Pamsi kolko i krzyzyk Denis Firat

Denis Firat 249031

May 2020

1 Wstep

Moim zadaniem było stworzenie prostego programu, gry komputerowej, kółko i krzyżyk. Wyzwaniem i celem projektu było zaimplementowanie algorytmu minimax, który imitowałby sztuczna inteligencje, bota, z którym gracz mógłby grać. Program umożliwia zmiane wielkości planszy, na której gra sie z botem. Plansza wyświetlana jest w terminalu z pomoca znaków ASCII.

2 Opis algorytmu

Funkcja MiniMax(char **plansza, char PChar, int rozmiar, int depth, int a, int b) ocenia, jakość ruchu jaki wykonać ma sztuczna inteligencja. Funkcja działa rekurencyjnie, wybiera "najlepszy" ruch dla danej planszy, nastepnie jeżeli nie nastapiła wygrana, to "symuluje" najlepszy ruch przeciwnika, a potem własny i tak dalej, aż do głebokości 6. Dodatkowo zostały dodane ciecia alfa, beta, których zadaniem jest zatrzymywać rekurencje funkcji, gdy okaże sie, że ruch nie zmienia przewagi na planszy.

3 Opis gry

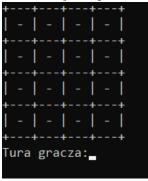
Na poczatek wita nas menu z trzema możliwościami:

- Start
- Zmien rozmiar planszy
- Wyjscie
- 1. Start
- Zmien rozmiar planszy
- Wyjscie

Po wybraniu Zmien rozmiar planszy, użytkownik ma okazje zmienić rozmiar planszy. Dostepne rozmiary sa od 3x3, ale zaleca sie nie przekraczać wiecej niż 10x10. Pole wyboru rozmiaru prezentuje sie tak:

Podaj rozmiar planszy:

Po rozpoczeciu gry wybierajac Start, naszym oczom ukazuje sie plansza o wybranym rozmiarze.



Zadaniem gracza jest wpisać współrzedne pola, do którego gracz chciałby wpisać swój znak. Pierwsza cyfra oznacza wiersz, a druga kolumne. Ważne jest, aby pamietać o tym, że współrzedne pól numerowane sa od 0. Gracz gra kółkiem i zaczyna rozgrywke. Po wybraniu pola, naszym oczom ukazuje sie uaktualniona plansza wraz z wyborem komputera, którego reprezentuje znak X. Proces wybierania pól trwa, aż ktoś wygra albo uzupełni sie cała plansza.



Program, po każdym ruchu sprawdza czy nastapiła wygrana jednej z stron, gdy ma to miejsce ukazuje sie napis o wygranej i możliwość rozpoczecia gry jeszcze raz.

```
+---+--+

| o | o | x |

+---+---+

| o | x | - |

+---+---+

| x | - | - |

+---+---+

X wygrywa

1. Start

2. Zmien rozmiar planszy

3. Wyjscie
```

4 Wnioski

Projekt ten, był chyba jednym z ciekawszych, nad jakim pracowałem w ramach tego kursu. Najważniejszy, wniosek dla mnie z tego projektu to zrozumienie, że świat sztucznych inteligencji, programów decyzyjnych, nie jest aż tak niedostepny dla poczatkujacego programisty. Zawsze przekonany, że coś takiego jak bot do gry jest już highendowym osiagnieciem zszokowałem sie, że minimax wystarczajacy do obsługi kółka i krzyżyk jest parolinijkowym algorytmem. Dzieki temu zainteresowałem sie praktycznym wdrażaniem AI do swoich projektów, a nie tylko ogladaniem jak efektów końcowych programistów na youtube. Jeżeli chodzi o wydajność działania algorytmu to w przypadku 3x3, 4x4 czas działania algorytmu jest prawie niezauważalny. Przy 5x5 czas na ruch przeciwnika trwał 14,88 sekundy, dla wiekszych plansz czas jest już na tyle długi, że granie mija sie z celem.

References

- [1] https://pl.wikipedia.org/wiki/Algorytm_min-max
- [2] https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/
- [3] https://stackoverflow.com/