

Dokumentacja zadania laboratoryjnego -  
algorytm minimax w grze w Kółko i krzyżyk

Krzysztof Wyrzykowski, nr indeksu 331455

27 listopada 2024

## 0.1 Implementacja Gry

W swoim rozwiązaniu zastosowałem 4 klasy: Game, State, Player oraz Display.

Klasa Player reprezentuje gracza i przechowuje imię oraz głębokość przeszukiwania drzewa gry. Posiada metodą `make_move()`, która na podstawie stanu gry zwraca ruch gracza, korzystając z algorytmu minimax dla zadanej głębokości.

Klasa State przechowuje informacje o stanie gry. W szczególności zawiera heurystykę, która ocenia obecny jej stan.

Klasa Game rozpoczyna rozgrywkę i kontroluje jej przebieg. Jej implementacja pozwala zarówno na korzystanie z klasy Player jak i innej klasy implementującej jej interfejs. Możliwe jest również użycie klasy gracza która będzie pobierała ruch od użytkownika. Klasa Game nie wyświetla przebiegu gry, ale do jej konstruktora można podać klasę implementującą interfejs klasy Display, którą może wyświetlać przebieg.

## 0.2 Przyjęta heurystyka

Przyjąłem heurystykę, która ocenia stan gry na podstawie typu pól na których znajdują się znaki X oraz O, które w obliczeniach są reprezentowane odpowiednio jako 1 i -1, a puste pola mają wartość 0. Poszczególnym typom pól przyporządkowałem wartości:

X	O		1	-1	0
O	X	X	-1	1	1
		O	0	0	-1

Rysunek 1: Reprezentacja stanu gry

- pole środkowe - 4
- pole w rogu - 3
- pole na krawędzi - 2

3	2	3
2	4	2
3	2	3

Rysunek 2: Wartości pól

Wartość heurystyki to suma iloczynów wartości pola oraz wartości znaku znajdującego się na nim.

3	-2	0
-2	4	2
0	0	-3

Rysunek 3: Stan gry z uwzględnieniem wartości pól

Dla tego stanu gry wartość heurystyki wynosi:  $3 - 2 + 0 - 2 + 4 + 2 + 0 + 0 - 3 = 2$ . Ponadto, aby uniknąć sytuacji w której gracz wygra grę ale ocena sytuacji końcowej jest gorsza niż ocena innej sytuacji, która nie prowadzi do zwycięstwa tego gracza, takie sytuacji przydzielam wartość 100 (lub -100 dla drugiego gracza).

O		X
O	X	
O		X

Rysunek 4: Stan gry o wartości -100

### 0.3 Badanie wpływu głębokości przeszukiwania na wynik gry

Zebrałem wyniki gier dla różnych głębokości przeszukiwania na próbie 1000 rozgrywek i wyniki zestawilem w tabeli poniżej.

Rozkład wyników na 1000 rozegranych gier (wygrane X/remisy/wygrane O)					
X\O	1	2	3	4	5
1	355/0/645	0/0/1000	0/0/1000	0/0/1000	0/0/1000
2	1000/0/0	0/1000/0	0/1000/0	0/1000/0	0/1000/0
3	1000/0/0	0/950/50	0/938/62	0/945/55	0/937/63
4	1000/0/0	0/1000/0	0/1000/0	0/1000/0	0/1000/0
5	1000/0/0	0/1000/0	0/1000/0	0/1000/0	0/1000/0

Tabela 1: Rozkład wyników na 1000 rozegranych gier, w zależności od głębokości przeszukiwania.

Na podstawie zebranych danych można zaobserwować, że:

- jeśli głębokość przeszukiwania obu graczy jest większa od 1 to gra w większości przypadków kończy się remisem
- jeśli jeden z graczy ma głębokość przeszukiwania równą jeden, a drugi gracz większą od jeden to wygrywa gracz z większą głębokością

Nieoczekiwanie dla obu głębokości równych 1, gra w 2/3 przypadków kończy się zwycięstwem drugiego gracza. Wynika to z tego, że w sytuacji przedstawionej poniżej, gracz X wybiera pole w rogu, które ma największą heurystykę, ponieważ nie wie, że w następnym ruchu gracz O wygra.

O		O
	X	
		X

Rysunek 5: Głębokość X = 1, głębokość O = 1

Drugim nieoczekiwanym wynikiem jest zwycięstwo gracza O, kiedy głębokość przeszukiwania gracza X jest równa 3. Wynika to z tego, że w sytuacji przedstawionej na rysunku 6. zagranie przez gracza X pola będącego krawędzią rozwija się do sytuacji takiej jak przedstawiona na rysunku 7, a zagranie pola będącego w rogu daje sytuację która jest przedstawiona na rysunku 8. Obie te sytuacje mają wartość heurystyczną równą 4, dlatego gracz wybiera dowolną z nich.

	X	
O		

Rysunek 6: Gracz O wybiera pole w rogu

		X
X	X	O
O		O

Rysunek 7: Gracz X w 3. ruchu wybrał pole będące krawędzią

X	O	
	X	
O	X	O

Rysunek 8: Gracz X w 3. ruchu wybrał pole będące krawędzią

Natomiast jedną z sytuacji do których może doprowadzić zagranie przez X pola będącego krawędzią jest sytuacja na rysunku 9. W tej sytuacji wygrywa gracz O, lecz nie jest tego w stanie przewidzieć gracz którego głębokość przeszukiwania wynosi 3. Szansa na zaistnienie tej sytuacji wynosi  $2/7 * 1/6 = 1/21$ , co zgadza się z danymi przedstawionymi w tabeli.

X		
X	X	O
O		O

Rysunek 9: Niezależnie od ruchu X wygrywa O

## Wnioski

Przyjęta heurystyka nie jest idealna. Prowadzi do sytuacji, w której zwiększenie poziomu przeszukiwania może prowadzić do znalezienia gorszego rozwiązania. Pomijając zachowanie algorytmu dla głębokości 3, algorytm prowadzi do remisu dla głębokości większej bądź równej 2, czyli do wyniku, który zawsze osiągną gracz podejmując optymalne decyzje.