AlphaDeepChess: motor de ajedrez basado en podas alpha-beta
AlphaDeepChess: chess engine based on alpha-beta pruning

Trabajo de Fin de Grado
Curso 2024–2025

Autores
Juan Girón Herranz
Yi Wang Qiu

Directores
Ignacio Fábregas Álfaro
Rubén Rafael Rubio Cuéllar

Grado en Ingeniería de Computadores
Facultad de Informática
Universidad Complutense de Madrid

Grado en Desarrollo de Videojuegos
Facultad de Informática
Universidad Complutense de Madrid

AlphaDeepChess: motor de ajedrez basado en podas alpha-beta
AlphaDeepChess: chess engine based on alpha-beta pruning

Trabajo de Fin de Grado en Ingeniería de Computadores
Trabajo de Fin de Grado en Desarrollo de Videojuegos

Autores
Juan Girón Herranz
Yi Wang Qiu

Directores
Ignacio Fábregas Álfaro
Rubén Rafael Rubio Cuéllar

Convocatoria: *Junio 2025*

Grado en Ingeniería de Computadores
Facultad de Informática
Universidad Complutense de Madrid

Grado en Desarrollo de Videojuegos
Facultad de Informática
Universidad Complutense de Madrid

28 de marzo de 2025

Dedication

To our younger selves, for knowing the art of chess

Acknowledgments

To our family members for their support and for taking us to chess tournaments to compete.

Resumen

AlphaDeepChess: motor de ajedrez basado en podas alpha-beta

Los motores de ajedrez han influido notablemente en el desarrollo de estrategias computacionales y algoritmos de juego desde mediados del siglo XX. Informáticos de la talla de Alan Turing y Claude Shannon sentaron las bases para el desarrollo de este campo. Posteriormente, las mejoras de hardware y software y la evolución de la heurística se asentarían sobre estos cimientos, incluida la introducción de la poda alfa-beta, una optimización del algoritmo minimax que reducía significativamente el número de nodos evaluados en un árbol de juego. Con el aumento de la potencia de cálculo, motores modernos como Stockfish o Komodo aprovechan no sólo las optimizaciones de búsqueda, sino también los avances en heurística y, en algunos casos, la inteligencia artificial mediante redes neuronales.

Palabras clave

motor de ajedrez,poda alfa-beta,algoritmo minimax,búsqueda árbol de juego,evaluación heurística,ordenación de jugadas,optimización de búsqueda,tablas de transposición,zobrist hashing

Abstract

AlphaDeepChess: chess engine based on alpha-beta pruning

Chess engines have significantly influenced the development of computational strategies and game-playing algorithms since the mid-20th century. Computer scientists as renowned as Alan Turing and Claude Shannon set the foundations for the development of the field. Thereafter, hardware and software improvements and the evolution of heuristics would build upon these foundations, including the introduction of alpha-beta pruning, an optimization of the minimax algorithm that significantly reduced the number of nodes evaluated in a game tree. With increasing computational power, modern engines such as Stockfish or Komodo leverage not only search optimizations but also advancements in heuristics and, in some cases, artificial intelligence using neuronal networks.

Keywords

chess engine,alpha-beta pruning,minimax algorithm,game tree search,heuristic evaluation,move ordering,search optimization,transposition tables,zobrist hashing

Contents

List of figures

List of tables

Chapter 1

Introduction

"The most powerful weapon in chess is to have the next move"
— David Bronstein

Chess, one of the oldest and most strategic games in human history, has long been a domain for both intellectual competition and computational research. The pursuit of creating a machine that could compete with the best human players, chess Grandmasters (GM), was present. It was only a matter of time before computation surpassed human computational capabilities.

In 1997, the chess engine Deep Blue made history by defeating the world champion at the time, Garry Kasparov, marking the first time a computer had defeated a reigning world champion in a six-game match under standard chess tournament time controls[1].

Since then, the development of chess engines has advanced rapidly, moving from rule-based systems to AI-driven models. However, classical search algorithms, such as alpha-beta pruning, continue to be fundamental to understanding the basics of efficient search and evaluation of game trees.

1.1. Objetives

- Develop a functional chess engine using alpha-beta pruning as the core search algorithm.
- Optimize search efficiency by implementing move ordering, quiescence search, and iterative deepening to improve pruning effectiveness.
- Implement transposition tables using Zobrist hashing to store and retrieve previously evaluated board positions efficiently.
- Ensure modularity and efficiency so that the engine can be tested, improved, and integrated into chess-playing applications.

[1] `https://en.wikipedia.org/wiki/Deep_Blue_(chess_computer)`

- Compare performance metrics against other classical engines to evaluate the impact of implemented optimizations.

1.2. Work plan

1. Research phase and basic implementation: understand the fundamentals of alpha-beta pruning with minimax and position evaluation. Familiarize with the UCI (Universal Chess Interface) and implement the move generator with its specific exceptions and rules.

2. Optimization: improve search efficiency using transposition tables and Zobrist hashing.

3. Comparation: use Stockfish to compare efficiency generating tournaments between chess engines and a profiler to detect possible bottlenecks.

Chapter 2

Status of the art

Chapter 3

Work description

3.1. Where to begin?

Let's start from the beginning. What is chess? Chess is a board game where two players who take white pieces and black pieces respectively compete to first checkmate[1] the opponent.

What about a chess engine? A chess engine consists of a software program that analyzes chess positions and returns optimal moves depending on its configuration. In order to help users to use these engines, chess community agreed on creating an open communication protocol called **Universal Chess Interface** or commonly referred to as UCI, that provides the interaction with chess engines through user interfaces.

In the following section 3.2, we will talk about the basic concepts of chess, but if you already have the knowledge we recommend you to advance directly to the section 3.3.

3.2. Basic concepts

Chess is a game of strategy that takes place on a chessboard with specific rules governing the movement and interaction of the pieces. This section introduces the fundamental concepts necessary to understand how chess is played.

3.2.1. Chessboard

A chessboard is a game board of 64 squares, 8 rows by 8 columns. To refer to each of the squares we mostly use **algebraic notation**[2] using the numbers from 1 to 8 and the letters from "a" to "h". There are also other notations like descriptive notation[3] which is obsolete or ICCF numeric notation[4] due to chess pieces have

[1] https://en.wikipedia.org/wiki/Checkmate
[2] https://en.wikipedia.org/wiki/Algebraic_notation_(chess)
[3] https://en.wikipedia.org/wiki/Descriptive_notation
[4] https://en.wikipedia.org/wiki/ICCF_numeric_notation

different abbreviations depending on language.

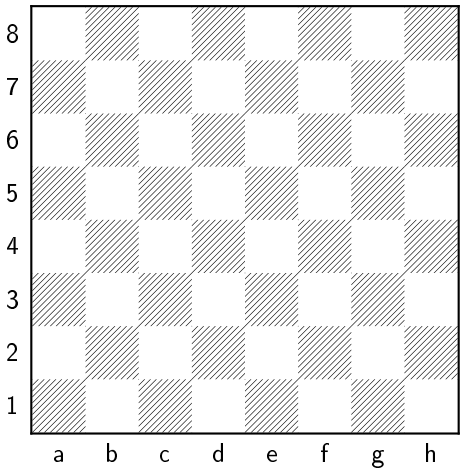

Figure 3.1: Empty chessboard.

For example, **g5** refers to the following square:

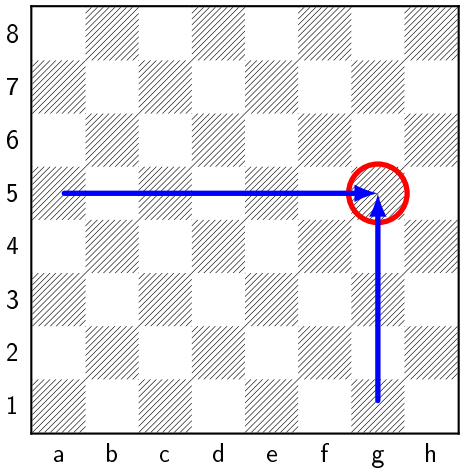

Figure 3.2: Example: square **g5** highlighted and arrows pointing to it.

It is important to know that when placing a chessboard in the correct orientation, there should always be a **white square in the bottom-right corner** or a **black square in the bottom-left corner**.

3.2.2. Chess pieces

There are 6 types of chess pieces: king, queen, rook, bishop, knight and pawn, and each side has 16 pieces:

- 1 king ♔ ♚

- 1 queen ♕♛
- 2 rooks ♖♖♜♜
- 2 bishops ♗♗♝♝
- 2 knights ♘♘♞♞
- 8 pawns ♙♙♙♙♙♙♙♙♟♟♟♟♟♟♟♟

The starting position of the chess pieces on a chessboard is the following:

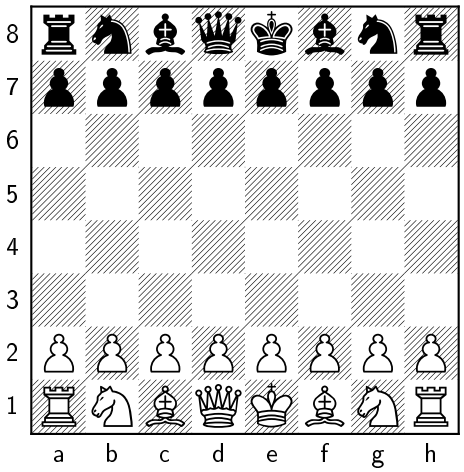

Figure 3.3: Starting position.

If we take a closer look at the positioning of each piece, we can see that the queen and king are placed in the center columns. The queen is placed on a square of its color, while the king is placed on the remaining central column. The rest of the pieces are positioned symmetrically, as shown in Figure 3.3.

This means that the chessboard is divided into two sides relative to the positions of the king and queen at the start of the game:

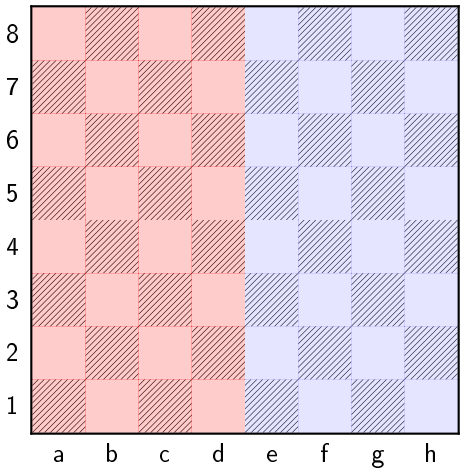

Figure 3.4: King's side (blue) and Queen's side (red).

3.2.3. Movement of the pieces

3.2.4. Pawn

The pawn can move one square forward, but it can only capture pieces diagonally. On its first move, the pawn has the option to move two squares forward. If a pawn reaches the last row of the opponent's side, it can be promoted[5] to any other piece (except for a king).

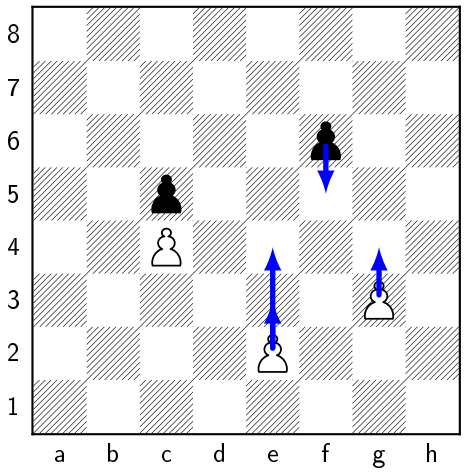

Figure 3.5: Pawn's movement.

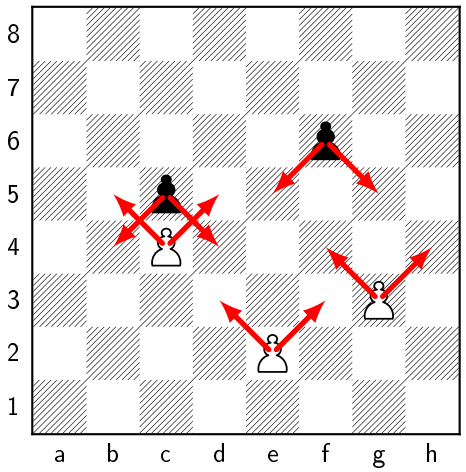

Figure 3.6: Pawn attack.

[5] https://en.wikipedia.org/wiki/Promotion_(chess)

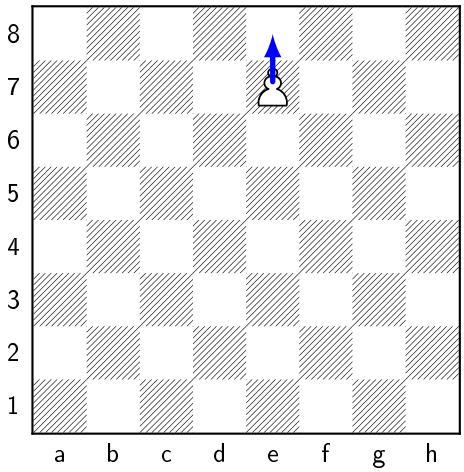

Figure 3.7: Promotion.

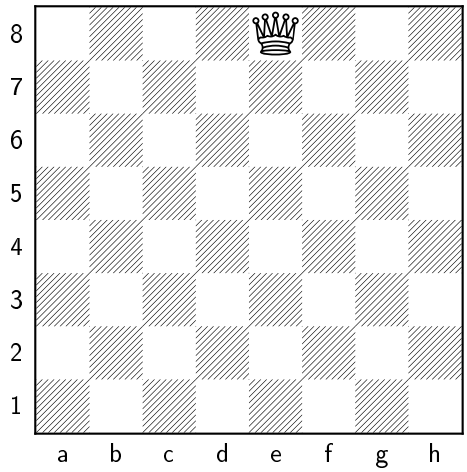

Figure 3.8: Pawn promotes to queen.

3.2.4.1. Rook

The rook can move any number of squares horizontally or vertically. It can also capture pieces in the same way.

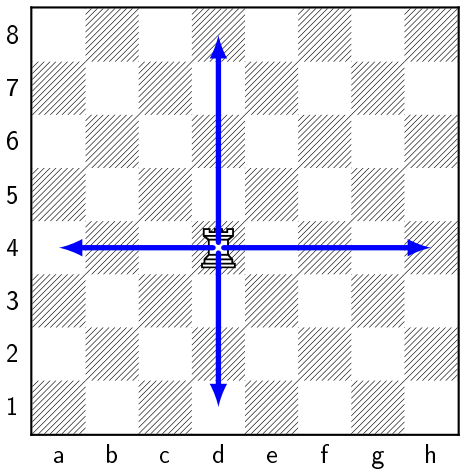

Figure 3.9: Rook's movement.

3.2.4.2. Knight

The knight moves in an L-shape: two squares in one direction and then one square perpendicular to that direction. The knight can jump over other pieces, making it a unique piece in terms of movement. It can also capture pieces in the same way.

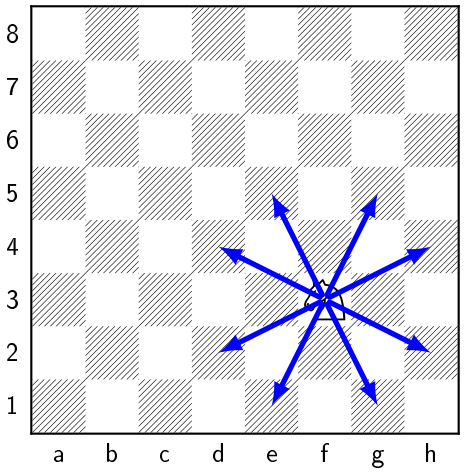

Figure 3.10: Knight's movement.

3.2.4.3. Bishop

The bishop can move any number of squares diagonally. It can also capture pieces in the same way. Considering that each side has two bishops, one bishop moves on light squares and the other on dark squares.

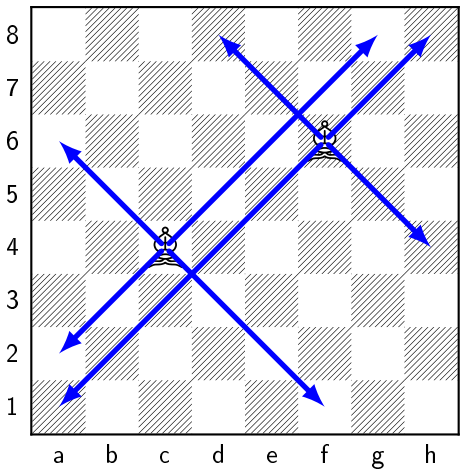

Figure 3.11: Bishop's movement.

3.2.4.4. King

The king can move one square in any direction: horizontally, vertically, or diagonally. However, the king cannot move to a square that is under attack by an opponent's piece. The king can also capture pieces in the same way. The king is a crucial piece in chess, as the game ends when one player checkmates the opponent's king.

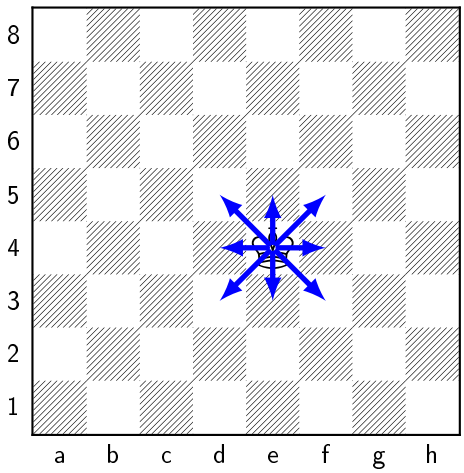

Figure 3.12: King's movement.

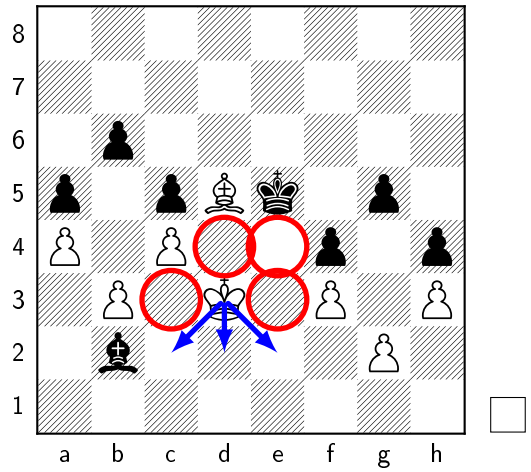

Figure 3.13: White King's movement in a game.

In Figure 3.13, the white king cannot move to **e4** because the black king is attacking that square. When two kings are positioned close to each other, neither can move to a square adjacent to the other.

Additionally, the king can perform a special move called **castling**, which involves moving the king two squares towards a rook and moving the rook to the square next to the king. Castling has specific conditions which are:

- Neither the king nor the rook involved in castling must have moved previously.
- There must be no pieces between the king and the rook.
- The king cannot be in check, move through a square under attack, or end up in check.

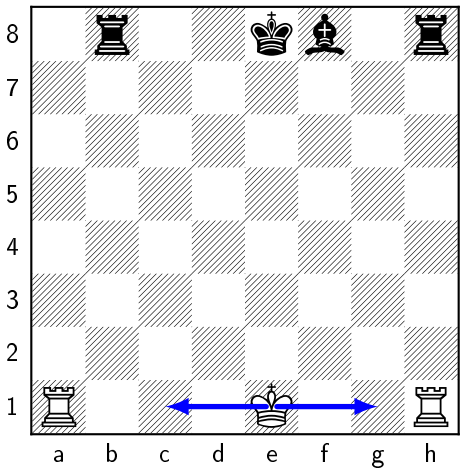

Figure 3.14: Castling

In this case, the white king can castle on either the king's side or the queen's side as long as the rooks have not been moved from their starting position, but the black king cannot castle because there is a bishop on **f8** interfering with the movement and the rook on the queen's side has been moved to **b8**.

3.2.4.5. Queen

The queen can move any number of squares in any direction: horizontally, vertically, or diagonally. It can also capture pieces in the same way.

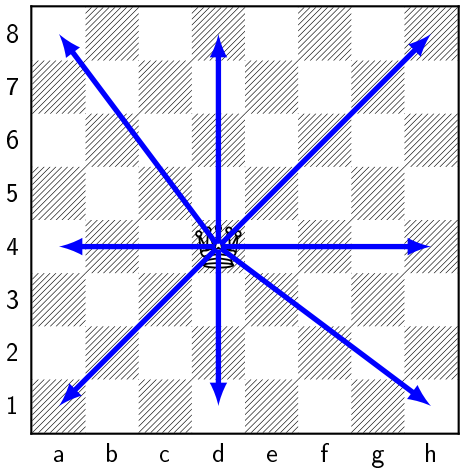

Figure 3.15: Queen's movement.

3.2.5. Rules

In every game, white starts first and the possible results of each game can be win for white, win for black or draw.

...

3.2.6. Notation

In addition to the **algebraic notation** of the squares in section 3.2.1, each piece is identified by an uppercase letter, which may vary across different languages:

Piece	English Notation	Spanish Notation
Pawn	*P*	*P* (peón)
Rook	*R*	*T* (torre)
Knight	*K*	*C* (caballo)
Bishop	*B*	*A* (alfil)
Queen	*Q*	*D* (dama)
King	*K*	*R* (rey)

Table 3.1: Chess piece notation in English and Spanish.

For the normal moves (not captures nor promoting), it is written using the piece uppercase letter plus the coordinate of destination:

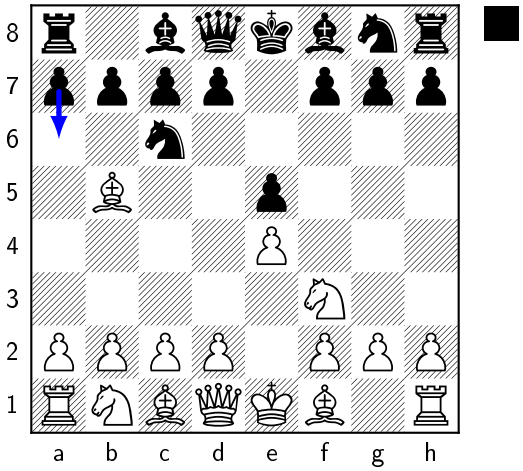

Figure 3.16: Pawn goes to a6.

In Figure 3.16, the pawn's movement is written as **Pa6** or directly **a6**.

Captures are written with an **"x"** between the piece uppercase letter and coordinate of destination/the captured piece coordinate:

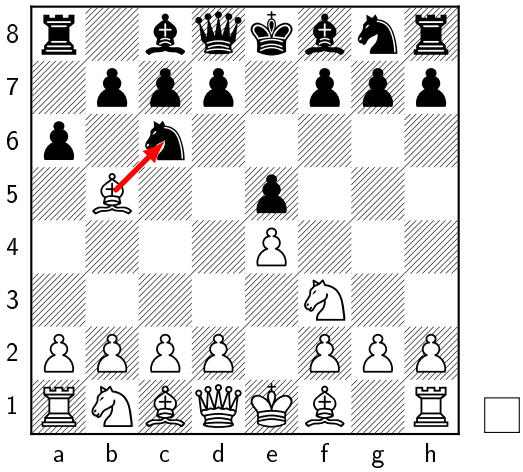

Figure 3.17: Bishop captures knight.

In Figure 3.17, the white bishop capturing the black knight is written as **Bxc6**. If it were black's turn, the pawn on **a6** could capture the white bishop, and it would be written as **Pxb5** or simply **axb5**, indicating the pawn's column.

Pawn promotion is written as the pawn's movement to the last row, followed by the piece to which it is promoted:

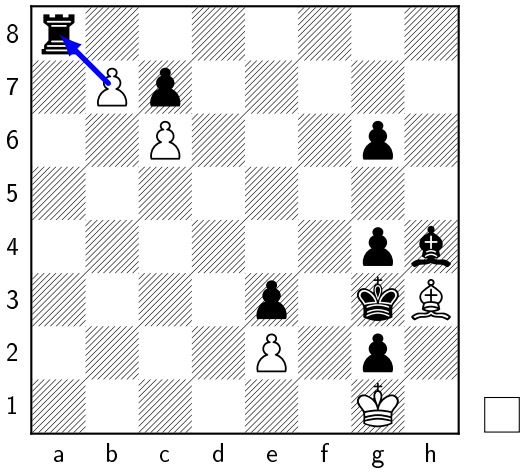

Figure 3.18: Pawn captures rook.

In Figure 3.18, white pawn capturing and promoting in **a8** to a queen is written as **bxa8Q** or **bxa8=Q**.

When castling, depending on whether it is on the king's side or the queen's side, it is written as **0-0** and **0-0-0**, respectively.

Moves that check and checkmates opponent's king are written by adding a + sign for check or ++ for checkmate, respectively.

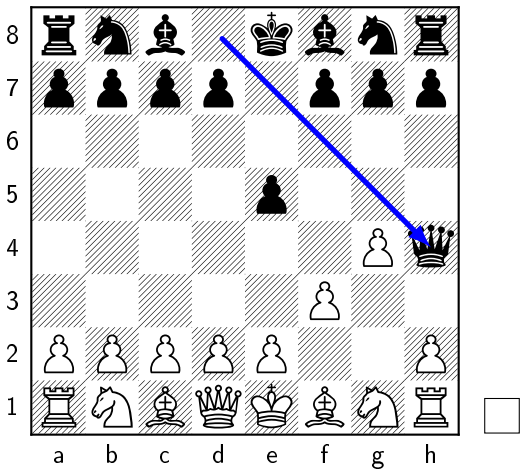

Figure 3.19: Black queen checkmates.

In Figure 3.19, black queen movement checkmates and it is written as **Dh4**++.

The end of game notation indicates the result of the game. It is typically written as:

- **1-0**: White wins.
- **0-1**: Black wins.
- **1/2-1/2**: The game ends in a draw.

For more information about notation, refer to the Wikipedia page: `https://en.wikipedia.org/wiki/Algebraic_notation_(chess)`.

3.3. Modules

3.3.1. Board

3.3.2. Move generator

3.3.3. Move ordering

3.3.4. Evaluation

3.3.5. Search

3.4. Code implementation

3.4.1. Data representation

Use of uint64_t as bitboards to store the board information and other code structures and classes justifying why is efficient.

3.4.2. Initialized memory

Some tables are memory initialized instead of computed, explain it.

3.5. Other

3.5.1. Board visualizer using Python

3.5.2. Comparation using Cutechess and Stockfish engine

3.5.3. Profiling

Chapter 4

Conclusions and Future Work

Conclusiones del trabajo y líneas de trabajo futuro.

Antes de la entrega de actas de cada convocatoria, en el plazo que se indica en el calendario de los trabajos de fin de grado, el estudiante entregará en el Campus Virtual la versión final de la memoria en PDF.

Personal contributions

Student1

Al menos dos páginas con las contribuciones del estudiante 1.

Student 2

Al menos dos páginas con las contribuciones del estudiante 2.

Apéndice A

Título del Apéndice A

Los apéndices son secciones al final del documento en las que se agrega texto con el objetivo de ampliar los contenidos del documento principal.

Apéndice B

Título del Apéndice B

Se pueden añadir los apéndices que se consideren oportunos.

Este texto se puede encontrar en el fichero Cascaras/fin.tex. Si deseas eliminarlo, basta con comentar la línea correspondiente al final del fichero TFGTeXiS.tex.

–¿Qué te parece desto, Sancho? – Dijo Don Quijote –
Bien podrán los encantadores quitarme la ventura,
pero el esfuerzo y el ánimo, será imposible.

Segunda parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes

–Buena está – dijo Sancho –; fírmela vuestra merced.
–No es menester firmarla – dijo Don Quijote–,
sino solamente poner mi rúbrica.

Primera parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes