Sztuczna Inteligencja i Inżynieria Wiedzy

Sprawozdanie numer 3

Autor: Patryk Konopka, 237980

Data: 14.04.2019

1. Cel ćwiczenia

Celem ćwiczenia było zapoznanie się z podstawowymi algorytmami stosowanymi w grach strategicznych (minmax oraz alfabeta).

1. Gra Stratego

Mój algorytm składa się z następujących klas:

* **Player** – klasa ta składa się z dwóch pól:
  + Number – numer gracza
  + Points – liczba punktów zdobyta przez gracza
* **Board** – jest to główna klasa programu składa się z następujących pól i metod:
  + Size – rozmiar tablicy gry
  + Board - tablica w której przechowywane są wszystkie ruchy
  + Place – metoda, która umieszcza punkt na planszy i zlicza ewentualne punkty za taki ruch
  + Can\_be\_placed – metoda, która sprawdza czy możliwe jest umieszczenie takiego punktu na planszy
  + Calculate\_points – metoda zliczająca punkty za dany ruch
  + Print – metoda drukująca planszę z grą
  + Random\_game – metoda startująca grę z wykorzystaniem losowej heurystyki przez obu graczy
  + Is\_not\_full – metoda sprawdzająca czy tablicy gry jest zapełniona
  + Minmax\_game – metoda startująca grę z wykorzystaniem heurystyki minmax
  + Minmax\_decision – metoda zwracająca następny ruch wg. heurystyki minmax
  + Minmax – w tej metodzie toczy się gra minmax z wykorzystaniem funkcji minmax\_decision to podjęcia decyzji o następnym ruchu.

1. Działanie algorytmu

W moim programie zaimplementowałem tylko heurystykę minmax oraz grę losową. W poniższej tabeli przedstawię liczbę punktów zdobytych przez graczy w 10 losowych grach, oraz wg. strategii minmax przy tablicy 5x5.

|  |  |  |
| --- | --- | --- |
| Runda | Gracz 1 | Gracz 2 |
| **1** | **56** | **45** |
| **2** | **50** | **51** |
| **3** | **47** | **54** |
| **4** | **57** | **44** |
| **5** | **43** | **58** |
| **6** | **59** | **42** |
| **7** | **52** | **49** |
| **8** | **50** | **51** |
| **9** | **47** | **54** |
| **10** | **42** | **59** |

Dla planszy 5x5 algorytm minmax zakończył się wynikiem 58 do 43. Możemy tutaj zauważyć, że w rundzie 6-tej heurystyka losowa zwróciła lepsze wyniki od minmax. Może to wynikać ze zbyt małej głębokości do której działa algorytm minmax.

W poniższej tabeli sprawdzam analogiczną sytuację lecz dla planszy 6x6.

|  |  |  |
| --- | --- | --- |
| Runda | Gracz 1 | Gracz 2 |
| **1** | **76** | **83** |
| **2** | **82** | **77** |
| **3** | **90** | **69** |
| **4** | **66** | **93** |
| **5** | **82** | **77** |
| **6** | **69** | **90** |
| **7** | **84** | **75** |
| **8** | **79** | **80** |
| **9** | **73** | **86** |
| **10** | **72** | **87** |

Dla planszy 6x6 algorytm minmax zakończył się wynikiem 91 do 68. W tych symulacjach heurystyka losowa częściej znajduję się w okolicach wyniku algorytmu minmax. Niestety ze względu na czas obliczeń nie udało mi się przeprowadzić symulacji na większych planszach.

1. Wnioski

Realizacja tego ćwiczenia znacznie poszerzyła moją wiedzę. Nauczyłem się nowej gry, w którą można grywać w deszczowe wieczory we dwóch. Z rzeczy związanych z przedmiotem nauczyłem się działania algorytmu minmax oraz alfabeta. Niestety ze względu na duże braki czasu związane z ciągłymi zjazdami przez cały maj nie udało mi się zaimplementować algorytmu alfa-beta.

Złożoność mojego algorytmu nie pozwoliła mi również na przeprowadzenie symulacji minmax dla tablic większych niż 6x6 (duży czas) oczekiwania. Domyślną głębokością mojego algorytmu jest 2. Przy próbie wywołania algorytmu dla głębokości większej niż 3 algorytm pomimo dłuższego czasu (ponad 5 minut) ciągle nie zwrócił wyniku.