Sprawozdanie Projekt 3 - Warcaby		
Rafał Jasiński 259384	13.06.2022r	Dr hab. inż. Andrzej Rusiecki
Zadania na ocenę bdb (5.0)		
Link do repozytorium: `https://gitlab.com/JasinskiR259384/pamsi-2022\`		

1 Wstęp

Celem zadania było napisanie gry w warcaby z wykorzystaniem algorytmu MinMax z alfa-beta cięciami. Należało również doprecyzować zasady gry.
Gra została napisana z wykorzystanie graphical user interface (GUI) w Java'ie za pośrednictwem JavaFX oraz programu obsługującego tworzenie scen graficznych SceneBuilder.

2 Ustawienia

W grze po naciśnięciu przycisku ***PLAY*** pojawiają się ustawienia rozgrywki.

2.1 Poziom trudności (Set Difficulty)

Za pomocą przycisków można wybrać "poziom" trudności.
Poziom ten jest definiowany w oparciu o klasę *Evaluation*, która odpowiada za przyznawanie punktów w zależności od pozycji na planszy.
Dostępne są 3 poziomy trudności:

- Medium - wszystkie wartości mają stałą wartość równą 1 - wartość domyślna
- Hard - wartości zostały dobrane ręcznie
- Random Fiesta - w tym trybie wartości poszczególnych ustawień są losowane za pomocą funkcji *Random*

2.2 Głębokość funkcji MinMax (Depth)

Ta opcja odpowiada za głębokość przeszukiwań rekurencji algorytmu MinMax
Dostępne są 3 poziomy przeszukiwań:

- 2
- 4 - wartość domyślna
- 6

Większa głębokość sprawia, że czas potrzebny na ruch bota się zwiększa, co powoduje opóźnienie. (Powyżej 6 sama aplikacja również przechodziła w stan zwieszenia)

3 Algorytm

Algorytm AI napisany jest na bazie algorytmu MinMax wraz z alfa-beta cięciami.

3.1 MinMax

Algorytm polegający na minimalizowaniu, maksymalnie możliwych strat. Aby jego działanie miało jakikolwiek sens, musi istnieć dwóch graczy, którzy otrzymują pewną ilość punktów za wykonanie czynności. Gracz A będzie się starał wybrać taki ruch, aby stracić jak najmniej punktów oraz ograniczyć ilość zdobytych punktów przez gracza B. (Gracz B analogicznie) [odwrotna wersja tego algorytmu nosi nazwę MaxMin]
Zasadę działania algorytmu można przedstawić za pomocą drzewa binarnego, w których każdy kolejny poziom zawiera naprzemienne punkty graczy. Korzeniem natomiast jest wartość, którą zdobędzie dany gracz po przeanalizowaniu możliwych strat.

3.2 Alfa-Beta cięcia

Algorytm ten polega na "wycinaniu" (nie braniu pod uwagę podczas obliczeń) pewnej części przypadków, jeżeli zostaną spełnione określone warunki. Wiąże się to z założeniem, że to, czy dana ścieżka może uzyskać mniejszą wartość, nie ma znaczenia, ponieważ możemy przyjąć pewną górną granicę (z poprzednich odgałęzień drzewa), która będzie spełniała warunek odcięcia - w tym przypadku algorytm MinMax nigdy nie wejdzie w daną ścieżkę.

3.3 Heurystyka

Jest to metoda pozwalająca znaleźć pewne rozwiązanie, które nie koniecznie będzie rozwiązaniem optymalnym, a nawet prawidłowym. W programie heurystyka ta powiązana jest z procesem ewaluacji, czyli oceny aktualnej pozycji oraz oceny pozycji po wykonaniu jednego z możliwych ruchów. Po zliczeniu ilości punktów za pozycję, wartość ta jest przekazywana do algorytmu MinMax, który "podejmuje" decyzję, czy "opłaca się" podążać tą ścieżką.

4 Testy i uwagi

4.1 Uwagi

1. Instrukcja o tym jak należy uruchomić grę znajduje się na repozytorium w sekcji *readme.md*, natomiast zasady gry są dostępne w samej grze pod zakładką **HOW TO PLAY**

2. Jeżeli podczas gry użytkownik nie będzie posiadał już możliwych ruchów do wykonania, po tym jak AI wykona swój ruch, a sytuacja użytkownika się nie zmieni, gra się kończy i pojawia się ekran z informacją o porażce. Następuje to dość nagle i można odnieść wrażenie, że w grze występuje błąd, jednakże jest to celowa zasada działania.

4.2 Testy

Działanie algorytmu zostało poddane różnym metodom grania: bezpieczne podejście (unikanie bić), pchanie pionków pod zbicie (agresywna gra), mieszane (staranie się uzyskać jak najkwiększej przewagi). Założenia działania punktacji zawartej w ewaluacji oraz "decyzji" podjętych przez AI zmieniały się w zależności od sytuacji na planszy.

5 Wnioski

1. Podczas zwiększania głębokości rekurencji algorytmu MinMax decyzje podjęte przez AI mogą wydawać się gorsze ze względu na fakt, że podczas oceny, algorytm ten porównuje punktacje za pozycję w oparciu o ruchy zgodne z przydzielonymi punktami w procesie ewaluacji.

Literatura

[1] Lidraughts. `https://lidraughts.org`. [Online; accessed 12-June-2022].

[2] Algorithms explained – minimax and alpha-beta pruning. `https://youtu.be/l-hh51ncgDI`, 2018. [Online; accessed 9-June-2022].

[3] J. Mańdziuk, K. WALĘDZIK, M. Kusiak. Evolutionary-based heuristic generators for checkers and give-away checkers. 24(4), 2007.

[4] Wikipedia. Heuristic (computer science) — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Heuristic%20(computer%20science)&oldid=1082833841`, 2022. [Online; accessed 12-June-2022].

[5] Wikipedia. Minimax — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Minimax&oldid=1088723729`, 2022. [Online; accessed 12-June-2022].