

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

1. Wstęp	9
1.1. Wprowadzenie	9
1.2. Cel i zakres	10
1.3. Układ pracy	10
2. Implementacja silnika szachowego	11
2.1. Uniwersalny Interfejs Szachowy	11
2.1.1. Dane wejściowe	11
2.1.2. Komunikacja z interfejsu do silnika	13
2.1.3. Komunikacja z silnika do interfejsu	13
2.1.4. Przykład użycia	13
2.2. Reprezentacja pozycji	13
2.2.1. Reprezentacja szachownicy	13
2.2.2. Reprezentacja stanu	13
2.2.3. Reprezentacja ruchu	13
2.3. Generowanie ruchów	13
2.3.1. Generowanie ruchów pseudolegalnych	13
2.3.2. Generowanie ruchów legalnych	13
2.4. Algorytm wyszukiwania	13
2.4.1. Algorytm minimax	13
2.4.2. Iteracyjne pogłębianie wyszukiwania	13
2.5. Ocena heurystyczna	13
2.5.1. Heurystyka stanu gry	13
2.5.2. Heurystyka liczebności bierek	13
2.6. Zarządzanie czasem	13
3. Ulepszenia dla silnika szachowego	14
3.1. Ulepszenia dla wyszukiwania	14
3.1.1. Biblioteka otwarć	14
3.1.2. Alfa-Beta cięcie	14
3.1.3. Ewaluacja cichych stanów	14
3.1.4. Sortowanie ruchów	14
3.1.5. Tabela transpozycji	14
3.1.6. Okno estymacji	14
3.1.7. Rozszerzanie wyszukiwania	14
3.2. Ulepszenia dla oceny heurystycznej	14
3.2.1. Tablice figur	14
3.2.2. Ochrona króla	15
3.2.3. Struktura pionów	15
3.2.4. Moment gry	15
3.2.5. Mobilność	15

4. Ocena siły silnika	16
4.1. Porównanie wersji silnika	16
5. Zakończenie	17
Bibliografia	18

Spis rysunków

2.1. Przykładowe pozycje szachowe: a) startowa, b) po ruchu e2e4	12
--	----

Spis tabel

Spis listingów

Skróty

UCI (ang. *Universal Chess Interface*) Uniwersalny Interfejs Szachowy

FEN (ang. *Forsyth–Edwards Notation*) Notacja Forsytha-Edwardsa

LAN (ang. *Long Algebraic Notation*) Pełna Algebraiczna Notacja Szachowa

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. Uniwersalny Interfejs Szachowy

Aby umożliwić komunikację między programem a innymi aplikacjami szachowymi należy zdefiniować wspólny protokół wymiany informacji. Standardem, przyjętym nie tylko w silnikach szachowych, ale także w interfejsach graficznych oraz systemach zarządzania rozgrywkami, jest

2.1.1. Dane wejściowe

Notacja Forsytha-Edwardsa

Aby umożliwić komunikację z programem, należy w pierwszej kolejności sprecyzować, w jakim formacie dostarczone zostaną dane wejściowe reprezentujące konkretną pozycję szachową. Standardem, wykorzystywanym nie tylko w większości silników, ale także w pojedynkach rozgrywanych online, jest Notacja Forsytha-Edwardsa (ang. *Forsyth-Edwards Notation*, w skrócie FEN). Jest to format stworzony specjalnie na potrzeby komputerów. FEN pozwala na jednoznaczne określenie pozycji na szachownicy. Przykład notacji dla pozycji startowej:

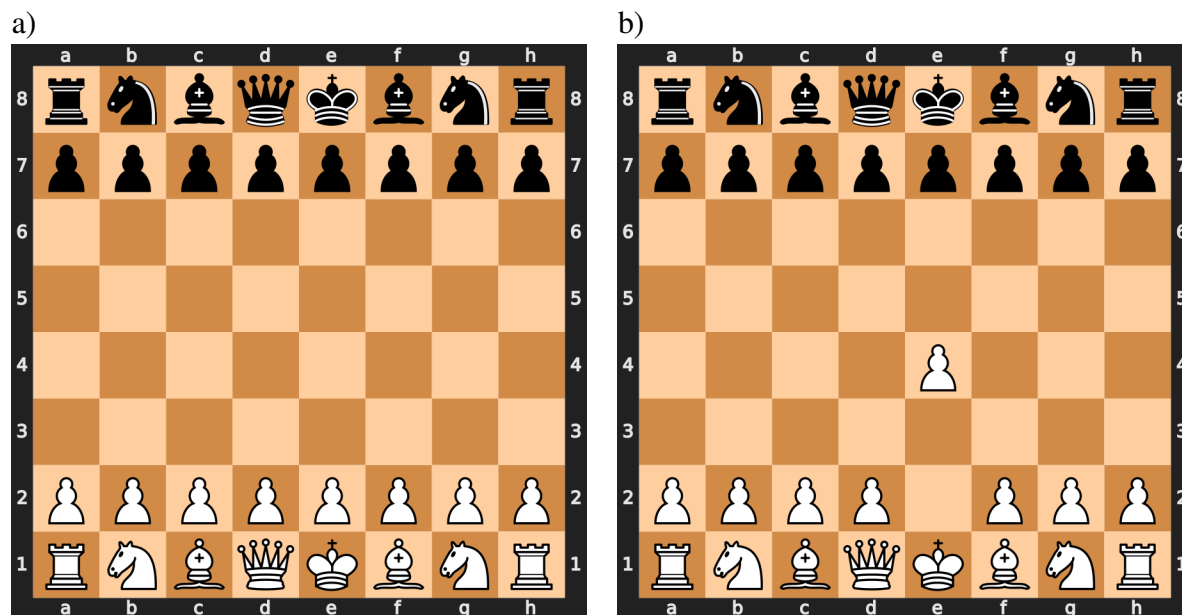
- 2.1 a) rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1
- 2.1 b) rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq e3 0 1

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

1. Reprezentacja 64 pól szachownicy. Każdy z wierszy szachownicy oddzielony jest „/”. Odpowiednio dla białych i czarnych figur.
2. Czyj jest ruch (w — białe, b — czarne)
3. Możliwości roszady (K/k — krótka roszada, Q/q — długa roszada)
4. Możliwości bicia w przelocie, znanego szerzej jako ruch en passant. Sprecyzowane pole jest polem atakowanym. W przypadku braku możliwości bicia w przelocie w polu widnieje „-”.
5. Liczba posunięć od ostatniego bicia bądź ruchu pionem. Istotna w regule 50 posunięć.
6. Liczba pełnych ruchów.

Szachowa notacja algebraiczna

Przykład odniesienia do bibliografii[2, 1]. Udało się?



Rys. 2.1: Przykładowe pozycje szachowe: a) startowa, b) po ruchu e2e4

2.1.2. Komunikacja z interfejsu do silnika

2.1.3. Komunikacja z silnika do interfejsu

2.1.4. Przykład użycia

2.2. Reprezentacja pozycji

2.2.1. Reprezentacja szachownicy

Reprezentacja ilościowa

Reprezentacja tablicowa

Reprezentacja bitowa

2.2.2. Reprezentacja stanu

2.2.3. Reprezentacja ruchu

2.3. Generowanie ruchów

2.3.1. Generowanie ruchów pseudolegalnych

Generowanie ruchów króla i skoczka

Generowanie ruchów hetmana, wieży i gońca

Generowanie ruchów piona

2.3.2. Generowanie ruchów legalnych

Technika usuwania ruchów pseudo-legalnych

Perft test

2.4. Algorytm wyszukiwania

2.4.1. Algorytm minimax

Lorem ipsum dolor sit amet, consectetur adipiscing elit

2.4.2. Iteracyjne pogłębianie wyszukiwania

2.5. Ocena heurystyczna

2.5.1. Heurystyka stanu gry

2.5.2. Heurystyka liczebności bierek

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. Ewaluacja cichych stanów

Ewaluacja cichych stanów

3.1.4. Sortowanie ruchów

Sortowanie ruchów

3.1.5. Tabela transpozycji

Tabela transpozycji

3.1.6. Okno estymacji

Okno estymacji

3.1.7. Rozszerzanie wyszukiwania

Rozszerzanie wyszukiwania

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

Bibliografia

- [1] G. D. Greenwade. “The Comprehensive Tex Archive Network (CTAN)”. W: *TUGBoat* 14.3 (1993), s. 342–351.
- [2] D. E. Knuth. *The T_EXbook*. Addison – Wesley, 1990. ISBN: 0-201-13447-0.

Dodatek A

Instrukcja wdrożenia

To jest dodatek A