

LBYCPA1 FINAL PROJECT DOCUMENTATION

CHESS GAME

Section EQ8

Vincent Gabriel P. Mamac
Zach Marius S. De Guzman

For our final project, we created a chess game in Python that implemented the same conventional mechanics as an ordinary chess game would have. We promised that the code would allow you to play against an AI opponent or a real opponent (locally). We had also stated that the program would implement castling, en passant, pawn promotion, etc. All those promises, we delivered through several tutorial videos on YouTube and a thorough understanding of the functions of the Python module, Pygame.

In our code, we first had to draw the chessboard and design the chess pieces by making sure that the dimensions and the sizes were correct so that the chessboard would be made up of an 8x8 grid (or that when we run the program, the window isn't overlapping the screen) and that the chess pieces would perfectly fit in each square. Then we loaded up some images of chess pieces we installed from the web and colored the board appropriately.

After that, we made it so that the user could click a chess piece and be able to move and control it, and then we started touching more on the mechanics of the game, such as checkmate, check, or stalemate, and the movements of each piece (under the ChessEngine.py file), making sure that their movements are as accurate as possible and as similar as can be to the actual chess game. For instance, the pawn's movement was restricted to vertical shifts, with diagonal movement permitted only when capturing an opponent's piece. Similarly, the rook could traverse horizontally and vertically, while the bishop moves diagonally. The queen's movement combined the capabilities of both the rook and bishop. Finally, the king's movement was limited to a single square in any direction.

We then made it so that the program would allow the user to undo his movements by pressing the ("z") button and worked further on the mechanics of the game, expanding the possibilities of what the user could do in the game. There is also a way for the user to reset the whole game back to its starting position by pressing the ("r") button. We implemented castling, en passant, and pawn promoting, making for an authentic chess experience.

Then we started to work on the animation of the chess pieces, which made it seem like the chess pieces were actually moved to a specific square, not teleported there, adding to the realism of the game. Additionally, we added a feature that made it possible to find out where your selected chess piece can move next by highlighting the squares where the pieces are allowed to move.

Finally, we worked on the most difficult part of the project, which was the AI opponent. However, through a series of difficulties and a YouTube guide, we were able to implement a fairly intelligent AI that could surely put up a challenge, at least for the average chess player.

All these aforementioned features make up for a chess game that is functional, authentic, and very enjoyable to play.

Although we were satisfied with the overall product, we wish to implement more features in the near future that could significantly improve our chess program. Features such as the option for each player to forfeit when he feels that he is losing or a feature that allows each player to see how many pieces were captured from him and how many pieces he captured.