

Warcaby z AI w Haskellu

Programowanie Funkcyjne 2025L

Autor: Maciej Scheffer 325 222

Temat:

Gra w warcaby z wykorzystaniem sztucznej inteligencji

Temat wybrałem ponieważ jestem entuzjastą szachowym i nieraz miałem już styczność z programowaniem gier na szachownicy, przykładowo szachy w pythonie na PIPR w I semestrze.

Implementacja:

Jako język programowania użyłem Haskellu, ponieważ jest to najpopularniejsze rozwiązanie na jakie natrafiałem podczas przygotowywania się do projektu.

UI

Początkowo projekt zakładał UI w C++, jednak problemy z bibliotekami Haskellu po dniu zmagają odwiodły mnie od tego pomysłu, spróbowałem więc UI w Pythonie z podobnym rezultatem, dlatego ostatecznie interfejs jest w konsoli z podświetlaniem dozwolonych ruchów oraz menu.

AI

Wybrany przeze mnie algorytmem AI jest minimax z odcinaniem alfa-beta, ponieważ dobrze nadaje się do gier w których w prosty sposób można wyciągnąć funkcję heurystyczną oraz przez ograniczoną złożoność gry w warcaby. Maksymalna głębokość którą udało mi się osiągnąć z w miarę płynną rozgrywką to 10. Na tej głębokości algorytm bezproblemowo radzi sobie z warcabowymi zagadkami logicznymi z forów internetowych.

Zasady

Zaimplementowałem zasady klasyczne ponieważ najbardziej były zbliżone do tych które znam, a zatem

- piony poruszają się w jednym kierunku po skosach
- bicie po skosach w przód i w tył
- pion jeżeli zakończy ruch na ostatnim polu staje się damką
- damki poruszają się nieograniczoną liczbę pól oraz
- damki po biciu mogą stanąć nieograniczoną liczbę pól za zbitym pionkiem
- damki mogą bić na odległość
- jeżeli możliwe jest bicie to trzeba je wykonać
- jedyne legalne bicie to te które zbije możliwie największą liczbę pól

Program bierze pod uwagę te zasady udostępniając graczom tyle te ruchy które je spełniają.

Uruchomienie

W celu jak największego uproszczenia projektu, nie wykorzystałem żadnych bibliotek zewnętrznych, a zatem uruchomienie jest możliwe za pomocą komendy **runghc ./Main.hs**

```
AI moves: E8 -> G6

  A B C D E F G H
1 . b . . . b . .
2 . . . . . . . .
3 . . . w . . . b
4 b . . . . . . .
5 . . . . . . . b
6 b . . . . . B .
7 . . . . . . . .
8 . . . . . . . .

Welcome to Minimax Haskell Checkers!
Author: Maciej Scheffer

Choose difficulty level:
1) Easy (depth 2)
2) Medium (depth 3)
3) Hard (depth 5)
4) Extreme (depth 8)
Enter level number: |

  A B C D E F G H
1 . b . b . b . b
2 b . b . b . b .
3 . b . b . b . b
4 . . . . . . . .
5 . . . . . . . .
6 w . w . w . w .
7 . w . w . w . w
8 w . w . w . w .

Available Moves:
A6 -> B5
C6 -> B5
C6 -> D5
E6 -> D5
E6 -> F5
G6 -> F5
G6 -> H5

Enter move (e.g. A3 B4): |
```

Testowanie:

Do testowania użyłem różnych ustawień szachownicy i testuję zachowanie algorytmu sprawdzającego poprawność ruchów razem z algorytmem sztucznej inteligencji.

Wyszukiwanie najdłuższego bicia - w tym przypadku istnieje jedna sekwencja ruchów w których damka zbija aż 9 pionków i kończy sekwencję na polu A6

```
  A B C D E F G H
1 . . . . . . . .
2 . . b . . . b .
3 . . . b . b . .
4 . . . . b . . .
5 . . . b . b . .
6 . . b . . . b .
7 . b . . . . . .
8 . . . . w . . .

Available Moves:
E8 -> A6

Enter move (e.g. A3 B4): |
```

Wymuszenie bicia w ruchu - możliwe do wykonania są tylko te ruchy które prowadzą do bicia pionów

```
  A B C D E F G H
1  . . . . . . . .
2  . . . . . . . .
3  . . . . . . . .
4  . . b . . . . .
5  . w . . w . w .
6  . . . b . . . .
7  . . . . . . . .
8  . . . . . . . .

Available Moves:
B5 -> D3
E5 -> C7

Enter move (e.g. A3 B4):
```

Zagadki warcabowe do testowania AI - wykorzystałem powszechne zagadki z forów internetowych do wytestowania głębi minimax. Po błędzie gracza: ruch białych G7 na F6, w ośmiu ruchach pozycja jest jednostronnie wygrana przez czarne (na lewo zagadka, na prawo pozycja po ośmiu ruchach)

Ruchy: wG6->F6 ; bB4->C5 | wymuszone wD6->B4 ; bD4->E5 | wymuszone wF6->D4 ; B2->C3 | wymuszone B4->C2 lub wymuszone D4->B2 ; wymuszone potrójne bicie C1->A7.

Czarne więc poświęcają 3 pionki aby potem w jednym biciu je odbić i stanąć na polu przed promocją

```
  A B C D E F G H
1  . . b . . . . .
2  . b . . . . . .
3  . . . . . . . .
4  . b . b . . . .
5  . . . . . . . .
6  . w . w . . . .
7  . . . . . w . .
8  . . . w . . . .

Available Moves:
B6 -> A5
B6 -> C5
D6 -> C5
D6 -> E5
G7 -> F6
G7 -> H6
D8 -> C7
D8 -> E7

Enter move (e.g. A3 B4):
```

```
AI moves: C1 -> A7

  A B C D E F G H
1  . . . . . . . .
2  . . . . . . . .
3  . . . . . . . .
4  . . . . . . . .
5  . . . . . . . .
6  . . . . . . . .
7  b . . . . . . .
8  . . . w . . . .

Available Moves:
D8 -> C7
D8 -> E7

Enter move (e.g. A3 B4):
```