

PROJEKTOWANIE ALGORYTMÓW I METODY SZTUCZNEJ INTELIGENCJI

PROJEKT 3

MATEUSZ TKACZ

248 961

1. Wprowadzenie

Projekt polegał na stworzeniu gry i zastosowaniu w niej metod sztucznej inteligencji. Sztuczna inteligencja ma symulować zachowania człowieka. W najprostszych przypadkach jest to algorytm, który analizuje przyszłe możliwości i wybiera optymalne dla siebie, jak np. minimax. Opisywaną grą jest „kółko i krzyżyk”, jednak sztuczną inteligencję można stosować do wielu innych gier.

2. Opis gry

Gra posiada opcję wybierania rozmiaru planszy (kwadratowej) oraz wyboru ile znaków w linii jest wymaganych do wygranej. Jednak na te opcje zostały naniesione pewne ograniczenia – minimalny rozmiar planszy to 3x3, gdyż w innym wypadku wygrywa zawsze ten kto zaczyna. Dodatkowo liczba znaków potrzebnych do wygranej z dołu jest ograniczona do 3, a z góry do rozmiaru planszy. Grę rozpoczyna gracz, kolejny ruch należy do komputera. Gracz ma też możliwość wyboru, czy chce posługiwać się „X” czy „O”.

3. Wykorzystane techniki SI

Podstawą SI w wykonanym projekcie jest algorytm minimax. Jego działanie polega na tym, że tworzy on drzewo przyszłych stanów gry i analizuje je pod kątem tego, w jakim stopniu dany ruch wpłynie na sytuację w dalszym etapie gry. Składa się on z kilku elementów. Pierwszy przypisuje danemu stanowi jakąś wartość, a kolejne dwa służą do wyboru bardziej opłacalnego ruchu. Jeden maksymalizuje zysk gracza, a drugi go minimalizuje. Maksymalna wartość oznacza zysk komputera, a minimalna zysk jego przeciwnika. W pewnym sensie komputer sam „rozgrywa” wszystkie możliwe opcje, a następnie wybiera ruch, który prowadzi do najbardziej opłacalnego dla niego stanu. W programie wywoływanie algorytmu zaimplementowane jest w sposób rekurencyjny.

Działanie algorytmu można również optymalizować. W tym przypadku wykorzystywana jest maksymalna głębokość wywołań rekurencyjnych algorytmu co znacznie poprawia szybkość działania algorytmu w przypadku plansz o większych rozmiarach. Ma to związek z tym, że liczba wszystkich elementów drzewa jest równa $n!$, gdzie n – liczba wolnych pól na planszy. Jednak w tym wypadku algorytm może podejmować mniej optymalne decyzje. Gdy głębokość rekurencji jest ograniczona, algorytm minimax, wykona się tylko określoną liczbę razy, co będzie odpowiadało bliskiej przyszłości w grze, więc SI może wykonywać ruchy, które w dalszej przyszłości okażą się mniej optymalne.

Kolejnym sposobem na zaoszczędzenie czasu jest wykorzystanie cięć alfa beta. Usprawniają one przeszukiwanie drzewa przez wykorzystanie dwóch wielkości: alfa – która reprezentuje najlepszą wartość dla MAX, oraz beta, która reprezentuje najlepszą wartość dla MIN. Cięcia te polegają na tym, że jeśli w jednym fragmencie drzewa znajdzie się bardziej optymalna wartość dla jednego z członów, to nie ma sensu przeszukiwać

dalszej części drzewa, gdyż wiadomo, że i tak zostanie wybrana gałąź z bardziej optymalną wartością.

Dodatkowo do wartości danego ruchu dodawana jest głębokość rekurencji (głębokość zadawana jest na początku i zmniejsza się do 0). Sprawia to, że wartość ruchu staje się większa gdy wygrana następuje szybciej. Analogicznie jest w przypadku przegranej – mniejsza wartość gdy porażka następuje wcześniej.

4. Podsumowanie

W trakcie pisania projektu było wyraźnie widać, jak duży wpływ ma rozmiar planszy na czas wykonywania algorytmu. Można było też zauważyć jaki wpływ miały usprawnienia algorytmu minimax na czas jego wykonywania. Najskuteczniejsze okazały się cięcia alfa beta, które skróciły czasy oczekiwania z rzędu minut do sekund. Głębokość rekurencji nie była aż tak skuteczna jednak wyraźnie było widać poprawę czasu działania programu.

5. Bibliografia

- 1) <https://www.youtube.com/watch?v=l-hh51ncgDI&> [dostęp : 22.05.2020r.]
- 2) Materiały z wykładu
- 3) <https://stackoverflow.com/questions/1403150/how-do-you-dynamically-allocate-a-matrix> [dostęp: 22.05.2020r.]