Sztuczna Inteligencja

Sprawozdanie

Dzianis Dziurdz

Danila Rubleuski

Mikita Zenevich

1 Wstęp

Celem projektu było stworzenie partii szachów w wersji 2D, w której sztuczna inteligencja walczy się ze sobą.

SI wykorzystuje drzewo Minimax, którego wartości określane są funkcją heurystyczną. Drzewo jest na bieżąco rozszerzane w głąb z każdym następnym ruchem SI.

Gra jest napisana w języku JAVA z pomocą bibliotek:

- JFRAME — obsługa grafiki
- CHESSLIB — dla generowania legalnych ruchów szachowych na podstawie pozycji na szachownicy.

2 Implementacja

Korzystanie z biblioteki CHESSLIB ułatwia nam pobieranie ruchów dostępnych dla szachów i ładowanie ikon szachów.

Stworzyliśmy trzy różne sztuczne intelekty o różnej złożoności heurystyk. Przyjrzyjmy się teraz na te heurystyki.

2.1 Heurystyka

Drzewo Minimax z ograniczoną głębokością wymaga efektywnej heurystyki, która bardzo dobrze rozpozna silne i słabe punkty SI na planszy. Dla nas liczy się tylko wygrana, bądź niedopuszczenie do zwycięstwa innej SI, w zależności od tego, co znajduje się bliżej.

Punkty dodatnie przyznawane są w zależności od liczby pionów, podczas gdy punkty ujemne są przyznawane na podstawie liczby bierek przeciwnika. Następnie punkty są sumowane i możemy w ten sposób wybrać najlepszy ruch, który poprowadzi nas do zwycięstwa lub spróbuje doprowadzić nas do remisu.

Aby uniknąć powtarzających się gier, dodaliśmy ruchy tasowania (shuffle), które może zrobić min max. W takim przypadku, jeśli ma kilka najlepszych opcji do ruchu, wybierze pierwszą opcję z losowych, w przeciwieństwie do wyboru tylko pierwszej, jeśliby nie było randomizacji.

Rozpatrzymy trzy implementacji sztucznej inteligencji.

- Stosując „prosty” minimaks z alfa beta algorytm ten bierze pod uwagę tylko liczbę pionków SI i przeciwnika, nie biorąc pod uwagę ich pozycji na planszy.
- Wykorzystując „inteligentny” minimax z alfa-betą, algorytm ten uwzględnia nie tylko liczbę pionów SI i przeciwnika, ale także przyznaje dodatkowe punkty za różne pozycje na planszy za różne pionki, pozwalając na realizację różnych strategii.
- Trzecia opcja, losowy SI, ten algorytm nie jest minimaxem, ale działa losowo wybierając jeden ruch ze wszystkich dostępnych. Nie liczy figur i nie uwzględnia ich pozycji na planszy.

3 Porównanie wyników

Rozpatrzymy wyniki gry dla wszystkich trzech algorytmów na różnej liczbie gier i różnej głębokości.

Prosty minimaks.

	wygrał	przegrał	remis	głębokość	prosty, głębokość
inteligentny	72	23	5	3	3
inteligentny	39	42	19	3	4
losowy	0	100	0	-	3
losowy	0	87	13	-	1

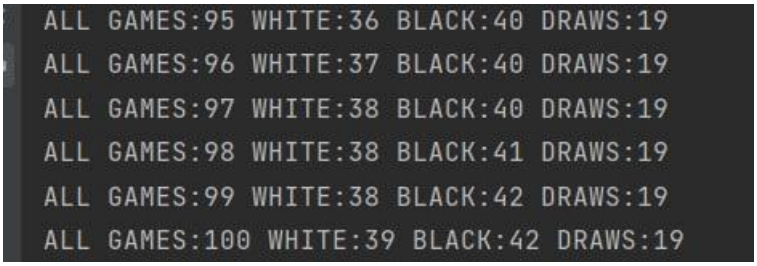
Inteligentny minimaks.

	wygrał	przegrał	remis	głębokość	inteligentny, głębokość
prosty	23	72	5	3	3
prosty	42	39	19	4	3
losowy	100	0	0	-	3
losowy	89	0	11	-	1

Możemy zauważyć, że przy tej samej głębokości algorytm „inteligentny” wygrywa ze znacznie większymi szansami niż dwa pozostałe algorytmy. Można było również zaobserwować, że przy głębokości 3 dla inteligentnego i 4 dla normalnego, inteligentny algorytm trochę traci w stosunku do zwykłego, a to ze względu na to, że zwykły, ze względu na jego świetną głębokość, może rozważać i wykonywać lepsze ruchy, w przeciwieństwie do innych algorytmów.

4 Przykładowy wynik dla prostego i inteligentnego algorytmu

Głębokość prostego 4 i inteligentnego 3.



White – inteligentny

Black - prosty