



Department of Computer Science and Engineering

Course Code: CSE-3636

Course Title: Artificial Intelligence Lab

Project Title: AI based Chess Game using Python

Project Documentation

Week 3: Undo, Valid & Pawn Moves

Submitted by

Lamisa Mashiat (C201249)

Sohana Tasneem (C201266)

Sadia Haque Chowdhury (C201270)

Submitted to

Subrina Akter

Assistant Professor

Dept. of CSE

1st April, 2023

We have updated both the ChessEngine.py & ChessMain.py script this week with including functionalities of undo moves, valid moves & legal pawn moves.

ChessEngine.py

- We have added a dictionary **moveFunctions** that holds the reference to the functions that generate moves for each type of chess piece. The keys in the dictionary represent the type of chess piece, and the values represent the function that generates moves for that piece type.
- Function **undoMove** undoes the last move made by the player. It checks if there is any move to undo in the **moveLog** list, and if so, it pops the last move object from the list and updates the chess board accordingly. The player's turn is also switched back to the previous player.
- Functions **getValidMoves** and **getAllPossibleMoves** generate all the possible moves for the player whose turn it is. The **getValidMoves** function is not considering checks for now, while the **getAllPossibleMoves** function considers all the possible moves without checking for checks. It loops through all the squares on the chess board and calls the appropriate move function based on the piece type on that square. The generated moves are stored in the **moves** list and returned.
- **getPawnMoves** function get all the pawn moves for the pawn located at row, col and add these moves to the list. It modifies the `moves` list to include all valid moves for the pawn. This function adds all possible pawn moves to the `moves` list based on the current board state. It takes into account the color of the pawn and whether it can capture an enemy piece. If a pawn can make a two-square move, it will be added to the `moves` list as well.

ChessMain.py

The while loop is the main game loop that runs until the game is over or the player quits. Within the loop, the function checks for user events such as mouse clicks or key presses and updates the game state accordingly.

The if-else conditions within the loop handle these events. For example, if the user clicks the mouse, the function first determines which square was clicked and adds it to a list of player clicks. If two squares have been clicked, the function creates a new Move object using the two squares and checks if the move is valid. If the move is valid, the function updates the game state by making the move and sets the moveMade flag to True.

If the user presses the 'z' key, the function undoes the last move and sets the moveMade flag to True.

The drawGameState() function is called at the end of each iteration of the loop to update the game board on the screen. The clock.tick() function sets the maximum frame rate for the game. Finally, the p.display.flip() function updates the display to show any changes made during the iteration.

Overall, the while loop and the if-else conditions within it serve to keep the game running, handle user input events, and update the screen accordingly.