

PROJEKTOWANIE ALGORYTMÓW SZTUCZNEJ INTELIGENCJI

PROJEKT 4 – GRY i AI

27.05.2020r.

Sebastian Robak 248994

Prowadząca zajęcia: Mgr inż. Magda Skoczeń

Termin zajęć: Środa 18:55

1. Krótkie wprowadzenie

Celem projektu było stworzenie gry dwuosobowej z użyciem sztucznej inteligencji, która wykorzystuje algorytm Minimax. Wykorzystuje się go w grach stworzonych dla dwóch osób. Polega on na zyskaniu największego zysku i zminimalizowania strat dla strony dla której jest wykorzystywany.

Część graficzna została zaimplementowana za pomocą programu graficznego Photoshop oraz przy użyciu biblioteki graficznej SFML.

2. Opis tworzonej gry

Gra w warcaby odbywa się na planszy 8x8 pól. Każde pole zawiera czerwony lub czarny pion lub damkę. Gra kończy się, gdy jeden z graczy straci wszystkie swoje pionki lub nie będzie mógł wykonać ruchu, ponieważ będzie zablokowany. Gracze zremisuje jeśli obydwójce wykonają po 10 ruchów damkami bez żadnego zbijania. Jeśli pionek dojdzie do ostatniego rzędu (czyli najdalej oddalonego od niego) pionek staje się damką. Przyjęto zasadę bicia obowiązkowego. Pionki mogą poruszać się tylko do przodu, ale mogą bić do tyłu. Damka po zbijaniu pionka może dalej się poruszać. Możliwe jest bicie kilkukrotne.

Plansza jest zaimplementowana za pomocą tablicy, która zawiera odpowiednią ilość pól oraz informację o tym co znajduje się na danym polu planszy. Klasa „Board” odpowiadająca planszy zawiera metody wykorzystywane do przesuwania się pionków i sposobów bicia. Do wykonania ruchu została użyta klasa zawierająca współrzędne pionka przed i po ruchu. Pionek może wykonać ruch jeśli znajduje się on na liście możliwych ruchów dla niego i jeśli na tej liście znajduje się bicie to musi ono zostać wykonane i nie ma możliwości użycia innego pionka w takiej sytuacji.

3. Opis wykorzystania AI

Do implementacji AI wykorzystano algorytm Minimax. Polega on na wybieraniu ruchów, które polegają na minimalizowaniu strat oraz osiągnięciu największego zysku dla danej strony. Polega to na stworzeniu drzewa wszystkich możliwych sytuacji dla następnych X ruchów (które określane są jako głębokość). Każdej możliwej przewidzianej sytuacji przypisywane są odpowiednie wartości zysku lub straty w zależności od wyniku ruchów. Następnie algorytm wybiera sytuację z największym zyskiem i realizuje ruch który prowadzi do takowej sytuacji w grze.

Algorytm polega na założeniu, że strona (gracz) dla której wywołany jest algorytm stara się zdobyć największą ilość punktów, a drugi gracz stara się przeszkodzić w tym. Drzewo w każdym momencie dostaje informację czy aktualny gracz stara się minimalizować czy maksymalizować zyski i w zależności od tego wybiera odpowiednią sytuację.

Zysk jest określany na podstawie punktów, które są przydzielane za określone ruchy i doprowadzenie do konkretnej sytuacji:

1. Wygrana +1000
2. Przegrana -1000
3. Remis 0
4. Pionek na bocznym polu +2
5. Pionek oddalony o jedno pola bocznego +1
6. Każdy pionek +10
7. Każda damka +30
8. Pionek z możliwością bicia +15

Do algorytmu Minimax wykorzystany został dodatkowo algorytm Alfa-Beta. Modyfikuje on algorytm Minimax poprzez wprowadzenie dodatkowych zmiennych oznaczających minimalną i maksymalną

wartość jaką algorytm Minimax może zapewnić na danej głębokości drzewa lub wyżej. Alfa oznacza najlepszą wartość jaką strona maksymalizująca może uzyskać a Beta najlepszą wartość jaką strona minimalizująca może uzyskać. Algorytm Alfa-Beta polega na przerywaniu oceniania danej sytuacji, gdy zostanie znaleziona co najmniej jedna inna możliwość, która dowodzi że ruch jest gorszy niż poprzednio badany. To oznacza, że taki ruch nie musi być dalej analizowany i można go porzucić.

Minimax razem z dodatkowym algorytmem zwraca ten sam wynik, który byłby zwrócony w standardowej wersji, ale jest szybszy, ponieważ przycina gałęzie, które nie mogą wpłynąć na ostateczną decyzję.

4. Podsumowanie i wnioski

- Złożoność algorytmu jest dość duża, przy większej głębokości ruchy mogą wykonywać się bardzo powoli, więc dla nieuciążliwej rozgrywki trzeba zmniejszyć dopuszczalną maksymalną głębokość algorytmu w zależności od możliwości obliczeniowych komputera
- Algorytm nie jest niepokonany, ponieważ zależy od wartości przypisywanych danym ruchom, a te nie zawsze są idealne dla danej sytuacji na planszy

5. Literatura

https://pl.wikipedia.org/wiki/Algorytm_min-max

<https://en.wikipedia.org/wiki/Minimax>

https://kcir.pwr.edu.pl/~witold/aiarr/2009_projekty/warcaby/

<https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/>

https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning

<https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-4-alpha-beta-pruning/>