

Wprowadzenie do sztucznej inteligencji

Laboratorium - lista nr 3

Termin oddania: piąte pełne laboratorium

Termin przysyłania przez MS Teams na konkurs: 10 czerwca, godzina 23:59.

Zadanie 1

Rozważmy następującą modyfikację gry w kółko i krzyżyk:

- gra jest na planszy 5×5 ,
- wygrywa gracz, który pierwszy ustawi cztery swoje symbole w jednej linii bez przerw,
- jeśli jednak gracz ustawi nieprzerwaną linię tylko z trzech swoich symboli, to przegrywa (o ile ostatni ruch nie ustawił jakiejś czwórki).

W załączonym pliku labor3.zip znajduje się prosta implementacja serwera gry, klienta dla gracza i losowego bota. Protokół gry przewiduje:

1. połączenie się klientów z serwerem (serwer obsługuje tylko jedną grę z dwoma graczami),
2. oczekiwanie przez klienta na komunikat z serwera 700, na który należy odpowiedzieć numerem gracza (1 dla X lub 2 dla O) oraz nazwą gracza (w tej samej linii),
3. gracz 1 otrzymuje następnie komunikat 600, że może rozpocząć grę,
4. gracze na przemian wysyłają do serwera swoje ruchy, oznaczające wybrane pole według poniższej tabeli

11	12	13	14	15
12	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

5. serwer po sprawdzeniu ruchu gracza przesyła go do przeciwnika (jeśli nie powoduje on końca gry) lub przesyła najpierw przeciwnikowi a następnie graczowi komunikat o końcu gry zgodnie z poniższymi regułami
 - 1xx - wygrał gracz, który otrzymał ten komunikat (xx=00 jeśli było to w wyniku jego ruchu, w p.p. xx to ruch przeciwnika powodujący koniec gry),
 - 2xx - przegrał gracz, który otrzymał ten komunikat (xx=00 jeśli było to w wyniku jego ruchu, w p.p. xx to ruch przeciwnika powodujący koniec gry),
 - 3xx - zremisował gracz, który otrzymał ten komunikat (xx=00 jeśli było to w wyniku jego ruchu, w p.p. xx to ruch przeciwnika powodujący koniec gry),
 - 400 - wygrana z powodu błędu przeciwnika,
 - 500 - przegrana z powodu własnego błędu.

Napisz program klienta gry implementujący strategię minimax, który jako parametry przyjmuje po kolei:

1. adres ip serwera,
2. numer portu serwera,
3. numer gracza (1 lub 2),
4. nazwę gracza (napis do 9 znaków),
5. głębokość przeszukiwania dla algorytmu minimax (od 1 do 10).

Program powinien działać z linii poleceń z podanymi parametrami, tylko w trybie tekstowym.

W kodzie programu w komentarzach powinien być szczegółowy opis użytej funkcji oceny heurystycznej.

W programie dozwolona jest biblioteka `otwarć` działająca jednak tylko do ruchu nieprzekraczającego podaną głębokość przeszukiwań. Także losowy wybór ruchu w przypadku jednakowej wartości funkcji oceny heurystycznej jest dozwolony.

Kryteria oceny:

- Program poprawnie gra z użyciem serwera i ma zaimplementowany algorytm minimax - ocena 3.0.
- Program ma zaimplementowane α - β -cięcia - ocena zwiększona o 0.5.
- Program został wysłany do wykładowcy przez MS Teams (zadanie na stronie wykładu) do 10 czerwca oraz grał poprawnie w konkursie i wygrał
 - mniej niż 25% gier - ocena zwiększona o 0.5,
 - co najmniej 25% gier - ocena zwiększona o 1.0,
 - co najmniej 50% gier - ocena zwiększona o 1.5,
 - co najmniej 75% gier - ocena zwiększona o 2.0,

W przypadku stwierdzenia gry nieodpowiadającej podanym parametrom (głębokość przeszukania) zadanie dostanie karę -2.0.

Wysyłany do wykładowcy program powinien być w postaci pliku zip, nazwanym numerem indeksu studenta, zawierać tylko konieczne pliki źródłowe oraz plik `ReadMe` zawierający dane autora oraz sposób kompilacji i uruchamiania programu (a także narzędzi i bibliotek użytych w programie). Programy będą uruchamiane w systemie Ubuntu automatycznie więc rodzaj i kolejność parametrów muszą być identyczne ze specyfikacją.

Ostateczna ocena przysłanych programów zostanie podana 17 czerwca.