Ćwiczenie nr 3
Wprowadzenie do sztucznej inteligencji

Autor
Maciej Prostak

1 Wprowadzenie

Zadanie polega na napisaniu algorytmu minimax, który zostanie użyty do wykonywania odpowiednich ruchów w grze "Wilki i Owce". Zostaną sprawdzone trzy przypadki: wilk porusza się przy użyciu algorytmu, a owce mają losowe ruchy (minimax vs random); wilk porusza się w sposób losowy i owce poruszając się za pomocą algorytmu (random vs minimax); wilk i owce poruszają się za pomocą algorytmu (minimax vs minimax). Następnie sprawdzę jak głębokość przeszukiwania drzewa wpłynie na wyniki działania algorytmu.

1.1 Technologia

Rozwiązanie zostało zaimplementowane w języku Python (3.8.0) z wykorzystaniem bibliotek pygame (2.0.3). Dodatkowo dla poprawy czytelności kodu zostały dołączone biblioteki black (21.9b0), flake8 (4.0.1) oraz isort (5.9.3).

1.2 Wizualizacja gry

Gra odbywa się na szachownicy 8x8. Wilk jest oznaczony kolorem czerwonym, natomiast kolorem zielonym oznaczone zostały owce.

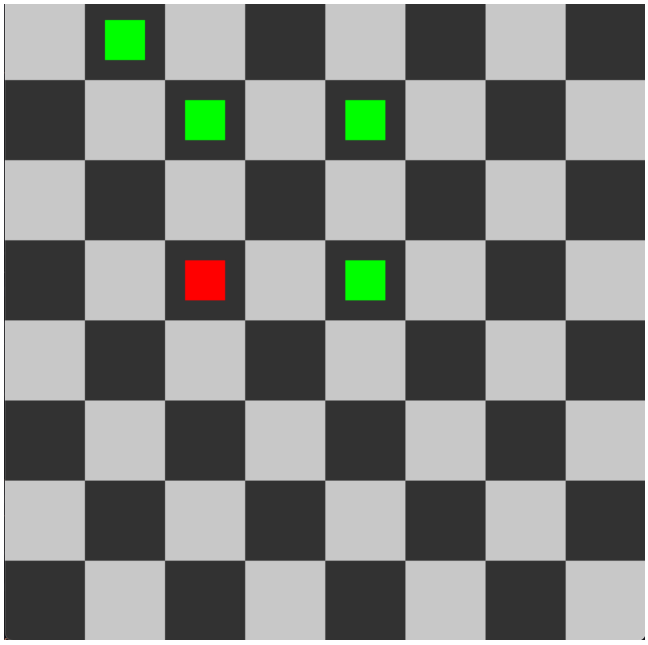

Fig. 1 Wizualizacja gry w wilki i owce. Owce oznaczone są kolorem zielonym a wilk kolorem czerwonym.

2 Wyniki

2.1 Minimax vs Random

Przeprowadzone zostało łącznie 100 prób, w których wilk używa algorytmu minimax, a owce poruszają się w sposób losowy. Wilk przy każdej próbie był losowo umieszczany na którymś z czarnych pól znajdujących się na dole szachownicy. Głębokość drzewa w algorytmie minimax dla wilka odpowiednio zostały zwiększane. Jeżeli jednak dla głębokości=1 wilk wygrywał za każdym razem, można było się spodziewać jeszcze lepszych wyników dla głębszego drzewa. Warto wspomnieć, że w wybranej grze nie istnieją remisy i zawsze któraś strona musi wygrać.

Wyniki umieszczone w tabeli (Tabela 1) pokazują, że jeżeli owce grają w sposób losowy, nie były w stanie wygrać ani jednej gry.

Ilość prób	Głębokość drzewa	Wygrana wilka	Wygrana owiec
25	1	25	0
25	2	25	0
25	4	25	0
25	6	25	0

Tabela 1 Wyniki gry "Wilki i Owce" (minimax vs random)

2.2 Random vs Minimax

W przypadku gdy owce używają algorytmu minimax i wilk porusza się w sposób losowy, widzimy, że znaczącą przewagę mają owce. Przy głębokości d=1 oraz d=2, wilk jednak uzyskuje jakieś wygrane. Również w grze gdzie owce mają dość rozbudowane drzewo wilkowi udało się wygrać jedną grę. Może wynikać to z funkcji heurystycznej owiec, którą na pewno mogłaby lepiej określać stan gry.

Ilość prób	Głębokość drzewa	Wygrana wilka	Wygrana owiec
25	1	11	14
25	2	5	20
25	4	0	25
25	6	1	24

Tabela 2 Wyniki gry "Wilki i Owce" (random vs minimax)

2.3 Minimax vs Minimax

Wykonano 110 prób (Tabela 2), w którym modyfikowane zostały parametry d1 (głębokość drzewa wilka) oraz d2 (głębokość drzewa owiec) podając końcowe wyniki ilości wygranych gier dla wilka oraz dla owiec. Można zauważyć, że wraz z równoczesnym zwiększaniem obydwu wartości d1 oraz d2 owce zaczynały wygrywać więcej gier. Widać jednak, że mimo wszystko przewagę cały czas ma Wilk w grach, w których wartość d1 = d2. Dopiero w grach gdzie d1=2, d2=6 zauważymy przewagę owiec, ale jednak nie dominuje gry. Może to wynikać z zaimplementowanej funkcji heurystycznej dla owiec, która mogłaby być ulepszona.

Ilość prób	d1 (Wilk)	d2 (Owce)	Wygrana wilka	Wygrana owiec
30	2	2	29	1
30	4	4	25	5
25	6	6	15	10
25	2	6	10	15

Tabela 3 Wyniki gry "Wilki i Owce" (minimax vs minimax), gdzie d1=głębokość drzewa (wilka), d2=głębokość drzewa (owce).

2.4 Wartości minimax dla różnych pozycji

Dla wszystkich podanych w tym podrozdziale przypadków wysokość drzewa minimax wynosi 6 dla owiec i wilka. Najlepszy wynik minimax dla ruchu wilka w tym przypadku (Fig. 2) to wartość 1700. Dla mojej zaimplementowanej funkcji heurystycznych oznacza ona, że wykonany ruch daje absolutną pewność wygrania.

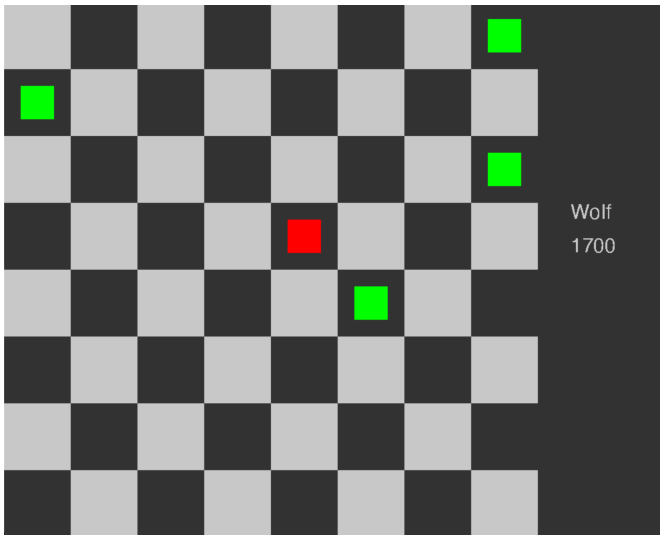

Fig. 2 Wizualizacja gry w wilki i owce z najlepszą wartością minimax.

Dla przypadku na rysunku (Fig. 3) widzimy, że owce posiadają dużą przewagę i mają dużą szansę na wygraną.

Wartość -1300 widoczną na rysunku (Fig. 4) dla owiec oznacza dla nich wygraną.

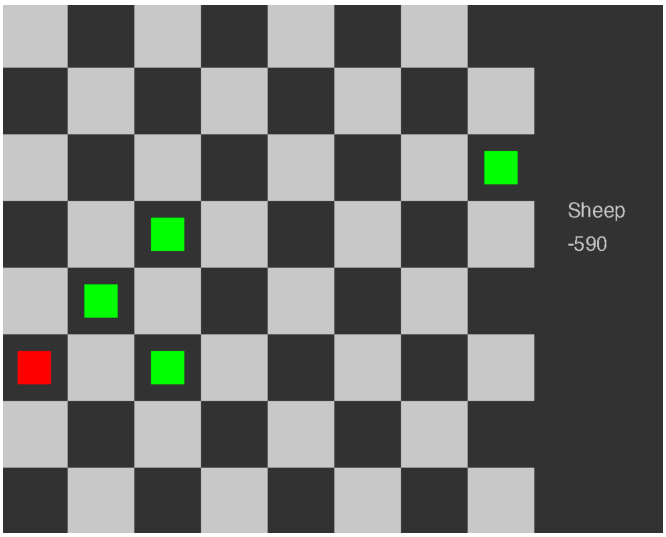

Fig. 3 Wizualizacja gry w wilki i owce z najlepszą wartością minimax.

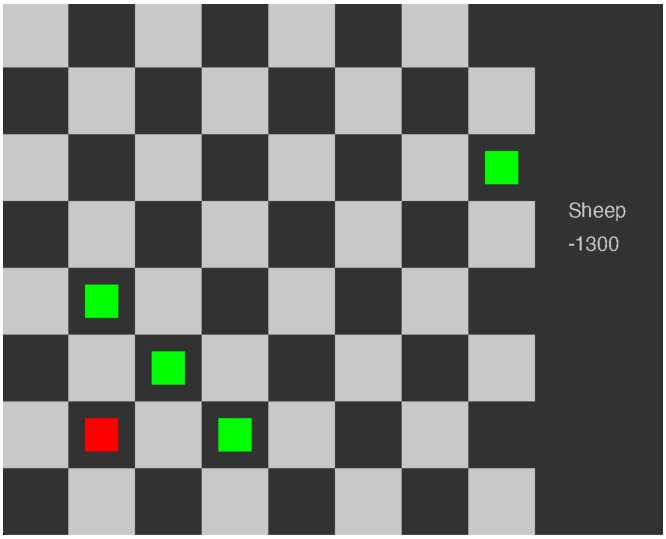

Fig. 4 Wizualizacja gry w wilki i owce z najlepszą wartością minimax.

Jako, że głębokość drzewa d=6, to przewidujemy 3 ruchy wilka do przodu. Widzimy, że w najlepszym przypadku po 3 ruchach wilka nie osiągniemy jego wygranej, ale znajdziemy się w 1 rzędzie (rząd 0 oznacza wygraną wilka) i wartość funkcji heurystycznej rzeczywiście wyniesie wartość 600. Ten przypadek potwierdza poprawnie wykonany ruch wilka.

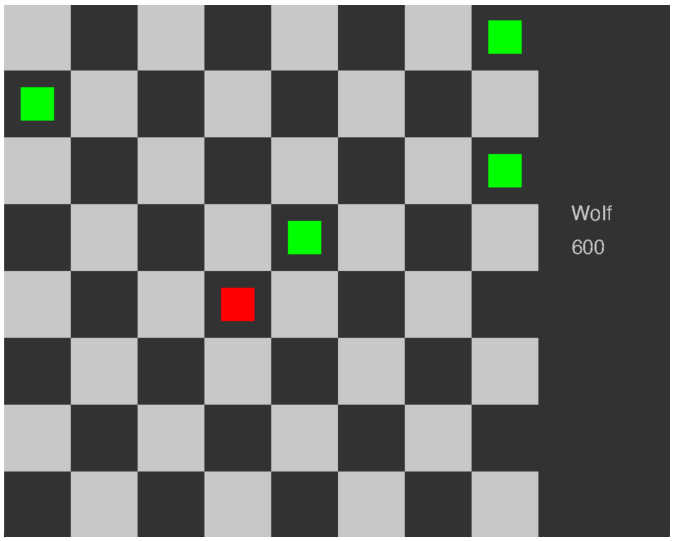

Fig. 5 Wizualizacja gry w wilki i owce z najlepszą wartością minimax.

3 Wnioski

Zaimplementowany algorytm mimimax oraz przedstawione wyniki pokazują, że budowane drzewa przeszukiwań potrafią wybierać najlepsze ruchy dla graczy. W moim przypadku maksymalna wysokość drzewa równała się 6, jako że dla większej wartości program działał bardzo wolno i pobierał dużą ilość pamięci. Należałoby zaimplementować dodatkowo do algorytmu minimax mechanizmu alfa-beta, która mogłaby znacznie przyspieszyć działanie oraz zmniejszyć użycie pamięci komputera.

W moich zaimplementowanych funkcjach heurystycznych wilka i owcy widzimy, że jeżeli dajemy równą szansę obydwu graczom, faworyzowany jest wilk, który w większej ilości przypadków wygrywa. Istnieje możliwość, że przy lepszej funkcji heurystycznej dla owcy wynik mógłby być znacznie inny. Poprawa wyniku owiec mogłaby również się pojawić jeżeli budowane drzewo byłoby zdecydowanie większe.