

PAMSI - Projekt 4 - Gry i SI

Kółko i krzyżyk

name

17 czerwca 2022

Spis treści

1	Cel ćwiczenia	2
2	Opis wybranej gry	2
3	Użyte techniki SI	2
3.1	Algorytm MinMax	2
3.2	Algorytm znajdowania najlepszego ruchu	2
4	Wnioski	2
5	Bibliografia	3

1 Cel ćwiczenia

Celem ćwiczenia było zaimplementowanie gry ze sztuczną inteligencją, której działanie oparte będzie na strategii MinMax.

2 Opis wybranej gry

Zaimplementowana przeze mnie gra nosi nazwę "kółko i krzyżyk". Gra toczy się w schemacie 1 na 1, sekwencyjnie. Polega na naprzemiennym stawianiu symboli przez obu graczy (jeden gracz ma symbole kółka, drugi krzyżyka) do momentu aż zostaną spełnione warunki wygranej, bądź do remisu (sytuacji w której nikt nie spełnił warunków zwycięstwa i jednocześnie nie można wykonać już żadnych ruchów). Polem gry jest kwadratowa plansza, w najpopularniejszej odsłonie zawierająca 9 kwadratowych pól zawartych w 3 rzędach i 3 kolumnach. Warunkiem zwycięstwa jest ułożenie 3 takich samych symboli w:

- jednej kolumnie
- jednym rzędzie
- na jednej z przekątnych

W naszym wariantcie gry wielkość pola jest wartością modyfikowalną, tak jak i liczba symboli potrzebnych do zwycięstwa.

3 Użyte techniki SI

3.1 Algorytm MinMax

Algorytm MinMax jest w tym wypadku głównym algorytmem na którym program opiera całą swoją strukturę działania. Algorytm polega na przeszukiwaniu głębokości "drzewka" decyzji, przyjęto w nim koncepcje dwóch wartości:

- MIN - wartość dotycząca przeciwnika, inaczej mówiąc minimalizacja szans na zwycięstwo
- MAX - maksymalizacja szansy na zwycięstwo

Sprawdzając coraz to dalsze symulacje stanu gry (naprzemiennie) porównujemy wartości symbolizujące skalę poprawności wyboru, szukając ruchu który będzie najbardziej optymalny. Algorytm jest wywoływany rekurencyjnie, w momencie znalezienia najbardziej optymalnego ruchu jest on przekazywany "wyżej" do korzenia drzewa decyzji.

3.2 Algorytm znajdowania najlepszego ruchu

Algorytm pomocniczy który steruje wywoływaniem pętli MinMax, w momencie znalezienia najlepszego ruchu zapamiętuje daną sekwencję i nakazuje jej wykonanie.

4 Wnioski

Program działa dla dowolnego rozmiaru planszy i dla dowolnej liczby znaków potrzebnych do zwycięstwa (przy założeniu, że liczba znaków potrzebna do zwycięstwa nie przekroczy ilości pól w jednej z linii). Zauważono, że największą skuteczność algorytm przejawia w sytuacjach w których liczba pól jest równa kwadratowi znaków potrzebnych do osiągnięcia wygranej (przykładowo pole 3x3 i wygrana przy 3 symbolach, 4x4 i 4 symbolach..). W takim wypadku nie zauważono spadku efektywności pracy algorytmu dla wyższych instancji.

Gorzej natomiast prezentowały się sytuację kiedy te proporcje były zniekształcone (przykładowo pole 6x6 i 4 znaki potrzebne do wygranej). W tym wypadku reakcja działania algorytmu była nadal szybka jednak widać było niedokładność w jego wyborach.

5 Bibliografia

- Piotr Wróblewski, W: Helion, Algorytmy, struktury danych i techniki programowania", 2009
- https://eduinf.waw.pl/inf/alg/001_search/index.php
- Michael T.G,Roberto T., David M.,W: John Wiley and Sons Inc. , "Data structures and algorithms in *C++*", 2009