

# Sprawozdanie WSI Lab3 Tobiasz Kownacki

---

## 1. Zadanie:

Zaimplementować algorytm min-max z przycinaniem alfa-beta. Algorytm ten należy zastosować do gry w proste warcaby (checkers/draughts). Niech funkcja oceny planszy zwraca różnicę pomiędzy stanem planszy gracza a stanem przeciwnika. Za pion przyznajemy 1 punkt, za damkę 10 p. Proszę nie zapomnieć o znacznej nagrodzie za zwycięstwo.

Zasady gry:

- bicie nie jest wymagane
- dozwolone jest tylko pojedyncze bicie (bez serii)

## Pytania:

1. Czy gracz sterowany przez AI zachowuje się rozsądnie z ludzkiego punktu widzenia? Jeśli nie to co jest nie tak?

Niech komputer gra z komputerem (bez wizualizacji), zmieniamy parametry jednego z oponentów, badamy jak zmiany te wpłyną na liczbę jego wygranych. Należy zbadać wpływ:

2. Głębokości drzewa przeszukiwań.
3. Alternatywnych funkcji oceny stanu (nadal ocena jest różnicą pomiędzy oceną stanu gracza a oceną przeciwnika), np.: a) nagrody jak w wersji podstawowej + nagroda za stopień zwartości grupy (jest dobrze jak wszyscy są blisko siebie lub przy krawędzi planszy) b) za każdy pion na własnej połowie planszy otrzymuje się 5 nagrody, na połowie przeciwnika 7, a za każdą damkę 10. c) za każdy nasz pion otrzymuje się nagrodę w wysokości:  $(5 + \text{numer wiersza, na którym stoi pion})$  (im jest bliżej wroga tym lepiej), a za każdą damkę dodatkowe 10.

---

## 2. Odpowiedzi

### 1. Czy gracz sterowany przez AI zachowuje się rozsądnie z ludzkiego punktu widzenia? Jeśli nie to co jest nie tak?

Gracz sterowany przez AI zachowuję się zazwyczaj rozsądnie. Jednak często komputer posiadając damki nie wykorzystuje ich pełnego potencjału tzn. zostawia je na końcu planszy i używa zwykłych pionków żeby zapolować na ostatnie pionki przeciwnika. Dla człowieka bardziej intuicyjne jest użycie damek w powyższym przypadku, szczególnie przy dużej przewadze nad przeciwnikiem. Komputer również często wykonuje te same ruchy damką w końcowej fazie gry, posiadając w ostatnim rzędzie cały rząd nieruszonych wcześniej swoich pionków.

---

### AI vs AI

### 2. Głębokości drzewa przeszukiwań.

Black \ White Depth	1	2	3	4	5	6
1	<u>Black</u>	<b>White</b>	<b>White</b>	<b>White</b>	<b>White</b>	<b>White</b>
2	<b>White</b>	<u>Draw</u> ( <u>Black</u> ).	<b>White</b>	<b>Draw</b> ( <b>White</b> )	<b>White</b>	<b>White</b>
3	<b>Draw</b> ( <b>White</b> )	<u>Draw</u> ( <u>Black</u> ).	<b>White</b>	<b>Draw</b> ( <b>White</b> )	<b>White</b>	<b>White</b>
4	<u>Draw</u> ( <u>Black</u> ).	<u>Draw</u> ( <u>Black</u> ).	<b>Draw</b> ( <b>White</b> )	<u>Draw</u> ( <u>Black</u> ).	<b>Draw</b> ( <b>White</b> )	<b>White</b>
5	<u>Black</u>	<u>Draw</u> ( <u>Black</u> ).	<b>White</b>	<u>Black</u>	<b>White</b>	<b>White</b>
6	<u>Black</u>	<u>Draw</u> ( <u>Black</u> ).	<u>Draw</u> ( <u>Black</u> ).	<u>Black</u>	<u>Black</u>	<u>Draw</u> ( <u>Black</u> ).

Wynik Draw(X) oznacza zapętlenie się algorytmów, ale gracz X ma większą wartość oceny stanu gry, obliczonej za pomocą basic\_ev\_func.

**Wnioski:** Większa głębokość drzewa przeszukiwań zwiększa szanse na wygraną, ale nie w każdym przypadku. Dla Black\_Depth = 1, gracz biały wygrywa w prawie wszystkich przypadkach. Są jednak przypadki gdy gracz z większą głębokością drzewa przegrywa np. Black\_Depth = 5 i White\_Depth = 3. Patrząc na komórki o jednakowej głębokości dla obu graczy, nie widać wyraźnej przewagi żadnego z nich. Po podliczeniu wyłącznie jednoznacznych wygranych można zauważyć, że gracz biały, rozpoczynając grę jako pierwszy, ma większe szanse na wygraną.

3. Alternatywnych funkcji oceny stanu Parametry:

- Głębokość drzewa przeszukiwań = 5

Black \ White ev_func	Basic	Group Price	Push to opp half	Push forward
Basic	<b>White</b>	<b>Draw (White)</b>	<b>White</b>	<b>Draw (White)</b>
Group Price	<u>Draw</u> ( <u>Black</u> ).	<u>Draw</u> ( <u>Black</u> ).	<b>Draw (White)</b>	<b>Draw (White)</b>
Push to opp half	<u>Draw</u> ( <u>Black</u> ).	<u>Draw</u> ( <u>Black</u> ).	<u>Draw</u> ( <u>Black</u> ).	<u>Draw</u> ( <u>Black</u> ).
Push forward	<b>Draw (White)</b>	<u>Draw</u> ( <u>Black</u> ).	<b>White</b>	<u>Draw</u> ( <u>Black</u> ).

Wynik Draw(X) oznacza zapętlenie się algorytmów, ale gracz X ma większą wartość oceny stanu gry, obliczonej za pomocą basic\_ev\_func.

**Wnioski:** Widać wyraźny wpływ funkcji oceny na szansę na wygraną. **Group Price** jest bardzo defensywną funkcją oceny, która za każdym razem doprowadza do zapętlenia się algorytmu i remisu, ale z wskazaniem częściej na przeciwnika. Każdy z graczy ma taką samą liczbę wygranych gier, ale gracz czarny nie wygrał ani razu jednoznacznie. Białemu udało się to 3 razy. Pokazuje to jak duże znaczenie ma zaczynanie gry. Pomimo zmiany funkcji oceny, algorytm dalej się zapętla i wykonuje te same ruchy. Najlepszą funkcją oceny okazuje

się **Push to opp half**, która nagradza za obecność na połowie przeciwnika. Gracz czarny, korzystając z tej metody, zawsze osiąga remis z przewagą punktową dla siebie. Z kolei biały jednoznacznie wygrywa w dwóch przypadkach, raz remisuje z przewagą punktową (grając przeciwko defensywnej funkcji **Group Price**) i raz przegrywa, gdy przeciwnik również korzysta z funkcji **Push to opp half**.