

# Sztuczna inteligencja i inżynieria wiedzy

## Laboratorium

### Ćwiczenie 3. Algorytmy rozwiązywania gier

Opracowanie: J. Bielaniewicz, P. Miłkowski

#### Cel ćwiczenia

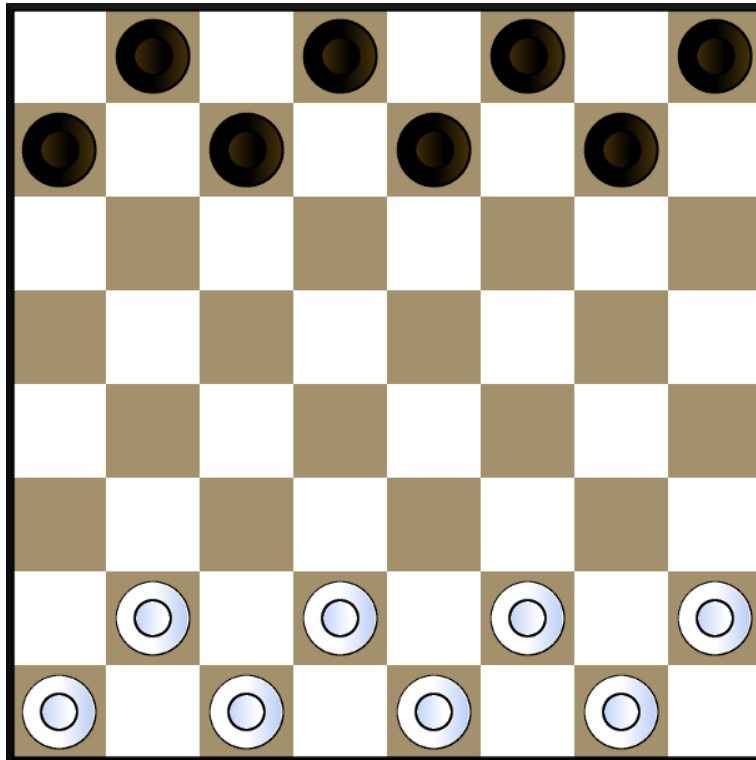
Zapoznanie się z głównymi pojęciami gier – gra i rozgrywka, stan i przestrzeń stanów gry, strategia oraz drzewo i węzły drzewa gry, wypłata, przeszukiwanie drzewa gry, racjonalne działanie graczy jako podmiotów decyzyjnych za pomocą własnej implementacji dwuosobowej gry logiczno-strategicznej Warcaby i zbadanie właściwości wykonanej implementacji tej gry.

#### Realizacja ćwiczenia

- Zapoznanie się z głównymi pojęciami gier – gra i partia gry, stan i przestrzeń stanów gry, strategia oraz drzewo i węzły drzewa gry, bicie, przeszukiwanie drzewa gry, racjonalne działanie graczy jako podmiotów decyzyjnych, na przykładzie dwuosobowej planszowej gry logiczno-strategicznej Warcaby.
- Sformułowanie funkcji oceny stanu planszy dla której – im większa wartość tej funkcji, tym lepiej dla jednego gracza a gorzej dla przeciwnika.
- Dobór strategii optymalnej gracza – gracz powinien dobierać swoją strategię tak, by w danym momencie gry zmaksymalizować swój zysk, czyli zminimalizować stratę, mając zarazem świadomość tego, jak może postąpić przeciwnik.
- Zapoznanie się z algorytmami znajdowania rozwiązań gier o sumie zerowej – sprawiedliwych (*Zero-sum game solving algorithms*), czyli optymalnych strategii gier, w szczególności z algorytmami *min-max* oraz *alfa-beta*.
- Wybór i opracowanie mechanizmu ustalania parametrów rozgrywki, zadawania ruchów do wykonania przez kontroler (i silnik), wizualizacji stanu planszy, oraz ogłaszania wyniku rozgrywki.
- Implementacja silnika gry z więzami ruchów na planszy.
- Samodzielna implementacja algorytmów *min-max* oraz *alfa-beta* z funkcją oceny stanu planszy.
- Implementacja kontrolera gry w trybie rozgrywki AI przeciw człowiekowi (do testowania działania silnika gry i oceny działania kontrolera), z odpowiednią wizualizacją stanów planszy oraz ogłaszania wyników rozgrywki. Testy funkcjonalne kontrolera gry i omówienie wyników testów.
- Uzupełnienie implementacji kontrolera gry do celu rozgrywek w trybie AI przeciw AI (do badań wydajności kontrolera), z odpowiednią wizualizacją stanów planszy oraz ogłaszania wyników rozgrywki.
- (Opcjonalne!) Przygotowanie interfejsu graficznego GUI wraz z możliwością grania człowiek-AI.
- Sformułowanie dla gry 2 do 3 heurystyk i/lub ich kombinacji „ważonych” dla funkcji oceny stanów planszy w grze. Implementacja heurystyk w kontrolerach gry z algorytmem *min-max* *alfa-beta* cięć.
- Sformułowanie dla gry 2 do 3 heurystyk i/lub ich kombinacji „ważonych” dla funkcji oceny stanów planszy w grze. Implementacja heurystyk w kontrolerach gry z algorytmem *min-max* *alfa-beta* cięć.
- Zbadanie czasów przetwarzania i ilości ruchów graczy w rozgrywkach w grze dla implementacji kontrolera z algorytmem *min-max* oraz *alfa-beta* w trybie pracy AI przeciw AI.
- Zbadanie wpływu zastosowanych heurystyk na czas przetwarzania przez kontroler z algorytmem *min-max* i przez kontroler z algorytmem *alfa-beta*.

- Prezentacja najciekawszych (zdaniem studenta) wyników.
- Dyskusja otrzymanych wyników.
- Przygotowanie sprawozdania (przyrostowo, z poszczególnych etapów realizacji ćwiczenia) zawierającego powyższe punkty.

### Gra warcaby



Plansza Warcabów 8x8

## Zasady gry

W ramach implementacji gry Warcaby, należy posłużyć się zasadami gry według Polskiego Związku Warcabowego:

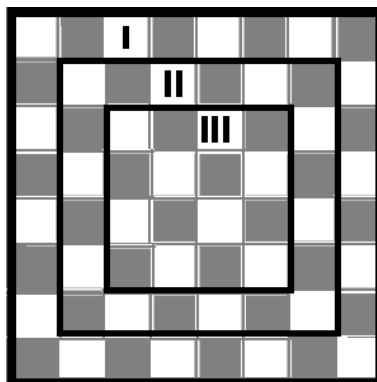
1. Gra toczy się pomiędzy dwoma graczami.
2. Plansza składa się z 64 równych pól.
3. Gra toczy się jedynie na ciemnych polach.
4. Przesunięcie pionka czy damki jest ruchem.
5. Pierwszy ruch wykonuje gracz z białymi pionkami.
6. Nie można nie wykonać ruchu w swojej turze.
7. Ruch pionkiem można wykonywać tylko do przodu, po przekątnej, na wolne miejsca.
8. Gdy pionek osiągnie przeciwny kraniec planszy w stosunku do krańca początkowego to staje się damką.
9. Damki poruszają się po zarówno do przodu jak i do tyłu, po przekątnych na dowolne wolne pole.
10. Jeśli gracz może w swojej turze wykonać bicie, to musi wybrać ruch będący biciem.
11. Bicia można wykonywać do przodu i do tyłu.
12. W przypadku wielokrotnego bicia należy wykonać bicie o największej wartości pól wchodzących w obszar bicia.
13. Jeśli podczas bicia pionek odwiedzi pole krańcowe to nie staje się damką.
14. Gra jest wygrana jeśli przeciwnik nie posiada już żadnych pionków lub jeśli jego pionki nie mogą wykonać żadnego ruchu.
15. Gra kończy się remisem jeśli przez 15 tur ruchów damkami nie nastąpi koniec gry.

## Ocena stanu gry

W celu ewaluacji stanu planszy należy zaproponować taką metodę zliczania pionków i damek posiadanych przez obie strony, która pozwoli liczbowo ocenić jakość pozycji. Typowo przyjmuje się, że wartości dodatnie oznaczają przewagę białych, a ujemne - czarnych (zero oznacza pozycję równą, zbalansowaną); oczywiście większe wartości oznaczają większą przewagę.

Należy zaproponować co najmniej dwie metody (*heurystyki*) oceny stanu planszy, zbadać je i porównać.

Zliczanie wcale nie musi (i nie powinno) mieć charakteru absolutnego (tzn. 5 pionków vs 4 pionki = +1) – możemy wykorzystać intuicję, że posiadanie pionków w różnych „strefach” planszy jest mniej lub bardziej korzystne. Przykładowo, można podzielić planszę na 3 różnie ważone strefy wg poniższego schematu:



## Algorytm min-max i algorytm cięć alfa-beta, zapis w pseudokodzie

Algorytmy *min-max* oraz *alfa-beta* cięć są podstawą działania kontrolerów gier logiczno-strategicznych oraz realizacji rozgrywek w tych grach. Implementują one poruszanie się w przestrzeni stanów gry realizując strategię wykonywania ruchów w drzewie gry **z uwzględnieniem racjonalnego zachowania się graczy**. Zbadanie kilku ważnych cech własnych implementacji kontrolera gry z algorytmami *min-max* oraz *alfa-beta* oraz usprawnienie działania kontrolera przez zastosowanie kilku heurystyk jest podstawowym celem ćwiczenia 3.

Algorytmy *min-max* oraz *alfa-beta* należy gruntownie przestudiować posiłkując się materiałami z wykładów [1] i [3]. Podstawowym źródłem są materiały z wykładu [1]. Źródło [3] to prezentacja video z transkrypcją wykładu na Massachusetts Institute of Technology (MIT). Jest dobrym dodatkiem do materiałów [1].

Bardzo rzetelny opis algorytmów *min-max* oraz *alfa-beta* i szkice fragmentów ich zapisów w pseudokodzie można też znaleźć w podręczniku akademickim [2]. Zalecane jest utrwalenie wiadomości z wykładu o grach przez przejrzenie rozdziału 5 Adversarial Search (str. 180), punkty: 5.1 Games omawiający gry i pojęcia z nimi związane, 5.2 Decyzje optymalne w grach, algorytm *min-max*, oraz cięcia alpha-beta, oraz ich pseudokody, 5.3 Imperfect Real-Time Decision o decyzjach redukcji głębokości przeszukiwania przestrzeni stanów gry, funkcji oceny stanów w przestrzeni stanów, oraz heurystycznej funkcji oceny stanów.

## Podział realizacji zadania na etapy

**Zajęcia 1** – Trening w rozgrywkach w *warcaby*. Analiza gry – cel, zasady, logika, stan i przestrzeń stanów gry, rozgrywka i notacja przebiegu rozgrywki, funkcja oceniająca. Algorytmy *min-max* i *alfa-beta* w pseudokodzie.

Wybór i opracowanie mechanizmu ustalania parametrów rozgrywki, zadawania ruchów do wykonania przez silnik, wizualizacji stanu planszy, oraz ogłaszania wyniku rozgrywki. Implementacja silnika gry z kontrolą spełniania reguł wykonywania ruchów na planszy. **Za realizację: max. 1pkt**

Dodanie do silnika trybu pracy z algorytmem *min-max* oraz funkcji oceniającej stanu planszy. Testy funkcjonalne kontrolera gry z algorytmem *min-max* z funkcją oceny stanu planszy. **Za realizację: max. 2pkt**

**Zajęcia 2** - Uzupełnienie implementacji kontrolera gry do celu rozgrywek w trybie AI vs AI (do badań wydajności kontrolera), z odpowiednią wizualizacją stanów planszy oraz ogłaszania wyników rozgrywki.

Implementacja interfejsu człowiek – AI w celu prezentacji działania aplikacji.

**Za realizację: max. 2pkt**

**Zajęcia 3** – Dodanie do kontrolera gry trybu pracy z algorytmem *alfa-beta* cięć. Testy funkcjonalne kontrolera z algorytmem *alfa-beta*. **Za realizację: max. 1pkt**

Dodanie do kontrolera losowości wykonywania pierwszego ruchu gracza rozpoczynającego rozgrywkę. Zbadanie średnich czasów przetwarzania i liczby ruchów gracza wygrywającego w rozgrywkach w grze dla implementacji kontrolera z algorytmem *min-max* oraz *alfa-beta* w trybie pracy AI przeciw AI, dla kilku/kilkunastu/kilkudziesięciu rozgrywek. **Za realizację: max. 2pkt**

Interfejs graficzny GUI wraz z funkcjonalnością gry człowiek – AI. **Za realizację: max. 2 pkt dodatkowe.**

**Zajęcia 4** – Sformułowanie dla gry 2 do 3 heurystyk i/lub ich kombinacji „ważonych” dla funkcji oceny stanów planszy w grze. Implementacja heurystyk w kontrolerze gry z algorytmem *min-max* oraz *alfa-beta* cięć. Testy wydajnościowe kontrolerów z algorytmem *min-max* oraz *alfa-beta* z różnymi heurystykami. pomiary czasów przetwarzania i ilości ruchów gracza wygrywającego.

Opracowanie dokumentacji porównawczej z badań wydajności (czasów przetwarzania i ilości ruchów gracza wygrywającego) w trybie działania AI vs AI z algorytmem *min-max* vs *alfa-beta*. Badania powinny ukazać, jak różne heurystyki i głębokość przeglądania drzewa gry wpływa na skuteczność gracza. Na podstawie wykonanych badań zaproponowanie graczy AI dla różnych poziomów trudności gry.

**Za realizację: max. 2pkt**

## Literatura

1. Prof. Halina Kwaśnicka: Materiały do wykładu “Sztuczna inteligencja i inżynieria wiedzy”, edycja 2022.
2. Stuart J. Russell, Peter Norvig, Ernest Davis, Artificial Intelligence: A Modern Approach, 3rd ed., Prentice Hall Series in Artificial Intelligence, Boston, USA.  
<https://faculty.psau.edu.sa/filedownload/doc-7-pdf-a154ffbcec538a4161a406abf62f5b76-original.pdf>
3. Courses, Electrical Engineering and Computer Science, Artificial Intelligence, Lecture Videos, Lecture 6: Search: Games, Minimax, and Alpha-Beta, Instructor: Patrick H. Winston.  
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/lecture-videos/lecture-6-search-games-minimax-and-alpha-beta/>
4. A Knowledge-based Approach of Connect-Four. The Game is Solved (1988).  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.38.2778&rep=rep1&type=pdf>