

**BABEȘ-BOLYAI UNIVERSITY CLUJ-NAPOCA**  
**FACULTY OF MATHEMATICS AND COMPUTER**  
**SCIENCE**  
**SPECIALIZATION COMPUTER SCIENCE**

## **DIPLOMA THESIS**

**Using artificial intelligence to assist  
chess players**

**Supervisor**  
**Asist. univ. dr. Florentin Bota**

*Author*  
*Cadar Eduard*

2023

**BABEȘ-BOLYAI UNIVERSITY CLUJ-NAPOCA  
FACULTY OF MATHEMATICS AND COMPUTER  
SCIENCE  
SPECIALIZATION INFORMATICA**

## **DIPLOMA THESIS**

**Utilizarea inteligenței artificiale în  
asistarea jucătorilor de șah**

**Supervisor  
Asist. univ. dr. Florentin Bota**

*Author  
Cadar Eduard*

2023



---

## ABSTRACT

---

The usual way of searching for a move from a chess position is through tactics or strategy. Tactics are series of moves that would bring an immediate advantage, while strategy refers to a general 'sense' of advantage on the board (having better developed pieces, better pawn structure etc.).

Chess engines are usually built using a minimax algorithm. This approach is able to find the moves that would bring material gain (capturing pieces) in a given depth, but cannot find moves that would slowly improve the position. A neural network trained on professional chess games is able to find good positional moves, but may be weak at finding tactical moves.

An algorithm combining these two approaches should yield better results than each of them on their own.

Abstract: un rezumat în limba engleză cu prezentarea, pe scurt, a conținutului pe capitole, punând accent pe contribuțiile proprii și originalitate

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>2</b>
2.1	Techniques/Algorithms used . . . . .	2
2.2	State of the art chess engines . . . . .	2
<b>3</b>	<b>Methodology</b>	<b>3</b>
3.1	Training . . . . .	3
3.1.1	Min-max algorithm . . . . .	3
3.2	Optimizing . . . . .	3
3.2.1	Alpha-Beta pruning . . . . .	3
<b>4</b>	<b>Technologies</b>	<b>4</b>
4.1	Chess game . . . . .	4
4.2	Chess engine . . . . .	4
<b>5</b>	<b>Results and evaluation</b>	<b>5</b>
<b>6</b>	<b>Conclusions</b>	<b>6</b>
	<b>Bibliography</b>	<b>7</b>

# Chapter 1

## Introduction

Introducere: obiectivele lucrării și descrierea succintă a capitolelor, prezentarea temei, prezentarea contribuției proprii, respectiv a rezultatelor originale și menționarea (dacă este cazul) a sesiunii de comunicări unde a fost prezentată sau a revistei unde a fost publicată.

# Chapter 2

## Background

### 2.1 Techniques/Algorithms used

Techniques/algorithms used in programming and training chess engines.

### 2.2 State of the art chess engines

Stockfish, AlphaZero etc. - overview and AI techniques used in them

# Chapter 3

## Methodology

Description of the approach taken to build the chess engine

Explanation of the AI techniques used and why they were chosen

### 3.1 Training

Algorithms/techniques used for training the engine

#### 3.1.1 Min-max algorithm

Used to search for best move to a given depth

### 3.2 Optimizing

Algorithms/techniques used for optimizing the engine

#### 3.2.1 Alpha-Beta pruning

Used to detect and cut off branches that will lead to worse results than the ones already analyzed



# Chapter 4

## Technologies

Details of the programming languages, libraries, and tools used

### 4.1 Chess game

Description of tools used in building the chess game - Unity, C#

### 4.2 Chess engine

Description of tools used in building the chess engine - Python

# Chapter 5

## Results and evaluation

Description of the testing methodology used

Analysis of the results obtained

Comparison with existing chess engines

Evaluation of the strengths and weaknesses of the chess engine

# Chapter 6

## Conclusions

Summary of the main findings and contributions of the thesis

Discussion of potential future improvements to the chess engine

# Bibliography