

Politechnika wrocławska
Wydział Elektroniki, Fotoniki i Mikrosystemów
Projektowanie Algorytmów i Metody Sztucznej Inteligencji
Automatyka i Robotyka

PROJEKT 3

Autor: Jakub Jankowiak

Nr. indeksu: 258965

Grupa: Y03-51a, Pn 15:15

Prowadzący: Mgr inż. Marta Emirajslów

15 czerwca 2022

Spis treści

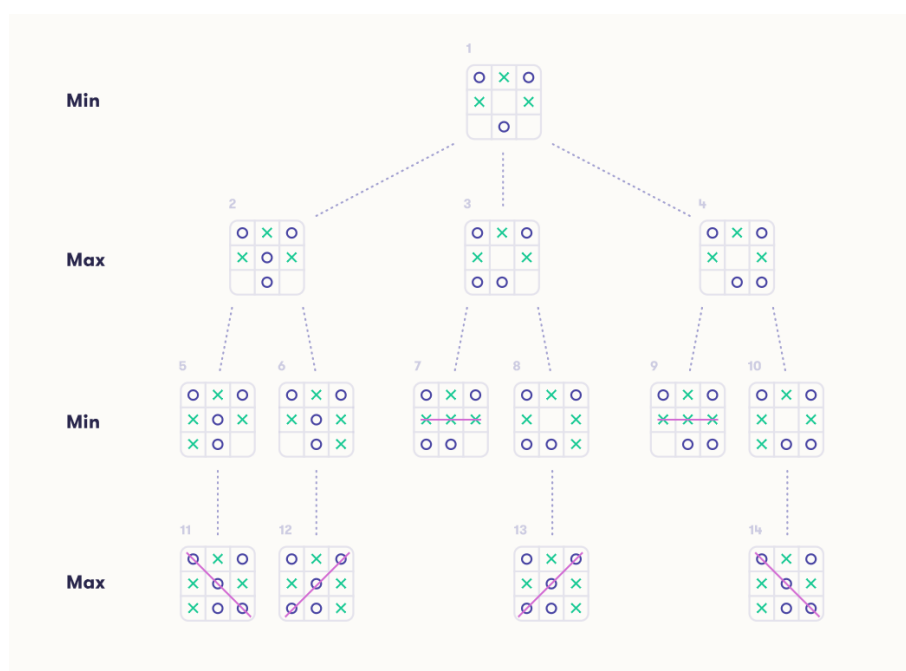
1	Wprowadzenie	2
2	Algorytm Minimax z cięciami alpha-beta	2
3	Przebieg gry	3
4	Wnioski	6
5	Bibliografia	6

1 Wprowadzenie

Zadaniem projektowym było odtworzenie i zaimplementowanie odpowiednio wcześniej wybranej gry, w tym przypadku jest to kółko i krzyżyk. Zaprojektowano grę wykorzystując do tego algorytm Minimax z cięciami alpha-beta. Program został napisany w języku C++.

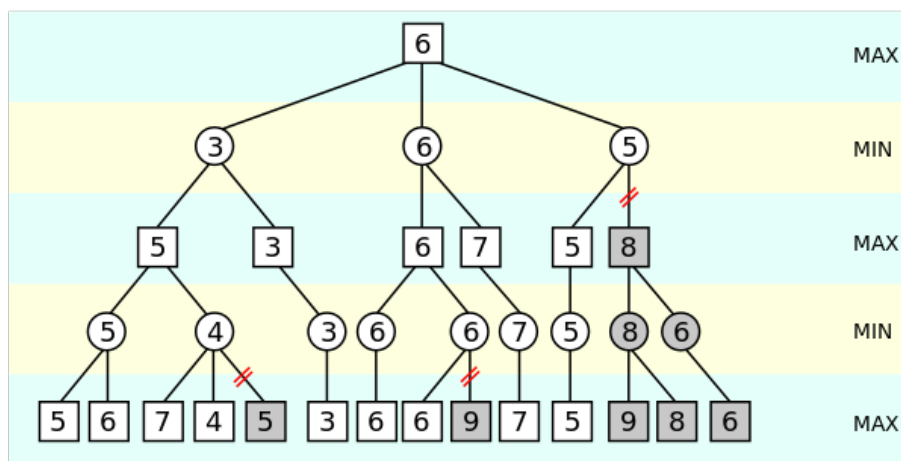
2 Algorytm Minimax z cięciami alpha-beta

Algorytm Minimax zwykle stosowany jest w grach, gdzie grającymi są gracz i komputer. Ocenia on stan gry danego poziomu na podstawie szukania możliwości potoczenia się gry. Podczas szukania przyszłych scenariuszy gry, algorytm przypisuje realne, możliwe korzyści dla samego siebie jak i przeciwnika. Ma to na celu zmaksymalizowanie szansy swojego zwycięstwa jednocześnie minimalizując szanse zwycięstwa przeciwnika. Algorytm opiera się o, można by powiedzieć, najgorszy scenariusz ponieważ zakłada, że przeciwnik wybierze najlepszy dla niego ruch zmuszając komputer do zagrania najgorszego. Fundamentalnym parametrem jest głębokość, czyli ilość przeszukiwanych poziomów rozgałęzień drzewa możliwych zagrań. Wraz z wzrostem głębokości rośnie również skuteczność maksymalizacji i minimalizacji, jednak wzrasta też czas przeszukiwania owych scenariuszy.



Rysunek 1: Działanie algorytmu Minimax dla gry kółko i krzyżyk

Cięcia alpha-beta służą do usprawnienia, optymalizacji działania algorytmu Minimax poprzez zmniejszenie gałęzi do przeszukania drzewa możliwości. Czas przeszukiwania ograniczony zostaje do przeszukania najbardziej obiecujących poddrzew, w związku z czym możemy zejść głębiej w tym samym czasie. Pozwala nam to znacznie zoptymalizować czas działania algorytmu nie zmniejszając przy tym jego skuteczności.



Rysunek 2: Działanie algorytmu Minimax dla gry kółko i krzyżyk

Algorytm utrzymuje dwie wartości alfa i beta, które reprezentują minimalny wynik gracza MAX i maksymalny wynik gracza MIN. Początkowo alfa jest -nieskończonością, a beta $+$ nieskończonością. W miarę postępowania rekursji przedział $(\alpha; \beta)$ staje się mniejszy i kiedy beta staje się mniejsze niż alfa, oznacza to, że obecna pozycja nie może być wynikiem najlepszej gry przez obu graczy i wskutek tego nie ma potrzeby przeszukiwania głębiej.

3 Przebieg gry

Na samym początku musimy zdefiniować rozmiar planszy na jakiej będziemy grać, minimalna plansza dla której można zacząć rozgrywkę to 3x3 ponieważ dla mniejszych plansz gra nie miała by sensu. Minimalna liczba znaków w linii jest równa 3 ponieważ dla mniejszej ilości gra również nie miałaby sensu. Gdy gracz wpisze wartości spoza zakresu wyświetlą mu się odpowiednie komunikaty, dotyczy to nie tylko rozmiarów planszy i ilości znaków w ciągu ale również wyboru kolumny i wiersza.

```

Witaj w grze kółko i krzyżyk!
Podaj dlugosc planszy (ilosc pol w kolumnie lub wierszu): 2
Za mala wartosc!!
Podaj dlugosc planszy (ilosc pol w kolumnie lub wierszu): 3
Podaj liczbe znakow w rzedzie wystarczajaca do wygranej: 2
Za mala wartosc!!
Podaj liczbe znakow w rzedzie wystarczajaca do wygranej: 4
Wiekksza od rozmiaru planszy?
Podaj liczbe znakow w rzedzie potrzebna do wygranej jeszcze raz: 3

1  2  3  K/W
|  |  |  1
---+---+---
|  |  |  2
---+---+---
|  |  |  3

```

Rysunek 3: Komunikaty dla błędnych wartości długości planszy i ilości znaków w ciągu

```

1  2  3  K/W
|  |  |  1
---+---+---
|  |  |  2
---+---+---
|  |  |  3

Twój ruch:
Nr Kolumny: 0
Kolumna poza zakresem planszy! Spróbuj ponownie.
Nr Kolumny: 1
Nr Wiersza: 0
Wiersz poza zakresem planszy! Spróbuj ponownie.
Nr Wiersza: 1
-----

```

Rysunek 4: Komunikaty dla błędnych wartości numerów kolum i wierszy

Program nie pozwala także na wpisywanie elementów w pola które już zostały wykorzystane. Po podaniu wykorzystanej wcześniej współrzędnej pojawia się odpowiedni komunikat po czym mamy ponownie szansę na podanie ponownie współrzędnej.

```

1  2  3  K/W
X |  |  |  1
---+---+---
| 0 |  |  2
---+---+---
|  |  |  3

Twój ruch:
Nr Kolumny: 2
Nr Wiersza: 2
To pole jest już wybrane, wybierz inne
-----

```

Rysunek 5: Komunikaty dla zajętych współrzędnych

Gracz gra "O" natomiast komputer "X", miejsce postawnienia znaku definiowane jest wyborem poszczególnej kolumny oraz wiersza. Po chwili otrzymujemy odpowiedź kompuera, przed każdym pojawieniem się planszy z wykonanym ruchem pojawia się linia oddzielająca która, zapobiega zlewaniu się ekranu.

```

Twoj ruch:
Nr Kolumny: 2
Nr Wiersza: 2
-----

 1  2  3  K/W
  |  |  1
--+---+---
  | 0 |  2
--+---+---
  |  |  3
-----

 1  2  3  K/W
X |  |  1
--+---+---
  | 0 |  2
--+---+---
  |  |  3

Twoj ruch:
Nr Kolumny: 

```

Rysunek 6: Przebieg rozgrywki

Podczas grania można uzyskać 3 różne wyniki, zwycięstwo komputera, zwycięstwo gracza oraz remis, dla każdego z tych scenariuszy wyświetlają się trzy różne komunikaty. Wyświetla nam się też pytanie o dalszą rozgrywkę.

```

 1  2  3  K/W
X | 0 | X  1
--+---+---
 0 | 0 | X  2
--+---+---
 0 | X | 0  3
-----
Koniec gry! Remis!
Chcesz zagrać jeszcze raz? (T/N)

 1  2  3  K/W
X |  | 0  1
--+---+---
X | 0 |  2
--+---+---
X |  | 0  3
-----
Koniec gry! Wygrywa X
Chcesz zagrać jeszcze raz? (T/N)

 1  2  3  K/W
X | X | 0  1
--+---+---
  | 0 |  2
--+---+---
 0 |  |  3
-----
Koniec gry! Wygrywa 0
Chcesz zagrać jeszcze raz? (T/N)

```

Rysunek 7: Scenariusze rozgrywki

Poziom trudności można zdefiniować poprzez głębokość przeszukiwania gałęzi drzewa przewidywań, im większa głębokość tym ciężiej będzie nam wygrać. Głębokość definiujemy w kodzie programu.

```
if(glebokosc == 6) return 0;
```

Rysunek 8: Definicja głębokości

4 Wnioski

Gracz grając z komputerem wyposażonym w algorytm Minimax jest w stanie wygrać ale tylko, gdy poziom głębokości przeszukiwania możliwości jest ustawiony na 5 lub mniej, gdy głębokość zostanie ustawiona na 6 lub wyżej gracz może jedynie zremisować z komputerem. Jest to wynik dla planszy 3x3, wraz z wzrostem rozmiarów planszy zwiększają się nasze szanse na zwycięstwo przy stałej głębokości. Wzrost ilości pól na planszy będzie także miał wpływ na czas wyboru ruchu przez komputer, jest to spowodowane wzrostem możliwych ruchów gdyż ilość pól również będzie większa. Cięcia alpha-beta znacznie optymalizują czas pracy algorytmu. Przy planszach 5x5 i większych czas może być naprawdę duży, w zależności od głębokości przeszukiwań. Na czas pracy ma również wpływ ilość potrzebnych znaków w ciągu, wraz z wzrostem ich ilości rośnie również czas.

5 Bibliografia

- https://pl.wikipedia.org/wiki/Algorytm_alpha-beta
- https://pl.wikipedia.org/wiki/Algorytm_min-max
- <https://course.elementsofai.com/pl/2/3>
- <https://eduinf.waw.pl/inf/utills/0012008/0415.php>
- <https://cpp0x.pl/forum/temat/?id=28543>