# Sprawozdanie z zadania 3. Aleksander Pizoń

## Wstęp

Gra w kółko i krzyżyk (Tic-Tac-Toe) to klasyczna gra dwuosobowa o sumie zerowej o pełnej informacji. Podczas wykonywania zadania zaimplementowałem algorytm MinMax do gry w kółko i krzyżyk. Algorytm ten jest jednym z podstawowych algorytmów stosowanych w grach które wymagają podejmowania decyzji w oparciu o przewidywania ruchów przeciwnika. Celem implementacji jest stworzenie programu zdolnego do podejmowania optymalnych ruchów w grze tj. takich które prowadzą do zwycięstwa lub remisu.

## Algorytm MinMax

Algorytm MinMax jest używany w grach dwuosobowych o sumie zerowej, gdzie jedna strona stara się maksymalizować wynik, a druga minimalizować. Działa on na zasadzie przeszukiwania drzewa stanów gry, gdzie węzłami są stany gry, a krawędziami są możliwe ruchy. Algorytm przeszukuje drzewo stanów, maksymalizując potencjalny wynik dla gracza maksymalizującego i minimalizując potencjalny wynik dla gracza minimalizującego. Algorytm zwraca optymalny ruch dla aktualnego stanu gry, zakładając że przeciwnik również wybiera optymalne ruchy.

Przycinanie alfa-beta

Przycinanie alfa-beta to technika optymalizacyjna wykorzystywana w algorytmie MinMax do redukcji liczby węzłów przeszukiwanego drzewa stanów gry. Technika ta polega na eliminowaniu niepotrzebnych węzłów w drzewie stanów poprzez przycinanie gałęzi, które nie wpłynęłyby na ostateczny wynik decyzji. W algorytmie MinMax, jeżli wartość beta (najlepszy wynik przeciwnika) w węźle jest mniejsza niż alfa (najlepszy wynik gracza maksymalizującego) w innych fragmencie drzewa możemy pominąć przeszukiwanie tego fragmentu, ponieważ przeciwnik nie wybierze tej ścieżki. Przycinanie alfa-beta znacząco redukuje czas potrzebny na przeszukanie drzewa stanów gry, co prowadzi do znacznego przyspieszenia działania algorytmu MinMax.

## Opisz programu

W pliku main.py znajduje się część odpowiedzialna za wywoływaniu gry oraz eksperymentów. Pobiera ona parametry z pliku parameters.py i wywołuje eksperymenty znajdujące się w pliku experiments.py. W pliku tools.py znajdują się pomocnicze funkcje potrzebne do przeprowadzenia eksperymentów oraz do zapisu danych w nich uzyskanych. W pliku game.py znajduje się klasa TicTacToe w której zaimplementowana jest logika gry w kółko i krzyżyk. W klasie zaimplementowane są takie metody jak: create\_board odpowiedzalna za tworzenie początkowej planszy, ask\_for\_move odpowiedzialna za pytanie graczy o ruchy, eval\_state zwracająca wartość gry w momencie gdy gra została zakończona. W pliku players.py zaimplementowane zostały klasy Player oraz Oponent. Klasa Player przedstawia gracza który swoje ruchy wpisuje poprzez konsolę, a klasa Oponent przedstawia komputer który swoje ruchy wybiera wywołując algorytm MinMax dla obecnego stanu gry. Właściwy algorytm MinMax został zaimplementowany w pliku minmax.py. Znajduje się tam klasa MinMax w której podczas wywołania można określić czy algorytm ma wykorzystywać przycinanie alfa-beta czy nie. W klasie tej zaimplementowana jest metoda eval, która dla zadanego stanu gry i ruchu zwraca wartość -1, 0 lub 1. Na podstawie wartości zwróconych przez metodę eval dla każdego obecnie możliwego ruchu gracz Oponent wybiera optymalny dla siebie ruch w zależności od tego czy jest graczem maksymalizującym czy minimalizującym.

## Eksperymenty

Pierwszy eksperyment miał za zadanie zbadać czas wykonania algorytmu MinMax dla każdego z początkowych ruchów oraz dla określonych ruchów dla trzech wybranych stanów w grze. W ekperymencie zbadano również liczbę odwiedzonych węzłów oraz średnią głębokość drzewa. Podczas każdego pomiaru algorytm był uruchamiany 10 razy a średnia wraz z odchyleniem zapisywane w pliku. W tabeli 1 przedstawiono wyniki punktów początkowych dla algorytmu z wyłączonym przycinaniem alfa-beta. W tabeli 2 znajdują się wyniki otrzymane dla algorytmu z włączonym przycinaniem alfa-beta.

Tabela 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nr. Ruchu | czas [s] | odch. st. [s] | węzły | śr. Głębokość |
| 1. | 0,559 | 0,043 | 59704 | 8,24 |
| 2. | 0,545 | 0,040 | 63904 | 8,28 |
| 3. | 0,484 | 0,022 | 59704 | 8,24 |
| 4. | 0,558 | 0,044 | 63904 | 8,28 |
| 5. | 0,453 | 0,022 | 55504 | 8,19 |
| 6. | 0,514 | 0,022 | 63904 | 8,28 |
| 7. | 0,477 | 0,025 | 59704 | 8,24 |
| 8. | 0,513 | 0,011 | 63904 | 8,28 |
| 9. | 0,482 | 0,017 | 59704 | 8,24 |

Tabela 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nr. Ruchu | czas [s] | odch. st. [s] | węzły | śr. Głębokość |
| 1. | 0,0237 | 0,0021 | 2337 | 7,96 |
| 2. | 0,0293 | 0,0030 | 2868 | 8,10 |
| 3. | 0,0311 | 0,0019 | 3274 | 8,09 |
| 4. | 0,0362 | 0,0054 | 3573 | 8,01 |
| 5. | 0,0214 | 0,0029 | 2315 | 7,92 |
| 6. | 0,0337 | 0,0032 | 3589 | 8,11 |
| 7. | 0,0359 | 0,0021 | 3808 | 7,97 |
| 8. | 0,0466 | 0,0056 | 4980 | 8,22 |
| 9. | 0,0359 | 0,0026 | 3956 | 8,00 |

Poniżej są przedstawione 3 wybrane stany gry.

Stan 1.

|  |  |  |
| --- | --- | --- |
| o | 2 | 3 |
| x | o | 6 |
| 7 | 8 | 9 |

Stan 2.

|  |  |  |
| --- | --- | --- |
| o | 2 | 3 |
| 4 | o | 6 |
| x | 8 | x |

Stan 3.

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | x | 6 |
| o | o | 9 |

W tabeli 3 i 4 znajdują się wyniki otrzymane w wyniku wykonania algorytmu MinMax dla powyższych stanów dla określonych ruchów z odpowiednio wyłączonym oraz włączonym przycinaniem alfa-beta.

Tabela 3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stan | ruch | czas [s] | odch. st. | węzły | śr. głębokość |
| 1. | 9 | 2,53E-03 | 5,10E-04 | 237 | 5,49 |
| 2. | 6 | 3,58E-04 | 4,51E-05 | 37 | 4,35 |
| 3. | 9 | 1,83E-03 | 3,35E-04 | 197 | 5,28 |

Tabela 4.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stan | ruch | czas [s] | odch. st. | węzły | śr. głębokość |
| 1. | 9 | 9,00E-04 | 2,58E-04 | 56 | 5,32 |
| 2. | 6 | 3,92E-04 | 6,96E-05 | 30 | 4,21 |
| 3. | 9 | 4,05E-04 | 6,75E-05 | 48 | 4,91 |

W drugim eksperymencie badano czas wykonywania algorytmu w zależności od postępu w grze. W tabeli 5 oraz 6 zabrano czasy wykonywania algorytmu dla kolejnych ruchów podczas 10 uruchomień gry z odpowiednio wyłączonym oraz włączonym przycinaniem alfa-beta.

Tabela 5.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| postęp | czas [s] | | | | | | | | | | śr. czas [s] | odch. st. [s] |
| 1. | 4,40E+00 | 5,17E+00 | 5,01E+00 | 5,15E+00 | 4,78E+00 | 5,59E+00 | 4,89E+00 | 4,50E+00 | 4,49E+00 | 4,62E+00 | 4,86E+00 | 3,78E-01 |
| 2. | 4,87E-01 | 6,24E-01 | 5,75E-01 | 4,90E-01 | 5,30E-01 | 6,70E-01 | 4,87E-01 | 4,75E-01 | 4,73E-01 | 5,21E-01 | 5,33E-01 | 6,84E-02 |
| 3. | 7,96E-02 | 8,02E-02 | 5,81E-02 | 6,10E-02 | 7,04E-02 | 8,44E-02 | 6,94E-02 | 7,03E-02 | 6,91E-02 | 6,94E-02 | 7,12E-02 | 8,26E-03 |
| 4. | 1,37E-02 | 7,35E-03 | 7,13E-03 | 7,27E-03 | 2,07E-02 | 1,12E-02 | 7,46E-03 | 7,91E-03 | 6,89E-03 | 7,01E-03 | 9,66E-03 | 4,48E-03 |
| 5. | 1,99E-03 | 1,94E-03 | 2,18E-03 | 3,03E-03 | 1,95E-03 | 1,97E-03 | 3,63E-03 | 2,85E-03 | 3,03E-03 | 2,03E-03 | 2,46E-03 | 6,16E-04 |
| 6. | 5,85E-04 | 4,81E-04 | 6,63E-04 | 5,36E-04 | 6,73E-04 | 4,78E-04 | 8,05E-04 | 5,19E-04 | 7,31E-04 | 5,51E-04 | 6,02E-04 | 1,11E-04 |
| 7. | 1,99E-04 | 1,25E-04 | 2,31E-04 | 1,52E-04 | 2,19E-04 | 1,75E-04 | 5,42E-04 | 2,33E-04 | 2,61E-04 | 2,11E-04 | 2,35E-04 | 1,15E-04 |
| 8. | 4,68E-04 | 7,05E-05 | 1,06E-04 | 1,17E-04 | 1,11E-04 | 1,04E-04 | 1,19E-04 | 1,09E-04 | 1,32E-03 | 1,40E-04 | 2,66E-04 | 3,86E-04 |
| 9. | 7,08E-05 | 2,53E-05 | 6,42E-05 | 2,69E-05 | 6,77E-05 | 6,12E-05 | 3,27E-04 | 6,75E-05 | 4,71E-05 | 4,56E-04 | 1,21E-04 | 1,47E-04 |

Tabela 6.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| postęp | czas [s] | | | | | | | | | | śr. czas [s] | och. st. [s] |
| 1. | 2,54E-01 | 2,77E-01 | 2,61E-01 | 2,83E-01 | 2,67E-01 | 3,30E-01 | 2,56E-01 | 2,80E-01 | 2,62E-01 | 2,78E-01 | 2,75E-01 | 2,19E-02 |
| 2. | 4,86E-02 | 5,89E-02 | 5,08E-02 | 4,93E-02 | 4,71E-02 | 3,60E-02 | 4,57E-02 | 4,70E-02 | 4,53E-02 | 4,72E-02 | 4,76E-02 | 5,65E-03 |
| 3. | 1,47E-02 | 1,41E-02 | 1,72E-02 | 1,38E-02 | 1,40E-02 | 1,68E-02 | 1,49E-02 | 1,41E-02 | 1,40E-02 | 1,41E-02 | 1,48E-02 | 1,24E-03 |
| 4. | 1,85E-03 | 1,80E-03 | 1,94E-03 | 2,22E-03 | 1,83E-03 | 3,19E-03 | 1,75E-03 | 1,75E-03 | 2,63E-03 | 1,75E-03 | 2,07E-03 | 4,82E-04 |
| 5. | 8,63E-04 | 8,35E-04 | 8,12E-04 | 8,09E-04 | 7,83E-04 | 9,85E-04 | 9,65E-04 | 7,68E-04 | 1,48E-03 | 7,91E-04 | 9,09E-04 | 2,13E-04 |
| 6. | 5,80E-04 | 3,14E-04 | 3,31E-04 | 3,48E-04 | 3,07E-04 | 3,65E-04 | 3,67E-04 | 3,10E-04 | 4,95E-04 | 3,06E-04 | 3,72E-04 | 9,24E-05 |
| 7. | 5,76E-04 | 1,62E-04 | 1,67E-04 | 1,70E-04 | 1,70E-04 | 1,74E-04 | 2,16E-04 | 1,66E-04 | 2,28E-04 | 1,75E-04 | 2,20E-04 | 1,27E-04 |
| 8. | 9,41E-05 | 5,56E-05 | 9,58E-05 | 5,02E-05 | 4,96E-05 | 1,05E-04 | 1,18E-04 | 8,57E-05 | 1,06E-04 | 5,25E-05 | 8,13E-05 | 2,67E-05 |
| 9. | 2,42E-05 | 5,83E-05 | 2,38E-05 | 5,59E-05 | 5,20E-05 | 6,86E-05 | 8,72E-05 | 5,88E-05 | 6,72E-05 | 6,36E-05 | 5,60E-05 | 1,94E-05 |

Na wykresie 1 przedstawiono zależność czasu wykonaniu algorytmu w zależności od postępu w grze.

Wykres 1.

Obraz zawierający tekst, diagram, Wykres, linia

Opis wygenerowany automatycznie

Na zdjęciach 1, 2, 3 przedstawiono screenshoty z gry.

Zdjęcie 1.

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

Zdjęcie 2.

Obraz zawierający tekst, zrzut ekranu, Czcionka, projekt graficzny

Opis wygenerowany automatycznie

Zdjęcie 3.

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

## Wnioski

Algorytm MinMax nadaje się do gry w kółko i krzyżyk. Zaimplementowany program potrafi odnaleźć optymalny ruch dla każdego z graczy. Metoda przycinania alfa-beta skutecznie przyspiesza czas wykonywania algorytmu co widać na wykresie 1. W porównaniu z podstawową wersją algorytmu która przeszukuje całe drzewo stanów gry metoda przycinania alfa-beta zmniejsza średnią głębokość przeszukiwanego drzewa oraz ilość odwiedzanych węzów. Czas wykonywania algorytmu maleje wraz z postępem gry ponieważ liczba stanów gry oraz możliwa głębokość drzewa maleje. Uruchomienie gry dla dwóch graczy korzystających z metody MinMax zawsze kończy się remisem. Wynika to z właściwości gry jaką jest kółko i krzyżyk. Obaj gracze wykonują najoptymalniejsze dla siebie ruchy nie pozwalając drugiemu graczowi wygrać.