

Kierunek: **Informatyka algorytmiczna (INA)**

**PRACA DYPLOMOWA**  
**INŻYNIERSKA**

**Bot dla gry w Szachy**

**Bot for Chess game**

Krzysztof Wiśniewski

Opiekun pracy  
**dr Maciej Gębala, prof. uczelni**

Słowa kluczowe: szachy, bot, teoria gier

# Streszczenie

Celem pracy jest opracowanie silnika szachowego w języku Java. Program ten zaprojektowano tak, aby analizować i oceniać pozycję na szachownicy, a następnie sugerować najlepszy ruch dla gracza, uwzględniając jego strategiczne i taktyczne aspekty. Interakcja z programem odbywa się za pośrednictwem wiersza poleceń, z wykorzystaniem Uniwersalnego Interfejsu Szachowego. Umożliwia to łatwą integrację z innymi aplikacjami szachowymi oraz pozwala na przeprowadzanie symulacji i analiz działania silnika bez potrzeby aplikowania interfejsu graficznego.

Niniejsza praca inżynierska składa się z trzech części, w których omówiono kolejne etapy pracy nad opracowaniem silnika szachowego. W pierwszej części przedstawiono podstawową wersję programu, która obejmuje generowanie możliwych ruchów zgodnie z zasadami gry w szachy, algorytm minimax wyszukiwania optymalnego ruchu oraz implementację naiwnej heurystyki. W części drugiej zaimplementowano ulepszenia algorytmów wyszukiwania i oceny pozycji, mające na celu zwiększenie efektywności i precyzji silnika. Ostatnia część pracy poświęcona jest zagadnieniom związanym z testowaniem siły silnika. Przeprowadzono analizę wydajności w odniesieniu do różnych jego wersji, uwzględniając testy porównawcze oraz metodologię oceny skuteczności.

**Słowa kluczowe:** szachy, bot, teoria gier

# Abstract

The aim of this thesis is to develop a chess engine in Java. This program is designed to analyze and evaluate position on the chessboard and subsequently suggest the best move for the player, considering its both strategic and tactical aspects. Interaction with the program is conducted via the command line, using the Universal Chess Interface. This allows for easy integration with other chess applications and facilitates the simulation and analysis of the engine's performance without the need to create graphical interface.

This engineering thesis consists of three parts, which discuss the successive stages of developing the chess engine. The first part presents the basic version of the program, which includes generating possible moves according to the rules of chess, the minimax algorithm for searching for the optimal move, and the implementation of a naive heuristic. The second part implements improvements to the search and position evaluation algorithms, aiming to increase the efficiency and precision of the engine. The final part of the thesis is dedicated to issues related to testing the engine's strength. An analysis of performance was conducted with respect to various versions of the engine, including comparative tests and methodology for assessing effectiveness.

**Keywords:** chess, bot, game theory

# Spis treści

<b>1. Wstęp</b>	<b>7</b>
1.1. Wprowadzenie	7
1.2. Cel i zakres	8
1.3. Układ pracy	8
<b>2. Implementacja silnika szachowego</b>	<b>9</b>
2.1. Reprezentacja pozycji	9
2.1.1. Dane wejściowe	9
2.1.2. Reprezentacja szachownicy	10
2.1.3. Reprezentacja stanu	10
2.1.4. Reprezentacja ruchu	10
2.2. Generowanie ruchów	10
2.2.1. Generowanie ruchów pseudolegalnych	10
2.2.2. Generowanie ruchów legalnych	10
2.3. Algorytm wyszukiwania	10
2.3.1. Algorytm minimax	10
2.3.2. Iteracyjne pogłębianie wyszukiwania	10
2.4. Ocena heurystyczna	10
2.4.1. Heurystyka stanu gry	10
2.4.2. Heurystyka liczebności bierek	10
2.5. Uniwersalny Interfejs Szachowy	10
2.6. Zarządzanie czasem	10
<b>3. Ulepszenia dla silnika szachowego</b>	<b>11</b>
3.1. Ulepszenia dla wyszukiwania	11
3.1.1. Biblioteka otwarc	11
3.1.2. Alfa-Beta cięcie	11
3.1.3. Sortowanie ruchów	11
3.1.4. Tabela transpozycji	11
3.2. Ulepszenia dla oceny heurystycznej	11
3.2.1. Tablice figur	11
3.2.2. Ochrona króla	11
3.2.3. Struktura pionów	11
3.2.4. Moment gry	11
3.2.5. Mobilność	12
<b>4. Ocena siły silnika</b>	<b>13</b>
4.1. Porównanie wersji silnika	13
<b>5. Zakończenie</b>	<b>14</b>

# Spis rysunków

# Spis tabel

# Spis listingów

# Rozdział 1

## Wstęp

### 1.1. Wprowadzenie

Początki szachów, wywodzących się z Indii, datuje się na VI wiek. Od tamtych czasów gra królewska znacząco ewoluowała, przemierzając Persję i świat arabski, aż w końcu dotarła do Europy, gdzie stała się cenioną rozgrywką dworską. Dzięki prostocie zasad, a zarazem złożoności strategii, gra ta szybko zyskała zainteresowanie wielu, zarówno profesjonalnych graczy biorących udział w turniejach, jak i amatorów poszukujących intelektualnych wyzwań. Obecnie szachy stały się najpopularniejszą grą planszową w historii ludzkości.

W swojej 1500-letniej historii zmieniały się nie tylko oficjalne zasady gry, jak na przykład wprowadzenie roszady, czy ruchu en-passant, ale także stosowane techniki i strategie, mające zagwarantować zwycięstwo. Szczególne znaczenie, dla rozwoju tych strategii, miał rozwój maszyn liczących w XX wieku, co otworzyło możliwość zautomatyzowania procesu analizy partii szachowych.

Za pioniera w tej dziedzinie uważany jest amerykański matematyk Claude Shannon, który w roku 1950 opublikował pracę o teoretycznych aspektach programowania silników szachowych, opartych o ocenę heurystyczną oraz algorytm min-max. Istotny wkład w rozwój szachowej sztucznej inteligencji miał także ojciec informatyki, Alan Turing, który rok później zaprojektował pierwszy program komputerowy, w pełni zdolny do gry w szachy. Ograniczenia techniczne tamtych czasów nie pozwoliły jednak przetestowania programu na maszynie. Rozegrano parę partii szachowych, w których każdy ruch był obliczany analogowo.

Najstarszy program uruchomiony na komputerze, który pozwalał na przeprowadzenie pełnej rozgrywki, powstał w 1958 roku. Od tamtego momentu wiele silników szachowych biło rekordy swoich poprzedników. Aż do pamiętnej zimy 1997 roku, kiedy to silnik szachowy DeepBlue wygrał pojedynek  $3\frac{1}{2} - 2\frac{1}{2}$  z ówczesnym mistrzem świata, Garrym Kasparovem.

Po tym wydarzeniu wkroczyliśmy w erę super silników. Wykorzystanie komputerów stanowi nieodłączny element analizy partii szachowych. Maszyny zadziwiają swoją precyzją, znajdując ruchy, których nie potrafią dostrzec najlepsi współcześni gracze. Szachy stały się nie tylko areną dla ludzkiego intelektu, ale także polem testowym zaawansowanych technologii. Zastosowanie najnowocześniejszych rozwiązań, takich jak sieci neuronowe, zrewolucjonizowało sposób, w jaki rozumiemy tę gre oraz pokazało, jak wiele jeszcze można w tej dziedzinie osiągnąć.

Era informatyki jest nie tylko nowym rozdziałem w historii szachów, ale także dowodem na to, jak daleko może zaprowadzić nas współpraca człowieka z maszyną.

## **1.2. Cel i zakres**

## **1.3. Układ pracy**

Układ pracy



## Rozdział 2

# Implementacja silnika szachowego

### 2.1. Reprezentacja pozycji

#### 2.1.1. Dane wejściowe

##### Notacja Forsytha-Edwardsa

Aby móc reprezentować dowolną pozycję szachową, należy w pierwszej kolejności określić, w jakim formacie zostanie ona do programu. Standardem wykorzystywanym w większości silników szachowych jest Notacja Forsytha-Edwardsa (ang. *Forsyth-Edwards Notation*, w skrócie FEN), stworzona specjalnie na potrzeby komputerów. Pozwala ona na jednoznaczne określenie pozycji na szachownicy. Przykład FEN dla pozycji startowej:

`rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1`

Całość informacji, zawartych w jednej linii znaków kodowanych ASCII, ma 6 pól oddzielonych spacjami. Pola te informują, o różnych aspektach danej pozycji:

- biały,
- niebieski,
- czerwony.

##### Szachowa notacja algebraiczna

asd

### **2.1.2. Reprezentacja szachownicy**

Reprezentacja ilościowa

Reprezentacja tablicowa

Reprezentacja bitowa

### **2.1.3. Reprezentacja stanu**

### **2.1.4. Reprezentacja ruchu**

## **2.2. Generowanie ruchów**

### **2.2.1. Generowanie ruchów pseudolegalnych**

Generowanie ruchów króla i skoczka

Generowanie ruchów hetmana, wieży i gońca

Generowanie ruchów piona

### **2.2.2. Generowanie ruchów legalnych**

Technika usuwania ruchów pseudo-legalnych

Perft test

## **2.3. Algorytm wyszukiwania**

### **2.3.1. Algorytm minimax**

Lorem ipsum dolor sit amet, consectetur adipiscing elit

### **2.3.2. Iteracyjne pogłębianie wyszukiwania**

## **2.4. Ocena heurystyczna**

### **2.4.1. Heurystyka stanu gry**

### **2.4.2. Heurystyka liczebności bierek**

## **2.5. Uniwersalny Interfejs Szachowy**

## **2.6. Zarządzanie czasem**

## **Rozdział 3**

# **Ulepszenia dla silnika szachowego**

### **3.1. Ulepszenia dla wyszukiwania**

#### **3.1.1. Biblioteka otwarć**

Biblioteka otwarć

#### **3.1.2. Alfa-Beta cięcie**

Alfa-Beta cięcie

#### **3.1.3. Sortowanie ruchów**

Sortowanie ruchów

#### **3.1.4. Tabela transpozycji**

Tabela transpozycji

### **3.2. Ulepszenia dla oceny heurystycznej**

#### **3.2.1. Tablice figur**

Tablice figur

#### **3.2.2. Ochrona króla**

Ochrona króla

#### **3.2.3. Struktura pionów**

Struktura pionów

#### **3.2.4. Moment gry**

Moment gry

### **3.2.5. Mobilność**

Mobilność

## **Rozdział 4**

# **Ocena siły silnika**

### **4.1. Porównanie wersji silnika**

## **Rozdział 5**

# **Zakończenie**

Zakończenie