Chess Engines: How to Make Chess Accessible to the General Public

A Technical Report
presented to the faculty of the
School of Engineering and Applied Science
University of Virginia

by

Stephen Johnson

May 11, 2023

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Stephen Johnson

Technical advisor: Rosanne Vrugtman, Department of Computer Science
Technical advisor: Briana Morrison, Department of Computer Science

Chess Engines: How to Make Chess Accessible to the General Public

CS 4991 Capstone Report, 2022
Stephen Johnson
Computer Science
University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
Smj8jqu@virginia.edu

Abstract
In the past two years, the popularity of chess has begun to rise as a result of pandemic policies, but some of the main tools available to players are locked away due to technical or economic constraints. To remedy this issue, I propose an application which would run in Docker, a combination of a chess engine and UI, which would function either locally or deployed to the web. The architecture would be a ReactJS user interface and a Java Spring Boot backend service connected using WebSockets. It would also be easily monetizable through Google AdSense in order to pay for hosting and compute power for the engine. Using a series of algorithms for building efficient tree data structures, the engine can be incrementally improved over time. The two potential outcomes from this project are the app itself and the skills that I have and will gain along the way. Going forward there are many ways to continuously improve the product. Among them are more algorithms to improve both the engine, and the user interface as industry trends change.

1. Introduction

The game of Chess is not merely an idle amusement. Several very valuable qualities of the mind, useful in the course of human life, are to be acquired or strengthened by it, so as to become habits, ready on all occasions. For life is a kind of chess…
—Benjamin Franklin [2]

Chess has existed for centuries and has always been considered intellectually intensive. This being the case, historically it was played by the wealthy elite. As the striations in society gradually weakened, more people had access and the desire to engage with chess. Over the past five years this has been especially true, and players want to have the capacity to improve. A main driver to this end is the availability of chess engines, which allow players to understand the dynamics at play in any given chess position, and to see the different ways they could improve it as the game continues.

Chess is an extremely important game. It provides the best analogue for strategy throughout literature and media, while also being an intellectually stimulating activity on its own. The more accessible the tools necessary to development as a player are, the more this activity can be made available to people of all ages and backgrounds.

2. Related Work

Turing (1953) used chess as an example to show the capabilities of his "Turing Machine" and used the game as a theme throughout his life's work. He explores a set of ideas about the limitations of computers, mainly writing about chess. After playing out a game according to his algorithm he concluded that his program would not fare well against strong human competitors [1]. He believed that the flaws

in his program were due to his own lack of skill in the game [1].

With slow, incremental algorithmic advancement over the course of the next 40 years, the next and most visible advancement in computer chess as reported by Campbell, et. al. (2002) was Deep Blue [3]. IBM created this system, which played against Kasparov in 1996 and 1997, the then-reigning World Chess Champion [3]. This system is massively more complex than Turing's, specifically designed around parallel processing for searching the tree of legal moves. Deep Blue II, which beat Kasparov 3.5-2.5 in 1997, used over 500 processors for tree searching [3].

Since Deep Blue, there has been much more open-source development on chess engines. The most recognizable of this class is called Stockfish (2022) [4]. A 2008 fork of the Glaurung engine, Stockfish is currently one of the most popular chess engines in the world. Unlike Deep Blue, it is purely software-based, meaning it can run on any hardware. Its main limitation is that it is an engine with no user interface, which is a massive barrier for entry to anyone who is not familiar with a command line.

3. System Design

The system designed has a simple frontend-backend architecture. The frontend consists of two screens, a login/landing page and the chessboard screen. The user is authenticated with the Google authentication system, which gives them access to the page with all of the functionality. It is built in ReactJS and uses libraries such as react-chessboard in order to provide the visuals for the functionality provided in the software. Making the UI browser-based rather than a custom desktop interface also allows for the system to be usable on a wider variety of devices and operating systems with minimal platform-specific testing.

3.1 System Logic

The backend holds the logic of the system. The system builds a tree of legal chess moves and searches for the nodes which give the given player the highest advantage according to an evaluation function.

3.2 Evaluation

The evaluation function allows the system to decide whether a move is "good" or "bad." The evaluation function currently used by the system is solely dependent on the values of the pieces. The value returned by the function is what is stored in each node, where a positive value favors white and a negative value signifies an advantage for black. The nodes are then searched for an advantage for the player who moves next, and does a depth-first search for the most favorable position at a given depth. This value then propagates up the tree in order to find the best move for the moving player. It does this by also calculating the best move for the opposing player at each depth in order to "steel man" the competition.

A limitation associated with this approach is that for most beginner players, computed "best moves" are indecipherable. This is because a move is only as good as the moves it allows you to make in the future. If someone is a beginner to chess and the engine tells them to play like a grandmaster, they will be lost and confused.

3.3 Communication

The communication between the frontend and the backend is currently done through REST API endpoints between the ReactJS frontend and the Java Spring backend. This means that the tree must be reconstructed for every call to the backend. This has the effect that the tool is only effective at low

depths, a suboptimal approach that and will be refactored to be based on WebSockets in future iterations.

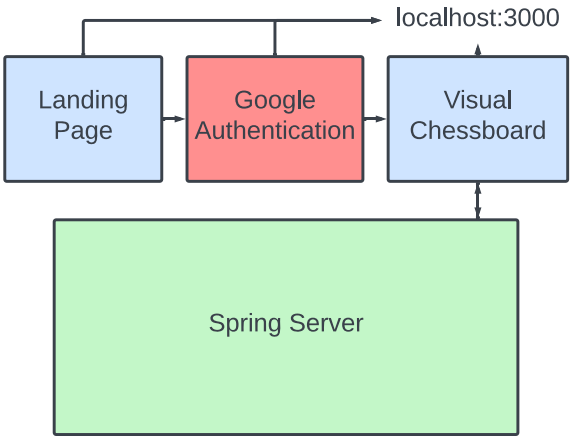

Figure 1: Architecture Diagram

3.3 Environment

The entire system is run in docker, and for ease of use it was configured with docker compose. This means that the entire thing can run locally on any machine with docker installed and use the host machine's resources to do the computation. This saves the product from being run in one of the major compute engine services where it would undoubtedly rack up extreme fees for the computationally expensive tree building. The docker compose file is in the UI repository and builds both simultaneously in order to deliver both the frontend and therefore the backend to the user via the browser.

4. Results

The result of this project is a functional app which runs a web-like format used locally on a host machine. This project continues in development and will likely continue to be for some time. We have great ambitions for the level of complexity of the engine itself as well as accessibility for the general public. Ideally, with proper funding acquired, we could host it on one of the major web service platforms and allow it to be used by the general public for training and learning purposes.

Generally, the project is still a work in progress but the proof of concept provided thus far is sufficient to argue that this tool can be sufficient to train against for beginner players.

5. Conclusion

Chess is an incredibly important game not only for the higher-level players, but also as a tool for people to improve their problem-solving skills. The current state-of-the-art fails to properly account for the technical abilities of the general population. In order to bring a product to market for the general populace, it needs to have an intuitive user interface and a lack of technical obstacles to get it running. This product allows someone of any technical background to train their brain with chess in order to get better at the game and become a better problem solver.

6. Future Work

This engine and interface are far from perfect. It needs much more styling and refinement before it could be a viable product in the market. One way it could be vastly improved is training machine learning models for the evaluation function, which would assist the system in building a tree that would more accurately predict the value of a given position. This is the way many of the current state-of-the-art systems work, and given enough compute being put into training the model, it could actually make the system itself somewhat computationally lighter than it otherwise would be.

Tree pruning could also be used to remove the obviously bad moves from the tree in order to better predict which moves the player will make. With beginner players this is a very limited improvement, though, because the deleted parts of the tree could still need to be traversed by the players

making bad moves, which may actually slow down the system. It is still a desirable improvement because as a player progresses it becomes more and more useful to improve efficiency.

References

[1] Alan Turing. 1953. Digital Computers applied to Games. Retrieved from https://turingarchive.kings.cam.ac.uk/publications-lectures-and-talks-amtb/amt-b-7.

[2] Benjamin Franklin. before June 1779. The Morals of Chess. Retrieved from https://founders.archives.gov/documents/Franklin/01-29-02-0608.

[3] Murray Campbell, A. Joseph Hoane Jr., and Feng-hsiung Hsu. 2002. Deep Blue. Artificial Intelligence 134, 1-2, 57-83. DOI: https://doi.org/10.1016/S0004-3702(01)00129-1.

[4] Stockfish. 2022. About. Retrieved from https://stockfishchess.org/about/.