Projektowanie algorytmów i metody sztucznej inteligencji

Prowadzący	Wykonawca	Nr. Indeksu	Data
dr inż. Łukasz Jeleń	Patryk Marciniak	248978	28.05.2020

1 Wstęp

Projekt polegał na zaimplementowaniu wybranej gry bazującej na algorytmie *minimax*. W wykonanym projekcie została zaimplementowana gra w warcaby o zasadach zbliżonych do warcab międzynarodowych:

- Plasza ma rozmiar 10×10 (100 pól).
- Pole na samym dole, po lewej powinno być ciemne.
- Każdy gracz ma 20 pionków.
- W startowej pozycji pionki ustawione są w czterech rzędach po przeciwnych stronach planszy, jedynie na polach ciemnych.
- Gracz grający jasnymi pionkami wykonuje ruch jako pierwszy.
- Zwykłe pionki mogą ruszać się do przodu, po skosie, na puste pola.
- Przeciwne pionki mogą być zbite poprzez przeskoczenie nad nimi, w jakimkolwiek kierunku, dwa pola dalej, na puste pole. Jeśli jest to możliwe po wykonaniu bicia należy wykonać kolejne bicie.
- Bicie jest obowiązkowe.
- Jeśli zwykły pionek dojdzie do ostatniego rzędu, to zmienia się w damkę.
- Jeśli pionek jest w ostatnim rzędzie tylko przelotnie (np. w wyniku kilku bić) to zmiana w damkę nie następuje.
- Damka ma możliwość przemieszczania się we wszystkich kierunkach, o nieograniczoną ilość pól, co również tyczy się bicia.
- Jeśli gracz nie ma możliwości ruchu, to przegrywa. Taka sytuacja może wystąpić jeśli przeciwne pionki blokują jakąkolwiek możliwość ruchu.
- Jeśli na planszy znajdują się damki i przez 20 tur stan planszy się nie zmieni, gra zostaje zremisowana.
- Jeśli gracz zostanie bez pionków, to przegrywa.

Dodatkowo istniała możliwość wykonania oprawy graficznej gry. W przypadku tego projektu użyto biblioteki SFML.

2 Opis algorytmu

2.1 Funkcja heurystyczna

Funkcja heurystyczna jest metodą znajdowania rozwiązań, dla której nie mamy gwarancji znalezienia rozwiązania optymalnego, czy nawet prawidłowego. Pozwala na przybliżenie rozwiązania danego problemu i to od jej implementacji zależy efektywność algorytmu wybierania najlepszych ruchów. Funkcji heurystycznej używa się w przypadku, gdy prawdziwy algorytm jest zbyt złożony lub nieznany.

Zaimplementowana funkcja heurystyczna w tym projekcie bierze pod uwagę zarówno liczbę posiadanych przez gracza pionków oraz damek, jak i ich pozycję na planszy. Pionki na dobrych pozycjach (np. przy krawędzi, która uniemożliwia ich zbicie) otrzymują dodatkowe punkty. Dodatkowo nagradzana jest sytuacja w której pionek ma możliwość bicia.

Punktowanie w zależności od czynników prezentuje się następująco :

- Zwykły pionek - 5 pkt.
- Damka - 20 pkt.
- Każdy zwykły pionek w rzędzie drugim od tyłu - 1 pkt.
- Każdy zwykły pionek w rzędzie trzecim od tyłu - 2 pkt.
- Każdy zwykły pionek w rzędzie czwartym od tyłu - 3 pkt.
- Każdy zwykły pionek w rzędach nie będącymi rzędami, w których pionki są na początku - 4 pkt.
- Każdy pionek w kolumnie przy krawędzi - 3 pkt.
- Każdy pionek w kolumnie drugiej od krawędzi - 1 pkt.
- Pionek, który ma możliwość bicia - 12 pkt.
- Wygrana - 999 pkt.
- przegrana - -999 pkt.
- Remis - 0 pkt.

2.2 Algorytm minimax

Algorytm **minimax** jest sposobem na minimalizowanie maksymalnych możliwych strat. Wywodzi się z teorii gry o sumie zerowej, czyli o grach, gdzie zysk jednego gracza wiąże się ze stratą drugiego gracza. Teoria **minimax** zakłada, że dla każdej gry o sumie zerowe, w której biorą udział dwie osoby, istnieje taka wartość, że biorąc pod uwagę strategię drugiego gracza, która jest najlepszą możliwą spłatą gracza pierwszego. Jednocześnie biorąc pod uwagę strategię gracza pierwszego, to ta sama wartość, ale o przeciwnym znaku jest najlepszą możliwą spłatą dla drugiego gracza. Reasumując, odpowiednia strategia pierwszego gracza gwarantuje mu spłatę niezależnie od strategii drugiego gracza. W tej samej sytuacji jest drugi gracz. Powoduje to, że warcaby jako gra o sumie zerowej, jeśli była by grana przez dwie osoby zawsze wybierające najoptymalniejszy ruch, skończyła by się remisem.

2.3 Algorytm alfa-beta

Algorytm alfa-beta (zwany też cięciami alfa-beta) jest algorytmem rozszerzającym algorytm **minimax** o redukcję węzłów, które nie muszą być rozważane w drzewie przeszukującym. Jego implementacja zakłada dodanie dwóch zmiennych (alfa i beta) do algorytmu **minimax**, które przechowują wartości maksymalną (alfa) i minimalną (beta), które algorytm może zapewnić na danej głębokości drzewa. Na początku działania algorytmu wartość "alfa" powinna być jak najmniejsza, zaś "beta" jak największa. W czasie działanai algorytmu te wartości są korygowane i na ich podstawie algorytm może zignorować dane poddrzewo, jeśli zajdzie warunek $\alpha \geqslant \beta$. W dalszych etapach działania algorytmu "odcięte" poddrzewo nie będzie rozważane co korzystnie wpływa na złożoność algorytmu, pozwalając na osiągnięcie krótszego czasu wyszukiwania optymalnego ruchu. Przy tych założeniach, złożoność obliczeniowa algorytmu będzie wynosić $\mathcal{O}(b^{d/2})$, gdzie b to średni lub stały współczynnik rozgałęzienia, a d to głębokość.

3 Podsumowanie i wnioski

- Zaimplementowane algorytmy dobrze radzą sobie w rozgrywce, jednak zauważyć można pewną powtarzalność w ruchach SI.
- Optymalność wybranego przez SI ruchu zależy w dużej mierze od funkcji heurystycznej oraz głębokości drzewa przeszukiwania. Z tego powodu nieodpowiednio dobrany system punktacji może powodować dziwne ruchy ze strony SI. Dodatkowo, algorytm zakłada, że jego przeciwnik wykona najoptymalniejszy dla niego ruch, dlatego też SI nie przewiduje innych ruchów gracza.
- Złożoność algorytmu jest dość duża, co powoduje, że jeśli chcielibyśmy zwiększyć poziom trudności SI (zwiększając głębokość drzewa przeszukiwania), wiązało by się to z dłuższym czasem oczekiwania na wyszukanie optymalnego ruchu.
- Aktualna wersja funkcji heurystycznej jest prosta i nie zawiera w sobie wszystkich czynników wpływających na strategię. Aby jak najdokładniej przybliżyć dany system należało by posiadać obszerną wiedzę na temat strategii gry w warcaby i być w stanie odpowiedno zaimplementować daną wiedzą tak, aby algorytm mógł z niej korzystać.

4 Literatura

- https://en.wikipedia.org/wiki/International_draughts
- https://en.wikipedia.org/wiki/Minimax
- https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning
- https://pl.wikipedia.org/wiki/Heurystyka_(informatyka)