

## Project 1 Refection document

The program assumes basic understanding of standard algebraic notation for manipulating chess pieces. Although it gives some simple instructions at the beginning of the game, if the user is unclear about anything, please refer to the wikipedia page referenced in game.

As examples of how to play the game after 'starting the game', here are two commands to enter to demonstrate the program's ability.

1. white starts: e4, c5, nf3, nc6, bb5, g6, bxc6, dxc6, o-o,...
2. white starts: d4, e5, nf3, bb4,...
3. white starts: f3, e5, g4, qh4.
4. white starts: nf3, h6, d4, a6, nd2,...

For case 1, basic manipulation, capturing, and castling is shown. In case 2, white king is in check, and for case 3, white is checkmated.

In case 4, the disambiguation ability is shown. This can also be achieved for castling, but that requires moving out all pieces other than rooks to the two sides of the king.

Although case 3 demonstrates the checkmate ability, testing was not exhaustive, and it is possible that it breaks.

I think I have successfully implemented most of the features described in the design document. A lot of details, such as disambiguation, was later added since it was not carefully considered when writing the design document. During the live session I was not able to demonstrate the checkmate ability too since it still had a few issues then. I managed to make it work for a test case, but without exhaustive testing, I cannot guarantee its robustness.

The most challenging parts of this project has been debugging the implementations. Since there were a lot of moving parts, and case checking, I had to insert a lot of debugging prints to check the state of the program when running it. I left some of these test prints commented in the checkmate function to show how I did the debugging. Also since the checkmate function is still a work in progress.