

Chess Engine

Generated by Doxygen 1.8.14

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Data Structure Index	5
3.1	Data Structures	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	Hash Namespace Reference	9
5.1.1	Detailed Description	9
5.1.2	Function Documentation	9
5.1.2.1	generatePosKey()	9
5.1.2.2	hashCa()	10
5.1.2.3	hashEP()	10
5.1.2.4	hashPce()	11
5.1.2.5	hashSide()	11
5.2	Init Namespace Reference	11
5.2.1	Detailed Description	12
5.2.2	Function Documentation	12
5.2.2.1	initAll()	12

5.2.2.2	initBitMasks()	12
5.2.2.3	initEvalMasks()	13
5.2.2.4	initFileRankBrd()	13
5.2.2.5	initHashKeys()	14
5.2.2.6	initMvvLva()	14
5.2.2.7	initSq120ToSq64()	14
5.3	IO Namespace Reference	15
5.3.1	Detailed Description	15
5.3.2	Function Documentation	15
5.3.2.1	parseMove()	15
5.3.2.2	printBestMove()	16
5.3.2.3	printBitBoard()	16
5.3.2.4	printBoard()	17
5.3.2.5	printMoveList()	17
5.3.2.6	printSearchDetails()	17
5.3.3	Variable Documentation	18
5.3.3.1	epstr	18
5.3.3.2	FileChar	18
5.3.3.3	PceChar	19
5.3.3.4	RankChar	19
5.3.3.5	SideChar	19
5.4	MoveFlags Namespace Reference	19
5.4.1	Detailed Description	19
5.5	MvvLva Namespace Reference	20
5.5.1	Detailed Description	20
5.6	PolyKeys Namespace Reference	20
5.6.1	Detailed Description	20
5.7	Value Namespace Reference	20
5.7.1	Detailed Description	21
5.7.2	Variable Documentation	21
5.7.2.1	BishopTable	21
5.7.2.2	KingEndGame	22
5.7.2.3	KingOpening	22
5.7.2.4	KnightTable	22
5.7.2.5	PawnTable	23
5.7.2.6	RookTable	23

6	Data Structure Documentation	25
6.1	Board Class Reference	25
6.1.1	Detailed Description	26
6.1.2	Member Function Documentation	26
6.1.2.1	flipBoard()	26
6.1.2.2	getAllCaptureMoves()	26
6.1.2.3	getAllMoves()	27
6.1.2.4	getEnPassant()	27
6.1.2.5	inCheck()	28
6.1.2.6	parseFEN()	28
6.1.2.7	resetBoard()	29
6.1.2.8	setUpCastlePerm()	29
6.1.2.9	setUpMoveCounters()	29
6.1.2.10	setUpPieces()	30
6.1.2.11	sqAttacked()	30
6.1.2.12	updatePieceLists()	31
6.2	ConsoleManager Class Reference	31
6.2.1	Detailed Description	31
6.2.2	Member Function Documentation	32
6.2.2.1	getProtocol()	32
6.2.2.2	loop()	32
6.3	Engine Class Reference	33
6.3.1	Detailed Description	33
6.3.2	Member Function Documentation	33
6.3.2.1	getBook()	33
6.3.2.2	getConfig()	34
6.3.2.3	getInstance()	34
6.4	EngineConfig Struct Reference	35
6.4.1	Detailed Description	35
6.5	Evaluator Class Reference	35

6.5.1	Detailed Description	36
6.5.2	Member Function Documentation	36
6.5.2.1	drawnMaterial()	36
6.5.2.2	evalBishops()	36
6.5.2.3	evalKings()	37
6.5.2.4	evalKnights()	37
6.5.2.5	evalPawns()	37
6.5.2.6	evalQueens()	38
6.5.2.7	evalRooks()	38
6.5.2.8	evaluatePosition()	38
6.6	MM Class Reference	39
6.6.1	Detailed Description	39
6.6.2	Member Function Documentation	39
6.6.2.1	addPiece()	39
6.6.2.2	clearPiece()	40
6.6.2.3	makeMove()	40
6.6.2.4	makeNullMove()	41
6.6.2.5	movePiece()	41
6.6.2.6	takeMove()	42
6.6.2.7	takeNullMove()	42
6.7	Move Class Reference	42
6.7.1	Detailed Description	43
6.7.2	Member Function Documentation	43
6.7.2.1	captured()	43
6.7.2.2	castle()	44
6.7.2.3	enPassant()	44
6.7.2.4	from()	45
6.7.2.5	isNull()	45
6.7.2.6	pawnStart()	45
6.7.2.7	promoted()	46

6.7.2.8	to()	46
6.7.2.9	toString()	46
6.7.2.10	wasCapture()	47
6.7.2.11	wasPromotion()	47
6.8	MoveList Class Reference	48
6.8.1	Detailed Description	49
6.8.2	Member Function Documentation	49
6.8.2.1	addCaptureMove()	49
6.8.2.2	addEnPasMove()	49
6.8.2.3	addPawnCaptureMove()	50
6.8.2.4	addPawnMove()	50
6.8.2.5	addQuietMove()	51
6.8.2.6	generateAllCaptureMoves()	51
6.8.2.7	generateAllMoves()	52
6.8.2.8	generateBishopMoves()	52
6.8.2.9	generateCastlingMoves()	53
6.8.2.10	generateKingMoves()	53
6.8.2.11	generateKnightMoves()	53
6.8.2.12	generateNonSlidingMoves()	54
6.8.2.13	generatePawnMoves()	54
6.8.2.14	generateQueenMoves()	55
6.8.2.15	generateRookMoves()	55
6.8.2.16	generateSlidingMoves()	56
6.8.2.17	reorderList()	56
6.8.2.18	size()	56
6.9	PerftTester Class Reference	57
6.9.1	Detailed Description	57
6.9.2	Member Function Documentation	57
6.9.2.1	perft()	57
6.9.2.2	perftTest()	58

6.9.2.3	perftTestAll()	58
6.10	PolyBook Class Reference	59
6.10.1	Detailed Description	60
6.10.2	Member Function Documentation	60
6.10.2.1	convertPolyMove()	60
6.10.2.2	endian_swap_u16()	60
6.10.2.3	endian_swap_u32()	61
6.10.2.4	endian_swap_u64()	61
6.10.2.5	enPasPossible()	61
6.10.2.6	getBookMove()	62
6.10.2.7	polyKeyFromBoard()	62
6.10.2.8	readBook()	62
6.10.3	Field Documentation	63
6.10.3.1	book	63
6.10.3.2	polyPiece	63
6.11	PolyglotEntry Struct Reference	63
6.11.1	Detailed Description	64
6.12	ProtocolManager Class Reference	64
6.12.1	Detailed Description	64
6.12.2	Member Function Documentation	64
6.12.2.1	getProtocol()	64
6.12.2.2	isOver()	65
6.12.2.3	loop()	65
6.13	PvEntry Class Reference	66
6.13.1	Detailed Description	66
6.14	PvTable Class Reference	66
6.14.1	Detailed Description	67
6.14.2	Member Function Documentation	67
6.14.2.1	clear()	67
6.14.2.2	get()	67

6.14.2.3	getHashEntry()	68
6.14.2.4	getPvLine()	68
6.14.2.5	insert() [1/2]	69
6.14.2.6	insert() [2/2]	69
6.14.2.7	size()	70
6.15	SearchAgent Class Reference	70
6.15.1	Detailed Description	71
6.15.2	Member Function Documentation	71
6.15.2.1	alphaBeta()	71
6.15.2.2	checkStop()	72
6.15.2.3	clearForSearch()	72
6.15.2.4	drawnMaterial()	73
6.15.2.5	isGameOver()	73
6.15.2.6	isRepetition()	74
6.15.2.7	quiescenceSearch()	75
6.15.2.8	searchPosition()	75
6.15.2.9	threeFoldRepetition()	76
6.15.3	Field Documentation	76
6.15.3.1	kInterval	76
6.16	SearchInfo Struct Reference	77
6.16.1	Detailed Description	77
6.17	Stopwatch Class Reference	77
6.17.1	Detailed Description	78
6.17.2	Member Function Documentation	78
6.17.2.1	getTimeInMilli()	78
6.17.2.2	start()	78
6.17.2.3	stop()	79
6.18	UCIManager Class Reference	79
6.18.1	Detailed Description	80
6.18.2	Member Function Documentation	80
6.18.2.1	getProtocol()	80
6.18.2.2	loop()	80
6.19	UndoMove Class Reference	81
6.19.1	Detailed Description	81
6.20	XBoardManager Class Reference	81
6.20.1	Detailed Description	82
6.20.2	Member Function Documentation	82
6.20.2.1	getProtocol()	82
6.20.2.2	loop()	82

7	File Documentation	85
7.1	/home/michael/Documents/Projects/C++/Chess-Engine/include/bitboard.h File Reference	85
7.1.1	Detailed Description	85
7.1.2	Function Documentation	86
7.1.2.1	clearBit()	86
7.1.2.2	countBits()	86
7.1.2.3	popBit()	87
7.1.2.4	setBit()	87
7.1.3	Variable Documentation	87
7.1.3.1	BitTable	88
7.2	/home/michael/Documents/Projects/C++/Chess-Engine/include/board.h File Reference	88
7.2.1	Detailed Description	88
7.3	/home/michael/Documents/Projects/C++/Chess-Engine/include/console.h File Reference	89
7.3.1	Detailed Description	89
7.4	/home/michael/Documents/Projects/C++/Chess-Engine/include/debug.h File Reference	89
7.4.1	Detailed Description	89
7.5	/home/michael/Documents/Projects/C++/Chess-Engine/include/defs.h File Reference	90
7.5.1	Detailed Description	91
7.6	/home/michael/Documents/Projects/C++/Chess-Engine/include/engine.h File Reference	92
7.6.1	Detailed Description	92
7.7	/home/michael/Documents/Projects/C++/Chess-Engine/include/eval.h File Reference	92
7.7.1	Detailed Description	93
7.8	/home/michael/Documents/Projects/C++/Chess-Engine/include/hash.h File Reference	94
7.8.1	Detailed Description	94
7.9	/home/michael/Documents/Projects/C++/Chess-Engine/include/init.h File Reference	94
7.9.1	Detailed Description	95
7.10	/home/michael/Documents/Projects/C++/Chess-Engine/include/io.h File Reference	95
7.10.1	Detailed Description	96
7.11	/home/michael/Documents/Projects/C++/Chess-Engine/include/move.h File Reference	96
7.11.1	Detailed Description	97

7.12	/home/michael/Documents/Projects/C++/Chess-Engine/include/movelist.h File Reference	97
7.12.1	Detailed Description	98
7.13	/home/michael/Documents/Projects/C++/Chess-Engine/include/movemaker.h File Reference	98
7.13.1	Detailed Description	98
7.13.2	Variable Documentation	99
7.13.2.1	CastlePerm	99
7.14	/home/michael/Documents/Projects/C++/Chess-Engine/include/polyglot.h File Reference	99
7.14.1	Detailed Description	99
7.15	/home/michael/Documents/Projects/C++/Chess-Engine/include/polyglotkeys.h File Reference	100
7.15.1	Detailed Description	100
7.16	/home/michael/Documents/Projects/C++/Chess-Engine/include/protocol.h File Reference	100
7.16.1	Detailed Description	101
7.17	/home/michael/Documents/Projects/C++/Chess-Engine/include/pvtable.h File Reference	101
7.17.1	Detailed Description	101
7.18	/home/michael/Documents/Projects/C++/Chess-Engine/include/search.h File Reference	101
7.18.1	Detailed Description	102
7.19	/home/michael/Documents/Projects/C++/Chess-Engine/include/searchinfo.h File Reference	102
7.19.1	Detailed Description	102
7.20	/home/michael/Documents/Projects/C++/Chess-Engine/include/stopwatch.h File Reference	102
7.20.1	Detailed Description	103
7.21	/home/michael/Documents/Projects/C++/Chess-Engine/include/tester.h File Reference	103
7.21.1	Detailed Description	103
7.22	/home/michael/Documents/Projects/C++/Chess-Engine/include/uci.h File Reference	103
7.22.1	Detailed Description	104
7.23	/home/michael/Documents/Projects/C++/Chess-Engine/include/utils.h File Reference	104
7.23.1	Detailed Description	105
7.23.2	Function Documentation	105
7.23.2.1	fileRankToSq()	105
7.23.2.2	InputWaiting()	105
7.23.2.3	isPiece()	106

7.23.2.4	randU64()	106
7.23.2.5	ReadInput()	107
7.23.2.6	showAttackedSqs()	107
7.23.2.7	sqOnBoard()	107
7.23.2.8	sqToString()	108
7.23.2.9	stringToLower()	108
7.23.3	Variable Documentation	109
7.23.3.1	WhiteToBlack	109
7.24	/home/michael/Documents/Projects/C++/Chess-Engine/include/xboard.h File Reference	109
7.24.1	Detailed Description	109
7.25	/home/michael/Documents/Projects/C++/Chess-Engine/src/bitboard.cc File Reference	110
7.25.1	Detailed Description	110
7.26	/home/michael/Documents/Projects/C++/Chess-Engine/src/board.cc File Reference	110
7.26.1	Detailed Description	110
7.27	/home/michael/Documents/Projects/C++/Chess-Engine/src/console.cc File Reference	111
7.27.1	Detailed Description	111
7.28	/home/michael/Documents/Projects/C++/Chess-Engine/src/engine.cc File Reference	111
7.28.1	Detailed Description	111
7.29	/home/michael/Documents/Projects/C++/Chess-Engine/src/eval.cc File Reference	112
7.29.1	Detailed Description	112
7.30	/home/michael/Documents/Projects/C++/Chess-Engine/src/hash.cc File Reference	112
7.30.1	Detailed Description	112
7.31	/home/michael/Documents/Projects/C++/Chess-Engine/src/init.cc File Reference	113
7.31.1	Detailed Description	113
7.32	/home/michael/Documents/Projects/C++/Chess-Engine/src/io.cc File Reference	113
7.32.1	Detailed Description	114
7.33	/home/michael/Documents/Projects/C++/Chess-Engine/src/main.cc File Reference	114
7.33.1	Detailed Description	114
7.34	/home/michael/Documents/Projects/C++/Chess-Engine/src/move.cc File Reference	114
7.34.1	Detailed Description	115

7.35	/home/michael/Documents/Projects/C++/Chess-Engine/src/movelist.cc File Reference	115
7.35.1	Detailed Description	115
7.36	/home/michael/Documents/Projects/C++/Chess-Engine/src/movemaker.cc File Reference	115
7.36.1	Detailed Description	116
7.37	/home/michael/Documents/Projects/C++/Chess-Engine/src/polyglot.cc File Reference	116
7.37.1	Detailed Description	116
7.38	/home/michael/Documents/Projects/C++/Chess-Engine/src/pvtable.cc File Reference	116
7.38.1	Detailed Description	117
7.39	/home/michael/Documents/Projects/C++/Chess-Engine/src/search.cc File Reference	117
7.39.1	Detailed Description	117
7.40	/home/michael/Documents/Projects/C++/Chess-Engine/src/stopwatch.cc File Reference	117
7.40.1	Detailed Description	118
7.41	/home/michael/Documents/Projects/C++/Chess-Engine/src/tester.cc File Reference	118
7.41.1	Detailed Description	118
7.42	/home/michael/Documents/Projects/C++/Chess-Engine/src/uci.cc File Reference	118
7.42.1	Detailed Description	119
7.43	/home/michael/Documents/Projects/C++/Chess-Engine/src/utils.cc File Reference	119
7.43.1	Detailed Description	120
7.43.2	Function Documentation	120
7.43.2.1	InputWaiting()	120
7.43.2.2	randU64()	120
7.43.2.3	ReadInput()	121
7.43.2.4	showAttackedSqs()	121
7.43.2.5	sqOnBoard()	121
7.43.2.6	sqToString()	122
7.43.2.7	stringToLower()	122
7.44	/home/michael/Documents/Projects/C++/Chess-Engine/src/xboard.cc File Reference	123
7.44.1	Detailed Description	123

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Hash	9
Init	Provides methods to initialize the non constant arrays used throughout the engine	11
IO	Various functions to print out engine data structures	15
MoveFlags	This namespace stores flags used to extract parameters from a move	19
MvvLva	20
PolyKeys	20
Value	Provides several constant values used for evaluating the board state	20

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Board	25
Engine	33
EngineConfig	35
Evaluator	35
MM	39
Move	42
MoveList	48
PerftTester	57
PolyBook	59
PolyglotEntry	63
ProtocolManager	64
ConsoleManager	31
UCIManager	79
XBoardManager	81
PvEntry	66
PvTable	66
SearchAgent	70
SearchInfo	77
Stopwatch	77
UndoMove	81

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Board	25
ConsoleManager	31
Engine	33
EngineConfig	
Stores the flags for the engine	35
Evaluator	35
MM	39
Move	42
MoveList	48
PerftTester	57
PolyBook	59
PolyglotEntry	63
ProtocolManager	64
PvEntry	66
PvTable	66
SearchAgent	70
SearchInfo	77
Stopwatch	77
UCIManager	79
UndoMove	
This class stores info needed to undo a move that was made	81
XBoardManager	81

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

/home/michael/Documents/Projects/C++/Chess-Engine/include/ bitboard.h	
Contains declarations of functions that manipulate bitboards	85
/home/michael/Documents/Projects/C++/Chess-Engine/include/ board.h	
Defines the internal Board representation used by the engine	88
/home/michael/Documents/Projects/C++/Chess-Engine/include/ console.h	
Contains declarations of functions for the console protocol	89
/home/michael/Documents/Projects/C++/Chess-Engine/include/ debug.h	
Defines an assert function for debugging	89
/home/michael/Documents/Projects/C++/Chess-Engine/include/ defs.h	
Contains declarations of various constant arrays used throughout the engine	90
/home/michael/Documents/Projects/C++/Chess-Engine/include/ engine.h	
Defines the central engine data structure	92
/home/michael/Documents/Projects/C++/Chess-Engine/include/ eval.h	
Contains declarations of functions that determine the strength of a given position	92
/home/michael/Documents/Projects/C++/Chess-Engine/include/ hash.h	
Contains declarations of functions that manipulate the board position key	94
/home/michael/Documents/Projects/C++/Chess-Engine/include/ init.h	
Contains declarations of functions that fill the non static constant arrays used throughout the engine	94
/home/michael/Documents/Projects/C++/Chess-Engine/include/ io.h	
Contains declarations of functions that print the various Engine data structures	95
/home/michael/Documents/Projects/C++/Chess-Engine/include/ move.h	
Defines the custom internal move representation	96
/home/michael/Documents/Projects/C++/Chess-Engine/include/ movelist.h	
Custom data structure to store possible moves	97
/home/michael/Documents/Projects/C++/Chess-Engine/include/ movemaker.h	
Contains declarations of functions that manipulate the position of pieces on the internal board	98
/home/michael/Documents/Projects/C++/Chess-Engine/include/ polyglot.h	
Contains declarations for the Polyglot Book class	99
/home/michael/Documents/Projects/C++/Chess-Engine/include/ polyglotkeys.h	
Contains all 781 polyglot hashkeys used in the opening	100
/home/michael/Documents/Projects/C++/Chess-Engine/include/ protocol.h	
Contains declaration of base class for default protocol functionality	100
/home/michael/Documents/Projects/C++/Chess-Engine/include/ pvtable.h	
Contains declarations for the Principal Variation/transposition table class used for caching	101

/home/michael/Documents/Projects/C++/Chess-Engine/include/ search.h	
Contains declarations of functions to search through the board state	101
/home/michael/Documents/Projects/C++/Chess-Engine/include/ searchinfo.h	
Contains information about the search constraints provided by the engine	102
/home/michael/Documents/Projects/C++/Chess-Engine/include/ stopwatch.h	
Contains declarations of functions for basic benchmarking and timing	102
/home/michael/Documents/Projects/C++/Chess-Engine/include/ tester.h	
Contains declarations of functions used for PerfT testing for testing accuracy of move generation and move making	103
/home/michael/Documents/Projects/C++/Chess-Engine/include/ uci.h	
Contains declarations of functions for the UCI protocol	103
/home/michael/Documents/Projects/C++/Chess-Engine/include/ utils.h	
Contains declarations of functions that perform various miscellaneous actions in the engine . .	104
/home/michael/Documents/Projects/C++/Chess-Engine/include/ xboard.h	
Contains declarations for the XBoard protocol	109
/home/michael/Documents/Projects/C++/Chess-Engine/src/ bitboard.cc	
Contains definitions of functions declared in bitboard.h	110
/home/michael/Documents/Projects/C++/Chess-Engine/src/ board.cc	
Contains definitions of functions declared in board.h	110
/home/michael/Documents/Projects/C++/Chess-Engine/src/ console.cc	
Contains definitions of functions declared in console.h	111
/home/michael/Documents/Projects/C++/Chess-Engine/src/ engine.cc	
Contains definitions of functions declared in engine.h	111
/home/michael/Documents/Projects/C++/Chess-Engine/src/ eval.cc	
Contains definitions of functions declared in eval.h	112
/home/michael/Documents/Projects/C++/Chess-Engine/src/ hash.cc	
Contains definitions of functions declared in hash.h	112
/home/michael/Documents/Projects/C++/Chess-Engine/src/ init.cc	
Contains definitions of functions declared in init.h	113
/home/michael/Documents/Projects/C++/Chess-Engine/src/ io.cc	
Contains definitions of functions declared in io.h	113
/home/michael/Documents/Projects/C++/Chess-Engine/src/ main.cc	
Entry point of the engine	114
/home/michael/Documents/Projects/C++/Chess-Engine/src/ move.cc	
Contains definitions of functions declared in move.h	114
/home/michael/Documents/Projects/C++/Chess-Engine/src/ movelist.cc	
Contains definitions of functions declared in movelist.h	115
/home/michael/Documents/Projects/C++/Chess-Engine/src/ movemaker.cc	
Contains definitions of functions declared in movemaker.h	115
/home/michael/Documents/Projects/C++/Chess-Engine/src/ polyglot.cc	
Contains definitions of functions declared in polyglot.h	116
/home/michael/Documents/Projects/C++/Chess-Engine/src/ pvtable.cc	
Contains definitions of functions declared in pvtable.h	116
/home/michael/Documents/Projects/C++/Chess-Engine/src/ search.cc	
Contains definitions of functions declared in search.h	117
/home/michael/Documents/Projects/C++/Chess-Engine/src/ stopwatch.cc	
Contains definitions of functions declared in stopwatch.h	117
/home/michael/Documents/Projects/C++/Chess-Engine/src/ tester.cc	
Contains definitions of functions declared in tester.h	118
/home/michael/Documents/Projects/C++/Chess-Engine/src/ uci.cc	
Contains definitions of functions declared in uci.h	118
/home/michael/Documents/Projects/C++/Chess-Engine/src/ utils.cc	
Contains definitions of functions declared in utils.h	119
/home/michael/Documents/Projects/C++/Chess-Engine/src/ xboard.cc	
Contains definitions of functions declared in xboard.h	123

Chapter 5

Namespace Documentation

5.1 Hash Namespace Reference

Functions

- `uint64_t generatePosKey` (const `Board` &pos)
Gets the position hash key for the current position.
- `void hashPce` (uint32_t pce, uint32_t sq, `Board` &pos) noexcept
Hashes in/out a piece on a given square.
- `void hashCa` (`Board` &pos) noexcept
Hashes in/out the castle permissions.
- `void hashSide` (`Board` &pos) noexcept
Hashes in/out the side to move.
- `void hashEP` (`Board` &pos) noexcept
Hashes in/out the enPassant square.

Variables

- `std::array< std::array< uint64_t, kBoardArraySize >, kNumPceTypes >` **PieceKeys**
- `uint64_t` **SideKey**
- `std::array< uint64_t, 16 >` **CastleKeys**

5.1.1 Detailed Description

This namespace provides various functions related to manipulating the board's hashkey

5.1.2 Function Documentation

5.1.2.1 generatePosKey()

```
uint64_t Hash::generatePosKey (  
    const Board & pos )
```

Gets the position hash key for the current position.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

The 64 bit position key.

Definition at line 12 of file hash.cc.

5.1.2.2 hashCa()

```
void Hash::hashCa (
    Board & pos ) [inline], [noexcept]
```

Hashes in/out the castle permissions.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

None.

Definition at line 46 of file hash.h.

5.1.2.3 hashEP()

```
void Hash::hashEP (
    Board & pos ) [inline], [noexcept]
```

Hashes in/out the enPassant square.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

None.

Definition at line 66 of file hash.h.

5.1.2.4 hashPce()

```
void Hash::hashPce (
    uint32_t pce,
    uint32_t sq,
    Board & pos ) [inline], [noexcept]
```

Hashes in/out a piece on a given square.

Parameters

<i>pce</i>	The piece to hash in/out.
<i>sq</i>	The square that the piece is/will be on
<i>pos</i>	The current board state

Returns

None

Definition at line 36 of file hash.h.

5.1.2.5 hashSide()

```
void Hash::hashSide (
    Board & pos ) [inline], [noexcept]
```

Hashes in/out the side to move.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

None.

Definition at line 56 of file hash.h.

5.2 Init Namespace Reference

Provides methods to initialize the non constant arrays used throughout the engine.

Functions

- void `initAll` () noexcept
Calls all of the other 'init' methods.
- void `initSq120ToSq64` () noexcept
Fills in the arrays that convert between array-120 to array-64 representations.
- void `initBitMasks` () noexcept
Fills in the arrays used for setting/clearing bits in bitboards.
- void `initHashKeys` () noexcept
Fills in the hashkeys arrays that will be used to for getting the board's hashkey.
- void `initFileRankBrd` () noexcept
Fills in the arrays that return the file/rank # for a given square.
- void `initEvalMasks` () noexcept
Fills in the arrays used for evaluating pawn structure during evaluation.
- void `initMvvLva` () noexcept
Fills in the arrays to determine most valuable victim least valuable attacker priority.

5.2.1 Detailed Description

Provides methods to initialize the non constant arrays used throughout the engine.

5.2.2 Function Documentation

5.2.2.1 `initAll()`

```
void Init::initAll ( ) [noexcept]
```

Calls all of the other 'init' methods.

Parameters

<i>None</i>	
-------------	--

Returns

None.

Definition at line 197 of file `init.cc`.

5.2.2.2 `initBitMasks()`

```
void Init::initBitMasks ( ) [noexcept]
```

Fills in the arrays used for setting/clearing bits in bitboards.

Parameters

<i>None</i>	
-------------	--

Returns

None.

Definition at line 97 of file init.cc.

5.2.2.3 initEvalMasks()

```
void Init::initEvalMasks ( ) [noexcept]
```

Fills in the arrays used for evaluating pawn structure during evaluation.

Parameters

<i>None</i>	
-------------	--

Returns

None.

Definition at line 122 of file init.cc.

5.2.2.4 initFileRankBrd()

```
void Init::initFileRankBrd ( ) [noexcept]
```

Fills in the arrays that return the file/rank # for a given square.

Parameters

<i>None</i>	
-------------	--

Returns

None.

Definition at line 54 of file init.cc.

5.2.2.5 initHashKeys()

```
void Init::initHashKeys ( ) [noexcept]
```

Fills in the hashkeys arrays that will be used to for getting the board's hashkey.

Parameters

None	
------	--

Returns

None.

Definition at line 106 of file init.cc.

5.2.2.6 initMvvLva()

```
void Init::initMvvLva ( ) [noexcept]
```

Fills in the arrays to determine most valuable victim least valuable attacker priority.

Parameters

None	
------	--

Returns

None.

Definition at line 186 of file init.cc.

5.2.2.7 initSq120ToSq64()

```
void Init::initSq120ToSq64 ( ) [noexcept]
```

Fills in the arrays that convert between array-120 to array-64 representations.

Parameters

None	
------	--

Returns

None.

Definition at line 73 of file init.cc.

5.3 IO Namespace Reference

Various functions to print out engine data structures.

Functions

- void `printBoard` (const `Board` &pos) noexcept
Prints out the board to stdio.
- void `printBitBoard` (const uint64_t bb) noexcept
Prints out the bitboard to stdio.
- void `printMoveList` (const `MoveList` &list) noexcept
Prints out the `MoveList` to stdio.
- void `printSearchDetails` (const `SearchInfo` &info, int32_t curDepth, int32_t bestScore, `PvTable` &pv, int32_t pvMoves) noexcept
Prints out the search details to stdio.
- void `printBestMove` (`Board` &pos, const `SearchInfo` &info, const `Move` &bestMove) noexcept
Prints out the best move stdio for the protocol manager.
- `Move` `parseMove` (std::string input, const `Board` &pos) noexcept
Reads in a move string and converts it to the internal representation.

Variables

- const std::string `PceChar` = ".PNBRQKpnbrqk"
- const std::string `SideChar` = "wb-"
- const std::string `RankChar` = "12345678"
- const std::string `FileChar` = "abcdefgh"
- const std::unordered_map< uint32_t, std::string > `epstr`

5.3.1 Detailed Description

Various functions to print out engine data structures.

5.3.2 Function Documentation

5.3.2.1 `parseMove()`

```
Move IO::parseMove (  
    std::string input,  
    const Board & pos ) [noexcept]
```

Reads in a move string and converts it to the internal representation.

Parameters

<i>input</i>	The move that was read in (e.g e2e4).
<i>pos</i>	The current board state.

Returns

The internal representation of the input move.

Definition at line 122 of file io.cc.

5.3.2.2 printBestMove()

```
void IO::printBestMove (
    Board & pos,
    const SearchInfo & info,
    const Move & bestMove ) [noexcept]
```

Prints out the best move stdio for the protocol manager.

Parameters

<i>pos</i>	The current board state.
<i>info</i>	The engine's searchInfo struct.
<i>bestMove</i>	The best move that was found.

Returns

None.

Definition at line 103 of file io.cc.

5.3.2.3 printBitBoard()

```
void IO::printBitBoard (
    const uint64_t bb ) [noexcept]
```

Prints out the bitboard to stdio.

Parameters

<i>bb</i>	The bitboard to print.
-----------	------------------------

Returns

None.

Definition at line 18 of file io.cc.

5.3.2.4 printBoard()

```
void IO::printBoard (
    const Board & pos ) [noexcept]
```

Prints out the board to stdio.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

None.

Definition at line 33 of file io.cc.

5.3.2.5 printMoveList()

```
void IO::printMoveList (
    const MoveList & list ) [noexcept]
```

Prints out the [MoveList](#) to stdio.

Parameters

<i>list</i>	The MoveList to print.
-------------	----------------------------------------

Returns

None.

Definition at line 61 of file io.cc.

5.3.2.6 printSearchDetails()

```
void IO::printSearchDetails (
    const SearchInfo & info,
```

```

    int32_t curDepth,
    int32_t bestScore,
    PvTable & pv,
    int32_t pvMoves ) [noexcept]

```

Prints out the search details to stdio.

Parameters

<i>info</i>	The engine's searchInfo struct
<i>curDepth</i>	The current search depth.
<i>bestScore</i>	The evaluation of the current board state
<i>pv</i>	The transposition table.
<i>pvMoves</i>	The length of the Pv line
<i>pos</i>	The current board state.

Returns

None.

Definition at line 73 of file io.cc.

5.3.3 Variable Documentation

5.3.3.1 epstr

```
const std::unordered_map<uint32_t, std::string> IO::epstr
```

Initial value:

```

=
    {{71,"a6"}, {72,"b6"}, {73,"c6"}, {74,"d6"}, {75,"e6"}, {76,"f6"}, {77,"g6"}, {78,"h6"},
    {41,"a3"}, {42,"b3"}, {43,"c3"}, {44,"d3"}, {45,"e3"}, {46,"f3"}, {47,"g3"}, {48,"h3"}, {99, "None"}}

```

Dictionary to provide string representations of enPassant sq's

Definition at line 40 of file io.h.

5.3.3.2 FileChar

```
const std::string IO::FileChar = "abcdefgh"
```

Dictionary to provide string representations of files

Definition at line 36 of file io.h.

5.3.3.3 PceChar

```
const std::string IO::PceChar = ".PNBRQKpnbrqk"
```

Dictionary to provide string representations of pieces

Definition at line 24 of file io.h.

5.3.3.4 RankChar

```
const std::string IO::RankChar = "12345678"
```

Dictionary to provide string representations of ranks

Definition at line 32 of file io.h.

5.3.3.5 SideChar

```
const std::string IO::SideChar = "wb-"
```

Dictionary to provide string representations of sides

Definition at line 28 of file io.h.

5.4 MoveFlags Namespace Reference

This namespace stores flags used to extract parameters from a move.

Variables

- constexpr int32_t **SQ** = 0x7F
- constexpr int32_t **EP** = 0x40000
- constexpr int32_t **PS** = 0x80000
- constexpr int32_t **CA** = 0x1000000
- constexpr int32_t **CAP** = 0x7C000
- constexpr int32_t **PROM** = 0xF00000

5.4.1 Detailed Description

This namespace stores flags used to extract parameters from a move.

5.5 MvvLva Namespace Reference

Variables

- `std::array< std::array< int32_t, kNumPceTypes >, kNumPceTypes >` **MvvLvaScore**
- `constexpr std::array< int32_t, kNumPceTypes >` **victimScore** {0, 100, 200, 300, 400, 500, 600, 100, 200, 300, 400, 500, 600}

5.5.1 Detailed Description

used in move ordering. MostValuableVictim-LeastValuableAttacker

5.6 PolyKeys Namespace Reference

Variables

- `constexpr std::array< uint64_t, 781 >` **Random64**

5.6.1 Detailed Description

This namespace holds all 781 polyglot hashkeys for use in the opening credits to http://hgm.nubati.net/book_format.html

5.7 Value Namespace Reference

Provides several constant values used for evaluating the board state.

Variables

- `constexpr int32_t` **kInfinity** = 30000
Highest possible score.
- `constexpr int32_t` **kMateScore** = 29000
Score for a Checkmate.
- `constexpr int32_t` **kIsolatedPawn** = -10
Score bonus for isolated pawns (no same color pawn on adjacent files)
- `constexpr int32_t` **kOpenRookFile** = 10
Score bonus for a Rook on an open file (no pawns)
- `constexpr int32_t` **kSemiOpenRookFile** = 5
Score bonus for a Rook on a semi-open file (no same color pawn)
- `constexpr int32_t` **kOpenQueenFile** = 5
Score bonus for a Queen on an open file (no pawns)
- `constexpr int32_t` **kSemiOpenQueenFile** = 3
Score bonus for a Queen on a semi-open file (no same color pawn)

- constexpr int32_t [kEndGameThreshold](#) = PieceInfo::PieceVal[wR] + 2 * PieceInfo::PieceVal[wB] + 2 * PieceInfo::PieceVal[wP]
Material threshold to determine when the endgame starts.
- constexpr int32_t [kBishopPair](#) = 30
Score bonus for the Bishop pair.
- constexpr std::array< int32_t, kNumFilesRanks > [passedPawnScore](#) {0, 5, 10, 20, 35, 60, 100, 200}
Score bonus for Passed pawns based on distance.
- constexpr std::array< int32_t, kChessboardSize > [PawnTable](#)
Scores the position of the Pawns.
- constexpr std::array< int32_t, kChessboardSize > [KnightTable](#)
Scores the position of the Knights.
- constexpr std::array< int32_t, kChessboardSize > [BishopTable](#)
Scores the position of the Bishops.
- constexpr std::array< int32_t, kChessboardSize > [RookTable](#)
Scores the position of the Rooks.
- constexpr std::array< int32_t, kChessboardSize > [KingEndGame](#)
Scores the king in the late game. Prioritizes the center.
- constexpr std::array< int32_t, kChessboardSize > [KingOpening](#)
Scores the king for the earlygame. Prioritizes staying safe.

5.7.1 Detailed Description

Provides several constant values used for evaluating the board state.

5.7.2 Variable Documentation

5.7.2.1 BishopTable

```
constexpr std::array<int32_t, kChessboardSize> Value::BishopTable
```

Initial value:

```
{
    0 , 0 , -10 , 0 , 0 , -10 , 0 , 0 ,
    0 , 0 , 0 , 10 , 10 , 0 , 0 , 0 ,
    0 , 0 , 10 , 15 , 15 , 10 , 0 , 0 ,
    0 , 10 , 15 , 20 , 20 , 15 , 10 , 0 ,
    0 , 10 , 15 , 20 , 20 , 15 , 10 , 0 ,
    0 , 0 , 10 , 15 , 15 , 10 , 0 , 0 ,
    0 , 0 , 0 , 10 , 10 , 0 , 0 , 0 ,
    0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 ,
}
```

Scores the position of the Bishops.

Definition at line 96 of file eval.h.

5.7.2.2 KingEndGame

```
constexpr std::array<int32_t, kChessboardSize> Value::KingEndGame
```

Initial value:

```
= {
    -50 ,  -10 ,   0 ,   0 ,   0 ,   0 ,  -10 ,  -50 ,
    -10 ,   0 ,  10 ,  10 ,  10 ,  10 ,   0 ,  -10 ,
     0 ,  10 ,  15 ,  15 ,  15 ,  15 ,  10 ,   0 ,
     0 ,  10 ,  15 ,  20 ,  20 ,  15 ,  10 ,   0 ,
     0 ,  10 ,  15 ,  20 ,  20 ,  15 ,  10 ,   0 ,
     0 ,  10 ,  15 ,  15 ,  15 ,  15 ,  10 ,   0 ,
    -10 ,   0 ,  10 ,  10 ,  10 ,  10 ,   0 ,  -10 ,
    -50 ,  -10 ,   0 ,   0 ,   0 ,   0 ,  -10 ,  -50 ,
}
```

Scores the king in the late game. Prioritizes the center.

Definition at line 123 of file eval.h.

5.7.2.3 KingOpening

```
constexpr std::array<int32_t, kChessboardSize> Value::KingOpening
```

Initial value:

```
= {
    0 ,   5 ,   5 ,  -10 ,  -10 ,   0 ,  10 ,   5 ,
   -30 ,  -30 ,  -30 ,  -30 ,  -30 ,  -30 ,  -30 ,  -30 ,
   -50 ,  -50 ,  -50 ,  -50 ,  -50 ,  -50 ,  -50 ,  -50 ,
   -70 ,  -70 ,  -70 ,  -70 ,  -70 ,  -70 ,  -70 ,  -70 ,
   -70 ,  -70 ,  -70 ,  -70 ,  -70 ,  -70 ,  -70 ,  -70 ,
   -70 ,  -70 ,  -70 ,  -70 ,  -70 ,  -70 ,  -70 ,  -70 ,
   -70 ,  -70 ,  -70 ,  -70 ,  -70 ,  -70 ,  -70 ,  -70 ,
}
```

Scores the king for the earlygame. Prioritizes staying safe.

Definition at line 137 of file eval.h.

5.7.2.4 KnightTable

```
constexpr std::array<int32_t, kChessboardSize> Value::KnightTable
```

Initial value:

```
{
    0 ,  -10 ,   0 ,   0 ,   0 ,   0 ,  -10 ,   0 ,
    0 ,   0 ,   0 ,   5 ,   5 ,   0 ,   0 ,   0 ,
    0 ,   0 ,  10 ,  10 ,  10 ,  10 ,   0 ,   0 ,
    0 ,   0 ,  10 ,  20 ,  20 ,  10 ,   5 ,   0 ,
    5 ,  10 ,  15 ,  20 ,  20 ,  15 ,  10 ,   5 ,
    5 ,  10 ,  10 ,  20 ,  20 ,  10 ,  10 ,   5 ,
    0 ,   0 ,   5 ,  10 ,  10 ,   5 ,   0 ,   0 ,
    0 ,   0 ,   0 ,   0 ,   0 ,   0 ,   0 ,   0 ,
}
```

Scores the position of the Knights.

Definition at line 82 of file eval.h.

5.7.2.5 PawnTable

```
constexpr std::array<int32_t, kChessboardSize> Value::PawnTable
```

Initial value:

```
{
    0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 ,
    10 , 10 , 0 , -10 , -10 , 0 , 10 , 10 ,
    5 , 0 , 0 , 5 , 5 , 0 , 0 , 5 ,
    0 , 0 , 10 , 20 , 20 , 10 , 0 , 0 ,
    5 , 5 , 5 , 10 , 10 , 5 , 5 , 5 ,
    10 , 10 , 10 , 20 , 20 , 10 , 10 , 10 ,
    20 , 20 , 20 , 30 , 30 , 20 , 20 , 20 ,
    0 , 0 , 0 , 0 , 0 , 0 , 0 , 0
}
```

Scores the position of the Pawns.

Definition at line 68 of file eval.h.

5.7.2.6 RookTable

```
constexpr std::array<int32_t, kChessboardSize> Value::RookTable
```

Initial value:

```
{
    0 , 0 , 5 , 10 , 10 , 5 , 0 , 0 ,
    0 , 0 , 5 , 10 , 10 , 5 , 0 , 0 ,
    0 , 0 , 5 , 10 , 10 , 5 , 0 , 0 ,
    0 , 0 , 5 , 10 , 10 , 5 , 0 , 0 ,
    0 , 0 , 5 , 10 , 10 , 5 , 0 , 0 ,
    0 , 0 , 5 , 10 , 10 , 5 , 0 , 0 ,
    25 , 25 , 25 , 25 , 25 , 25 , 25 , 25 ,
    0 , 0 , 5 , 10 , 10 , 5 , 0 , 0
}
```

Scores the position of the Rooks.

Definition at line 110 of file eval.h.

Chapter 6

Data Structure Documentation

6.1 Board Class Reference

```
#include <board.h>
```

Public Member Functions

- **Board** (const std::string &fen) noexcept
- void **parseFEN** (const std::string &) noexcept
*Set up the **Board** representation of the calling **Board** object.*
- uint32_t **sqAttacked** (const uint32_t sq, const uint32_t side) const noexcept
Gets the number of times a square is attacked by a given side.
- **MoveList** **getAllMoves** () const noexcept
Gets all the possible moves, including ones that happen during check.
- **MoveList** **getAllCaptureMoves** () const noexcept
Get all possible moves that involve capture, including ones that happen during check.
- void **flipBoard** () noexcept
*Flips the internal **Board** representation along the central file.*
- bool **inCheck** () noexcept
Checks if the current side is in check.

Data Fields

- std::vector< uint32_t > **pieces**
- std::vector< uint64_t > **pawns**
- std::vector< int32_t > **king_sq**
- uint32_t **side_to_move**
- uint32_t **en_pas**
- uint32_t **fifty_move**
- int32_t **ply**
- int32_t **hist_ply**
- uint64_t **pos_key**
- uint32_t **castle_perm**
- std::vector< std::vector< uint32_t > > **piece_list**
- std::vector< uint32_t > **big_pce**
- std::vector< uint32_t > **maj_pce**
- std::vector< uint32_t > **min_pce**
- std::vector< uint32_t > **material**
- std::vector< **UndoMove** > **history**
- std::vector< std::vector< int32_t > > **search_hist**
- std::vector< std::vector< **Move** > > **search_killers**

Private Member Functions

- void `resetBoard` (void) noexcept
Resets all board variables to default values.
- void `updatePieceLists` (void) noexcept
Updates all internal `Board` vectors related to pieces.
- void `setUpPieces` (const std::string &pieces) noexcept
Parses the first FEN argument and places the pieces on the board as indicated.
- void `setUpCastlePerm` (const std::string &perm) noexcept
Parses the castle argument of the FEN.
- void `getEnPassant` (const std::string &enPas) noexcept
Parses the enPassant argument of the FEN.
- void `setUpMoveCounters` (std::istream &stream, std::string §ion) noexcept
Parses the ply counter argument(s) of the FEN if they exist.

6.1.1 Detailed Description

Handles the entire board representation and all information regarding the board.

Definition at line 22 of file board.h.

6.1.2 Member Function Documentation

6.1.2.1 `flipBoard()`

```
void Board::flipBoard ( ) [noexcept]
```

Flips the internal `Board` representation along the central file.

Parameters

None	
------	--

Returns

None

Definition at line 355 of file board.cc.

6.1.2.2 `getAllCaptureMoves()`

```
MoveList Board::getAllCaptureMoves ( ) const [noexcept]
```

Get all possible moves that involve capture, including ones that happen during check.

Parameters

None	
------	--

Returns

A [MoveList](#) with all possible capture moves.

Definition at line 347 of file board.cc.

6.1.2.3 getAllMoves()

```
MoveList Board::getAllMoves ( ) const [noexcept]
```

Gets all the possible moves, including ones that happen during check.

Parameters

None	
------	--

Returns

A [MoveList](#) with all possible moves

Definition at line 340 of file board.cc.

6.1.2.4 getEnPassant()

```
void Board::getEnPassant (
    const std::string & enPas ) [private], [noexcept]
```

Parses the enPassant argument of the FEN.

Parameters

None	
------	--

Returns

None

Definition at line 176 of file board.cc.

6.1.2.5 inCheck()

```
bool Board::inCheck ( ) [noexcept]
```

Checks if the current side is in check.

Parameters

<i>None</i>	
-------------	--

Returns

true if the current side to move is in check, false otherwise.

Definition at line 385 of file board.cc.

6.1.2.6 parseFEN()

```
void Board::parseFEN (
    const std::string & fen ) [noexcept]
```

Set up the [Board](#) representation of the calling [Board](#) object.

Example: rnbqkbnr/pppppppp/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1

^	^	^	^	^
1	2	3	4	5 6

1. Piece locations
2. Current side to move
3. Castling permissions
4. EnPassant square
5. Halfmove clock (fifty move count)
6. Fullmove number

Parameters

<i>fen</i>	The Forsyth-Edwards Notation string
------------	-------------------------------------

Returns

None

Definition at line 202 of file board.cc.

6.1.2.7 resetBoard()

```
void Board::resetBoard (  
    void ) [private], [noexcept]
```

Resets all board variables to default values.

Parameters

None	
------	--

Returns

None

Definition at line 72 of file board.cc.

6.1.2.8 setUpCastlePerm()

```
void Board::setUpCastlePerm (  
    const std::string & perm ) [private], [noexcept]
```

Parses the castle argument of the FEN.

Parameters

None	
------	--

Returns

None

Definition at line 150 of file board.cc.

6.1.2.9 setUpMoveCounters()

```
void Board::setUpMoveCounters (  
    std::istringstream & stream,  
    std::string & section ) [private], [noexcept]
```

Parses the ply counter argument(s) of the FEN if they exist.

Parameters

None	
------	--

Returns

None

Definition at line 186 of file board.cc.

6.1.2.10 setUpPieces()

```
void Board::setUpPieces (
    const std::string & pieces ) [private], [noexcept]
```

Parses the first FEN argument and places the pieces on the board as indicated.

Parameters

<i>None</i>	
-------------	--

Returns

None

Definition at line 110 of file board.cc.

6.1.2.11 sqAttacked()

```
uint32_t Board::sqAttacked (
    const uint32_t sq,
    const uint32_t side ) const [noexcept]
```

Gets the number of times a square is attacked by a given side.

Parameters

<i>sq</i>	The target square.
<i>side</i>	The attacking side

Returns

The number of times the target square is attacked by the given side

Definition at line 262 of file board.cc.

6.1.2.12 updatePieceLists()

```
void Board::updatePieceLists (
    void ) [private], [noexcept]
```

Updates all internal [Board](#) vectors related to pieces.

Parameters

None	
------	--

Returns

None

Definition at line 232 of file board.cc.

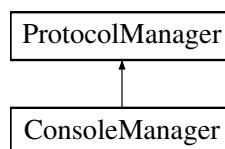
The documentation for this class was generated from the following files:

- [/home/michael/Documents/Projects/C++/Chess-Engine/include/board.h](#)
- [/home/michael/Documents/Projects/C++/Chess-Engine/src/board.cc](#)

6.2 ConsoleManager Class Reference

```
#include <console.h>
```

Inheritance diagram for ConsoleManager:



Public Member Functions

- void [loop](#) () override
Starts the protocol loop.
- int32_t [getProtocol](#) () override
Returns the protocol identifier for the current protocol.

Additional Inherited Members

6.2.1 Detailed Description

See [ProtocolManager](#) documentation

Definition at line 14 of file console.h.

6.2.2 Member Function Documentation

6.2.2.1 getProtocol()

```
int32_t ConsoleManager::getProtocol ( ) [override], [virtual]
```

Returns the protocol identifier for the current protocol.

Parameters

<i>None</i>	
-------------	--

Returns

The protocol identifier

Implements [ProtocolManager](#).

Definition at line 191 of file console.cc.

6.2.2.2 loop()

```
void ConsoleManager::loop ( ) [override], [virtual]
```

Starts the protocol loop.

Parameters

<i>None</i>	
-------------	--

Returns

None

Implements [ProtocolManager](#).

Definition at line 22 of file console.cc.

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

- [/home/michael/Documents/Projects/C++/Chess-Engine/include/console.h](#)
- [/home/michael/Documents/Projects/C++/Chess-Engine/src/console.cc](#)

6.3 Engine Class Reference

Public Member Functions

- **Engine** (const [Engine](#) &)=delete
- void **operator=** (const [Engine](#) &)=delete
- void **start** () noexcept

Static Public Member Functions

- static [Engine](#) & **getInstance** ()
Gets the instance of the [Engine](#) singleton.
- static [EngineConfig](#) & **getConfig** ()
Gets the instance of the [Engine](#) config.
- static [PolyBook](#) & **getBook** ()
Gets the Polyglot opening book that is being used by the engine.

Private Member Functions

- void **printGreeting** () const noexcept

Private Attributes

- std::unique_ptr< [ProtocolManager](#) > **protocol**
- [EngineConfig](#) **config**
- [PolyBook](#) **book**

6.3.1 Detailed Description

Definition at line 22 of file engine.h.

6.3.2 Member Function Documentation

6.3.2.1 getBook()

```
static PolyBook& Engine::getBook ( ) [inline], [static]
```

Gets the Polyglot opening book that is being used by the engine.

Parameters

<i>None</i>	
-------------	--

Returns

The Polyglot opening book that the engine is using.

Definition at line 49 of file engine.h.

6.3.2.2 getConfig()

```
static EngineConfig& Engine::getConfig ( ) [inline], [static]
```

Gets the instance of the [Engine](#) config.

Parameters

<i>None</i>	
-------------	--

Returns

The [Engine](#) config

Definition at line 40 of file engine.h.

6.3.2.3 getInstance()

```
static Engine& Engine::getInstance ( ) [inline], [static]
```

Gets the instance of the [Engine](#) singleton.

Parameters

<i>None</i>	
-------------	--

Returns

The [Engine](#) instance

Definition at line 30 of file engine.h.

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

- /home/michael/Documents/Projects/C++/Chess-Engine/include/[engine.h](#)
- /home/michael/Documents/Projects/C++/Chess-Engine/src/[engine.cc](#)

6.4 EngineConfig Struct Reference

Stores the flags for the engine.

```
#include <engine.h>
```

Data Fields

- bool **useBook**

6.4.1 Detailed Description

Stores the flags for the engine.

Definition at line 16 of file engine.h.

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

- [/home/michael/Documents/Projects/C++/Chess-Engine/include/engine.h](#)

6.5 Evaluator Class Reference

Public Member Functions

- int32_t [evaluatePosition](#) (const [Board](#) &pos) noexcept
Evaluates the score for the current board position for the side to move.

Private Member Functions

- int32_t [evalPawns](#) (const [Board](#) &pos) noexcept
Evaluates the strength of the Pawns.
- int32_t [evalBishops](#) (const [Board](#) &pos) noexcept
Evaluates the strength of the Bishop.
- int32_t [evalRooks](#) (const [Board](#) &pos) noexcept
Evaluates the strength of the Rook.
- int32_t [evalKnights](#) (const [Board](#) &pos) noexcept
Evaluates the strength of the Knights.
- int32_t [evalQueens](#) (const [Board](#) &pos) noexcept
Evaluates the strength of the Queens.
- int32_t [evalKings](#) (const [Board](#) &pos) noexcept
Evaluates the strength of the Kings.
- bool [drawnMaterial](#) (const [Board](#) &pos) noexcept
Determines if there is enough material to end in checkmate.

6.5.1 Detailed Description

Definition at line 159 of file eval.h.

6.5.2 Member Function Documentation

6.5.2.1 drawnMaterial()

```
bool Evaluator::drawnMaterial (
    const Board & pos ) [private], [noexcept]
```

Determines if there is enough material to end in checkmate.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

true if there is enough material on either side to lead to checkmate, false otherwise.

Definition at line 188 of file eval.cc.

6.5.2.2 evalBishops()

```
int32_t Evaluator::evalBishops (
    const Board & pos ) [private], [noexcept]
```

Evaluates the strength of the Bishop.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

The score attributed to the Bishop.

Definition at line 52 of file eval.cc.

6.5.2.3 evalKings()

```
int32_t Evaluator::evalKings (
    const Board & pos ) [private], [noexcept]
```

Evaluates the strength of the Kings.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

The score attributed to the Kings.

Definition at line 142 of file eval.cc.

6.5.2.4 evalKnights()

```
int32_t Evaluator::evalKnights (
    const Board & pos ) [private], [noexcept]
```

Evaluates the strength of the Knights.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

The score attributed to the Knights.

Definition at line 171 of file eval.cc.

6.5.2.5 evalPawns()

```
int32_t Evaluator::evalPawns (
    const Board & pos ) [private], [noexcept]
```

Evaluates the strength of the Pawns.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

The score attributed to the Pawns.

Definition at line 14 of file eval.cc.

6.5.2.6 evalQueens()

```
int32_t Evaluator::evalQueens (
    const Board & pos ) [private], [noexcept]
```

Evaluates the strength of the Queens.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

The score attributed to the Queens.

Definition at line 108 of file eval.cc.

6.5.2.7 evalRooks()

```
int32_t Evaluator::evalRooks (
    const Board & pos ) [private], [noexcept]
```

Evaluates the strength of the Rook.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

The score attributed to the Rook.

Definition at line 72 of file eval.cc.

6.5.2.8 evaluatePosition()

```
int32_t Evaluator::evaluatePosition (
    const Board & pos ) [noexcept]
```

Evaluates the score for the current board position for the side to move.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

The overall score of the position

Definition at line 243 of file eval.cc.

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

- /home/michael/Documents/Projects/C++/Chess-Engine/include/eval.h
- /home/michael/Documents/Projects/C++/Chess-Engine/src/eval.cc

6.6 MM Class Reference

Static Public Member Functions

- static bool [makeMove](#) ([Board](#) &pos, const [Move](#) &moveInfo) noexcept
Makes the provided move (move piece/capture/update board)
- static void [takeMove](#) ([Board](#) &pos) noexcept
Undos the last move that was made which is stored in Board::history.
- static void [makeNullMove](#) ([Board](#) &pos) noexcept
Similar in concept to Make move, but gives the opposing side a free move.
- static void [takeNullMove](#) ([Board](#) &pos) noexcept
Undos the last null move.

Static Private Member Functions

- static void [clearPiece](#) (const uint32_t sq, [Board](#) &pos) noexcept
Removes the piece on the given square from the internal board representation.
- static void [addPiece](#) (const uint32_t sq, [Board](#) &pos, const uint32_t pce) noexcept
Adds the piece on the given square to the internal board representation.
- static void [movePiece](#) (const uint32_t src, const uint32_t dest, [Board](#) &pos) noexcept
Moves the piece from the source square to the dest square and performs capture if applicable.

6.6.1 Detailed Description

Definition at line 34 of file movemaker.h.

6.6.2 Member Function Documentation

6.6.2.1 addPiece()

```
void MM::addPiece (
    const uint32_t sq,
    Board & pos,
    const uint32_t pce ) [static], [private], [noexcept]
```

Adds the piece on the given square to the internal board representation.

Parameters

<i>sq</i>	The square to add the piece to.
<i>pos</i>	The current board state.
<i>pce</i>	The piece to add to the target square

Returns

None

Definition at line 45 of file movemaker.cc.

6.6.2.2 clearPiece()

```
void MM::clearPiece (
    const uint32_t sq,
    Board & pos ) [static], [private], [noexcept]
```

Removes the piece on the given square from the internal board representation.

Parameters

<i>sq</i>	The square to clear.
<i>pos</i>	The current board state.

Returns

None

Definition at line 18 of file movemaker.cc.

6.6.2.3 makeMove()

```
bool MM::makeMove (
    Board & pos,
    const Move & moveInfo ) [static], [noexcept]
```

Makes the provided move (move piece/capture/update board)

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

true if move was successfully made (legal), false otherwise

Definition at line 90 of file movemaker.cc.

6.6.2.4 makeNullMove()

```
void MM::makeNullMove (  
    Board & pos ) [static], [noexcept]
```

Similar in concept to Make move, but gives the opposing side a free move.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

None

Definition at line 249 of file movemaker.cc.

6.6.2.5 movePiece()

```
void MM::movePiece (  
    const uint32_t src,  
    const uint32_t dest,  
    Board & pos ) [static], [private], [noexcept]
```

Moves the piece from the source square to the dest square and performs capture if applicable.

Parameters

<i>src</i>	The square the piece starts on.
<i>dest</i>	The square the piece is moving to.
<i>pos</i>	The current board state.

Returns

None

Definition at line 71 of file movemaker.cc.

6.6.2.6 takeMove()

```
void MM::takeMove (
    Board & pos ) [static], [nothrow]
```

Undos the last move that was made which is stored in Board::history.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

None

Definition at line 179 of file movemaker.cc.

6.6.2.7 takeNullMove()

```
void MM::takeNullMove (
    Board & pos ) [static], [nothrow]
```

Undos the last null move.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

None

Definition at line 272 of file movemaker.cc.

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

- /home/michael/Documents/Projects/C++/Chess-Engine/include/[movemaker.h](#)
- /home/michael/Documents/Projects/C++/Chess-Engine/src/[movemaker.cc](#)

6.7 Move Class Reference

Public Member Functions

- **Move** (uint32_t *from*, uint32_t *to*, uint32_t *captured*, uint32_t *prom*, uint32_t *flag*) noexcept
- **Move** (uint32_t *_move*, int32_t *_score*) noexcept
- **Move** (uint32_t *_move*) noexcept

- **Move** (const [Move](#) &o) noexcept
- **Move** ([Move](#) &&o) noexcept
- constexpr uint32_t [from](#) () const
Gets the square that the move is starting from.
- constexpr uint32_t [to](#) () const
Gets the square that the move is ending on.
- constexpr uint32_t [captured](#) () const
Gets the piece That was captured.
- constexpr uint32_t [enPassant](#) () const
Checks if the move was an EnPassant move.
- constexpr uint32_t [pawnStart](#) () const
Checks if the move was a pawn start (pawn double move).
- constexpr uint32_t [promoted](#) () const
Gets the piece that was promoted to if any.
- constexpr uint32_t [castle](#) () const
Checks if the move was a castle move.
- constexpr bool [isNull](#) () const
Checks if the move was NULLMOVE.
- constexpr bool [wasCapture](#) () const
Checks if the move was a capture.
- constexpr bool [wasPromotion](#) () const
Checks if the move was a promotion.
- const std::string [toString](#) () const noexcept
Gets the string representation of a move (e.g "e2e4").
- [Move](#) & **operator=** (const [Move](#) &o) noexcept
- [Move](#) & **operator=** ([Move](#) &&o) noexcept
- bool **operator==** (const [Move](#) &rhs) const noexcept
- bool **operator!=** (const [Move](#) &rhs) const noexcept

Data Fields

- uint32_t **move**
- int32_t **score**

6.7.1 Detailed Description

Definition at line 40 of file move.h.

6.7.2 Member Function Documentation

6.7.2.1 captured()

```
constexpr uint32_t Move::captured ( ) const [inline]
```

Gets the piece That was captured.

Parameters

<i>None</i>	
-------------	--

Returns

The piece that was captured.

Definition at line 69 of file move.h.

6.7.2.2 castle()

```
constexpr uint32_t Move::castle ( ) const [inline]
```

Checks if the move was a castle move.

Parameters

<i>None</i>	
-------------	--

Returns

true if there was a castle move, false otherwise.

Definition at line 93 of file move.h.

6.7.2.3 enPassant()

```
constexpr uint32_t Move::enPassant ( ) const [inline]
```

Checks if the move was an EnPassant move.

Parameters

<i>None</i>	
-------------	--

Returns

true if there was EnPassant, false otherwise.

Definition at line 75 of file move.h.

6.7.2.4 from()

```
constexpr uint32_t Move::from ( ) const [inline]
```

Gets the square that the move is starting from.

Parameters

<i>None</i>	
-------------	--

Returns

The source square

Definition at line 57 of file move.h.

6.7.2.5 isNull()

```
constexpr bool Move::isNull ( ) const [inline]
```

Checks if the move was NULLMOVE.

Parameters

<i>None</i>	
-------------	--

Returns

true if (*this) == NULLMOVE, false otherwise.

Definition at line 100 of file move.h.

6.7.2.6 pawnStart()

```
constexpr uint32_t Move::pawnStart ( ) const [inline]
```

Checks if the move was a pawn start (pawn double move).

Parameters

<i>None</i>	
-------------	--

Returns

true if there was a pawn start, false otherwise.

Definition at line 81 of file move.h.

6.7.2.7 promoted()

```
constexpr uint32_t Move::promoted ( ) const [inline]
```

Gets the piece that was promoted to if any.

Parameters

<i>None</i>	
-------------	--

Returns

The piece that was promoted to if any.

Definition at line 87 of file move.h.

6.7.2.8 to()

```
constexpr uint32_t Move::to ( ) const [inline]
```

Gets the square that the move is ending on.

Parameters

<i>None</i>	
-------------	--

Returns

The destination square

Definition at line 63 of file move.h.

6.7.2.9 toString()

```
const std::string Move::toString ( ) const [noexcept]
```

Gets the string representation of a move (e.g "e2e4").

Parameters

None	
------	--

Returns

The string representation of a move.

Definition at line 35 of file move.cc.

6.7.2.10 wasCapture()

```
constexpr bool Move::wasCapture ( ) const [inline]
```

Checks if the move was a capture.

Parameters

None	
------	--

Returns

true if there was a capture, false otherwise.

Definition at line 106 of file move.h.

6.7.2.11 wasPromotion()

```
constexpr bool Move::wasPromotion ( ) const [inline]
```

Checks if the move was a promotion.

Parameters

None	
------	--

Returns

true if there was a promotion, false otherwise.

Definition at line 112 of file move.h.

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

- /home/michael/Documents/Projects/C++/Chess-Engine/include/[move.h](#)
- /home/michael/Documents/Projects/C++/Chess-Engine/src/[move.cc](#)

6.8 MoveList Class Reference

Public Member Functions

- **MoveList** (const [MoveList](#) &o)
- **MoveList** ([MoveList](#) &&o) noexcept
- void [generateAllMoves](#) (const [Board](#) &pos) noexcept
Add all possible moves to the [MoveList](#).
- void [generateAllCaptureMoves](#) (const [Board](#) &pos) noexcept
Add all possible capture moves to the [MoveList](#).
- void [reorderList](#) (int32_t idx) noexcept
Reorders the movelist to maximize alpha-beta hits.
- uint32_t [size](#) () const noexcept
Get the number of possible moves in the [MoveList](#).
- const [Move](#) & **operator[]** (const int idx) const noexcept
- [Move](#) & **operator[]** (const int idx) noexcept

Static Public Attributes

- static constexpr uint32_t **kPvMoveBonus** = 2000000

Private Member Functions

- void [addQuietMove](#) (const [Board](#) &pos, [Move](#) &&move) noexcept
Add a move that does not capture to the [MoveList](#).
- void [addCaptureMove](#) (const [Board](#) &pos, [Move](#) &&move) noexcept
Add a move that does capture to the [MoveList](#).
- void [addEnPasMove](#) (const [Board](#) &pos, [Move](#) &&move) noexcept
Add a move that performs enPassant to the [MoveList](#).
- void [addPawnMove](#) (const [Board](#) &pos, uint32_t from, uint32_t to, uint32_t side) noexcept
Add a move that involves a pawn.
- void [addPawnCaptureMove](#) (const [Board](#) &pos, uint32_t from, uint32_t to, uint32_t cap, uint32_t side) noexcept
Add a capture move that involves a pawn.
- void [generatePawnMoves](#) (const [Board](#) &pos, uint32_t side) noexcept
Add all Pawn moves to the [MoveList](#).
- void [generateBishopMoves](#) (const [Board](#) &pos, uint32_t side) noexcept
Add all Bishop moves to the [MoveList](#).
- void [generateRookMoves](#) (const [Board](#) &pos, uint32_t side) noexcept
Add all Rook moves to the [MoveList](#).
- void [generateQueenMoves](#) (const [Board](#) &pos, uint32_t side) noexcept
Add all Queen moves to the [MoveList](#).
- void [generateKnightMoves](#) (const [Board](#) &pos, uint32_t side) noexcept
Add all Knight moves to the [MoveList](#).
- void [generateKingMoves](#) (const [Board](#) &pos, uint32_t side) noexcept
Add all King moves to the [MoveList](#).
- void [generateSlidingMoves](#) (const [Board](#) &pos, uint32_t side) noexcept
Add all Rook/Bishop/Queen moves to the [MoveList](#).
- void [generateNonSlidingMoves](#) (const [Board](#) &pos, uint32_t side) noexcept
Add all Knight/King moves to the [MoveList](#).
- void [generateCastlingMoves](#) (const [Board](#) &pos, uint32_t side) noexcept
Add all Castling moves to the [MoveList](#).

Private Attributes

- `std::vector< Move > moves`

Static Private Attributes

- `static constexpr uint32_t kMaxPossibleMoves = 256`
- `static constexpr uint32_t kCaptureBonus = 1000000`
- `static constexpr uint32_t kPrimaryKillerBonus = 900000`
- `static constexpr uint32_t kSecondaryKillerBonus = 800000`

6.8.1 Detailed Description

Definition at line 24 of file `movelist.h`.

6.8.2 Member Function Documentation

6.8.2.1 addCaptureMove()

```
void MoveList::addCaptureMove (
    const Board & pos,
    Move && move ) [private], [noexcept]
```

Add a move that does capture to the [MoveList](#).

Parameters

<i>pos</i>	The current board state.
<i>move</i>	The move to add.

Returns

None

Definition at line 41 of file `movelist.cc`.

6.8.2.2 addEnPasMove()

```
void MoveList::addEnPasMove (
    const Board & pos,
    Move && move ) [private], [noexcept]
```

Add a move that performs enPasant to the [MoveList](#).

Parameters

<i>pos</i>	The current board state.
<i>move</i>	The move to add.

Returns

None

Definition at line 47 of file movelist.cc.

6.8.2.3 addPawnCaptureMove()

```
void MoveList::addPawnCaptureMove (
    const Board & pos,
    uint32_t from,
    uint32_t to,
    uint32_t cap,
    uint32_t side ) [private], [noexcept]
```

Add a capture move that involves a pawn.

Parameters

<i>pos</i>	The current board state.
<i>from</i>	The source square.
<i>to</i>	The destination square.
<i>cap</i>	The piece that was captured.
<i>side</i>	The side to move.

Returns

None

Definition at line 86 of file movelist.cc.

6.8.2.4 addPawnMove()

```
void MoveList::addPawnMove (
    const Board & pos,
    uint32_t from,
    uint32_t to,
    uint32_t side ) [private], [noexcept]
```

Add a move that involves a pawn.

Parameters

<i>pos</i>	The current board state.
<i>from</i>	The source square.
<i>to</i>	The destination square.
<i>side</i>	The side the pawn belongs to

Returns

None

Definition at line 54 of file movelist.cc.

6.8.2.5 addQuietMove()

```
void MoveList::addQuietMove (
    const Board & pos,
    Move && move ) [private], [noexcept]
```

Add a move that does not capture to the [MoveList](#).

Parameters

<i>pos</i>	The current board state.
<i>move</i>	The move to add.

Returns

None

Definition at line 24 of file movelist.cc.

6.8.2.6 generateAllCaptureMoves()

```
void MoveList::generateAllCaptureMoves (
    const Board & pos ) [noexcept]
```

Add all possible capture moves to the [MoveList](#).

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

None

Definition at line 376 of file movelist.cc.

6.8.2.7 generateAllMoves()

```
void MoveList::generateAllMoves (
    const Board & pos ) [noexcept]
```

Add all possible moves to the [MoveList](#).

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

None

Definition at line 367 of file movelist.cc.

6.8.2.8 generateBishopMoves()

```
void MoveList::generateBishopMoves (
    const Board & pos,
    uint32_t side ) [private], [noexcept]
```

Add all Bishop moves to the [MoveList](#).

Parameters

<i>pos</i>	The current board state.
<i>side</i>	The side to move.

Returns

None

Definition at line 118 of file movelist.cc.

6.8.2.9 generateCastlingMoves()

```
void MoveList::generateCastlingMoves (
    const Board & pos,
    uint32_t side ) [private], [noexcept]
```

Add all Castling moves to the [MoveList](#).

Parameters

<i>pos</i>	The current board state.
<i>side</i>	The side to move.

Returns

None

Definition at line 317 of file movelist.cc.

6.8.2.10 generateKingMoves()

```
void MoveList::generateKingMoves (
    const Board & pos,
    uint32_t side ) [private], [noexcept]
```

Add all King moves to the [MoveList](#).

Parameters

<i>pos</i>	The current board state.
<i>side</i>	The side to move.

Returns

None

Definition at line 249 of file movelist.cc.

6.8.2.11 generateKnightMoves()

```
void MoveList::generateKnightMoves (
    const Board & pos,
    uint32_t side ) [private], [noexcept]
```

Add all Knight moves to the [MoveList](#).

Parameters

<i>pos</i>	The current board state.
<i>side</i>	The side to move.

Returns

None

Definition at line 224 of file movelist.cc.

6.8.2.12 generateNonSlidingMoves()

```
void MoveList::generateNonSlidingMoves (
    const Board & pos,
    uint32_t side ) [private], [noexcept]
```

Add all Knight/King moves to the [MoveList](#).

Parameters

<i>pos</i>	The current board state.
<i>side</i>	The side to move.

Returns

None

Definition at line 270 of file movelist.cc.

6.8.2.13 generatePawnMoves()

```
void MoveList::generatePawnMoves (
    const Board & pos,
    uint32_t side ) [private], [noexcept]
```

Add all Pawn moves to the [MoveList](#).

Parameters

<i>pos</i>	The current board state.
<i>side</i>	The side to move.

Returns

None

Definition at line 276 of file movelist.cc.

6.8.2.14 generateQueenMoves()

```
void MoveList::generateQueenMoves (
    const Board & pos,
    uint32_t side ) [private], [noexcept]
```

Add all Queen moves to the [MoveList](#).**Parameters**

<i>pos</i>	The current board state.
<i>side</i>	The side to move.

Returns

None

Definition at line 170 of file movelist.cc.

6.8.2.15 generateRookMoves()

```
void MoveList::generateRookMoves (
    const Board & pos,
    uint32_t side ) [private], [noexcept]
```

Add all Rook moves to the [MoveList](#).**Parameters**

<i>pos</i>	The current board state.
<i>side</i>	The side to move.

Returns

None

Definition at line 144 of file movelist.cc.

6.8.2.16 generateSlidingMoves()

```
void MoveList::generateSlidingMoves (
    const Board & pos,
    uint32_t side ) [private], [noexcept]
```

Add all Rook/Bishop/Queen moves to the [MoveList](#).

Parameters

<i>pos</i>	The current board state.
<i>side</i>	The side to move.

Returns

None

Definition at line 217 of file movelist.cc.

6.8.2.17 reorderList()

```
void MoveList::reorderList (
    int32_t idx ) [noexcept]
```

Reorders the movelist to maximize alpha-beta hits.

Parameters

<i>idx</i>	The index to start reordering the list at.
------------	--------------------------------------------

Returns

None

Definition at line 385 of file movelist.cc.

6.8.2.18 size()

```
uint32_t MoveList::size ( ) const [noexcept]
```

Get the number of possible moves in the [MoveList](#).

Parameters

<i>None</i>	
-------------	--

Returns

The size of the [MoveList](#).

Definition at line 402 of file movelist.cc.

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

- /home/michael/Documents/Projects/C++/Chess-Engine/include/[movelist.h](#)
- /home/michael/Documents/Projects/C++/Chess-Engine/src/[movelist.cc](#)

6.9 PerftTester Class Reference

Public Member Functions

- int [perftTest](#) (uint32_t depth, [Board](#) &pos, bool print) noexcept
- void [perftTestAll](#) ([Board](#) &pos) noexcept

Private Member Functions

- void [perft](#) (uint32_t depth, [Board](#) &pos) noexcept
Recursive helper method that backtracks through all possible moves and counts the number of leaf nodes (final positions) reached.

Private Attributes

- const uint32_t **depth_limit** = 6
- uint64_t **leafNodes**

6.9.1 Detailed Description

Definition at line 13 of file tester.h.

6.9.2 Member Function Documentation

6.9.2.1 perft()

```
void PerftTester::perft (
    uint32_t depth,
    Board & pos ) [private], [noexcept]
```

Recursive helper method that backtracks through all possible moves and counts the number of leaf nodes (final positions) reached.

Parameters

<i>depth</i>	Current search depth.
<i>pos</i>	The current board state.

Returns

None

Definition at line 19 of file tester.cc.

6.9.2.2 perftTest()

```
int PerftTester::perftTest (
    uint32_t depth,
    Board & pos,
    bool print = true ) [noexcept]
```

Calls the perft helper function to count number of leaf nodes reached. If print is enabled then the engine will also output the time taken and total number of positions reached from each possible initial move from the starting position

Parameters

<i>depth</i>	Desired search depth.
<i>pos</i>	The current board state.
<i>print</i>	Print enabled

Returns

Total number of leaf nodes reached

Definition at line 38 of file tester.cc.

6.9.2.3 perftTestAll()

```
void PerftTester::perftTestAll (
    Board & pos ) [noexcept]
```

Loops through all provided positions in the perftsuite.epd file and automatically performs a perft test to depth 5 (depth 6 takes too long for some positions) and verifies the results with the provided number in the file.

Parameters

<i>depth</i>	Desired search depth.
<i>pos</i>	The current board state.

Returns

None

Definition at line 70 of file tester.cc.

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

- /home/michael/Documents/Projects/C++/Chess-Engine/include/[tester.h](#)
- /home/michael/Documents/Projects/C++/Chess-Engine/src/[tester.cc](#)

6.10 PolyBook Class Reference

```
#include <polyglot.h>
```

Public Member Functions

- bool [readBook](#) ()
Reads in from lib/polyglot-collection/Performance.bin.
- [Move](#) [getBookMove](#) ([Board](#) &pos)
Randomly returns one of the possible opening book moves for the given position.

Private Member Functions

- bool [enPasPossible](#) (const [Board](#) &pos)
Checks if enPassant is possible and there is a pawn in pace to perform the capture.
- uint64_t [polyKeyFromBoard](#) (const [Board](#) &pos)
Generates the polyglot hashkey from the current board state.
- uint16_t [endian_swap_u16](#) (uint16_t bigEnd)
Returns the little-endian value of the big-endian input.
- uint32_t [endian_swap_u32](#) (uint32_t bigEnd)
Returns the little-endian value of the big-endian input.
- uint64_t [endian_swap_u64](#) (uint64_t bigEnd)
Returns the little-endian value of the big-endian input.
- [Move](#) [convertPolyMove](#) (uint16_t polyMove, const [Board](#) &pos)
Converts a polyglot move to the internal move representation.

Private Attributes

- std::vector< [PolyglotEntry](#) > [book](#)

Static Private Attributes

- static constexpr std::array< int32_t, 13 > [polyPiece](#) { -1, 1, 3, 5, 7, 9, 11, 0, 2, 4, 6, 8, 10 }

6.10.1 Detailed Description

This class holds the in memory representation of the polyglot book

Definition at line 26 of file polyglot.h.

6.10.2 Member Function Documentation

6.10.2.1 convertPolyMove()

```
Move PolyBook::convertPolyMove (
    uint16_t polyMove,
    const Board & pos ) [private]
```

Converts a polyglot move to the internal move representation.

Parameters

<i>polyMove</i>	The book move in Polyglot form.
<i>pos</i>	The current board state

Returns

The internal representation of the poly move.

Definition at line 136 of file polyglot.cc.

6.10.2.2 endian_swap_u16()

```
uint16_t PolyBook::endian_swap_u16 (
    uint16_t bigEnd ) [private]
```

Returns the little-endian value of the big-endian input.

Parameters

<i>bigEnd</i>	The bigEndian 16 bit number
---------------	-----------------------------

Returns

The little-endian value of the input number.

Definition at line 108 of file polyglot.cc.

6.10.2.3 endian_swap_u32()

```
uint32_t PolyBook::endian_swap_u32 (
    uint32_t bigEnd ) [private]
```

Returns the little-endian value of the big-endian input.

Parameters

<i>bigEnd</i>	The bigEndian 32 bit number
---------------	-----------------------------

Returns

The little-endian value of the input number.

Definition at line 114 of file polyglot.cc.

6.10.2.4 endian_swap_u64()

```
uint64_t PolyBook::endian_swap_u64 (
    uint64_t bigEnd ) [private]
```

Returns the little-endian value of the big-endian input.

Parameters

<i>bigEnd</i>	The bigEndian 64 bit number
---------------	-----------------------------

Returns

The little-endian value of the input number.

Definition at line 123 of file polyglot.cc.

6.10.2.5 enPasPossible()

```
bool PolyBook::enPasPossible (
    const Board & pos ) [private]
```

Checks if enPassant is possible and there is a pawn in pace to perform the capture.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

true if enPasant is possible with capture, false otherwise.

Definition at line 19 of file polyglot.cc.

6.10.2.6 getBookMove()

```
Move PolyBook::getBookMove (
    Board & pos )
```

Randomly returns one of the possible opening book moves for the given position.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

A possible book move for the position.

Definition at line 161 of file polyglot.cc.

6.10.2.7 polyKeyFromBoard()

```
uint64_t PolyBook::polyKeyFromBoard (
    const Board & pos ) [private]
```

Generates the polyglot hashkey from the current board state.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

The polyglot hashkey for a given board position.

Definition at line 41 of file polyglot.cc.

6.10.2.8 readBook()

```
bool PolyBook::readBook ( )
```

Reads in from lib/polyglot-collection/Performance.bin.

Parameters

None

Returns

true if the book was successfully read into memory, false otherwise.

Definition at line 79 of file polyglot.cc.

6.10.3 Field Documentation

6.10.3.1 book

```
std::vector<PolyglotEntry> PolyBook::book [private]
```

Provides a buffer to read in the polyglot entries

Definition at line 75 of file polyglot.h.

6.10.3.2 polyPiece

```
constexpr std::array<int32_t, 13> PolyBook::polyPiece { -1, 1, 3, 5, 7, 9, 11, 0, 2, 4, 6, 8, 10} [static], [private]
```

Since the polyglot book uses different values for the pieces, this array provides a conversion

Definition at line 32 of file polyglot.h.

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

- /home/michael/Documents/Projects/C++/Chess-Engine/include/[polyglot.h](#)
- /home/michael/Documents/Projects/C++/Chess-Engine/src/[polyglot.cc](#)

6.11 PolyglotEntry Struct Reference

```
#include <polyglot.h>
```

Data Fields

- uint64_t **key**
- uint16_t **move**
- uint16_t **weight**
- uint32_t **learn**

6.11.1 Detailed Description

This Struct represents how the polyglot entries are layed out in memory

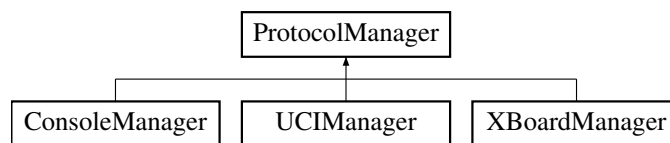
Definition at line 15 of file polyglot.h.

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

- </home/michael/Documents/Projects/C++/Chess-Engine/include/polyglot.h>

6.12 ProtocolManager Class Reference

Inheritance diagram for ProtocolManager:



Public Member Functions

- virtual void [loop](#) ()=0
Starts the protocol loop.
- virtual bool [isOver](#) ()
Checks if the engine has recieved a stop signal from the GUI.
- virtual int32_t [getProtocol](#) ()=0
Returns the protocol identifier for the current protocol.

Static Public Attributes

- static constexpr int32_t **kUCI** = 0
- static constexpr int32_t **kXBoard** = 1
- static constexpr int32_t **kConsole** = 2

Protected Attributes

- [Board](#) pos
- [SearchInfo](#) info
- [SearchAgent](#) sa

6.12.1 Detailed Description

Definition at line 15 of file protocol.h.

6.12.2 Member Function Documentation

6.12.2.1 getProtocol()

```
virtual int32_t ProtocolManager::getProtocol ( ) [pure virtual]
```

Returns the protocol identifier for the current protocol.

Parameters

<i>None</i>	
-------------	--

Returns

The protocol identifier

Implemented in [UCIManager](#), [ConsoleManager](#), and [XBoardManager](#).

6.12.2.2 isOver()

```
virtual bool ProtocolManager::isOver ( ) [inline], [virtual]
```

Checks if the engine has recieved a stop signal from the GUI.

Parameters

<i>None</i>	
-------------	--

Returns

true if a stop command has been sent, false otherwise

Definition at line 37 of file protocol.h.

6.12.2.3 loop()

```
virtual void ProtocolManager::loop ( ) [pure virtual]
```

Starts the protocol loop.

Parameters

<i>None</i>	
-------------	--

Returns

None

Implemented in [UCIManager](#), [ConsoleManager](#), and [XBoardManager](#).

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

- [/home/michael/Documents/Projects/C++/Chess-Engine/include/protocol.h](#)

6.13 PvEntry Class Reference

```
#include <pvtable.h>
```

Public Member Functions

- **PvEntry** (uint64_t key, [Move](#) _move, int32_t _score, int32_t _depth, int32_t _flags) noexcept
- **PvEntry** (uint64_t key, [Move](#) move) noexcept
- **PvEntry** (const [PvEntry](#) &o) noexcept
- **PvEntry** ([PvEntry](#) &&o) noexcept
- [PvEntry](#) & **operator=** (const [PvEntry](#) &o) noexcept
- [PvEntry](#) & **operator=** ([PvEntry](#) &&o) noexcept

Data Fields

- uint64_t **pos_key**
- [Move](#) **move**
- int32_t **score**
- int32_t **depth**
- int32_t **flags**

6.13.1 Detailed Description

This class holds the cached info of our searches for use with the Principal Variation Table

Definition at line 18 of file pvtable.h.

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

- /home/michael/Documents/Projects/C++/Chess-Engine/include/[pvtable.h](#)
- /home/michael/Documents/Projects/C++/Chess-Engine/src/[pvtable.cc](#)

6.14 PvTable Class Reference

```
#include <pvtable.h>
```

Public Member Functions

- [Move](#) **get** (const [Board](#) &pos) noexcept
Get the best move previously found for this position.
- void **insert** (const [Board](#) &pos, const [Move](#) &move) noexcept
Stashes the move associated with this position into the hash map.
- void **insert** (const [Board](#) &pos, const [Move](#) &move, int32_t score, int32_t depth, int32_t flags) noexcept
Stashes the move associated with this position into the hash map.
- int32_t **size** () const noexcept
Gets the number of entries in the transposition table.
- void **clear** () noexcept
Clears all entries in the transposition table.
- int32_t **getPvLine** ([Board](#) &pos, const uint32_t depth) noexcept
Fills in the pv array with the searched pv moves and returns the length of the line.
- bool **getHashEntry** ([Board](#) &pos, [Move](#) &pvMove, int32_t &score, int32_t alpha, int32_t beta, int32_t depth) noexcept
Checks the alpha beta bounds and returns if a better, previously cached move is found.

Data Fields

- `std::vector< Move > pv_arr`

Private Attributes

- `std::unordered_map< uint64_t, PvEntry > pv_table`

6.14.1 Detailed Description

This class holds all info for a given position with the best move found. Serves as a cache to improve evaluation speed. Also called a transposition table.

Definition at line 40 of file `pvtable.h`.

6.14.2 Member Function Documentation

6.14.2.1 `clear()`

```
void PvTable::clear ( ) [noexcept]
```

Clears all entries in the transposition table.

Parameters

<i>None</i>	
-------------	--

Returns

None

Definition at line 89 of file `pvtable.cc`.

6.14.2.2 `get()`

```
Move PvTable::get (
    const Board & pos ) [noexcept]
```

Get the best move previously found for this position.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

The best move previously found for this position.

Definition at line 58 of file pvtable.cc.

6.14.2.3 getHashEntry()

```
bool PvTable::getHashEntry (
    Board & pos,
    Move & pvMove,
    int32_t & score,
    int32_t alpha,
    int32_t beta,
    int32_t depth ) [noexcept]
```

Checks the alpha beta bounds and returns if a better, previously cached move is found.

Parameters

<i>pos</i>	The current board state.
<i>pvMove</i>	The move to store the found move in if found.
<i>score</i>	The score buffer to store the found score in if found
<i>alpha</i>	The alpha cutoff
<i>beta</i>	The beta cutoff
<i>depth</i>	The desired search depth

Returns

true if the position has an applicable cached move.

Definition at line 112 of file pvtable.cc.

6.14.2.4 getPvLine()

```
int32_t PvTable::getPvLine (
    Board & pos,
    const uint32_t depth ) [noexcept]
```

Fills in the pv array with the searched pv moves and returns the length of the line.

Parameters

<i>pos</i>	The current board state.
<i>depth</i>	The search depth.

Returns

The number of moves in the Pv line

Definition at line 94 of file pvtable.cc.

6.14.2.5 insert() [1/2]

```
void PvTable::insert (
    const Board & pos,
    const Move & move ) [noexcept]
```

Stashes the move associated with this position into the hash map.

Parameters

<i>pos</i>	The current board state.
<i>move</i>	The move to store for this position.

Returns

None

Definition at line 64 of file pvtable.cc.

6.14.2.6 insert() [2/2]

```
void PvTable::insert (
    const Board & pos,
    const Move & move,
    int32_t score,
    int32_t depth,
    int32_t flags ) [noexcept]
```

Stashes the move associated with this position into the hash map.

Parameters

<i>pos</i>	The current board state.
<i>move</i>	The move to store for this position.
<i>score</i>	The evaluation of this position.
<i>depth</i>	The depth that this position was previously searched to.
<i>flags</i>	The flags that are associated with the move.

Returns

None

Definition at line 70 of file pvtable.cc.

6.14.2.7 size()

```
int32_t PvTable::size ( ) const [noexcept]
```

Gets the number of entries in the transposition table.

Parameters

<i>None</i>	
-------------	--

Returns

The number of entries in the transposition table.

Definition at line 84 of file pvtable.cc.

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

- [/home/michael/Documents/Projects/C++/Chess-Engine/include/pvtable.h](#)
- [/home/michael/Documents/Projects/C++/Chess-Engine/src/pvtable.cc](#)

6.15 SearchAgent Class Reference

```
#include <search.h>
```

Public Member Functions

- void [searchPosition](#) ([Board](#) &pos, [SearchInfo](#) &info) noexcept
Performs iterative deepening alpha-beta search on the position.
- bool [isGameOver](#) ([Board](#) &pos) noexcept
Checks if there is checkmate, stalemate, insufficient material, or threefold repetition.

Data Fields

- [Evaluator](#) **eval**
- [PvTable](#) **pv**

Private Member Functions

- `int32_t isRepetition` (const `Board` &pos) noexcept
Checks if the position has happened before.
- `void checkStop` (`SearchInfo` &info)
Checks if the time limit has been passed.
- `void clearForSearch` (`Board` &pos, `SearchInfo` &info) noexcept
Clears the board's search_hist and search_killers vectors.
- `int32_t alphaBeta` (int32_t alpha, int32_t beta, uint32_t depth, `Board` &pos, `SearchInfo` &info, bool doNull) noexcept
Recursively performs the mini-max algorithm with alpha-beta pruning. If doNull is set, considers making a null move to improve search time.
- `int32_t quiescenceSearch` (int32_t alpha, int32_t beta, `Board` &pos, `SearchInfo` &info) noexcept
Searches similarly to alpha-beta but only on capture moves. Used to avoid the horizon effect.
- `bool threeFoldRepetition` (const `Board` &pos) noexcept
Checks if threefold repetition has occurred.
- `bool drawnMaterial` (const `Board` &pos) noexcept
Checks if there is sufficient material to checkmate either side.

Static Private Attributes

- static constexpr uint32_t `kInterval` = 0x7FF

6.15.1 Detailed Description

This class provides functions to search through the board state

Definition at line 20 of file search.h.

6.15.2 Member Function Documentation

6.15.2.1 alphaBeta()

```
int32_t SearchAgent::alphaBeta (
    int32_t alpha,
    int32_t beta,
    uint32_t depth,
    Board & pos,
    SearchInfo & info,
    bool doNull ) [private], [noexcept]
```

Recursively performs the mini-max algorithm with alpha-beta pruning. If doNull is set, considers making a null move to improve search time.

Parameters

<i>pos</i>	The current board state.
<i>alpha</i>	The alpha cutoff
<i>beta</i>	The beta cutoff
<i>depth</i>	The current depth of the search
<i>pos</i>	The current board state
<i>info</i>	The engine's searchInfo object
<i>doNull</i>	Null move flag

Returns

The evaluation of the current position

Definition at line 152 of file search.cc.

6.15.2.2 checkStop()

```
void SearchAgent::checkStop (
    SearchInfo & info ) [private]
```

Checks if the time limit has been passed.

Parameters

<i>None</i>	
-------------	--

Returns

None

Definition at line 114 of file search.cc.

6.15.2.3 clearForSearch()

```
void SearchAgent::clearForSearch (
    Board & pos,
    SearchInfo & info ) [private], [noexcept]
```

Clears the board's search_hist and search_killers vectors.

Parameters

<i>pos</i>	The current board state.
<i>info</i>	The engine's searchInfo object.

Returns

None

Definition at line 126 of file search.cc.

6.15.2.4 drawnMaterial()

```
bool SearchAgent::drawnMaterial (
    const Board & pos ) [private], [noexcept]
```

Checks if there is sufficient material to checkmate either side.

Parameters

<i>pos</i>	The current board state
------------	-------------------------

Returns

true if there is sufficient material on either side to cause checkmate.

Definition at line 34 of file search.cc.

6.15.2.5 isGameOver()

```
bool SearchAgent::isGameOver (
    Board & pos ) [noexcept]
```

Checks if there is checkmate, stalemate, insufficient material, or threefold repetition.

Parameters

<i>pos</i>	The current board state
------------	-------------------------

Returns

true if the game has concluded in one way or other, false otherwise

Definition at line 59 of file search.cc.

6.15.2.6 isRepetition()

```
int32_t SearchAgent::isRepetition (
    const Board & pos ) [private], [noexcept]
```

Checks if the position has happened before.

Parameters

<i>pos</i>	The current board state.
------------	--------------------------

Returns

true if the current position has happened before, false otherwise

Definition at line 47 of file search.cc.

6.15.2.7 quiescenceSearch()

```
int32_t SearchAgent::quiescenceSearch (
    int32_t alpha,
    int32_t beta,
    Board & pos,
    SearchInfo & info ) [private], [noexcept]
```

Searches similarly to alpha-beta but only on capture moves. Used to avoid the horizon effect.

Parameters

<i>alpha</i>	The alpha cutoff
<i>beta</i>	The beta cutoff
<i>pos</i>	The current board state
<i>info</i>	The engine's searchInfo object

Returns

The evaluation of the current position

Definition at line 303 of file search.cc.

6.15.2.8 searchPosition()

```
void SearchAgent::searchPosition (
    Board & pos,
    SearchInfo & info ) [noexcept]
```

Performs iterative deepening alpha-beta search on the position.

Parameters

<i>pos</i>	The current board state
<i>info</i>	The engine's searchInfo object

Returns

None

Definition at line 367 of file search.cc.

6.15.2.9 threeFoldRepetition()

```
bool SearchAgent::threeFoldRepetition (
    const Board & pos ) [private], [noexcept]
```

Checks if threefold repetition has occurred.

Parameters

<i>pos</i>	The current board state
------------	-------------------------

Returns

true if threefold repetition has occurred, false otherwise.

Definition at line 21 of file search.cc.

6.15.3 Field Documentation**6.15.3.1 kInterval**

```
constexpr uint32_t SearchAgent::kInterval = 0x7FF [static], [private]
```

Bitmask to check search flags every 2048 positions

Definition at line 26 of file search.h.

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

- /home/michael/Documents/Projects/C++/Chess-Engine/include/[search.h](#)
- /home/michael/Documents/Projects/C++/Chess-Engine/src/[search.cc](#)

6.16 SearchInfo Struct Reference

Data Fields

- uint64_t **startTime**
- uint64_t **stopTime**
- bool **timeLimit**
- uint32_t **depth**
- uint32_t **depthLimit**
- uint32_t **nodes**
- uint32_t **movesLeft**
- bool **infinite**
- bool **quit**
- bool **stopped**
- float **fh**
- float **fhf**
- int32_t **protocol**
- bool **doPrint**

6.16.1 Detailed Description

Definition at line 11 of file searchinfo.h.

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

- /home/michael/Documents/Projects/C++/Chess-Engine/include/[searchinfo.h](#)

6.17 Stopwatch Class Reference

Public Member Functions

- void [start](#) () noexcept
Starts the internal timer.
- float [stop](#) () noexcept
Stops the internal timer and returns the time elapsed.

Static Public Member Functions

- static uint64_t [getTimeInMilli](#) () noexcept
Get the time since the epoch in milliseconds.

Static Public Attributes

- static constexpr int32_t **kMilliPerSecond** = 1000
- static constexpr int32_t **kSecondsPerMinute** = 60

Private Types

- typedef std::chrono::high_resolution_clock **Time**
- typedef std::chrono::milliseconds **ms**

Private Attributes

- Time::time_point **start_time**

6.17.1 Detailed Description

Definition at line 13 of file stopwatch.h.

6.17.2 Member Function Documentation

6.17.2.1 getTimeInMilli()

```
uint64_t Stopwatch::getTimeInMilli ( ) [static], [noexcept]
```

Get the time since the epoch in milliseconds.

Parameters

None	
------	--

Returns

Get the time since the epoch in milliseconds

Definition at line 28 of file stopwatch.cc.

6.17.2.2 start()

```
void Stopwatch::start ( ) [noexcept]
```

Starts the internal timer.

Parameters

None	
------	--

Returns

None

Definition at line 16 of file stopwatch.cc.

6.17.2.3 stop()

```
float Stopwatch::stop ( ) [noexcept]
```

Stops the internal timer and returns the time elapsed.

Parameters

None	
------	--

ReturnsThe time elapsed since [start\(\)](#) was called in milliseconds.

Definition at line 21 of file stopwatch.cc.

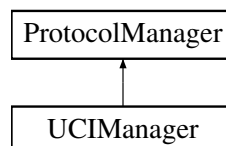
The documentation for this class was generated from the following files:

- /home/michael/Documents/Projects/C++/Chess-Engine/include/[stopwatch.h](#)
- /home/michael/Documents/Projects/C++/Chess-Engine/src/[stopwatch.cc](#)

6.18 UCIManager Class Reference

```
#include <uci.h>
```

Inheritance diagram for UCIManager:

**Public Member Functions**

- void **parseGoCmd** (const std::string &cmd)
- void **parsePosition** (const std::string &input)
- void **loop** () override
Starts the protocol loop.
- int32_t **getProtocol** () override
Returns the protocol identifier for the current protocol.

Additional Inherited Members

6.18.1 Detailed Description

See [ProtocolManager](#) documentation

Definition at line 15 of file uci.h.

6.18.2 Member Function Documentation

6.18.2.1 `getProtocol()`

```
int32_t UCIManager::getProtocol ( ) [override], [virtual]
```

Returns the protocol identifier for the current protocol.

Parameters

<i>None</i>	
-------------	--

Returns

The protocol identifier

Implements [ProtocolManager](#).

Definition at line 203 of file uci.cc.

6.18.2.2 `loop()`

```
void UCIManager::loop ( ) [override], [virtual]
```

Starts the protocol loop.

Parameters

<i>None</i>	
-------------	--

Returns

None

Implements [ProtocolManager](#).

Definition at line 134 of file uci.cc.

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

- [/home/michael/Documents/Projects/C++/Chess-Engine/include/uci.h](#)
- [/home/michael/Documents/Projects/C++/Chess-Engine/src/uci.cc](#)

6.19 UndoMove Class Reference

This class stores info needed to undo a move that was made.

```
#include <move.h>
```

Public Member Functions

- **UndoMove** (int32_t _move, const [Board](#) &pos)

Data Fields

- int32_t **move**
- int32_t **castle_perm**
- int32_t **en_pas**
- int32_t **fifty_move**
- uint64_t **pos_key**

6.19.1 Detailed Description

This class stores info needed to undo a move that was made.

Definition at line 17 of file move.h.

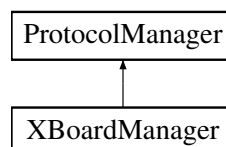
The documentation for this class was generated from the following files:

- [/home/michael/Documents/Projects/C++/Chess-Engine/include/move.h](#)
- [/home/michael/Documents/Projects/C++/Chess-Engine/src/move.cc](#)

6.20 XBoardManager Class Reference

```
#include <xboard.h>
```

Inheritance diagram for XBoardManager:



Public Member Functions

- void [loop](#) () override
Starts the protocol loop.
- int32_t [getProtocol](#) () override
Returns the protocol identifier for the current protocol.

Additional Inherited Members

6.20.1 Detailed Description

See [ProtocolManager](#) documentation

Definition at line 14 of file xboard.h.

6.20.2 Member Function Documentation

6.20.2.1 [getProtocol\(\)](#)

```
int32_t XBoardManager::getProtocol ( ) [override], [virtual]
```

Returns the protocol identifier for the current protocol.

Parameters

<i>None</i>	
-------------	--

Returns

The protocol identifier

Implements [ProtocolManager](#).

Definition at line 163 of file xboard.cc.

6.20.2.2 [loop\(\)](#)

```
void XBoardManager::loop ( ) [override], [virtual]
```

Starts the protocol loop.

Parameters

<i>None</i>	
-------------	--

Returns

None

Implements [ProtocolManager](#).

Definition at line 19 of file xboard.cc.

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

- /home/michael/Documents/Projects/C++/Chess-Engine/include/xboard.h
- /home/michael/Documents/Projects/C++/Chess-Engine/src/xboard.cc

Chapter 7

File Documentation

7.1 /home/michael/Documents/Projects/C++/Chess-Engine/include/bitboard.h File Reference

Contains declarations of functions that manipulate bitboards.

```
#include "defs.h"
```

Functions

- int [BB::popBit](#) (uint64_t &bb) noexcept
Sets the highest order 1 bit to 0.
- int [BB::countBits](#) (uint64_t bb) noexcept
Counts the number of 1's in the bitboard.
- void [BB::setBit](#) (uint64_t &bb, int index) noexcept
Sets the bit at index to 1.
- void [BB::clearBit](#) (uint64_t &bb, int index) noexcept
Sets the bit at index to 0.

Variables

- constexpr std::array< int32_t, kChessboardSize > [BB::BitTable](#)
- std::array< uint64_t, kChessboardSize > **BB::SetMask**
- std::array< uint64_t, kChessboardSize > **BB::ClearMask**

7.1.1 Detailed Description

Contains declarations of functions that manipulate bitboards.

Author

Michael Lee

Date

1/9/2019

7.1.2 Function Documentation

7.1.2.1 clearBit()

```
void BB::clearBit (
    uint64_t & bb,
    int index ) [noexcept]
```

Sets the bit at index to 0.

Parameters

<i>bb</i>	The desired bitboard.
<i>index</i>	The index to set to 0.

Returns

None.

Sets the bit at index to 0

Definition at line 45 of file bitboard.cc.

7.1.2.2 countBits()

```
int BB::countBits (
    uint64_t b ) [noexcept]
```

Counts the number of 1's in the bitboard.

Parameters

<i>bb</i>	The desired bitboard
-----------	----------------------

Returns

The number of 1's in the bitboard.

Returns the number of 1's in the bitboard

Definition at line 27 of file bitboard.cc.

7.1.2.3 popBit()

```
int BB::popBit (
    uint64_t & bb ) [noexcept]
```

Sets the highest order 1 bit to 0.

Parameters

<i>bb</i>	The desired bitboard
-----------	----------------------

Returns

The index of the bit that was flipped

Sets the highest order bit to 0 Returns the value after the operations

Definition at line 16 of file bitboard.cc.

7.1.2.4 setBit()

```
void BB::setBit (
    uint64_t & bb,
    int index ) [noexcept]
```

Sets the bit at index to 1.

Parameters

<i>bb</i>	The desired bitboard
<i>index</i>	The index to set to 1

Returns

None.

Sets the bit at index to 1

Definition at line 37 of file bitboard.cc.

7.1.3 Variable Documentation

7.1.3.1 BitTable

```
constexpr std::array<int32_t, kChessboardSize> BB::BitTable
```

Initial value:

```
{
    63, 30, 3, 32, 25, 41, 22, 33, 15, 50, 42, 13, 11, 53, 19, 34, 61, 29, 2,
    51, 21, 43, 45, 10, 18, 47, 1, 54, 9, 57, 0, 35, 62, 31, 40, 4, 49, 5, 52,
    26, 60, 6, 23, 44, 46, 27, 56, 16, 7, 39, 48, 24, 59, 14, 12, 55, 38, 28,
    58, 20, 37, 17, 36, 8
}
```

Magic numbers used for pop/set bit in bitboard

Definition at line 18 of file bitboard.h.

7.2 /home/michael/Documents/Projects/C++/Chess-Engine/include/board.h File Reference

Defines the internal [Board](#) representation used by the engine.

```
#include "defs.h"
#include "move.h"
#include "phtable.h"
#include <string>
#include <vector>
#include <sstream>
```

Data Structures

- class [Board](#)

7.2.1 Detailed Description

Defines the internal [Board](#) representation used by the engine.

Author

Michael Lee

Date

1/9/2019

7.3 /home/michael/Documents/Projects/C++/Chess-Engine/include/console.h File Reference

Contains declarations of functions for the console protocol.

```
#include "protocol.h"
```

Data Structures

- class [ConsoleManager](#)

7.3.1 Detailed Description

Contains declarations of functions for the console protocol.

Author

Michael Lee

Date

1/9/2019

7.4 /home/michael/Documents/Projects/C++/Chess-Engine/include/debug.h File Reference

Defines an assert function for debugging.

```
#include <stdlib.h>
```

Macros

- #define **ASSERT**(n)

7.4.1 Detailed Description

Defines an assert function for debugging.

Author

Michael Lee

Date

1/9/2019

7.5 /home/michael/Documents/Projects/C++/Chess-Engine/include/defs.h File Reference

Contains declarations of various constant arrays used throughout the engine.

```
#include "debug.h"
#include "move.h"
#include <string>
#include <array>
#include <cstdint>
```

Macros

- `#define STARTFEN "rnbqkbnr/pppppppp/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1"`

Enumerations

- `enum {
 EMPTY, wP, wN, wB,
 wR, wQ, wK, bP,
 bN, bB, bR, bQ,
 bK }`
- `enum {
 FILE_A, FILE_B, FILE_C, FILE_D,
 FILE_E, FILE_F, FILE_G, FILE_H,
 FILE_NONE }`
- `enum {
 RANK_1, RANK_2, RANK_3, RANK_4,
 RANK_5, RANK_6, RANK_7, RANK_8,
 RANK_NONE }`
- `enum { WHITE, BLACK, BOTH }`
- `enum {
 A1 = 21, B1, C1, D1,
 E1, F1, G1, H1,
 A2 = 31, B2, C2, D2,
 E2, F2, G2, H2,
 A3 = 41, B3, C3, D3,
 E3, F3, G3, H3,
 A4 = 51, B4, C4, D4,
 E4, F4, G4, H4,
 A5 = 61, B5, C5, D5,
 E5, F5, G5, H5,
 A6 = 71, B6, C6, D6,
 E6, F6, G6, H6,
 A7 = 81, B7, C7, D7,
 E7, F7, G7, H7,
 A8 = 91, B8, C8, D8,
 E8, F8, G8, H8,
 NO_SQ, OFFBOARD }`
- `enum { WKCA = 0b0001, WQCA = 0b0010, BKCA = 0b0100, BQCA = 0b1000 }`
- `enum { HFNONE, HFALPHA, HFBETA, HFEXACT }`

Functions

- `const Move NOMOVE (0)`

Variables

- constexpr uint32_t **kMaxSearchDepth** = 64
- constexpr uint32_t **kMoveLimit** = 2 << 10
- constexpr uint32_t **kBoardArraySize** = 120
- constexpr uint32_t **kChessboardSize** = 64
- constexpr uint32_t **kNumPlayers** = 2
- constexpr uint32_t **kNumFilesRanks** = 8
- constexpr uint32_t **kNumPceTypes** = 13
- const std::string **kAppName** = "ChessEngine"
- constexpr std::array< bool, kNumPceTypes > **PieceInfo::PieceBig** { false, false, true, true, true, true, true, false, true, true, true, true, true }
- constexpr std::array< bool, kNumPceTypes > **PieceInfo::PieceMaj** { false, false, false, false, true, true, true, false, false, false, true, true, true }
- constexpr std::array< bool, kNumPceTypes > **PieceInfo::PieceMin** { false, false, true, true, false, false, false, false, true, true, false, false, false }
- constexpr std::array< uint32_t, kNumPceTypes > **PieceInfo::PieceVal** { 0, 100, 325, 325, 550, 1000, 50000, 100, 325, 325, 550, 1000, 50000 }
- constexpr std::array< uint32_t, kNumPceTypes > **PieceInfo::PieceCol** { BOTH, WHITE, WHITE, WHITE, WHITE, WHITE, WHITE, BLACK, BLACK, BLACK, BLACK, BLACK, BLACK }
- constexpr std::array< bool, kNumPceTypes > **PieceInfo::PieceSlides** { false, false, false, true, true, true, false, false, false, true, true, true, false }
- constexpr std::array< bool, kNumPceTypes > **PieceInfo::PiecePawn** { false, true, false, false, false, false, false, false, true, false, false, false, false }
- constexpr std::array< bool, kNumPceTypes > **PieceInfo::PieceKing** { false, false, false, false, false, false, false, true, false, false, false, false, true }
- constexpr std::array< bool, kNumPceTypes > **PieceInfo::PieceRookQueen** { false, false, false, false, true, true, false, false, false, false, true, true, false }
- constexpr std::array< bool, kNumPceTypes > **PieceInfo::PieceBishopQueen** { false, false, false, true, false, true, false, false, false, true, false, true, false }
- constexpr std::array< bool, kNumPceTypes > **PieceInfo::PieceKnight** { false, false, true, false, false, false, false, false, false, true, false, false, false }
- constexpr std::array< int32_t, 2 > **Attack::wPCap** { -11, -9 }
- constexpr std::array< int32_t, 2 > **Attack::bPCap** { 11, 9 }
- constexpr std::array< int32_t, 2 > **Attack::PnMoves** { -10, 10 }
- constexpr std::array< int32_t, 8 > **Attack::KnMoves** { -8, -19, -21, -12, 8, 19, 21, 12 }
- constexpr std::array< int32_t, 4 > **Attack::RkMoves** { -1, -10, 1, 10 }
- constexpr std::array< int32_t, 4 > **Attack::BiMoves** { -9, -11, 11, 9 }
- constexpr std::array< int32_t, 8 > **Attack::KiMoves** { -1, -10, 1, 10, -9, -11, 11, 9 }

7.5.1 Detailed Description

Contains declarations of various constant arrays used throughout the engine.

Author

Michael Lee

Date

1/9/2019

7.6 /home/michael/Documents/Projects/C++/Chess-Engine/include/engine.h File Reference

Defines the central engine data structure.

```
#include <memory>
#include "protocol.h"
#include "polyglot.h"
```

Data Structures

- struct [EngineConfig](#)
Stores the flags for the engine.
- class [Engine](#)

7.6.1 Detailed Description

Defines the central engine data structure.

Author

Michael Lee

Date

1/9/2019

7.7 /home/michael/Documents/Projects/C++/Chess-Engine/include/eval.h File Reference

Contains declarations of functions that determine the strength of a given position.

```
#include "board.h"
#include "defs.h"
```

Data Structures

- class [Evaluator](#)

Namespaces

- [Value](#)
Provides several constant values used for evaluating the board state.

Variables

- constexpr int32_t [Value::kInfinity](#) = 30000
Highest possible score.
- constexpr int32_t [Value::kMateScore](#) = 29000
Score for a Checkmate.
- constexpr int32_t [Value::kIsolatedPawn](#) = -10
Score bonus for isolated pawns (no same color pawn on adjacent files)
- constexpr int32_t [Value::kOpenRookFile](#) = 10
Score bonus for a Rook on an open file (no pawns)
- constexpr int32_t [Value::kSemiOpenRookFile](#) = 5
Score bonus for a Rook on a semi-open file (no same color pawn)
- constexpr int32_t [Value::kOpenQueenFile](#) = 5
Score bonus for a Queen on an open file (no pawns)
- constexpr int32_t [Value::kSemiOpenQueenFile](#) = 3
Score bonus for a Queen on a semi-open file (no same color pawn)
- constexpr int32_t [Value::kEndGameThreshold](#) = PieceInfo::PieceVal[wR] + 2 * PieceInfo::PieceVal[wB] + 2 * PieceInfo::PieceVal[wP]
Material threshold to determine when the endgame starts.
- constexpr int32_t [Value::kBishopPair](#) = 30
Score bonus for the Bishop pair.
- constexpr std::array< int32_t, kNumFilesRanks > [Value::passedPawnScore](#) {0, 5, 10, 20, 35, 60, 100, 200}
Score bonus for Passed pawns based on distance.
- constexpr std::array< int32_t, kChessboardSize > [Value::PawnTable](#)
Scores the position of the Pawns.
- constexpr std::array< int32_t, kChessboardSize > [Value::KnightTable](#)
Scores the position of the Knights.
- constexpr std::array< int32_t, kChessboardSize > [Value::BishopTable](#)
Scores the position of the Bishops.
- constexpr std::array< int32_t, kChessboardSize > [Value::RookTable](#)
Scores the position of the Rooks.
- constexpr std::array< int32_t, kChessboardSize > [Value::KingEndGame](#)
Scores the king in the late game. Prioritizes the center.
- constexpr std::array< int32_t, kChessboardSize > [Value::KingOpening](#)
Scores the king for the earlygame. Prioritizes staying safe.
- std::array< uint64_t, kNumFilesRanks > [EvalBB::FileMask](#)
- std::array< uint64_t, kNumFilesRanks > [EvalBB::RankMask](#)
- std::array< uint64_t, kChessboardSize > [EvalBB::whitePassedMask](#)
- std::array< uint64_t, kChessboardSize > [EvalBB::blackPassedMask](#)
- std::array< uint64_t, kChessboardSize > [EvalBB::isolatedMask](#)

7.7.1 Detailed Description

Contains declarations of functions that determine the strength of a given position.

Author

Michael Lee

Date

1/9/2019

7.8 /home/michael/Documents/Projects/C++/Chess-Engine/include/hash.h File Reference

Contains declarations of functions that manipulate the board position key.

```
#include "board.h"
#include "defs.h"
#include <array>
```

Namespaces

- [Hash](#)

Functions

- `uint64_t Hash::generatePosKey (const Board &pos)`
Gets the position hash key for the current position.
- `void Hash::hashPce (uint32_t pce, uint32_t sq, Board &pos) noexcept`
Hashes in/out a piece on a given square.
- `void Hash::hashCa (Board &pos) noexcept`
Hashes in/out the castle permissions.
- `void Hash::hashSide (Board &pos) noexcept`
Hashes in/out the side to move.
- `void Hash::hashEP (Board &pos) noexcept`
Hashes in/out the enPassant square.

Variables

- `std::array< std::array< uint64_t, kBoardArraySize >, kNumPceTypes > Hash::PieceKeys`
- `uint64_t Hash::SideKey`
- `std::array< uint64_t, 16 > Hash::CastleKeys`

7.8.1 Detailed Description

Contains declarations of functions that manipulate the board position key.

Author

Michael Lee

Date

1/9/2019

7.9 /home/michael/Documents/Projects/C++/Chess-Engine/include/init.h File Reference

Contains declarations of functions that fill the non static constant arrays used throughout the engine.

Namespaces

- [Init](#)

Provides methods to initialize the non constant arrays used throughout the engine.

Functions

- void [Init::initAll](#) () noexcept
Calls all of the other 'init' methods.
- void [Init::initSq120ToSq64](#) () noexcept
Fills in the arrays that convert between array-120 to array-64 representations.
- void [Init::initBitMasks](#) () noexcept
Fills in the arrays used for setting/clearing bits in bitboards.
- void [Init::initHashKeys](#) () noexcept
Fills in the hashkeys arrays that will be used to for getting the board's hashkey.
- void [Init::initFileRankBrd](#) () noexcept
Fills in the arrays that return the file/rank # for a given square.
- void [Init::initEvalMasks](#) () noexcept
Fills in the arrays used for evaluating pawn structure during evaluation.
- void [Init::initMvvLva](#) () noexcept
Fills in the arrays to determine most valuable victim least valuable attacker priority.

7.9.1 Detailed Description

Contains declarations of functions that fill the non static constant arrays used throughout the engine.

Author

Michael Lee

Date

1/9/2019

7.10 /home/michael/Documents/Projects/C++/Chess-Engine/include/io.h File Reference

Contains declarations of functions that print the various [Engine](#) data structures.

```
#include "board.h"
#include "move.h"
#include "movelist.h"
#include "searchinfo.h"
#include <string>
```

Namespaces

- [IO](#)

Various functions to print out engine data structures.

Functions

- void `IO::printBoard` (const `Board` &pos) noexcept
Prints out the board to stdio.
- void `IO::printBitBoard` (const uint64_t bb) noexcept
Prints out the bitboard to stdio.
- void `IO::printMoveList` (const `MoveList` &list) noexcept
Prints out the `MoveList` to stdio.
- void `IO::printSearchDetails` (const `SearchInfo` &info, int32_t curDepth, int32_t bestScore, `PvTable` &pv, int32_t pvMoves) noexcept
Prints out the search details to stdio.
- void `IO::printBestMove` (`Board` &pos, const `SearchInfo` &info, const `Move` &bestMove) noexcept
Prints out the best move stdio for the protocol manager.
- `Move` `IO::parseMove` (std::string input, const `Board` &pos) noexcept
Reads in a move string and converts it to the internal representation.

Variables

- const std::string `IO::PceChar` = ".PNBRQKpnbrqk"
- const std::string `IO::SideChar` = "wb-"
- const std::string `IO::RankChar` = "12345678"
- const std::string `IO::FileChar` = "abcdefgh"
- const std::unordered_map< uint32_t, std::string > `IO::epstr`

7.10.1 Detailed Description

Contains declarations of functions that print the various `Engine` data structures.

Author

Michael Lee

Date

1/9/2019

7.11 /home/michael/Documents/Projects/C++/Chess-Engine/include/move.h File Reference

Defines the custom internal move representation.

```
#include <string>
```

Data Structures

- class `UndoMove`
This class stores info needed to undo a move that was made.
- class `Move`

Namespaces

- [MoveFlags](#)

This namespace stores flags used to extract parameters from a move.

Variables

- constexpr int32_t **MoveFlags::SQ** = 0x7F
- constexpr int32_t **MoveFlags::EP** = 0x40000
- constexpr int32_t **MoveFlags::PS** = 0x80000
- constexpr int32_t **MoveFlags::CA** = 0x1000000
- constexpr int32_t **MoveFlags::CAP** = 0x7C000
- constexpr int32_t **MoveFlags::PROM** = 0xF00000

7.11.1 Detailed Description

Defines the custom internal move representation.

Author

Michael Lee

Date

1/9/2019

7.12 /home/michael/Documents/Projects/C++/Chess-Engine/include/movelist.h File Reference

Custom data structure to store possible moves.

```
#include "defs.h"
#include "board.h"
#include "move.h"
```

Data Structures

- class [MoveList](#)

Namespaces

- [MvvLva](#)

Variables

- `std::array< std::array< int32_t, kNumPceTypes >, kNumPceTypes > MvvLva::MvvLvaScore`
- `constexpr std::array< int32_t, kNumPceTypes > MvvLva::victimScore {0, 100, 200, 300, 400, 500, 600, 100, 200, 300, 400, 500, 600}`

7.12.1 Detailed Description

Custom data structure to store possible moves.

Author

Michael Lee

Date

1/9/2019

7.13 /home/michael/Documents/Projects/C++/Chess-Engine/include/movemaker.h File Reference

Contains declarations of functions that manipulate the position of pieces on the internal board.

```
#include "defs.h"
#include "board.h"
```

Data Structures

- class [MM](#)

Variables

- `constexpr std::array< int32_t, 120 > CastlePerm`

7.13.1 Detailed Description

Contains declarations of functions that manipulate the position of pieces on the internal board.

Author

Michael Lee

Date

1/9/2019

7.13.2 Variable Documentation

7.13.2.1 CastlePerm

```
constexpr std::array<int32_t,120> CastlePerm
```

Initial value:

```
{
    15, 15, 15, 15, 15, 15, 15, 15, 15, 15,
    15, 15, 15, 15, 15, 15, 15, 15, 15, 15,
    15, 13, 15, 15, 15, 12, 15, 15, 14, 15,
    15, 15, 15, 15, 15, 15, 15, 15, 15, 15,
    15, 15, 15, 15, 15, 15, 15, 15, 15, 15,
    15, 15, 15, 15, 15, 15, 15, 15, 15, 15,
    15, 15, 15, 15, 15, 15, 15, 15, 15, 15,
    15, 15, 15, 15, 15, 15, 15, 15, 15, 15,
    15, 15, 15, 15, 15, 15, 15, 15, 15, 15,
    15, 15, 15, 15, 15, 15, 15, 15, 15, 15,
    15, 7, 15, 15, 15, 3, 15, 15, 11, 15,
    15, 15, 15, 15, 15, 15, 15, 15, 15, 15,
    15, 15, 15, 15, 15, 15, 15, 15, 15, 15
}
```

Indicates the change in castle permissions if a given square moves. Used with bitwise &

Definition at line 18 of file movemaker.h.

7.14 /home/michael/Documents/Projects/C++/Chess-Engine/include/polyglot.h File Reference

Contains declarations for the Polyglot Book class.

```
#include "defs.h"
#include <vector>
```

Data Structures

- struct [PolyglotEntry](#)
- class [PolyBook](#)

7.14.1 Detailed Description

Contains declarations for the Polyglot Book class.

Author

Michael Lee

Date

1/9/2019

7.15 /home/michael/Documents/Projects/C++/Chess-Engine/include/polyglotkeys.h File Reference

Contains all 781 polyglot hashkeys used in the opening.

```
#include "defs.h"
```

Namespaces

- [PolyKeys](#)

Macros

- `#define U64(u) (u##ULL)`

Variables

- `constexpr std::array< uint64_t, 781 > PolyKeys::Random64`

7.15.1 Detailed Description

Contains all 781 polyglot hashkeys used in the opening.

Author

Michael Lee

Date

1/9/2019

7.16 /home/michael/Documents/Projects/C++/Chess-Engine/include/protocol.h File Reference

Contains declaration of base class for default protocol functionality.

```
#include "board.h"
#include "searchinfo.h"
#include "search.h"
#include "movemaker.h"
```

Data Structures

- class [ProtocolManager](#)

7.16.1 Detailed Description

Contains declaration of base class for default protocol functionality.

Author

Michael Lee

Date

1/9/2019

7.17 /home/michael/Documents/Projects/C++/Chess-Engine/include/pvtable.h File Reference

Contains declarations for the Principal Variation/transposition table class used for caching.

```
#include "move.h"
#include <unordered_map>
#include <vector>
```

Data Structures

- class [PvEntry](#)
- class [PvTable](#)

7.17.1 Detailed Description

Contains declarations for the Principal Variation/transposition table class used for caching.

Author

Michael Lee

Date

1/9/2019

7.18 /home/michael/Documents/Projects/C++/Chess-Engine/include/search.h File Reference

Contains declarations of functions to search through the board state.

```
#include "defs.h"
#include "board.h"
#include "searchinfo.h"
#include "eval.h"
#include "polyglot.h"
#include "pvtable.h"
```

Data Structures

- class [SearchAgent](#)

7.18.1 Detailed Description

Contains declarations of functions to search through the board state.

Author

Michael Lee

Date

1/9/2019

7.19 [/home/michael/Documents/Projects/C++/Chess-Engine/include/searchinfo.h](#) File Reference

Contains information about the search constraints provided by the engine.

Data Structures

- struct [SearchInfo](#)

7.19.1 Detailed Description

Contains information about the search constraints provided by the engine.

Author

Michael Lee

Date

1/9/2019

7.20 [/home/michael/Documents/Projects/C++/Chess-Engine/include/stopwatch.h](#) File Reference

Contains declarations of functions for basic benchmarking and timing.

```
#include <chrono>
#include <unistd.h>
```

Data Structures

- class [Stopwatch](#)

7.20.1 Detailed Description

Contains declarations of functions for basic benchmarking and timing.

Author

Michael Lee

Date

1/9/2019

7.21 /home/michael/Documents/Projects/C++/Chess-Engine/include/tester.h File Reference

Contains declarations of functions used for Perft testing for testing accuracy of move generation and move making.

```
#include "board.h"  
#include <string>
```

Data Structures

- class [PerftTester](#)

7.21.1 Detailed Description

Contains declarations of functions used for Perft testing for testing accuracy of move generation and move making.

Author

Michael Lee

Date

1/9/2019

7.22 /home/michael/Documents/Projects/C++/Chess-Engine/include/uci.h File Reference

Contains declarations of functions for the UCI protocol.

```
#include "protocol.h"  
#include <string>
```

Data Structures

- class [UCIManager](#)

7.22.1 Detailed Description

Contains declarations of functions for the UCI protocol.

Author

Michael Lee

Date

1/9/2019

7.23 /home/michael/Documents/Projects/C++/Chess-Engine/include/utils.h File Reference

Contains declarations of functions that perform various miscellaneous actions in the engine.

```
#include "defs.h"
#include "board.h"
#include "searchinfo.h"
#include "move.h"
#include <string>
```

Functions

- bool [sqOnBoard](#) (uint32_t sq) noexcept
Checks if a square number in array-120 form is on the array-64 board representation.
- int [fileRankToSq](#) (int32_t file, int32_t rank) noexcept
Gets the array-120 square number of a given file and rank number.
- uint64_t [randU64](#) () noexcept
Generates a uniformly-distributed random 64-bit number.
- bool [isPiece](#) (int32_t piece) noexcept
Generates a uniformly-distributed random 64-bit number.
- void [showAttackedSqs](#) (const int side, [Board](#) &pos) noexcept
Prints the 8x8 board with numbers indicating how many times each square is attacked by the provided side.
- std::string [sqToString](#) (const int sq) noexcept
Returns the string representation of the square.
- void [stringToLower](#) (std::string &str) noexcept
Changes the given string to lowercase.
- int [InputWaiting](#) ()
Checks if the GUI interface has sent an interrupt.
- void [ReadInput](#) ([SearchInfo](#) &info)
Reads in the GUI signal from stdin.

Variables

- `std::array< int32_t, kBoardArraySize > BoardUtils::Sq120ToSq64`
- `std::array< int32_t, kChessboardSize > BoardUtils::Sq64ToSq120`
- `std::array< int32_t, kBoardArraySize > BoardUtils::FileBrd`
- `std::array< int32_t, kBoardArraySize > BoardUtils::RankBrd`
- `constexpr std::array< int32_t, kChessboardSize > BoardUtils::WhiteToBlack`

7.23.1 Detailed Description

Contains declarations of functions that perform various miscellaneous actions in the engine.

Author

Michael Lee

Date

1/9/2019

7.23.2 Function Documentation

7.23.2.1 fileRankToSq()

```
int fileRankToSq (
    int32_t file,
    int32_t rank ) [noexcept]
```

Gets the array-120 square number of a given file and rank number.

Parameters

<i>file</i>	FILE_A <= file <= FILE_H
<i>rank</i>	RANK_1 <= rank <= RANK_8

Returns

The array-120 square number of the given file and rank number

7.23.2.2 InputWaiting()

```
int InputWaiting ( )
```

Checks if the GUI interface has sent an interrupt.

Parameters

<i>None</i>	
-------------	--

Returns

true if there was a GUI interrupt, false otherwise.

Definition at line 91 of file utils.cc.

7.23.2.3 isPiece()

```
bool isPiece (
    int32_t piece ) [noexcept]
```

Generates a uniformly-distributed random 64-bit number.

Parameters

<i>piece</i>	A piece, OFFBOARD, EMPTY, or NO_SQ
--------------	------------------------------------

Returns

true if wP <= piece <= bK, false otherwise

7.23.2.4 randU64()

```
uint64_t randU64 ( ) [noexcept]
```

Generates a uniformly-distributed random 64-bit number.

Parameters

<i>None</i>	
-------------	--

Returns

A random 64 bit number

Definition at line 54 of file utils.cc.

7.23.2.5 ReadInput()

```
void ReadInput (
    SearchInfo & info )
```

Reads in the GUI signal from stdin.

Parameters

<i>info</i>	The engine's searchInfo instance
-------------	----------------------------------

Returns

None

Definition at line 125 of file utils.cc.

7.23.2.6 showAttackedSqs()

```
void showAttackedSqs (
    const int side,
    Board & pos ) [noexcept]
```

Prints the 8x8 board with numbers indicating how many times each square is attacked by the provided side.

Parameters

<i>side</i>	The attacking side
<i>pos</i>	The board state

Returns

None

Definition at line 65 of file utils.cc.

7.23.2.7 sqOnBoard()

```
bool sqOnBoard (
    uint32_t sq ) [noexcept]
```

Checks if a square number in array-120 form is on the array-64 board representation.

Parameters

<i>sq</i>	The square number in array-120 form.
-----------	--------------------------------------

Returns

true if the square is on the array-64 board, false otherwise.

Definition at line 29 of file utils.cc.

7.23.2.8 sqToString()

```
std::string sqToString (
    const int sq ) [noexcept]
```

Returns the string representation of the square.

Parameters

<i>sq</i>	A square in array-120 form.
-----------	-----------------------------

Returns

The string representation of the square (e.g "A1").

Definition at line 45 of file utils.cc.

7.23.2.9 stringToLower()

```
void stringToLower (
    std::string & str ) [noexcept]
```

Changes the given string to lowercase.

Parameters

<i>str</i>	the string to change to lower.
------------	--------------------------------

Returns

None

Definition at line 84 of file utils.cc.

7.23.3 Variable Documentation

7.23.3.1 WhiteToBlack

```
constexpr std::array<int32_t, kChessboardSize> BoardUtils::WhiteToBlack
```

Initial value:

```
{
    56 , 57 , 58 , 59 , 60 , 61 , 62 , 63 ,
    48 , 49 , 50 , 51 , 52 , 53 , 54 , 55 ,
    40 , 41 , 42 , 43 , 44 , 45 , 46 , 47 ,
    32 , 33 , 34 , 35 , 36 , 37 , 38 , 39 ,
    24 , 25 , 26 , 27 , 28 , 29 , 30 , 31 ,
    16 , 17 , 18 , 19 , 20 , 21 , 22 , 23 ,
    8  , 9  , 10 , 11 , 12 , 13 , 14 , 15 ,
    0  , 1  , 2  , 3  , 4  , 5  , 6  , 7  ,
}
```

Gives the corresponding square to the Black side from the White side

Definition at line 27 of file utils.h.

7.24 /home/michael/Documents/Projects/C++/Chess-Engine/include/xboard.h File Reference

Contains declarations for the XBoard protocol.

```
#include "protocol.h"
```

Data Structures

- class [XBoardManager](#)

7.24.1 Detailed Description

Contains declarations for the XBoard protocol.

Author

Michael Lee

Date

1/9/2019

7.25 /home/michael/Documents/Projects/C++/Chess-Engine/src/bitboard.cc File Reference

Contains definitions of functions declared in [bitboard.h](#).

```
#include "bitboard.h"  
#include "defs.h"  
#include "utils.h"  
#include <cstdio>
```

7.25.1 Detailed Description

Contains definitions of functions declared in [bitboard.h](#).

Author

Michael Lee

Date

1/9/2019

7.26 /home/michael/Documents/Projects/C++/Chess-Engine/src/board.cc File Reference

Contains definitions of functions declared in [board.h](#).

```
#include "board.h"  
#include "defs.h"  
#include "debug.h"  
#include "hash.h"  
#include "utils.h"  
#include "bitboard.h"  
#include "movelist.h"  
#include <string>  
#include <sstream>  
#include <cstdio>
```

7.26.1 Detailed Description

Contains definitions of functions declared in [board.h](#).

Author

Michael Lee

Date

1/9/2019

7.27 /home/michael/Documents/Projects/C++/Chess-Engine/src/console.cc File Reference

Contains definitions of functions declared in [console.h](#).

```
#include "console.h"
#include "stopwatch.h"
#include "defs.h"
#include "movemaker.h"
#include "io.h"
#include "searchinfo.h"
#include "polyglot.h"
#include "engine.h"
#include <iostream>
#include <sstream>
```

7.27.1 Detailed Description

Contains definitions of functions declared in [console.h](#).

Author

Michael Lee

Date

1/9/2019

7.28 /home/michael/Documents/Projects/C++/Chess-Engine/src/engine.cc File Reference

Contains definitions of functions declared in [engine.h](#).

```
#include "engine.h"
#include "defs.h"
#include "init.h"
#include "uci.h"
#include "xboard.h"
#include "console.h"
#include "protocol.h"
#include <iostream>
```

7.28.1 Detailed Description

Contains definitions of functions declared in [engine.h](#).

Author

Michael Lee

Date

1/9/2019

7.29 /home/michael/Documents/Projects/C++/Chess-Engine/src/eval.cc File Reference

Contains definitions of functions declared in [eval.h](#).

```
#include "eval.h"
#include "defs.h"
#include "utils.h"
#include <iostream>
#include <cmath>
```

7.29.1 Detailed Description

Contains definitions of functions declared in [eval.h](#).

Author

Michael Lee

Date

1/9/2019

7.30 /home/michael/Documents/Projects/C++/Chess-Engine/src/hash.cc File Reference

Contains definitions of functions declared in [hash.h](#).

```
#include "defs.h"
#include "hash.h"
#include "utils.h"
#include "board.h"
```

7.30.1 Detailed Description

Contains definitions of functions declared in [hash.h](#).

Author

Michael Lee

Date

1/9/2019

7.31 /home/michael/Documents/Projects/C++/Chess-Engine/src/init.cc File Reference

Contains definitions of functions declared in [init.h](#).

```
#include "defs.h"
#include "utils.h"
#include "init.h"
#include "hash.h"
#include "bitboard.h"
#include "io.h"
#include "stopwatch.h"
#include "movelist.h"
#include <vector>
#include <iostream>
```

Namespaces

- [Hash](#)
- [MvvLva](#)

7.31.1 Detailed Description

Contains definitions of functions declared in [init.h](#).

Author

Michael Lee

Date

1/9/2019

7.32 /home/michael/Documents/Projects/C++/Chess-Engine/src/io.cc File Reference

Contains definitions of functions declared in [io.h](#).

```
#include "io.h"
#include "phtable.h"
#include "defs.h"
#include "stopwatch.h"
#include "utils.h"
#include "protocol.h"
#include "movemaker.h"
#include <bitset>
#include <iostream>
#include <sstream>
```

7.32.1 Detailed Description

Contains definitions of functions declared in [io.h](#).

Author

Michael Lee

Date

1/9/2019

7.33 /home/michael/Documents/Projects/C++/Chess-Engine/src/main.cc File Reference

Entry point of the engine.

```
#include "engine.h"
#include "tester.h"
#include "stopwatch.h"
#include "polyglot.h"
#include "bitboard.h"
#include <iostream>
```

Functions

- void **runTest** ()
- int **main** ()

7.33.1 Detailed Description

Entry point of the engine.

Author

Michael Lee

Date

1/9/2019

7.34 /home/michael/Documents/Projects/C++/Chess-Engine/src/move.cc File Reference

Contains definitions of functions declared in [move.h](#).

```
#include "move.h"
#include "defs.h"
#include "board.h"
#include "utils.h"
#include <iostream>
#include <sstream>
#include <utility>
```


7.34.1 Detailed Description

Contains definitions of functions declared in [move.h](#).

Author

Michael Lee

Date

1/9/2019

7.35 /home/michael/Documents/Projects/C++/Chess-Engine/src/movelist.cc File Reference

Contains definitions of functions declared in [movelist.h](#).

```
#include "movelist.h"
#include "utils.h"
#include <bitset>
#include <iostream>
#include <utility>
#include <algorithm>
```

7.35.1 Detailed Description

Contains definitions of functions declared in [movelist.h](#).

Author

Michael Lee

Date

1/9/2019

7.36 /home/michael/Documents/Projects/C++/Chess-Engine/src/movemaker.cc File Reference

Contains definitions of functions declared in [movemaker.h](#).

```
#include "movemaker.h"
#include "defs.h"
#include "bitboard.h"
#include "debug.h"
#include "utils.h"
#include "move.h"
#include "hash.h"
#include <cstdio>
#include <algorithm>
```

7.36.1 Detailed Description

Contains definitions of functions declared in [movemaker.h](#).

Author

Michael Lee

Date

1/9/2019

7.37 /home/michael/Documents/Projects/C++/Chess-Engine/src/polyglot.cc File Reference

Contains definitions of functions declared in [polyglot.h](#).

```
#include "defs.h"
#include "polyglotkeys.h"
#include "polyglot.h"
#include "board.h"
#include "io.h"
#include "move.h"
#include "utils.h"
#include <sstream>
#include <fstream>
#include <iostream>
```

7.37.1 Detailed Description

Contains definitions of functions declared in [polyglot.h](#).

Author

Michael Lee

Date

1/9/2019

7.38 /home/michael/Documents/Projects/C++/Chess-Engine/src/pvtable.cc File Reference

Contains definitions of functions declared in [pvtable.h](#).

```
#include "pvtable.h"
#include "defs.h"
#include "board.h"
#include "movemaker.h"
#include "utils.h"
#include "eval.h"
#include <utility>
```

7.38.1 Detailed Description

Contains definitions of functions declared in [pvtable.h](#).

Author

Michael Lee

Date

1/9/2019

7.39 /home/michael/Documents/Projects/C++/Chess-Engine/src/search.cc File Reference

Contains definitions of functions declared in [search.h](#).

```
#include "search.h"
#include "stopwatch.h"
#include "movelist.h"
#include "utils.h"
#include "searchinfo.h"
#include "engine.h"
#include "movemaker.h"
#include "io.h"
#include <sstream>
#include <iostream>
#include <cstdio>
```

7.39.1 Detailed Description

Contains definitions of functions declared in [search.h](#).

Author

Michael Lee

Date

1/9/2019

7.40 /home/michael/Documents/Projects/C++/Chess-Engine/src/stopwatch.cc File Reference

Contains definitions of functions declared in [stopwatch.h](#).

```
#include "stopwatch.h"
```

7.40.1 Detailed Description

Contains definitions of functions declared in [stopwatch.h](#).

Author

Michael Lee

Date

1/9/2019

7.41 /home/michael/Documents/Projects/C++/Chess-Engine/src/tester.cc File Reference

Contains definitions of functions declared in [tester.h](#).

```
#include "defs.h"
#include "tester.h"
#include "io.h"
#include "movelist.h"
#include "movemaker.h"
#include "move.h"
#include <fstream>
#include <iostream>
#include <sstream>
```

7.41.1 Detailed Description

Contains definitions of functions declared in [tester.h](#).

Author

Michael Lee

Date

1/9/2019

7.42 /home/michael/Documents/Projects/C++/Chess-Engine/src/uci.cc File Reference

Contains definitions of functions declared in [uci.h](#).

```
#include "uci.h"
#include "defs.h"
#include "io.h"
#include "move.h"
#include "utils.h"
#include "stopwatch.h"
#include "engine.h"
#include <iostream>
#include <sstream>
```

7.42.1 Detailed Description

Contains definitions of functions declared in [uci.h](#).

Author

Michael Lee

Date

1/9/2019

7.43 /home/michael/Documents/Projects/C++/Chess-Engine/src/utils.cc File Reference

Contains definitions of functions declared in [utils.h](#).

```
#include "utils.h"
#include "defs.h"
#include "io.h"
#include "movelist.h"
#include "movemaker.h"
#include <unistd.h>
#include <random>
#include <sstream>
#include <iostream>
#include <string>
#include <algorithm>
#include <cstring>
#include "sys/time.h"
#include "sys/select.h"
#include "string.h"
```

Functions

- bool [sqOnBoard](#) (uint32_t sq) noexcept
Checks if a square number in array-120 form is on the array-64 board representation.
- int [fileRankToSq](#) (int file, int rank) noexcept
- bool [isPiece](#) (int piece) noexcept
- std::string [sqToString](#) (const int sq) noexcept
Returns the string representation of the square.
- uint64_t [randU64](#) () noexcept
Generates a uniformly-distributed random 64-bit number.
- void [showAttackedSqs](#) (const int side, [Board](#) &pos) noexcept
Prints the 8x8 board with numbers indicating how many times each square is attacked by the provided side.
- void [stringToLower](#) (std::string &str) noexcept
Changes the given string to lowercase.
- int [InputWaiting](#) ()
Checks if the GUI interface has sent an interrupt.
- void [ReadInput](#) ([SearchInfo](#) &info)
Reads in the GUI signal from stdin.

7.43.1 Detailed Description

Contains definitions of functions declared in [utils.h](#).

Author

Michael Lee

Date

1/9/2019

7.43.2 Function Documentation

7.43.2.1 InputWaiting()

```
int InputWaiting ( )
```

Checks if the GUI interface has sent an interrupt.

Parameters

None	
------	--

Returns

true if there was a GUI interrupt, false otherwise.

Definition at line 91 of file utils.cc.

7.43.2.2 randU64()

```
uint64_t randU64 ( ) [noexcept]
```

Generates a uniformly-distributed random 64-bit number.

Parameters

None	
------	--

Returns

A random 64 bit number

Definition at line 54 of file utls.cc.

7.43.2.3 ReadInput()

```
void ReadInput (
    SearchInfo & info )
```

Reads in the GUI signal from stdin.

Parameters

<i>info</i>	The engine's searchInfo instance
-------------	----------------------------------

Returns

None

Definition at line 125 of file utls.cc.

7.43.2.4 showAttackedSqs()

```
void showAttackedSqs (
    const int side,
    Board & pos ) [noexcept]
```

Prints the 8x8 board with numbers indicating how many times each square is attacked by the provided side.

Parameters

<i>side</i>	The attacking side
<i>pos</i>	The board state

Returns

None

Definition at line 65 of file utls.cc.

7.43.2.5 sqOnBoard()

```
bool sqOnBoard (
    uint32_t sq ) [noexcept]
```

Checks if a square number in array-120 form is on the array-64 board representation.

Parameters

<i>sq</i>	The square number in array-120 form.
-----------	--------------------------------------

Returns

true if the square is on the array-64 board, false otherwise.

Definition at line 29 of file utils.cc.

7.43.2.6 sqToString()

```
std::string sqToString (  
    const int sq ) [noexcept]
```

Returns the string representation of the square.

Parameters

<i>sq</i>	A square in array-120 form.
-----------	-----------------------------

Returns

The string representation of the square (e.g "A1").

Definition at line 45 of file utils.cc.

7.43.2.7 stringToLower()

```
void stringToLower (  
    std::string & str ) [noexcept]
```

Changes the given string to lowercase.

Parameters

<i>str</i>	the string to change to lower.
------------	--------------------------------

Returns

None

Definition at line 84 of file utils.cc.

7.44 /home/michael/Documents/Projects/C++/Chess-Engine/src/xboard.cc File Reference

Contains definitions of functions declared in [xboard.h](#).

```
#include "xboard.h"
#include "io.h"
#include "stopwatch.h"
#include <iostream>
#include <sstream>
#include <cstdio>
#include "searchinfo.h"
```

7.44.1 Detailed Description

Contains definitions of functions declared in [xboard.h](#).

Author

Michael Lee

Date

1/9/2019

Index

/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/bitboard.h, 85
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/board.h, 88
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/console.h, 89
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/debug.h, 89
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/defs.h, 90
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/engine.h, 92
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/eval.h, 92
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/hash.h, 94
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/init.h, 94
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/io.h, 95
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/move.h, 96
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/movelist.h, 97
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/movemaker.h, 98
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/polyglot.h, 99
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/polyglotkeys.h, 100
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/protocol.h, 100
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/pvtable.h, 101
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/search.h, 101
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/searchinfo.h, 102
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/stopwatch.h, 102
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/tester.h, 103
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/uci.h, 103
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/utils.h, 104
/home/michael/Documents/Projects/C++/Chess-↵
Engine/include/xboard.h, 109
/home/michael/Documents/Projects/C++/Chess-↵

Engine/src/bitboard.cc, 110
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/board.cc, 110
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/console.cc, 111
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/engine.cc, 111
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/eval.cc, 112
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/hash.cc, 112
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/init.cc, 113
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/io.cc, 113
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/main.cc, 114
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/move.cc, 114
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/movelist.cc, 115
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/movemaker.cc, 115
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/polyglot.cc, 116
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/pvtable.cc, 116
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/search.cc, 117
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/stopwatch.cc, 117
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/tester.cc, 118
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/uci.cc, 118
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/utils.cc, 119
/home/michael/Documents/Projects/C++/Chess-↵
Engine/src/xboard.cc, 123

addCaptureMove
MoveList, 49
addEnPasMove
MoveList, 49
addPawnCaptureMove
MoveList, 50
addPawnMove
MoveList, 50
addPiece
MM, 39

- addQuietMove
 - MoveList, 51
- alphaBeta
 - SearchAgent, 71
- BishopTable
 - Value, 21
- BitTable
 - bitboard.h, 87
- bitboard.h
 - BitTable, 87
 - clearBit, 86
 - countBits, 86
 - popBit, 86
 - setBit, 87
- Board, 25
 - flipBoard, 26
 - getAllCaptureMoves, 26
 - getAllMoves, 27
 - getEnPassant, 27
 - inCheck, 27
 - parseFEN, 28
 - resetBoard, 28
 - setUpCastlePerm, 29
 - setUpMoveCounters, 29
 - setUpPieces, 30
 - sqAttacked, 30
 - updatePieceLists, 30
- book
 - PolyBook, 63
- captured
 - Move, 43
- castle
 - Move, 44
- CastlePerm
 - movemaker.h, 99
- checkStop
 - SearchAgent, 72
- clear
 - PvTable, 67
- clearBit
 - bitboard.h, 86
- clearForSearch
 - SearchAgent, 72
- clearPiece
 - MM, 40
- ConsoleManager, 31
 - getProtocol, 32
 - loop, 32
- convertPolyMove
 - PolyBook, 60
- countBits
 - bitboard.h, 86
- drawnMaterial
 - Evaluator, 36
 - SearchAgent, 73
- enPasPossible
 - PolyBook, 61
- enPassant
 - Move, 44
- endian_swap_u16
 - PolyBook, 60
- endian_swap_u32
 - PolyBook, 60
- endian_swap_u64
 - PolyBook, 61
- Engine, 33
 - getBook, 33
 - getConfig, 34
 - getInstance, 34
- EngineConfig, 35
- epstr
 - IO, 18
- evalBishops
 - Evaluator, 36
- evalKings
 - Evaluator, 36
- evalKnights
 - Evaluator, 37
- evalPawns
 - Evaluator, 37
- evalQueens
 - Evaluator, 38
- evalRooks
 - Evaluator, 38
- evaluatePosition
 - Evaluator, 38
- Evaluator, 35
 - drawnMaterial, 36
 - evalBishops, 36
 - evalKings, 36
 - evalKnights, 37
 - evalPawns, 37
 - evalQueens, 38
 - evalRooks, 38
 - evaluatePosition, 38
- FileChar
 - IO, 18
- fileRankToSq
 - utils.h, 105
- flipBoard
 - Board, 26
- from
 - Move, 44
- generateAllCaptureMoves
 - MoveList, 51
- generateAllMoves
 - MoveList, 52
- generateBishopMoves
 - MoveList, 52
- generateCastlingMoves
 - MoveList, 52
- generateKingMoves

- MoveList, [53](#)
- generateKnightMoves
 - MoveList, [53](#)
- generateNonSlidingMoves
 - MoveList, [54](#)
- generatePawnMoves
 - MoveList, [54](#)
- generatePosKey
 - Hash, [9](#)
- generateQueenMoves
 - MoveList, [55](#)
- generateRookMoves
 - MoveList, [55](#)
- generateSlidingMoves
 - MoveList, [55](#)
- get
 - PvTable, [67](#)
- getAllCaptureMoves
 - Board, [26](#)
- getAllMoves
 - Board, [27](#)
- getBook
 - Engine, [33](#)
- getBookMove
 - PolyBook, [62](#)
- getConfig
 - Engine, [34](#)
- getEnPassant
 - Board, [27](#)
- getHashEntry
 - PvTable, [68](#)
- getInstance
 - Engine, [34](#)
- getProtocol
 - ConsoleManager, [32](#)
 - ProtocolManager, [64](#)
 - UCIManager, [80](#)
 - XBoardManager, [82](#)
- getPvLine
 - PvTable, [68](#)
- getTimeInMilli
 - Stopwatch, [78](#)
- Hash, [9](#)
 - generatePosKey, [9](#)
 - hashCa, [10](#)
 - hashEP, [10](#)
 - hashPce, [10](#)
 - hashSide, [11](#)
- hashCa
 - Hash, [10](#)
- hashEP
 - Hash, [10](#)
- hashPce
 - Hash, [10](#)
- hashSide
 - Hash, [11](#)
- inCheck
 - Board, [27](#)
- Init, [11](#)
 - initAll, [12](#)
 - initBitMasks, [12](#)
 - initEvalMasks, [13](#)
 - initFileRankBrd, [13](#)
 - initHashKeys, [13](#)
 - initMvvLva, [14](#)
 - initSq120ToSq64, [14](#)
- initAll
 - Init, [12](#)
- initBitMasks
 - Init, [12](#)
- initEvalMasks
 - Init, [13](#)
- initFileRankBrd
 - Init, [13](#)
- initHashKeys
 - Init, [13](#)
- initMvvLva
 - Init, [14](#)
- initSq120ToSq64
 - Init, [14](#)
- InputWaiting
 - utils.cc, [120](#)
 - utils.h, [105](#)
- insert
 - PvTable, [69](#)
- IO, [15](#)
 - epstr, [18](#)
 - FileChar, [18](#)
 - parseMove, [15](#)
 - PceChar, [18](#)
 - printBestMove, [16](#)
 - printBitBoard, [16](#)
 - printBoard, [17](#)
 - printMoveList, [17](#)
 - printSearchDetails, [17](#)
 - RankChar, [19](#)
 - SideChar, [19](#)
- isGameOver
 - SearchAgent, [73](#)
- isNull
 - Move, [45](#)
- isOver
 - ProtocolManager, [65](#)
- isPiece
 - utils.h, [106](#)
- isRepetition
 - SearchAgent, [73](#)
- kInterval
 - SearchAgent, [76](#)
- KingEndGame
 - Value, [21](#)
- KingOpening
 - Value, [22](#)
- KnightTable
 - Value, [22](#)

- loop
 - ConsoleManager, [32](#)
 - ProtocolManager, [65](#)
 - UCIManager, [80](#)
 - XBoardManager, [82](#)
- makeMove
 - MM, [40](#)
- makeNullMove
 - MM, [41](#)
- MM, [39](#)
 - addPiece, [39](#)
 - clearPiece, [40](#)
 - makeMove, [40](#)
 - makeNullMove, [41](#)
 - movePiece, [41](#)
 - takeMove, [41](#)
 - takeNullMove, [42](#)
- Move, [42](#)
 - captured, [43](#)
 - castle, [44](#)
 - enPassant, [44](#)
 - from, [44](#)
 - isNull, [45](#)
 - pawnStart, [45](#)
 - promoted, [46](#)
 - to, [46](#)
 - toString, [46](#)
 - wasCapture, [47](#)
 - wasPromotion, [47](#)
- MoveFlags, [19](#)
- MoveList, [48](#)
 - addCaptureMove, [49](#)
 - addEnPasMove, [49](#)
 - addPawnCaptureMove, [50](#)
 - addPawnMove, [50](#)
 - addQuietMove, [51](#)
 - generateAllCaptureMoves, [51](#)
 - generateAllMoves, [52](#)
 - generateBishopMoves, [52](#)
 - generateCastlingMoves, [52](#)
 - generateKingMoves, [53](#)
 - generateKnightMoves, [53](#)
 - generateNonSlidingMoves, [54](#)
 - generatePawnMoves, [54](#)
 - generateQueenMoves, [55](#)
 - generateRookMoves, [55](#)
 - generateSlidingMoves, [55](#)
 - reorderList, [56](#)
 - size, [56](#)
- movePiece
 - MM, [41](#)
- movemaker.h
 - CastlePerm, [99](#)
- MvvLva, [20](#)
- parseFEN
 - Board, [28](#)
- parseMove
 - IO, [15](#)
- pawnStart
 - Move, [45](#)
- PawnTable
 - Value, [22](#)
- PceChar
 - IO, [18](#)
- perft
 - PerftTester, [57](#)
- perftTest
 - PerftTester, [58](#)
- perftTestAll
 - PerftTester, [58](#)
- PerftTester, [57](#)
 - perft, [57](#)
 - perftTest, [58](#)
 - perftTestAll, [58](#)
- PolyBook, [59](#)
 - book, [63](#)
 - convertPolyMove, [60](#)
 - enPasPossible, [61](#)
 - endian_swap_u16, [60](#)
 - endian_swap_u32, [60](#)
 - endian_swap_u64, [61](#)
 - getBookMove, [62](#)
 - polyKeyFromBoard, [62](#)
 - polyPiece, [63](#)
 - readBook, [62](#)
- polyKeyFromBoard
 - PolyBook, [62](#)
- PolyKeys, [20](#)
- polyPiece
 - PolyBook, [63](#)
- PolyglotEntry, [63](#)
- popBit
 - bitboard.h, [86](#)
- printBestMove
 - IO, [16](#)
- printBitBoard
 - IO, [16](#)
- printBoard
 - IO, [17](#)
- printMoveList
 - IO, [17](#)
- printSearchDetails
 - IO, [17](#)
- promoted
 - Move, [46](#)
- ProtocolManager, [64](#)
 - getProtocol, [64](#)
 - isOver, [65](#)
 - loop, [65](#)
- PvEntry, [66](#)
- PvTable, [66](#)
 - clear, [67](#)
 - get, [67](#)
 - getHashEntry, [68](#)
 - getPvLine, [68](#)

- insert, [69](#)
 - size, [70](#)
- quiescenceSearch
 - SearchAgent, [75](#)
- randU64
 - utils.cc, [120](#)
 - utils.h, [106](#)
- RankChar
 - IO, [19](#)
- readBook
 - PolyBook, [62](#)
- ReadInput
 - utils.cc, [121](#)
 - utils.h, [106](#)
- reorderList
 - MoveList, [56](#)
- resetBoard
 - Board, [28](#)
- RookTable
 - Value, [23](#)
- SearchAgent, [70](#)
 - alphaBeta, [71](#)
 - checkStop, [72](#)
 - clearForSearch, [72](#)
 - drawnMaterial, [73](#)
 - isGameOver, [73](#)
 - isRepetition, [73](#)
 - kInterval, [76](#)
 - quiescenceSearch, [75](#)
 - searchPosition, [75](#)
 - threeFoldRepetition, [76](#)
- SearchInfo, [77](#)
- searchPosition
 - SearchAgent, [75](#)
- setBit
 - bitboard.h, [87](#)
- setUpCastlePerm
 - Board, [29](#)
- setUpMoveCounters
 - Board, [29](#)
- setUpPieces
 - Board, [30](#)
- showAttackedSqs
 - utils.cc, [121](#)
 - utils.h, [107](#)
- SideChar
 - IO, [19](#)
- size
 - MoveList, [56](#)
 - PvTable, [70](#)
- sqAttacked
 - Board, [30](#)
- sqOnBoard
 - utils.cc, [121](#)
 - utils.h, [107](#)
- sqToString
 - utils.cc, [122](#)
 - utils.h, [108](#)
- start
 - Stopwatch, [78](#)
- stop
 - Stopwatch, [79](#)
- Stopwatch, [77](#)
 - getTimelnMilli, [78](#)
 - start, [78](#)
 - stop, [79](#)
- stringToLower
 - utils.cc, [122](#)
 - utils.h, [108](#)
- takeMove
 - MM, [41](#)
- takeNullMove
 - MM, [42](#)
- threeFoldRepetition
 - SearchAgent, [76](#)
- to
 - Move, [46](#)
- toString
 - Move, [46](#)
- UCIManager, [79](#)
 - getProtocol, [80](#)
 - loop, [80](#)
- UndoMove, [81](#)
- updatePieceLists
 - Board, [30](#)
- utils.cc
 - InputWaiting, [120](#)
 - randU64, [120](#)
 - ReadInput, [121](#)
 - showAttackedSqs, [121](#)
 - sqOnBoard, [121](#)
 - sqToString, [122](#)
 - stringToLower, [122](#)
- utils.h
 - fileRankToSq, [105](#)
 - InputWaiting, [105](#)
 - isPiece, [106](#)
 - randU64, [106](#)
 - ReadInput, [106](#)
 - showAttackedSqs, [107](#)
 - sqOnBoard, [107](#)
 - sqToString, [108](#)
 - stringToLower, [108](#)
 - WhiteToBlack, [109](#)
- Value, [20](#)
 - BishopTable, [21](#)
 - KingEndGame, [21](#)
 - KingOpening, [22](#)
 - KnightTable, [22](#)
 - PawnTable, [22](#)
 - RookTable, [23](#)

wasCapture
 Move, [47](#)
wasPromotion
 Move, [47](#)
WhiteToBlack
 utils.h, [109](#)

XBoardManager, [81](#)
 getProtocol, [82](#)
 loop, [82](#)