Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Projekt 3

Imię i nazwisko: Jakub Zając

Nr indeksu: 259362

Dane prowadzącego: Marta Emirsajłow

Termin zajęć: Poniedziałek 15:15

1. Wprowadzanie

Zadanie projektowe polegało na stworzeniu prostej gry pozwalającej na rozgrywkę między komputerem a graczem. Komputer do ustalenia swojego najlepszego ruchu w danej pozycji używa metod sztucznej inteligencji – algorytmu mini-max wspomaganego poprzez cięcia alfa-beta. Do opcji gry należało dodać zmienną wielkość planszy oraz możliwą do ustalenia długość ciągu znaków wygrywających.

2. Opis stworzonej gry

W celu wykonaniu zadania stworzyłem klasę *Game* stanowiącą najważniejszy element programu. Jej głównym elementem jest tablica board[][] przechowująca aktualny stan planszy. Do obsługi gry wykorzystałem wiele funkcji pomocniczych:

- *display_board()* wyświetla sformatowaną tablicę w terminalu zawierającą aktualny stan planszy

- *play()* pozwala umieścić podany znak na konkretne pole planszy. Pobiera jako parametry znak i indeksy pola, zwraca informację o powodzeniu operacji

- *evaluate()* sprawdza czy któraś ze stron wygrała i zwraca odpowiednią wartość. Funkcja sprawdza planszę poziomo, pionowo oraz po przekątnych malejących i rosnących

- *is_draw()* sprawdza czy sytuacja na planszy jest patowa

- *mini_max()* główna funkcja realizująca algorytm mini-max.

Na podstawie tych funkcji stworzyłem funkcje główne, pozwalające na wykonanie ruchu gracza, komputera oraz funkcję sprawdzającą wynik gry.

3. Opisy technik SI

Najważniejszym elementem i zarazem klu zadania było zastosowanie algorytmu mającego naśladować wykonywanie inteligentnych decyzji. Technika wykorzystywana w programie to algorytm MiniMax, mający za zadanie przeanalizować sytuację na planszy kilka/kilkanaście kroków w przód i za pomocą ewaluacji planszy wybrać ruch, którego ścieżka jest najbardziej obiecująca i maksymalizuje szanse na wygranie jednego z uczestników. Analiza zaczyna się od maksymalizacji wyniku komputera, następnie z każdą kolejną rekurencją role się obracają – na przemian algorytm minimalizuje ocenę przeciwnika i następnie maksymalizuje ocenę komputera.

Problemem algorytmu jest jego bardzo niekorzystna złożoność obliczeniowa, co czyni go bardzo wolnym dla większych niż standardowa plansz. Dodatkową techniką pozwalającą nam na zredukowanie czasu działania algorytmu są cięcia alfa-beta. Dzięki nim nie musimy przeglądać każdego możliwego rozwinięcia gry, a ograniczamy się tylko do tych, które należą do okna wywołania funkcji. Za każdym razem dla gracza maksymalizującego wynik staramy się ograniczyć przedział parametrem beta, a dla minimalizującego parametrem alfa. Parametry te są przekazywane w parametrach, tak aby dotarły do każdego poziomu rekurencji.

4. Podsumowanie i wnioski

Algorytm mini-max najbardziej sprawdza się dla planszy gry wymiarów 3x3. Dla niej zawsze znajduje strategię remisującą, a w przypadku błędu gracza zawsze wygrywa. Problemem stają się plansze wyższego rzędu. Nawet po zastosowaniu cięć alfa-beta pierwsze ruchy zajmują dłuższy czas co wymaga ograniczanie poziomów rekurencji.

Techniki sztucznej inteligencji pozwalają nam w prosty sposób rozwiązać bardziej skomplikowane problemy. Kluczem jednak staje się dużo moc obliczeniowa, która potrzebna jest do ich działania dla gorszych przypadków.

Do stworzenia wydajniejszej SI należałoby użyć dodatkowych metod, co znacznie skomplikowałoby program.

5. Literatura

1) https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/
2) https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-3-tic-tac-toe-ai-finding-optimal-move/
3) https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-4-alpha-beta-pruning/
4) https://en.wikipedia.org/wiki/Tic-tac-toe
5) https://en.wikipedia.org/wiki/Minimax
6) https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning
7) Wykłady i prezentacje dr inż. Łukasza Jelenia