

WSI

Ćwiczenie 3

RAPORT

1. Opis zadania

Należało napisać skrypt, który buduje drzewo zadanej gry (dwuosobowej gry deterministycznej – w moim przypadku „Wilk i owce”), a następnie gra sam ze sobą w sposób losowy (losuje wynik) i odrzuca ścieżki prowadzące do przegranej.

2. Zasady gry „Wilk i owce”

Jeden z graczy dysponuje czterema pionami białymi (owcami), drugi dysponuje jednym pionem czarnym (wilkiem). Pozycja początkowa wygląda następująco: owce obsadzają pierwszy rząd szachownicy, wilk staje na jednym dowolnym wybranym polu krańcowego rzędu szachownicy.

Celem gracza-wilka jest przedostanie się na pierwszy rząd pół szachownicy. Gracz dysponujący owcami wygrywa, gdy uda mu się zablokować wilka tak, żeby nie mógł on wykonać posunięcia.

Grę rozpoczyna gracz grający wilkiem. W jednym ruchu może przejść o jedno pole po przekątnej w dowolną stronę.

Piony gracza grającego owcami w kolejnych posunięciach przechodzą także o jedno pole po przekątnej, lecz tylko do przodu. Wilk może cofać się, owce nie mogą się cofać.

Źródło: wikipedia

3. Opis działania programu

Gracze sterowani przez komputer kolejno za pomocą algorytmu min-max budują drzewo gry dla zadanej głębokości. Jeśli istnieje kilka ścieżek, które zostały tak samo dobrze ocenione, to gracz dokonuje losowego wyboru najlepszej ścieżki. Wilk zawsze wybiera ścieżki najbardziej optymalne, a owce w swoim ruchu mają element losowości, gdyż może zdarzyć się (z zadaniem prawdopodobieństwem), że owce dokonają ruchu nieoptymalnego.

Po każdym ruchu wypisywane są nowe pozycje graczy. Na koniec, na podstawie historii ruchów, w konsoli wyrysowywane są kolejne stany gry w postaci szachownicy z pionami.

4. Opis heurystyki

Wilk – max, owce – min

Ocena pozycji wilka:

- im bliżej 8 rzędu, tym więcej punktów (np. 5 rząd = 5p, 6 rząd = 6p, itd.)

- im więcej możliwości ruchu na danym polu, tym więcej punktów (3p za możliwość ruchu do przodu, 1p za możliwość ruchu do tyłu; np. jeśli wilk na danej pozycji ma możliwość ruszyć się w każdym kierunku to otrzymuje łącznie 8p)

Ocena pozycji owiec:

- jeśli wszystkie owce w 1 rzędzie -> -4p, 3 owce -> -3p, 2 owce -> -2p

- jeśli owca blokuje ruch wilka -> -3p dla ruchu wilka do przodu, -1p dla ruchu wilka do tyłu

- następuje ocena odległości owcy od wilka, gdzie $\text{dist} = \text{abs}(\text{wolf_pos.x} - \text{sheep.x}) + \text{abs}(\text{wolf_pos.y} - \text{sheep.y})$, a punkty przyznawane są ze słownika $\text{points} = \{2: -12, 4: -10, 6: -8, 8: -6, 10: -4, 12: -2, 14: 0\}$, gdzie kluczem jest dystans, a wartością liczba punktów (im bliżej tym lepiej; owca jest min)

Ocena stanu gry to suma punktów uzyskanych z powyższych punktów

5. Przykładowe wykonanie programu (dla głębokości 4)

- Cała historia dostępna w plikach tekstowych dołączonych wraz z raportem i programem

a) Wygrana owiec (pozycja startowa i końcowa)

```
[ '.', 'S', '.', 'S', '.', 'S', '.', 'S']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
['W', '.', '.', '.', '.', '.', '.', '.']
```

```
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', 'S', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', 'S', '.', 'S']
[ '.', '.', '.', '.', 'S', '.', 'W', '.']
```

b) Wygrana wilka (pozycja startowa i końcowa)

```
[ '.', 'S', '.', 'S', '.', 'S', '.', 'S']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
['W', '.', '.', '.', '.', '.', '.', '.']
```

```
[ '.', '.', '.', 'W', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', 'S', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ '.', '.', 'S', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '.']
[ 'S', '.', 'S', '.', '.', '.', '.', '.']
```

