**Chess game with c language**

**Using an array, we initialized the board –using a function called ‘board game’- with the characters of both players and the borders to indicate the place from ‘a to h’ and from ‘1 to 8’.**

**We’ve created a function called ’print board’ to print the board by adding additional characters: grey color to represent the white squares and green to represent the black squares.**

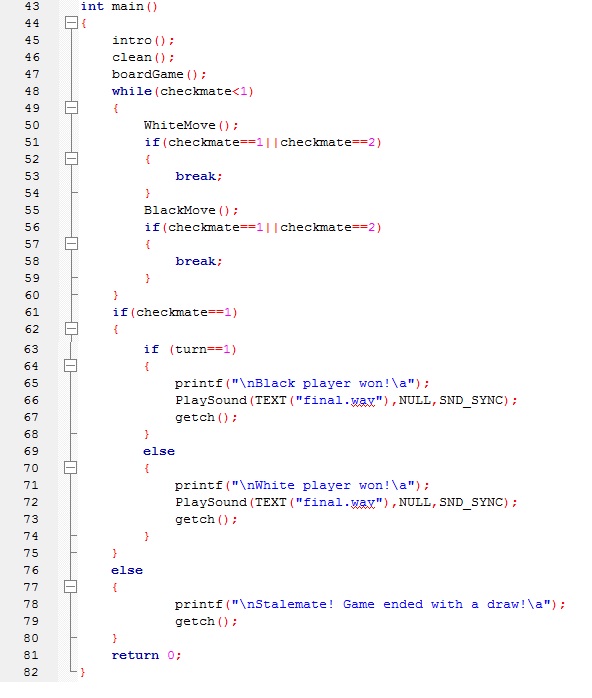
**We’ve made a function called ’clean’ to clean the screen after each print, for printing the board with the last move.**

**We’ve created a function called ‘modify letter’ to convert the user’s input for a place -using the ascII table- to the right place in the board.**

**We’ve created a static integer to indicate the player’s turn and it’s equal to: 1 for the white and 0 for the black, then we’ve created two functions, one to control the white moves and the other to control the black moves.**

**We’ve created a function for each piece to control its moves (if they’re valid or not).**

**By using a while loop, we’re calling the white moves function then the black moves function until there is a stalemate or a checkmate.**



**If it’s the white’s turn for example, when we get an input from the user, we check the origin place he entered contains one of his own characters or not, if it contains, we check the place where he wants to go and if this path is valid or not. If the path is valid, we check what the destination contains: the piece can move if the destination is empty or contains a character of the other player (while it’s not the king), but if it contains one of his own characters, it won’t be a valid move.**

**We’ve created a function called ‘storing’ to get the eaten pieces and store them in 4 columns (2 for each player) that are placed in the right and the left, and we place the eaten pieces with the order of the initial board: a column for the pawns and the other for the other pieces placed in order.**

**PROMOTION: for the promotion, if a pawn gets the end of the board, the player enters the piece that he wants to exchange the pawn with, then we initialize this place with the new piece.**

**UNDO: for the undo, we’ve put the board in a three dimensional array, two dimensions for the board, and the third dimension to store the board in each play with a counter that increases by 1 after each valid move, if the player enters ‘u’, we print the board with the third dimension ‘counter-n’ (undo n times).**

**REDO: we print the board with the ‘counter-n + n’ (undo & redo n times) with a function called ‘redo’.**

**SAVE: we save the board with its moves in a text file using a function called ‘save’.**

**LOAD: we upload the saved file if we typed ‘save’ at least once using a function called ‘load’.**

**CHECK: after each play, with a function called ‘check king’, we find the king’s coordinates and check if the king is threatened by a piece from the other player or not, if the player whose king is checked doesn’t save his king in his turn, by a function called ‘check the check’, we undo his last play and ask him to enter a play to save his king with.**

**STALEMATE: by checking all the non-eaten elements for each player, if all this pieces don’t have any valid moves (they are surrounded by characters from their team), and the king should move but the next step it will be eaten (there isn’t any safe place around it), the game ends, it’s a draw.**

**CHECKMATE: if the king is checked in its place and cannot save itself, and there isn’t any piece from its team that can make a valid move to save it (to eat a piece that checks the king), the game ends and the winner is the other player.**

**Extras:**

**- We arrange the eaten pieces in the same order as they are in the original board to be more comfortable to the eyes.**

**- We’ve created an intro to the game.**

**- The players enter their names and play the game with their names (instead of white and black).**

**- We colored the screen and the words.**

**- We added a music that is played in the intro, and another one when the winner wins.**

**Made by: - Mina Achraf Ishac Barsoum 71**

**- Pierre Maged Mounir Ibrahim 22**

