Chess Endgames With Retrograde Analysis

Aim & Objectives

The overall aim of the project is to provide solutions to endgame scenarios in chess with 5 or fewer pieces remaining on the board.

Utilise the technique of Retrograde Analysis to generate the tables which are used for solving chess endgames

Compute in advance and store chess endgame tables to improve performance times compared to calculating repeatedly at runtime

Implement an application which is able to output from any user provided scenario, covered by the scope of the project, whether it is winnable and if so in how many moves

Separate the project into four different levels to ensure there are fall-backs provided as a contingency plan of failing to complete all features by the deadline

Implement an existing Chess GUI library in my application to provide a more user-friendly display than the terminal for displaying the chess board and moves being performed

Learn how to use C++ to implement inheritance, polymorphism and dynamically allocated data structures

Method

- I first researched the techniques of retrograde analysis in chess and how they are implemented.
- Researched the different technologies available for implementation and selected C++ as my choice of language
- Designed and implemented my Pieces for use in the endgame algorithm and game playing algorithm
- Developed and tested the endgame generation algorithms using tools and statistics online
- Developed and tested the game playing aspect using the console for input/output.

Motivation

To understand the importance behind this project you must understand the 50-move rule in chess, this states that if within 50 moves (or 100 plies) neither player has

Captured an opponent piece

Moved a pawn

that either player may declare a draw. This creates scenarios where one player may be in an advantageous position, however due to their inability to play perfectly, they are unable to seize the victory.

This project is personally interesting to me as I attended a tournament and was seeded against somebody who required a draw

Future Work

There are several considerations for future work regarding this project, the first thing I would like to focus on is the implementation of a GUI to provide the user a better visual representation than the current console input/output which it uses. I feel as though this would improve the usability of the application, including the ability to drag and drop pieces instead entering the moves on the keyboard.

Additionally, as I did not complete up to 5-pieces, I would like to continue my work to include all 5-piece combinations.

Finally, I would like to improve the tablebase generation algorithms to use symmetry to improve runtimes, whilst this would not affect the user experience, it would allow me to generate larger endgame tables quicker than I am currently able to.