

Sztuczne inteligencja i inżynieria wiedzy

Laboratorium

Ćwiczenie 3. Algorytmy rozwiązywania gier o sumie zerowej

Opracowanie: Jacek Gruber

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 *Connect 4* 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, wypłata, przeszukiwanie drzewa gry, racjonalne działanie graczy jako podmiotów decyzyjnych, na przykładzie dwuosobowej planszowej gry logiczno-strategicznej *Connect 4*.
- 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.
- 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.

- 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 Connect 4

Status gry: rozwiązana w 1988 roku. W grze optymalnej graczy zawsze wygrywa gracz rozpoczynający. Gra ma stosunkowo małą przestrzeń stanów, rzędu 10^{15} , oraz małą złożoność drzewa gry, rzędu 10^{21} .

Gra *Connect Four* (1974) posiada stosunkowo niską ocenę Geek Rating 1.21/5 na BoardGameGeek (platforma gier online i zasobów dla społeczności graczy) [3]. Mimo niskiego współczynnika ratingowego gra jest wciąż bardzo popularna. Występuje ona pod bardzo wieloma (ponad 58) nazwami, z których najczęściej spotykane to: *Connect Four*, *Captain's Mistress* (wykonanie ekskluzywne), *Four Up*, *Plot Four*, *Find Four*, *Fourplay*, *Four in Row*, *Four in a Line*, *Connect 4*, *Connect-4*. Dla gry będziemy używać różnych nazw, starając się zachowywać te przyjęte w cytowanych w instrukcji publikacjach naukowych, oraz nazw niektórych realizacji produktowych gry.

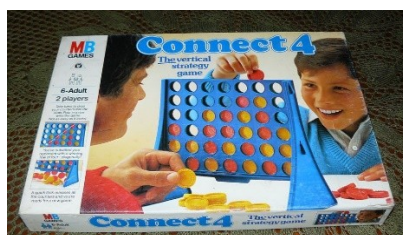
Autorami gry *Connect Four* są Ned Strongin oraz Howard Wexler. Gra ma status publikacyjny Public Domain i posiada ponad 72 realizacje produktowe.

Connect 4 jest bardzo znaną pionową grą podobną do najprostszych planszowych gier strategicznych z ustawianiem bez przestawiania pionów – *Tic Tac Toe* (*Kółko i Krzyżyk*) oraz *Go Moku*. Plansza jest umieszczana pionowo na stojaku, a gracze upuszczają do planszy pionów w postaci żetonów. Ilustrują to rysunki 1 do 3. Każdy z graczy ma pionów swojego koloru czarnego i białego (w realizacjach produktowych najczęściej czerwone i żółte). Plansza posiada na górnej krawędzi 7 slotów ponumerowanych od literami od A (lewy slot) do G. Każdy slot ma miejsce na wrzucenie łącznie 6 pionów gracza i przeciwnika lokowanych w pionowych rzędach planszy ponumerowanych cyframi od 1 (dół) do 6 (góra). Zatem pionowa szachownica ma rozmiar 7 na 6 pól. Numeracja umożliwia zapisywanie przebiegu rozgrywki gracza i przeciwnika żetonami czarnymi i białymi. Plansza z oznaczeniami slotów i pionowych wierszy jest przedstawiona na rysunku 3. Poszczególne miejsca w slotach są widoczne przez otwory o nieco mniejszej średnicy niż pion. W każdej turze zawodnicy (gracz i przeciwnik) mogą wrzucać swoje pionów do tych samych lub do innych slotów, na przemian po jednym żetonie.

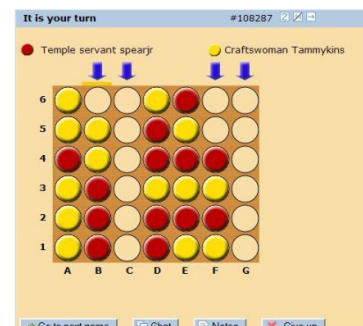
Gra toczy się dopóki jeden z graczy nie zdobędzie ułożenia 4 pionów w rzędzie – poziomo, pionowo lub po przekątnej, co jest celem gry.



Rysunek 1



Rysunek 2



Rysunek 3

Rysunek 1 Źródło: <https://boardgamegeek.com/image/614060/connect-four>

Zasady gry *Connect 4*, jej pobieżną analizę matematyczną i najważniejsze heurystyki gry przedstawiono w prezentacji [4]. W wielu miejscach w Internecie są dostępne różne implementacje i wersje online gry *Connect 4*. Pozwala to szybko wytrenować się w tej grze, ocenić jakość implementacji w rozgrywkach treningowych i kompletność funkcjonalności, oraz ocenić implementację interfejsu użytkownika GUI. Można utworzyć konto i zalogować się na stronie <https://boardgamegeek.com/boardgame/2719/connect-four>, aby grać z graczami ze społeczności.

Inne przykładowe wersje online gry *Connect 4* są osiągalne pod adresami <https://c4arena.com/>, <https://www.mathsisfun.com/games/connect4.html>, oraz <https://www.helpfulgames.com/subjects/brain-training/connect-four.html>. Można tam grać w trybie gracz przeciw graczowi, oraz gracz przeciw AI, z różnymi stopniami trudności.

Ze strony

<https://www.microsoft.com/en-us/p/connect-4-hd-free/9nblgggz60xd?activetab=pivot:overviewtab>

można pobrać i zainstalować na komputerze lub na urządzeniu mobilnym bezpłatną wersję gry *Connect 4* (*Connect-4 HD Free*).

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] oraz [5]. Podstawowym źródłem są materiały z wykładu [1]. Źródło [5] 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.

Przykłady heurystyk i wzorowych rozgrywek w *Connect 4*, analiza gry

Bardzo interesująca jest praca [6] dotycząca koncepcji zastosowania bazy wiedzy z regułami ruchów i heurystykami w rozgrywkach w *Connect-Four*. W pracy przedstawiono przykłady efektywnych heurystyk a także przykłady rozgrywek „perfect” gry w *Connect-Four*. Najważniejsze heurystyki można wychwycić w prezentacji [4].

Bardzo wartościową publikacją na temat heurystyk w grach *Connect-K* jest pozycja [7].

Podział realizacji zadania na etapy

Zajęcia 1 – Trening w rozgrywkach w *Connect 4*. Analiza gry – cel, zasady, logika, stan i przestrzeń stanów gry, rozgrzywka 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: 0 – 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: 0 – 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: 0 – 2pkt

Zajęcia 3 – Dodanie do kontrolera gry trybu pracy z algorytmem *alfa-beta* cięć. Testy funkcjonalne kontrolera z algorytmem *alfa-beta*. **Za realizację: 0 – 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: 0 – 2pkt**

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. **Za realizację: 0 – 3pkt**

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. **Za realizację: 0 – 1pkt**

Ocena ćwiczenia – max 12pkt

0 – 1 pkt	Trening w rozgrywkach w Connect 4. Analiza gry – cel, zasady, logika, stan i przestrzeń stanów gry, rozgrywka i notacja przebiegu rozgrywki, funkcja oceniająca. Algorytmy <i>min-max</i> i <i>alfa-beta</i> 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 kontrola spełniania reguł wykonywania ruchów na planszy. Dokumentacja – kod i syntetyczne sprawozdanie.
0 – 2 pkt	Dodanie do silnika trybu pracy z algorytmem <i>min-max</i> oraz funkcji oceniającej stanu planszy. Testy funkcjonalne kontrolera gry z algorytmem <i>min-max</i> z funkcją oceny stanu planszy. Dokumentacja – kod i zbiór odpowiednich widoków stanów planszy i/lub ich sekwencji. Syntetyczne sprawozdanie.
0 – 1 pkt	Dodanie do kontrolera gry trybu pracy z algorytmem <i>alfa-beta</i> cięć. Testy funkcjonalne kontrolera z algorytmem <i>alfa-beta</i> . Dokumentacja – kod i zbiór odpowiednich widoków stanów planszy i/lub ich sekwencji. Syntetyczne sprawozdanie.
0 – 2 pkt	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. Dodanie do kontrolera losowości wykonywania pierwszego ruchu gracza rozpoczynającego rozgrywkę. Zbadanie średnich czasów przetwarzania i ilości ruchów gracza wygrywającego w rozgrywkach w grze dla implementacji kontrolera z algorytmem <i>min-max</i> oraz <i>alfa-beta</i> w trybie pracy AI przeciw AI, dla kilku/kilkunastu/kilkudziesięciu rozgrywek. Dokumentacja wyników badań w postaci odpowiednich zestawień i/lub wykresów. Syntetyczne sprawozdanie.
0 – 2 pkt	Interfejs graficzny GUI wraz z funkcjonalnością gry człowiek – AI.

0 – 3 pkt	<p>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 <i>min-max</i> oraz <i>alfa-beta</i> cięć. Testy wydajnościowe kontrolerów z algorytmem <i>min-max</i> oraz <i>alfa-beta</i> z różnymi heurystykami. Pomiary czasów przetwarzania i ilości ruchów gracza wygrywającego.</p> <p>Dokumentacja testów w postaci odpowiednich zestawień i/lub wykresów. Syntetyczne sprawozdanie.</p>
0 – 1 pkt	<p>Opracowanie dokumentacji porównawczej z badań wydajności (czasów przetwarzania i liczby ruchów gracza wygrywającego) w trybie działania AI vs AI z algorytmem <i>min-max</i> vs <i>alfa-beta</i>. Badania powinny ukazać, jak różne heurystyki i głębokość przeglądania drzewa gry wpływa na skuteczność gracza.</p> <p>Sprawozdanie z wykorzystaniem wyników uzyskanych poprzednio.</p>

Materiały pomocnicze i literatura

1. Prezentacja i notatki z wykładu “Sztuczna inteligencja i inżynieria wiedzy”, edycja 2020.
2. Stuart J. Russell, Peter Norvig, Ernest Davis, Artificial Intelligence: A Modern Approach, 3rd ed., Prentice Hall Series in Artificial Intelligence, Boston, USA. Podręcznik akademicki.
<https://faculty.psau.edu.sa/filedownload/doc-7-pdf-a154ffbcec538a4161a406abf62f5b76-original.pdf>
3. Platforma BoardGameGeek. <https://boardgamegeek.com/boardgame/2719/connect-four>
4. Prezentacja video. <https://boardgamegeek.com/video/37287/connect-four/connect-four-numberphile-mathematical-analysis>
5. 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/>
6. 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>
7. Heuristics in the game of Connect-K.
http://inside.mines.edu/~huawang/CSCI404_Projects/Project2/connectk.pdf