ChessManager

- nb args : int - args : char*[]
- + ChessManager(int argc, int argv)
- + go() : int

GameManager

- # player1 : Player # player2 : Player
- # board : ChessBoard
- # nb turn no move : int # observer : Observer
- + GameManager(Player p1, Player p2)
- + set observer(Observer o)
- + play(): int
- check move(): bool
- check rules(): bool
- add history(Chessboard cb): void
- clear history(): void

Options

- argc : int
- argv_ : char*[] is pgn:bool
- + Options(int argc, char* argv[])
- + parse(): bool + is pgn(): bool
- + get_pgn() : std::string + get_listeners() : std::vector<std::string>

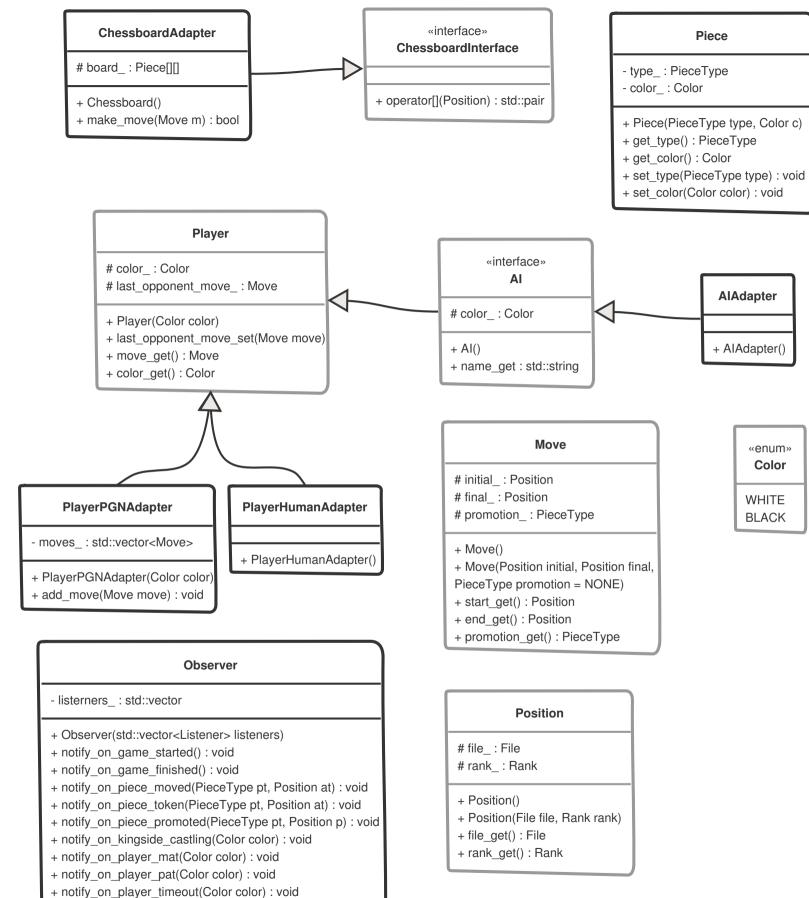
ParserPGN

- file : std::string
- white_player_ : PlayerPGNAdapter
- black player : PlayerPGNAdapter
- + Parser(std::string filename)
- + parse(): bool
- + get_white_player : Player
- + get black player : Player

«interface»

Listener

- + register_chessboard_interface(ChessboardInterface cbi): void
- + on game started(): void
- + on_game_finished(): void
- + on_piece_moved(PieceType pt, Position at) : void
- + on_piece_token(PieceType pt, Position at) : void
- + on_piece_promoted(PieceType pt, Position p) : void
- + on kingside castling(Color color): void
- + on_player_mat(Color color) : void
- + on_player_pat(Color color) : void
- + on player timeout(Color color): void
- + on_player_disqualified(Color color) : void
- + on draw(): void



+ notify_on_player_disqualified(Color color) : void

+ notify_on_draw(): void

«enum» Rank

RANK FIRST **EINS** ZWEI DREI **VIER FUNF** SECHS SIEBEN **ACHT** RANK LAST

> «enum» File

FILE FIRST **ANNA BELLA CESAR** DAVID EVA **FELIX GUSTAV HECTOR** FILE LAST

«enum»

Color

WHITE

BLACK

«enum» PieceType

KING QUEEN ROOK **BISHOP** KNIGHT NONE

PAWN

http://sketchboard.io