SETU

Ollscoil
Teicneolaíochta
an Oirdheiscirt

South East
Technological
University

Herm0ni Chess Bot
Final Report 2024-2025

Student Name: Seán Rourke

Student Number: C00251168

Supervisor: Joseph Kehoe

Contents

Introduction 3
Project Description 3
  Board Representation 3
  Heuristic Algorithm 3
    Seach Function 4
Challenges 4
  Method Selection 4
  Testing 4
  Implementing the API Client 5
Learning Outcomes 5
  Technical Learnings 5
    Code Writing 5
    Doxygen Documentation 5
    API Usage 5
  Personal Learnings 5
    Time Management 5
    Forward Thinking 6
    Independence and Reliance 6
Project Review 6
  Elements Achieved 6
  Elements Not Achieved 6
  Other Enhancements Identified 6
Differences 7
Personal Reflection 7
Conclusion 7
Acknowledgements 8
Declaration 8

Introduction

This is the final report on the outcome of the Herm0ni chess bot project. The purpose of this report is to provide a description of the Herm0ni chess bot project, along with detailing the successes and failures of the project, how it the project changed over the along the way and the lessons learned during the completion of the project.

Project Description

The goal of the Herm0ni chess bot project was to develop and artificial intelligence (AI) chess bot and integrate it into the lichess.org website so that the bot was able to play games against both humans and other chess bots. AI has been used in chess for many years, to analyse games and provide different ratings of bots for players to play against for practice or for fun. Many of the bots available to today are restricted behind a paid subscription. The purpose of the Herm0ni bot is to provide another free to play chess bot for players.

In the research phase of the project, it was determined that the bot would be writing in C++ due to its fast computational speed. The research phase also revealed that there would be three main aspects of the Herm0ni chess bot:

1. Board Representation
2. Heuristic Algorithm
3. Search Algorithm

Board Representation

Board representation is how the bot stores the positions of the pieces on the chessboard. It is used to track where each piece is and in the move generation of each piece. The Herm0ni bot uses a combination of bitboards and a piece array to represent the board.

A bitboard is a 64-bit integer where the 1 bit represents the presence of a piece. Each chess piece has its own bitboard for each colour, for example there is a bitboard for white bishops. There is also a bitboard for all of the pieces of each colour and a bitboard for all the pieces in general. These are used during move generation. As moves are made, bitwise operations are used to update the bitboards to represent the updated state of the board.

The piece array is an array of 64 elements, where each element is of type Piece, an enum representing a chess piece or an empty square. This is used in conjunction with the bitboards when generating and making moves.

Heuristic Algorithm

The heuristic algorithm is how the bot determines the evaluation of the current position. The Herm0ni bot's heuristic algorithm is based on five aspects:

1. Material Count: The value of all the pieces each player has. The bot uses standard valuations for the pieces.

2. Centre Presence: The number of pieces each player has in the centre of the board. In most scenarios, pieces are stronger in the middle of the board as they control more squares. The bot gives higher value to having pieces in the middle of the board.
3. Centre Attacks: The number of pieces each player has attacking the centre. Having more pieces attacking the centre helps with central control along with centre presence.
4. Development: How developed each players minor pieces (knights and bishops) are. The bot values having active knights and bishops, giving better scores to having these pieces off of the home square.
5. King Safety: How vulnerable each player's king is. The bot values having a pawn shield in front of the king and having the king castled.

The heuristic algorithm is used during the search function, to determine what the best move to make is.

Seach Function

The search function is how the bot chooses which move to make next. The Herm0ni bot uses the minimax algorithm with alpha-beta pruning and looks two moves for each player ahead. This involves traversing a tree of every move that the bot could make using depth-first search.

There are two values, alpha and beta. Alpha is the best score that the bot can guarantee so far, while beta is the best score the opponent can guarantee so far. The bot assumes the opponent makes the best move it has found for the opponent. This means that if at a node in the tree, alpha is greater than beta, the child nodes of that move are not simulated, as assuming best play from the opponent, that position would never be reached.

When a leaf node is reached in the tree, the bot runs the heuristic algorithm to determine the evaluation of the position and then selects the move that results in the best position for itself.

Challenges

Method Selection

Ai has been used in chess for over 60 years and there have been many different approaches in this time. This made the research phase for this project initially overwhelming. With so many different ways to create a bot, it was tough to decide what methods to use. Through thorough research and advice from my project supervisor, the design of the bot was decided.

Testing

Given the many rules in chess and how some rules only apply at specific times, it was tedious and time consuming to ensure that the Herm0ni bot adhered to all rules correctly, as, if a certain rule was to be tested, for example, en passant, the board would need to be setup in a position that allowed en passant, or a game would have to be played until the desired scenario occurred.

Implementing the API Client

Having very little experience with using APIs, it was initially quite difficult to figure out how to have the bot work on the website. To resolve this issue, the open-source stockfish executable was used as a temporary substitute for the Herm0ni executable, while the bot account was setup, and the API client was implemented. Once implementing the UCI protocol in the Hermoni bot, the executables were changed.

One issue that appeared during testing was the bot attempting to make an illegal move seemingly randomly. No solution was found to this issue, but it may be a problem with the API client itself rather than the bot as research has found that other users had a similar issue when using the stockfish executable.

Learning Outcomes

Technical Learnings

Code Writing

From undertaking this project, my ability to write concise, well-structured code has improved significantly. My knowledge of C++ in particular has vastly improved.

Doxygen Documentation

My experience with Doxygen prior to this project was very limited. Using Doxygen to document the code of this project has made me more familiar with the structure of Doxygen comments, and has improved my comment writing in general, making it more clear and well formatted.

API Usage

Before this project I had not used APIs extensively and was unaware of the considerations that needed to be made when using them. Due to the work done on this project, I am more cognizant of factors like security, as the lichess-bot api requires an authentication token that must be securely kept, and when communicating with the API, the data must be safe in the correct format.

Personal Learnings

Time Management

This is the first project that I have worked on for this length of time. With so many different parts of the project to work on, it was important to ensure that a sufficient amount of time was allocated to complete each part of the project, regardless of if I preferred working on one part more than another.

Forward Thinking

A project that goes on for this long is bound to change over time. As more research is done, or the feasibility of ideas are tested, the initial concept differs from the final product. This taught me to think ahead when working on the project, to ensure that sufficient research was done before committing to an idea that could potentially have to be removed later on down the line.

Independence and Reliance

As this was a largely solo project, it has improved my ability at working on my own, looking for answers myself, rather than having someone there to explain everything to me. However, this project has also taught me when to realise that I have reached a roadblock that I cannot pass on my own, and to seek guidance from someone who can share valuable advice.

Project Review

Elements Achieved

I believe that the Herm0ni chess bot has successfully fulfilled the core goal of this project. The bot is capable of playing games on the lichess.org website against both human and bot players. From having the Herm0ni bot play multiple games against the different levels of the Fairy-Stockfish bot on lichess.org, I would estimate that the Herm0ni bot has an Elo rating of 1500-1600.

The source code of the Herm0ni bot is commented using Doxygen, with the documentation being available in the GitHub repository.

Elements Not Achieved

As stated earlier, the Herm0ni heuristic algorithm factors five aspects of chess into its calculation. However, there are many other factors that could be considered for a heuristic algorithm, such as: a piece-square table, pawn structure, piece coordination, etc. If more time was available for the project, more work could be done on implementing these features to improve the heuristic function.

Other Enhancements Identified

The Herm0ni bot is currently only capable of playing standard chess games. The bot could be adapted to work with chess variants such as chess960.

As the bot is implemented on lichess.org, it is only capable of playing games. An additional, off-site resource could be created allowing the bot to evaluate positions that are provided by the user.

Differences

In the early stages of the project, I had the idea to have a bot of multiple levels of skill that could be selected from. When implementing the lichess-bot API client, I discovered that Difficulty selection cannot be added for a single account. This meant that having multiple options required separate code and executables, a separate lichess accounts and running the lichess-bot client separately for each level of the bot. Due to these reasons I opted to just offer a single skill level of bot.

Currently, in order to play against the bot on the lichess.org website, the lichess-bot API client must be run locally on my machine. I had initially considered hosting the bot on a separate platform but due to cost and uncertainty around the security of having the bot account API token in plain text in the config file, I decided against it.

Personal Reflection

If I was to start this project again, I think I would have started coding earlier. While a lot of research was required to determine the design of the bot and what methods it would use for board representation and searching, it can be hard to make these decisions from just research findings. Small code implementations of the possible concepts may have made it easier to come to decisions quicker.

I also would attempt to manage my time better at the start of the project. Spending such a large portion of time on the functional specification, design document and research poster led to less time later on for the coding. Dividing this time up better would leave more time for improvement of the heuristic function, improving the strength of the bot.

I also would try to do less work in more frequent intervals. Taking breaks of a few days sometimes led to a bit of time needed to refresh myself on what I was currently working on when I resumed project work. Spreading the work into shorter, more frequent intervals would allow the same amount of time to be committed to the work, while rectifying the issue of forgetting exactly what was being worked on at the time.

Conclusion

In conclusion, the Herm0ni chess bot project was an invaluable experience for me as a developer and one that I thoroughly enjoyed working on despite the challenges that arose throughout the project. I am proud that my goal of creating a fully functional chess bot has been achieved and that I've been able to have it play against me and my friends. This project has tested me in ways that I have never experienced before, and I appreciate the knowledge and skills that I have acquired in facing these challenges.

Acknowledgements

I would like to thank Dr. Joseph Kehoe, my project supervisor. From the very beginning of this project, Joseph has offered me incredible support and guidance, from highlighting the key aspects of chess programming before this topic was chosen, to answering any question I had throughout the project.

Declaration

- I declare that all works submitted in the completion of this project are of my own preparation and production.
- Where relevant, I have cited all external sources and included a Bibliography of terms as appropriate.
- I understand that failure to comply with the University's regulations governing plagiarism constitutes a serious offence.

Student Name: Seán Rourke

Student No: C00251168

Date: April 2025