisconnected, [117](#)
  - isGameOver, [119](#)
  - playerDisconnected, [116](#)
  - playerResigned, [117](#)
  - serverStoppedGame, [115](#)
  - setCustomStr, [115](#)
  - setDepth, [119](#)
  - setInCheck, [118](#)
  - stalemate, [114](#)
  - threeFoldRepetition, [115](#)
  - tieByAgreement, [114](#)
  - timedOut, [116](#)
  - toString, [119](#)
- GameStatus.GameStatusType, [120](#)
  - annotation, [122](#)
  - CHECK, [122](#)
  - GAME\_GOES\_ON, [122](#)
  - gameOverStr, [122](#)
  - GameStatusType, [121](#)
  - isGameOver, [121](#)
  - TIE, [122](#)
  - UNFINISHED, [122](#)
  - WIN\_OR\_LOSS, [122](#)
- GameStatus.SpecificStatus, [263](#)
  - Checkmate, [264](#)
  - FiftyMoveRule, [265](#)
  - GameGoesOn, [265](#)
  - gameStatusType, [266](#)
  - InsufficientMaterial, [265](#)
  - PlayerDisconnectedVsAi, [265](#)
  - PlayerDisconnectedVsReal, [266](#)
  - Resignation, [264](#)
  - ServerStoppedGame, [266](#)
  - SpecificStatus, [263](#)
  - Stalemate, [265](#)
  - ThreeFoldRepetition, [265](#)
  - TieByAgreement, [265](#)
  - TimedOut, [264](#)
  - TimedOutVsInsufficientMaterial, [264](#)
  - toString, [264](#)
- GameStatusType
  - GameStatus.GameStatusType, [121](#)
- gameStatusType
  - GameStatus.SpecificStatus, [266](#)
- GameTime, [123](#)
- GameView, [123](#)
- generate
  - IDsGenerator, [128](#)
- get
  - EfficientGen< K, V >, [91](#)
  - RegEx, [245](#)
- getAddedRes
  - DBResponse, [79](#)
- getAnnotation
  - Move, [199](#)
- getArgs
  - RequestBuilder, [254](#)
- getArgVal
  - RequestBuilder, [253](#)
- getAttackingDirections
  - PieceType, [224](#), [227](#)
- getAvailable
  - Message, [169](#)
- getBasicMoveAnnotation
  - BasicMove, [32](#)
- getBoard
  - Message, [174](#)
- getBuilder
  - DBRequest, [75](#)
- getCapturingPieceType
  - Move, [198](#)
- getCell
  - TableDBResponse, [297](#), [298](#)
- getCheckedKingLoc
  - GameStatus, [117](#)

- getColor
  - PlayerColor, 232
- getColString
  - Location, 138
- getColumnIndex
  - TableDBResponse, 297
- getColumns
  - TableDBResponse, 299
- getCombination
  - Direction, 84
- getCreatedAt
  - EstablishedGameInfo, 96
- getDBRequest
  - Message, 164
- getDBResponse
  - Message, 165
- getDefault
  - Config< V >, 67
- getDefaultAnswer
  - Question, 240
- getDefaultDesc
  - Config< V >, 68
- getDescribedDefault
  - Config< V >, 67
- getDetailedStr
  - GameStatus, 118, 119
- getDetails
  - RegEx, 245
  - StatusResponse, 273
- getDirectionByOffset
  - Direction, 83
- getDisabledCastling
  - Move, 194
- getEnPassantLoc
  - Move, 201
- getError
  - Message, 164
- getEval
  - Evaluation, 102
- getEvaluationDepth
  - Evaluation, 100
- getEvaluationFor
  - Evaluation, 103
- getFen
  - Piece, 220
  - Square, 271
- getFirstRow
  - TableDBResponse, 297
- getFlipped
  - BasicMove, 30
- getGameDesc
  - CreatedGame, 72
  - EstablishedGameInfo, 97
  - GameInfo, 110
- getGameSettings
  - Message, 170
- getGameStatus
  - Evaluation, 102
  - Message, 173
- getGameStatusType
  - GameStatus, 118
- getGameTime
  - Message, 172
- getHandledStr
  - MyError, 208
- getIntermediateMove
  - Move, 196
- getJoiningPlayerColor
  - GameInfo, 110
- getLoc
  - Location, 134–136
  - Square, 270
- getLocalAddress
  - AppSocket, 14
- getLoginInfo
  - Message, 174
- getLoginType
  - LoginInfo, 142
- getMaxDistance
  - Location, 139
- getMessagesHandler
  - AppSocket, 13
- getMessageType
  - Message, 175
- getMinSize
  - MyJButton, 212
- getMove
  - Message, 172
  - MinimaxMove, 187
- getMoveDepth
  - MinimaxMove, 187
- getMoveEvaluation
  - MinimaxMove, 188
  - Move, 197
- getMoveFlag
  - Move, 200
- getMoveStack
  - EstablishedGameInfo, 97
  - Message, 166
- getMovingColor
  - Move, 194
- getMovingFrom
  - BasicMove, 30
- getMovingTo
  - BasicMove, 31
- getMyAdapter
  - MyJFrame, 214
- getName
  - PlayerColor, 233
  - RequestBuilder, 254
- getOpponent
  - PlayerColor, 232
- getOtherPlayer
  - Message, 168
- getPassword
  - LoginInfo, 143



- getPiece
  - Board, [37](#)
  - Piece, [219](#)
  - Square, [270](#)
- getPieceFromFen
  - Piece, [218](#)
- getPiecelcon
  - Piece, [219](#)
  - PieceType, [225](#)
  - Square, [271](#)
- getPieceName
  - PieceType, [227](#)
- getPieceType
  - PieceType, [224](#)
- getPlayerColor
  - Message, [171](#)
- getPlayerFromFen
  - PlayerColor, [232](#)
- getPlayersCastling
  - CastlingRights, [46](#)
- getPort
  - StrUtils, [281](#)
- getPossibleAnswers
  - Question, [241](#)
- getPossibleMoves
  - Message, [173](#)
- getPostDescription
  - RequestBuilder, [254](#)
- getPreDescription
  - RequestBuilder, [254](#)
- getPreMoves
  - Message, [170](#)
- getPrevFullMoveClock
  - Move, [195](#)
- getPrevHalfMoveClock
  - Move, [195](#)
- getProfilePic
  - LoginInfo, [142](#)
- getPromotingTo
  - Move, [196](#)
- getQuestion
  - Message, [168](#)
- getQuestionStr
  - Question, [240](#)
- getRegex
  - RegEx, [246](#)
- getRelative
  - Direction, [83](#)
- getRemoteAddress
  - AppSocket, [14](#)
- getRequest
  - DBRequest, [76](#)
  - DBResponse, [78](#)
- getRequestVariations
  - PreMadeRequest, [236](#)
- getRespondingToMsgId
  - Message, [167](#)
- getRights
  - CastlingRights, [45](#)
- getRow
  - Board, [36](#)
- getRows
  - TableDBResponse, [299](#)
- getRowString
  - Location, [139](#)
- getShortPrintingStr
  - MinimaxMove, [189](#)
- getShouldSync
  - RequestBuilder, [253](#)
- getSpecificStatus
  - GameStatus, [120](#)
- getSquare
  - Board, [35](#)
- getStartingColor
  - GameInfo, [110](#)
- getStatement
  - SQLStatement, [268](#)
- getStatus
  - DBResponse, [78](#)
- getStr
  - Condition, [63](#)
- getSubject
  - Message, [171](#)
- getSubRequest
  - DBRequest, [75](#)
- getSyncedLists
  - Message, [166](#)
- getUrl
  - StrUtils, [283](#)
- getUsername
  - LoginInfo, [143](#)
  - Message, [169](#)
- getUsernameSuggestions
  - Message, [165](#)
- getValuesSuggestion
  - Config< V >, [67](#)
- getWalkingDirections
  - PieceType, [227](#)
- getWhitePieceFen
  - PieceType, [224](#)
- getWinner
  - ArchivedGameInfo, [17](#)
- getWinningColor
  - GameStatus, [117](#)
- Graphable, [124](#)
- GraphableDBResponse, [124](#)
  - toString, [124](#)
- GraphElement, [125](#)
- GraphElementType, [125](#)
- GUEST
  - AuthSettings, [27](#)
  - LoginType, [146](#)
- handle
  - ErrorHandler< E extends MyError >, [95](#)
- handledErr
  - EnvManager, [93](#)

- handledRun
  - AppSocket, 13
  - HandledThread, 127
  - MyThread, 217
- HandledThread, 125
  - handledRun, 127
  - HandledThread, 126
  - runInHandledThread, 126
  - setRunnable, 126
- handleErrors
  - ThreadsManager, 300
- HANGING\_PIECES
  - EvaluationParameters, 106
- hasAny
  - CastlingRights, 45
- hash
  - Location, 139
- hashCode
  - BasicMove, 31
  - MinimaxMove, 189
- header
  - MyJFrame.Closing< T >, 48
- htmlNewLines
  - StrUtils, 276
- Icon
  - RegEx, 247
- icon
  - MyJFrame.Closing< T >, 48
- ID
  - GameInfo, 109
  - SyncableItem, 291
  - UserInfo, 307
- IDsGenerator, 128
  - canUsed, 128
  - generate, 128
- ifTrue
  - SwitchCase, 290
- ignore
  - ErrorHandler< E extends MyError >, 94
- ignoreErrs
  - MyThread, 217
- indexOf2
  - PlayerColor, 234
- initDebugLoginValues
  - LoginInfo, 144
- initGame
  - Message, 156
- InsufficientMaterial
  - GameStatus.SpecificStatus, 265
- insufficientMaterial
  - GameStatus, 116
- interrupt
  - Message, 159
- interruptBlocking
  - MessagesHandler, 177
- interruptListener
  - AppSocket, 12
- IPPortAddress
  - RegEx, 247
- IS\_JAR
  - Environment, 93
- isAbsoluteUrl
  - StrUtils, 275
- isAnyData
  - DBResponse, 79
  - StatusResponse, 273
  - TableDBResponse, 298
- isAttack
  - PieceType, 227
- isBlackSquare
  - Location, 138
- isCapturing
  - Move, 198
- isCastling
  - Move.MoveFlag, 205
- isCheck
  - Evaluation, 101
  - GameStatus, 118
  - Move, 197
- isClosed
  - AppSocket, 15
- isConnected
  - AppSocket, 14
- isCreator
  - GameInfo, 108
- isCreatorToMove
  - UnfinishedGame, 305
- isDeeperAndBetterThan
  - MinimaxMove, 187
- isDiagonalPiece
  - PieceType, 226
- isDisconnected
  - GameStatus, 117
- isEmpty
  - Square, 271
  - StrUtils, 278
- isEnabled
  - CastlingRights, 44
- isGameOver
  - Evaluation, 101
  - GameStatus, 119
  - GameStatus.GameStatusType, 121
- isGreaterThan
  - Evaluation, 101
- isGuest
  - LoginInfo, 142
- isInBounds
  - Location, 137
- isLinePiece
  - PieceType, 226
- isOnMyTeam
  - Piece, 220
- isResponse
  - Message, 168
- isReversible
  - Move, 198

- isSavedDate
  - RegEx, [244](#)
- isSliding
  - PieceType, [230](#)
- isSquareEmpty
  - Board, [38](#)
- isSubject
  - Message, [175](#)
- isSuccess
  - DBResponse, [78](#)
- isUserInput
  - Arg, [19](#)
  - ArgType, [25](#)
- isWhiteSquare
  - Location, [138](#)
- iteration
  - StrUtils.IterationThingy< T >, [129](#)
- iterator
  - Board, [37](#)
- join
  - Selection, [259](#)
- JOINABLE\_GAMES
  - SyncedListType, [292](#)
- key
  - MyJFrame.MyAdapter.HeldDown, [128](#)
- KING
  - CastlingRights.Side, [261](#)
  - PieceType, [229](#)
- KING\_SAFETY
  - EvaluationParameters, [106](#)
- kingFinalLoc
  - CastlingRights.Side, [261](#)
- kingTravelDistance
  - CastlingRights.Side, [262](#)
- KNIGHT
  - PieceType, [228](#)
- L
  - Direction, [85](#)
- label
  - Col, [51](#)
- LazyHashSupplier< T >, [130](#)
- LinkLabel, [130](#)
  - setText, [130](#)
- Location, [131](#)
  - asInt, [140](#)
  - asLong, [140](#)
  - col, [140](#)
  - createMatIndicesStr, [133](#)
  - flip, [137](#), [138](#)
  - getColString, [138](#)
  - getLoc, [134](#)–[136](#)
  - getMaxDistance, [139](#)
  - getRowString, [139](#)
  - hash, [139](#)
  - isBlackSquare, [138](#)
  - isInBounds, [137](#)
  - isWhiteSquare, [138](#)
  - Location, [132](#)
  - locsMat, [133](#)
  - locsMatStr, [133](#)
  - matrixStr, [133](#)
  - row, [140](#)
  - toString, [138](#)
  - valueOf, [135](#)
- locsMat
  - Location, [133](#)
- locsMatStr
  - Location, [133](#)
- log
  - MathUtils, [153](#)
- LOGIN
  - LoginType, [145](#)
- LoginInfo, [140](#)
  - asUser, [144](#)
  - getLoginType, [142](#)
  - getPassword, [143](#)
  - getProfilePic, [142](#)
  - getUsername, [143](#)
  - initDebugLoginValues, [144](#)
  - isGuest, [142](#)
  - LoginInfo, [141](#)
  - setLoginType, [142](#)
  - setPassword, [143](#)
  - setProfilePic, [142](#)
  - setUsername, [143](#)
  - toString, [144](#)
- LoginType, [145](#)
  - asUser, [145](#)
  - CANCEL, [146](#)
  - GUEST, [146](#)
  - LOGIN, [145](#)
  - NOT\_SET\_YET, [146](#)
  - REGISTER, [146](#)
  - toString, [145](#)
- LongCastle
  - Move.MoveFlag, [205](#)
- LOSS\_EVAL
  - Evaluation, [105](#)
- main
  - CastlingRights, [43](#)
  - ConfirmDialogs, [69](#)
  - StrUtils, [286](#)
- MAJOR\_PIECES
  - PieceType, [229](#)
- MATERIAL
  - EvaluationParameters, [105](#)
- Math, [146](#)
  - apply, [150](#)
  - asFloat, [149](#)
  - col, [152](#)
  - Div, [152](#)
  - execute, [149](#)
  - formatNum, [147](#), [148](#)
  - Mult, [152](#)

- nullIf0, 147
- Plus, 152
- simpleOperation, 150
- str, 148
- strSource, 150
- strVal, 147
- zeroIfNull, 149
- math
  - Col, 55, 57
  - Condition, 62
- MathUtils, 153
  - log, 153
- matrixStr
  - Location, 133
- Message, 154
  - answerQuestion, 161
  - askForGameSettings, 159
  - askForLogin, 155
  - askQuestion, 161
  - bye, 160
  - checkUsernameAvailability, 161
  - dbRequest, 163
  - error, 160
  - gameOver, 158
  - getAvailable, 169
  - getBoard, 174
  - getDbRequest, 164
  - getDbResponse, 165
  - getError, 164
  - getGameSettings, 170
  - getGameStatus, 173
  - getGameTime, 172
  - getLoginInfo, 174
  - getMessageType, 175
  - getMove, 172
  - getMoveStack, 166
  - getOtherPlayer, 168
  - getPlayerColor, 171
  - getPossibleMoves, 173
  - getPreMoves, 170
  - getQuestion, 168
  - getRespondingToMsgId, 167
  - getSubject, 171
  - getSyncedLists, 166
  - getUsername, 169
  - getUsernameSuggestions, 165
  - initGame, 156
  - interrupt, 159
  - isResponse, 168
  - isSubject, 175
  - Message, 154, 155
  - messageID, 175
  - returnDBResponse, 163
  - returnGameSettings, 159
  - returnLogin, 156
  - returnMove, 160
  - returnUsernameAvailable, 162
  - returnUsernameNotAvailable, 162
  - setAvailable, 169
  - setBoard, 174
  - setDbRequest, 163
  - setDBResponse, 165
  - setError, 164
  - setGameSettings, 170
  - setGameStatus, 173
  - setGameTime, 172
  - setLoginInfo, 174
  - setMove, 172
  - setMoveStack, 166
  - setOtherPlayer, 168
  - setPlayerColor, 171
  - setPossibleMoves, 173
  - setPreMoves, 171
  - setQuestion, 169
  - setRespondingTo, 167
  - setRespondingToMsgId, 167
  - setSyncedLists, 166
  - setUsername, 170
  - setUsernameSuggestions, 165
  - syncLists, 164
  - throwError, 157
  - updateByMove, 157
  - waitForMatch, 158
  - waitForYourTurn, 158
  - welcomeMessage, 156
- MessageCallback, 176
- messageID
  - Message, 175
- MessagesHandler, 176
  - blockTilRes, 177
  - createDisconnectedError, 179
  - interruptBlocking, 177
  - MessagesHandler, 176
  - noBlockRequest, 177
  - onAddTime, 180
  - onAlive, 184
  - onAnyDisconnection, 179
  - onAnyMsg, 178
  - onBye, 183
  - onCancelQuestion, 177
  - onDBRequest, 183
  - onDBResponse, 183
  - onDisconnected, 178
  - onError, 182
  - onGameOver, 182
  - onGetGameSettings, 180
  - onGetMove, 181
  - onInitGame, 181
  - onInterrupt, 184
  - onIsAlive, 184
  - onLogin, 179
  - onOfferDraw, 180
  - onPlannedDisconnect, 179
  - onQuestion, 182
  - onResign, 179
  - onUnplannedDisconnect, 179

- onUpdateByMove, 182
  - onUpdateSyncedList, 184
  - onUsernameAvailability, 183
  - onWaitForMatch, 181
  - onWaitTurn, 181
  - onWelcomeMessage, 180
  - prepareForDisconnect, 178
  - receivedMessage, 178
  - socket, 185
- MessageType, 185
- MinimaxMove, 186
- compareTo, 189
  - equals, 189
  - getMove, 187
  - getMoveDepth, 187
  - getMoveEvaluation, 188
  - getShortPrintingStr, 189
  - hashCode, 189
  - isDeeperAndBetterThan, 187
  - MinimaxMove, 186, 187
  - setMove, 188
  - setMoveEvaluation, 188
  - toString, 189
- MINOR\_PIECES
- PieceType, 229
- Move, 190
- castling, 192
  - compareTo, 201
  - copyMove, 193
  - equals, 199
  - flipMove, 193
  - getAnnotation, 199
  - getCapturingPieceType, 198
  - getDisabledCastling, 194
  - getEnPassantLoc, 201
  - getIntermediateMove, 196
  - getMoveEvaluation, 197
  - getMoveFlag, 200
  - getMovingColor, 194
  - getPrevFullMoveClock, 195
  - getPrevHalfMoveClock, 195
  - getPromotingTo, 196
  - isCapturing, 198
  - isCheck, 197
  - isReversible, 198
  - Move, 191, 192
  - setCapturing, 198
  - setDisabledCastling, 194
  - setEnPassantLoc, 201
  - setIntermediateMove, 196
  - setMoveAnnotation, 193
  - setMoveEvaluation, 197
  - setMoveFlag, 200
  - setMovingColor, 194
  - setPrevFullMoveClock, 195
  - setPrevHalfMoveClock, 195
  - setPromotingTo, 196
  - setReversible, 198
  - setThreefoldOption, 194
  - strictEquals, 200
  - threefoldClaim, 192
  - toString, 199
- Move.MoveFlag, 203
- CASTLING\_FLAGS, 205
  - castlingSide, 206
  - DoublePawnPush, 205
  - EnPassant, 205
  - equals, 204
  - isCastling, 205
  - LongCastle, 205
  - MoveFlag, 204
  - NormalMove, 204
  - Promotion, 205
  - ShortCastle, 205
- MoveAnnotation, 202
- annotate, 202
  - basicAnnotate, 203
- MoveFlag
- Move.MoveFlag, 204
- MoveFlags, 206
- MOVEMENT\_ABILITY
- EvaluationParameters, 106
- MovesList, 206
- MovesList.CompareMoves, 59
- msgSocket
- AppSocket, 16
- Mult
- Math, 152
- mult
- CastlingRights.Side, 262
- MyError, 207
- errToString, 208
  - getHandledStr, 208
  - MyError, 207, 208
  - toString, 208
- MyJButton, 209
- getMinSize, 212
  - MyJButton, 209–211
  - replaceWithCancel, 212
  - resetState, 212
  - setFont, 212
  - setOnClick, 211
  - setText, 211
- MyJFrame, 213
- debugAdapter, 214
  - doXClick, 215
  - getMyAdapter, 214
  - MyJFrame, 213
  - onClose, 215
  - setOnExit, 214
  - setOnResize, 215
- MyJFrame.BooleanClosing, 39
- checkClosingVal, 40
  - closing, 40
  - show, 39
- MyJFrame.Closing< T >, 46

- checkClosingVal, [47](#)
- closing, [47](#)
- header, [48](#)
- icon, [48](#)
- show, [47](#)
- title, [48](#)
- tryClose, [47](#)
- MyJFrame.MyAdapter.HeldDown, [127](#)
  - endPress, [127](#)
  - key, [128](#)
  - startPress, [127](#)
- MyJFrame.StringClosing, [274](#)
  - checkClosingVal, [274](#)
  - show, [274](#)
- MyLbl, [215](#)
- MyLbl.StringModifier, [275](#)
- MyThread, [216](#)
  - currentThread, [216](#)
  - handledRun, [217](#)
  - ignoreErrs, [217](#)
  - MyThread, [216](#)
  - reactivateErrs, [217](#)
  - stopRun, [217](#)
- MyThread.ThreadStatus, [301](#)
  - DONE, [302](#)
  - NOT\_STARTED, [301](#)
  - RUNNING, [301](#)
- name
  - RequestBuilder, [255](#)
- nested
  - Col, [55](#)
- nestMe
  - Selection, [258](#)
- NEVER\_AUTH
  - AuthSettings, [27](#)
- NO\_AUTH
  - AuthSettings, [28](#)
- NO\_CASTLING\_ABILITY
  - CastlingRights, [46](#)
- NO\_PLAYER
  - PlayerColor, [234](#)
- noBlockRequest
  - MessagesHandler, [177](#)
- noNulls
  - Condition, [60](#)
- NormalMove
  - Move.MoveFlag, [204](#)
- normalPerspective
  - Direction, [90](#)
- NOT\_SET\_YET
  - LoginType, [146](#)
- NOT\_STARTED
  - MyThread.ThreadStatus, [301](#)
- notAFile
  - BitData, [33](#)
- notEquals
  - Condition, [62](#)
- notHFile
  - BitData, [33](#)
- nullIf0
  - Math, [147](#)
- NUM\_OF\_DIRECTIONS
  - Direction, [89](#)
- NUM\_OF\_DIRECTIONS\_WO\_KNIGHT
  - Direction, [89](#)
- NUM\_OF\_KNIGHT\_DIRECTIONS
  - Direction, [89](#)
- NUM\_OF\_PIECE\_TYPES
  - PieceType, [229](#)
- NUM\_OF\_PLAYERS
  - PlayerColor, [234](#)
- Number
  - ArgType, [24](#)
- Numbers
  - Regex, [247](#)
- numOfRows
  - TableDBResponse, [296](#)
- of
  - Col, [54](#)
- offset
  - Direction, [90](#)
- onAddTime
  - MessagesHandler, [180](#)
- onAlive
  - MessagesHandler, [184](#)
- onAnyDisconnection
  - MessagesHandler, [179](#)
- onAnyMsg
  - MessagesHandler, [178](#)
- onBye
  - MessagesHandler, [183](#)
- onCancelQuestion
  - MessagesHandler, [177](#)
- onClose
  - MyJFrame, [215](#)
- onDBRequest
  - MessagesHandler, [183](#)
- onDBResponse
  - MessagesHandler, [183](#)
- onDisconnected
  - MessagesHandler, [178](#)
- onError
  - MessagesHandler, [182](#)
- onGameOver
  - MessagesHandler, [182](#)
- onGetGameSettings
  - MessagesHandler, [180](#)
- onGetMove
  - MessagesHandler, [181](#)
- ONGOING\_GAMES
  - SyncedListType, [293](#)
- onInitGame
  - MessagesHandler, [181](#)
- onInterrupt
  - MessagesHandler, [184](#)
- onIsAlive

- MessagesHandler, 184
- onLogin
  - MessagesHandler, 179
- onOfferDraw
  - MessagesHandler, 180
- onPlannedDisconnect
  - MessagesHandler, 179
- onQuestion
  - MessagesHandler, 182
- onResign
  - MessagesHandler, 179
- onUnplannedDisconnect
  - MessagesHandler, 179
- onUpdateByMove
  - MessagesHandler, 182
- onUpdateSyncedList
  - MessagesHandler, 184
- onUsernameAvailability
  - MessagesHandler, 183
- onWaitForMatch
  - MessagesHandler, 181
- onWaitTurn
  - MessagesHandler, 181
- onWelcomeMessage
  - MessagesHandler, 180
- opponentUsername
  - EstablishedGameInfo, 98
- opposite
  - Direction, 84
- OptionalArg
  - ArgsUtil, 22
- orderBy
  - Selection, 259
- parseURLS
  - StrUtils, 280
- Password
  - ArgType, 24
  - Col, 58
  - RegEx, 247
- PAWN
  - PieceType, 228
- perspective
  - Direction, 84
- PictureUrl
  - ArgType, 25
- Piece, 217
  - ALL\_PIECES, 222
  - B\_B, 222
  - B\_K, 222
  - B\_N, 222
  - B\_P, 221
  - B\_Q, 222
  - B\_R, 222
  - getFen, 220
  - getPiece, 219
  - getPieceFromFen, 218
  - getPiecelcon, 219
  - isOnMyTeam, 220
  - pieceType, 222
  - playerColor, 222
  - W\_B, 221
  - W\_K, 221
  - W\_N, 221
  - W\_P, 221
  - W\_Q, 221
  - W\_R, 221
- PIECE\_TABLES
  - EvaluationParameters, 105
- PIECE\_TYPES
  - PieceType, 229
- PieceType, 223
  - asInt, 230
  - ATTACKING\_DIRECTIONS, 230
  - ATTACKING\_PIECE\_TYPES, 230
  - BISHOP, 228
  - blackIcon, 230
  - CAN\_PROMOTE\_TO, 229
  - compareMovementType, 225
  - getAttackingDirections, 224, 227
  - getPiecelcon, 225
  - getPieceName, 227
  - getPieceType, 224
  - getWalkingDirections, 227
  - getWhitePieceFen, 224
  - isAttack, 227
  - isDiagonalPiece, 226
  - isLinePiece, 226
  - isSliding, 230
  - KING, 229
  - KNIGHT, 228
  - MAJOR\_PIECES, 229
  - MINOR\_PIECES, 229
  - NUM\_OF\_PIECE\_TYPES, 229
  - PAWN, 228
  - PIECE\_TYPES, 229
  - QUEEN, 228
  - ROOK, 228
  - UNIQUE\_MOVES\_PIECE\_TYPES, 229
  - value, 230
  - whitelcon, 230
- pieceType
  - Piece, 222
- plainTextIgnoreCase
  - ArgsUtil, 22
- Player1
  - Col, 58
- Player2
  - Col, 58
- PLAYER\_COLORS
  - PlayerColor, 234
- PlayerColor, 231
  - asInt, 234
  - BLACK, 233
  - diff, 235
  - getColor, 232
  - getName, 233

- getOpponent, [232](#)
- getPlayerFromFen, [232](#)
- indexOf2, [234](#)
- NO\_PLAYER, [234](#)
- NUM\_OF\_PLAYERS, [234](#)
- PLAYER\_COLORS, [234](#)
- PlayerColor, [231](#)
- startingRow, [234](#)
- toString, [233](#)
- WHITE, [233](#)
- playerColor
  - Piece, [222](#)
- playerColorToMove
  - UnfinishedGame, [305](#)
- playerDisconnected
  - GameStatus, [116](#)
- PlayerDisconnectedVsAi
  - GameStatus.SpecificStatus, [265](#)
- PlayerDisconnectedVsReal
  - GameStatus.SpecificStatus, [266](#)
- playerResigned
  - GameStatus, [117](#)
- PlayerToMove
  - Col, [58](#)
- playerToMove
  - UnfinishedGame, [305](#)
- Plus
  - Math, [152](#)
- postDescription
  - RequestBuilder, [256](#)
- preDescription
  - RequestBuilder, [256](#)
- PreMadeRequest, [235](#)
  - authSettings, [237](#)
  - ChangeProfilePic, [237](#)
  - createBuilder, [236](#)
  - DeleteUnfGames, [236](#)
  - Games, [236](#)
  - getRequestVariations, [236](#)
  - statistics, [237](#)
  - StatsByTimeOfDay, [237](#)
  - TopPlayers, [236](#)
- PreMadeRequest.VariationCreator, [309](#)
  - create, [309](#)
- PremovesGenerator, [237](#)
- prepareForDisconnect
  - MessagesHandler, [178](#)
- print
  - Board, [37](#)
  - DBResponse, [80](#)
  - Evaluation, [104](#)
- ProfilePic
  - Col, [59](#)
- Promotion
  - Move.MoveFlag, [205](#)
- QUEEN
  - CastlingRights.Side, [261](#)
  - PieceType, [228](#)
- Question, [238](#)
  - Answer, [241](#)
  - drawOffer, [239](#)
  - equals, [241](#)
  - getDefaultAnswer, [240](#)
  - getPossibleAnswers, [241](#)
  - getQuestionStr, [240](#)
  - Question, [239](#)
  - questionStr, [242](#)
  - questionType, [242](#)
  - Rematch, [242](#)
  - setDefaultAnswer, [240](#)
  - Threefold, [242](#)
  - toString, [241](#)
- questionStr
  - Question, [242](#)
- questionType
  - Question, [242](#)
- R
  - Direction, [85](#)
- reactivateErrs
  - MyThread, [217](#)
- receivedMessage
  - MessagesHandler, [178](#)
- Regex, [243](#)
  - Any, [248](#)
  - canBeEmpty, [244](#)
  - check, [246](#)
  - dontMatch, [248](#)
  - DontSaveGame, [247](#)
  - Fen, [246](#)
  - get, [245](#)
  - getDetails, [245](#)
  - getRegex, [246](#)
  - Icon, [247](#)
  - IPPortAddress, [247](#)
  - isSavedDate, [244](#)
  - Numbers, [247](#)
  - Password, [247](#)
  - Regex, [243](#), [244](#)
  - setUseDontMatch, [245](#)
  - StrUtilSkip, [247](#)
  - URL, [247](#)
  - Username, [246](#)
- REGISTER
  - LoginType, [146](#)
- Rematch
  - Question, [242](#)
- repeat
  - StrUtils, [287](#)
- replnStr
  - Arg, [20](#)
- replace
  - Col, [55](#)
  - SQLStatement, [267](#)
- replaceWithCancel
  - MyJButton, [212](#)
- request



- DBResponse, 80
- RequestBuilder, 248
  - addShouldSync, 251
  - args, 255
  - build, 255
  - changePassword, 251
  - changeProfilePic, 251
  - createResponse, 253
  - createVariation, 251
  - deleteAllUnFinishedGames, 252
  - games, 252
  - getArgs, 254
  - getArgVal, 253
  - getName, 254
  - getPostDescription, 254
  - getPreDescription, 254
  - getShouldSync, 253
  - name, 255
  - postDescription, 256
  - preDescription, 256
  - RequestBuilder, 249, 250
  - setSubBuilder, 252
  - statement, 255
  - statsByTimeOfDay, 252
  - subBuilder, 256
  - TIE\_STR, 255
  - top, 253
- requestMessage
  - AppSocket, 12, 15
- resetState
  - MyJButton, 212
- Resignation
  - GameStatus.SpecificStatus, 264
- respond
  - AppSocket, 14
- RESUMABLE\_GAMES
  - SyncedListType, 292
- returnDBResponse
  - Message, 163
- returnGameSettings
  - Message, 159
- returnLogin
  - Message, 156
- returnMove
  - Message, 160
- returnUsernameAvailable
  - Message, 162
- returnUsernameNotAvailable
  - Message, 162
- ROOK
  - PieceType, 228
- rookStartingCol
  - CastlingRights.Side, 262
- row
  - Location, 140
- rows
  - TableDBResponse, 300
- rowToString
  - TableDBResponse, 299
- runInHandledThread
  - HandledThread, 126
- RUNNING
  - MyThread.ThreadStatus, 301
- SavedGame
  - Col, 58
- Selection, 256
  - createStatement, 258
  - join, 259
  - nestMe, 258
  - orderBy, 259
  - Selection, 257, 258
  - top, 259
- ServerAddress
  - ArgType, 24
- ServerStoppedGame
  - GameStatus.SpecificStatus, 266
- serverStoppedGame
  - GameStatus, 115
- setAddedRes
  - DBResponse, 79
- setAvailable
  - Message, 169
- setBoard
  - Message, 174
- setCapturing
  - Move, 198
- setColName
  - Col, 54
- setCreatedAt
  - EstablishedGameInfo, 96
- setCustomStr
  - GameStatus, 115
- setDbRequest
  - Message, 163
- setDBResponse
  - Message, 165
- setDefaultAnswer
  - Question, 240
- setDepth
  - GameStatus, 119
- setDisabledCastling
  - Move, 194
- setEmpty
  - Square, 270
- setEnPassantLoc
  - Move, 201
- setError
  - Message, 164
- setEval
  - Evaluation, 102
- setEvaluationDepth
  - Evaluation, 100
- setFont
  - MyJButton, 212
- setGameSettings
  - Message, 170

- setGameStatus
  - Message, 173
- setGameTime
  - Message, 172
- setInCheck
  - GameStatus, 118
- setIntermediateMove
  - Move, 196
- setLoginInfo
  - Message, 174
- setLoginType
  - LoginInfo, 142
- setMessagesHandler
  - AppSocket, 13
- setMove
  - Message, 172
  - MinimaxMove, 188
- setMoveAnnotation
  - Move, 193
- setMoveEvaluation
  - MinimaxMove, 188
  - Move, 197
- setMoveFlag
  - Move, 200
- setMoveStack
  - Message, 166
- setMovingColor
  - Move, 194
- setMovingFrom
  - BasicMove, 30
- setMovingTo
  - BasicMove, 31
- setOnClick
  - MyJButton, 211
- setOnExit
  - MyJFrame, 214
- setOnResize
  - MyJFrame, 215
- setOtherPlayer
  - Message, 168
- setPassword
  - LoginInfo, 143
- setPerspective
  - Evaluation, 103
- setPiece
  - Board, 35
  - Square, 270
- setPlayerColor
  - Message, 171
- setPossibleMoves
  - Message, 173
- setPreMoves
  - Message, 171
- setPrevFullMoveClock
  - Move, 195
- setPrevHalfMoveClock
  - Move, 195
- setProfilePic
  - LoginInfo, 142
- setPromotingTo
  - Move, 196
- setQuestion
  - Message, 169
- setRespondingTo
  - Message, 167
- setRespondingToMsgId
  - Message, 167
- setReversible
  - Move, 198
- setRunnable
  - HandledThread, 126
- setSquareEmpty
  - Board, 38
- setStr
  - Condition, 63
- setSubBuilder
  - RequestBuilder, 252
- setSubRequest
  - DBRequest, 76
- setSyncedList
  - Message, 166
- setText
  - LinkLabel, 130
  - MyJButton, 211
- setThreefoldOption
  - Move, 194
- setTimeFormat
  - AiParameters, 10
  - GameSettings, 112
- setUseDontMatch
  - Regex, 245
- setUserInput
  - Arg, 19
- setUsername
  - LoginInfo, 143
  - Message, 170
- setUsernameSuggestions
  - Message, 165
- setWrapped
  - Col, 53
- ShortCastle
  - Move.MoveFlag, 205
- show
  - MyJFrame.BooleanClosing, 39
  - MyJFrame.Closing< T >, 47
  - MyJFrame.StringClosing, 274
- Side
  - CastlingRights.Side, 260
- SIDES
  - CastlingRights.Side, 262
- simpleOperation
  - Math, 150
- socket
  - MessagesHandler, 185
- SpecificStatus
  - GameStatus.SpecificStatus, 263

- splitArr
  - StrUtils, [284](#), [285](#)
- SQLStatement, [266](#)
  - createStatement, [267](#)
  - getStatement, [268](#)
  - replace, [267](#)
  - SQLStatement, [267](#)
  - toString, [268](#)
  - type, [268](#)
- Square, [269](#)
  - EMPTY\_PIECE, [272](#)
  - getFen, [271](#)
  - getLoc, [270](#)
  - getPiece, [270](#)
  - getPieceIcon, [271](#)
  - isEmpty, [271](#)
  - setEmpty, [270](#)
  - setPiece, [270](#)
  - Square, [269](#)
  - toString, [271](#)
- SQUARE\_CONTROL
  - EvaluationParameters, [106](#)
- Stalemate
  - GameStatus.SpecificStatus, [265](#)
- stalemate
  - GameStatus, [114](#)
- startingFen
  - Board, [39](#)
- startingPos
  - Board, [36](#)
- startingRow
  - PlayerColor, [234](#)
- startPress
  - MyJFrame.MyAdapter.HeldDown, [127](#)
- statement
  - RequestBuilder, [255](#)
- statistics
  - PreMadeRequest, [237](#)
- StatsByTimeOfDay
  - PreMadeRequest, [237](#)
- statsByTimeOfDay
  - RequestBuilder, [252](#)
- status
  - DBResponse, [80](#)
- StatusResponse, [272](#)
  - clean, [273](#)
  - getDetails, [273](#)
  - isAnyData, [273](#)
  - StatusResponse, [272](#), [273](#)
- STOCKFISH\_SAYS
  - EvaluationParameters, [106](#)
- stopReading
  - AppSocket, [15](#)
- stopRun
  - MyThread, [217](#)
- str
  - Math, [148](#)
- strictEquals
  - Move, [200](#)
- strINN
  - StrUtils, [284](#)
- strSource
  - Math, [150](#)
- StrUtils, [275](#)
  - awful, [286](#)
  - clean, [278](#)
  - countMatches, [276](#)
  - createTimeGreeting, [286](#)
  - createTimeStr, [286](#)
  - dateTimePrefix, [287](#)
  - dontCapFull, [276](#)
  - dontCapWord, [280](#)
  - fitInside, [283](#)
  - fixHtml, [284](#)
  - format, [277](#)
  - formatDate, [279](#)
  - formatDateSQL, [281](#)
  - getPort, [281](#)
  - getUrl, [283](#)
  - htmlNewLines, [276](#)
  - isAbsoluteUrl, [275](#)
  - isEmpty, [278](#)
  - main, [286](#)
  - parseURLS, [280](#)
  - repeat, [287](#)
  - splitArr, [284](#), [285](#)
  - strINN, [284](#)
  - uppercase, [280](#)
- StrUtils.IterationThingy< T >, [129](#)
  - iteration, [129](#)
- StrUtilSkip
  - Regex, [247](#)
- strVal
  - Math, [147](#)
- subBuilder
  - RequestBuilder, [256](#)
- sum
  - Col, [51](#)
- SwitchCase, [288](#)
  - condition, [290](#)
  - defaultCase, [290](#)
  - equals, [288](#)
  - ifTrue, [290](#)
  - SwitchCase, [288](#)
  - toString, [290](#)
- switchCase
  - Col, [52](#)
- SyncableItem, [291](#)
  - ID, [291](#)
- SyncedItems< E extends SyncableItem >, [292](#)
- SyncedItems< E extends SyncableItem >.Remover< L >, [248](#)
- SyncedListType, [292](#)
  - CONNECTED\_USERS, [292](#)
  - JOINABLE\_GAMES, [292](#)
  - ONGOING\_GAMES, [293](#)

- RESUMABLE\_GAMES, 292
- syncLists
  - Message, 164
- Table, 293
  - cols, 295
  - escapeValues, 294
  - Games, 294
  - Table, 293
  - tableAndValues, 294
  - UnfinishedGames, 294
  - Users, 295
- tableAndValues
  - Table, 294
- TableDBResponse, 295
  - clean, 298
  - columns, 300
  - getCell, 297, 298
  - getColumnIndex, 297
  - getColumns, 299
  - getFirstRow, 297
  - getRows, 299
  - isAnyData, 298
  - numOfRows, 296
  - rows, 300
  - rowToString, 299
  - TableDBResponse, 296
  - toString, 298
- Text
  - ArgType, 24
- ThreadsManager, 300
  - createThread, 300
  - handleErrors, 300
- Threefold
  - Question, 242
- threefoldClaim
  - Move, 192
- ThreeFoldRepetition
  - GameStatus.SpecificStatus, 265
- threeFoldRepetition
  - GameStatus, 115
- throwError
  - Message, 157
- ThrowingCallback< T >, 302
  - callback, 302
- ThrowingRunnable, 303
- TIE
  - GameStatus.GameStatusType, 122
- TIE\_EVAL
  - Evaluation, 104
- TIE\_STR
  - RequestBuilder, 255
- TieByAgreement
  - GameStatus.SpecificStatus, 265
- tieByAgreement
  - GameStatus, 114
- time
  - Col, 52
- TimedOut
  - GameStatus.SpecificStatus, 264
- timedOut
  - GameStatus, 116
- TimedOutVsInsufficientMaterial
  - GameStatus.SpecificStatus, 264
- TimeFormat, 303
- TimeFormatComponent, 304
- title
  - MyJFrame.Closing< T >, 48
- top
  - RequestBuilder, 253
  - Selection, 259
- TopPlayers
  - PreMadeRequest, 236
- toString
  - ArchivedGameInfo, 17
  - Arg, 20
  - BasicMove, 32
  - Board, 38
  - CastlingRights, 44
  - CastlingRights.Side, 261
  - Col, 57
  - Condition, 64
  - Config< V >, 67
  - DBRequest, 76
  - DBResponse, 80
  - EstablishedGameInfo, 97
  - Evaluation, 103
  - GameInfo, 109
  - GameStatus, 119
  - GameStatus.SpecificStatus, 264
  - GraphableDBResponse, 124
  - Location, 138
  - LoginInfo, 144
  - LoginType, 145
  - MinimaxMove, 189
  - Move, 199
  - MyError, 208
  - PlayerColor, 233
  - Question, 241
  - SQLStatement, 268
  - Square, 271
  - SwitchCase, 290
  - TableDBResponse, 298
- tryClose
  - MyJFrame.Closing< T >, 47
- type
  - DBRequest, 77
  - SQLStatement, 268
- U
  - Direction, 85
- U\_L
  - Direction, 86
- U\_L\_L
  - Direction, 88
- U\_R
  - Direction, 86
- U\_R\_R

- Direction, 88
- U\_U
  - Direction, 86
- U\_U\_L
  - Direction, 87
- U\_U\_R
  - Direction, 87
- UNFINISHED
  - GameStatus.GameStatusType, 122
- UnfinishedGame, 304
  - isCreatorToMove, 305
  - playerColorToMove, 305
  - playerToMove, 305
  - UnfinishedGame, 304
- UnfinishedGames
  - Table, 294
- UNIQUE\_MOVES\_PIECE\_TYPES
  - PieceType, 229
- Update, 306
  - createStatement, 307
  - Update, 306
- updateByMove
  - Message, 157
- uppercase
  - StrUtils, 280
- URL
  - RegEx, 247
- Url
  - ArgType, 25
- USER
  - AuthSettings, 27
- UserInfo, 307
  - ID, 307
- Username
  - ArgType, 24
  - Col, 58
  - RegEx, 246
- Users
  - Table, 295
- value
  - PieceType, 230
- ValueBtn
  - ValueBtn< T >, 308
- ValueBtn< T >, 308
  - ValueBtn, 308
- valueOf
  - Location, 135
- ver14::SharedClasses::DBActions::Condition::Relation, 59
- ver14::SharedClasses::DBActions::DBRequest::DBRequest::Type, 74
- ver14::SharedClasses::DBActions::DBResponse::DBResponse::Status, 77
- ver14::SharedClasses::DBActions::Statements::Selection::Join, 257
- ver14::SharedClasses::DBActions::Statements::Selection::Order, 257
- ver14::SharedClasses::Game::GameSetup::AiParameters::AiType, 9
- ver14::SharedClasses::Game::GameSetup::GameSettings::GameType, 111
- ver14::SharedClasses::Game::Moves::Move::ThreefoldStatus, 190
- ver14::SharedClasses::Misc::Question::QuestionType, 238
- VoidCallback, 310
  - callback, 310
- W\_B
  - Piece, 221
- W\_K
  - Piece, 221
- W\_N
  - Piece, 221
- W\_P
  - Piece, 221
- W\_Q
  - Piece, 221
- W\_R
  - Piece, 221
- waitForMatch
  - Message, 158
- waitForYourTurn
  - Message, 158
- weight
  - EvaluationParameters, 106
- welcomeMessage
  - Message, 156
- WHITE
  - PlayerColor, 233
- whitelcon
  - PieceType, 230
- whosCastling
  - CastlingRights, 43
- WIN\_EVAL
  - Evaluation, 104
- WIN\_OR\_LOSS
  - GameStatus.GameStatusType, 122
- Winner
  - Col, 58
- wrap
  - Col, 52
  - Condition, 62
- writeMessage
  - AppSocket, 14
- zeroIfNull
  - Math, 149