PAMSI - Projekt 3

Strategia MinMax

Prowadzący	mgr inż. Marta Emirsajłow
Autor	Patryk Szydlik ind 248949 Wydział Elektroniki W4
Termin zajęć	Piątek 7:30 - 9:00
Data oddania sprawozdania	29 Maja 2020 r

Spis treści

1 Wprowadzenie

Celem metod sztucznej inteligencji jest podejmowanie optymalnych decyzji przy rozwiązywaniu danych problemów. Niezależnie od skomplikowania zadania zakłada się, że idealny algorytm sztucznej inteligencji będzie w stanie znaleźć optymalną odpowiedź w skończonym chociaż niekoniecznie krótkim czasie. Poniżej przedstawiona jest w praktyce jedna z popularnych strategii podejmowania decyzji przez AI czyli strategia MinMax.

2 Strategia MinMax

Powyższa strategia zakłada wybór optymalnego ruchu w danej sytuacji poprzez analize wszystkich możliwych kombinacji ruchów wraz z przewidywaniem ruchów przeciwnika.
W programie zakłada się, że dla obecnego gracza optymalny ruch do wykonania to taki, który ma najwyższą wartość przypisaną mu przez funkcję oceniającą, natomiast dla przewidywanego ruchu przeciwnika optymalna pozycja to taka, która ma najmniejszą wartość punktową uzyskaną przez pierwszego gracza. Sposób wyboru kolejnych ruchów przez symulacje najpierw zagrania o maksymalnej punktacji, a potem zagrania o minimalnej punktacji w ramach ruchu przeciwnika jest główną cechą tej strategii i właśnie stąd przyjęła ona nazwę MinMax.

3 Zarys gry

3.1 Uruchamianie

Po skompilowaniu programu z użyciem makefile zostaje wyświetlona w terminalu prośba o wprowadzenie parametrów definiujących rozmiar pola gry oraz ilość znaków w rzędzie potrzebnych do wygrania. Zdecydowałem się na to rozwiązanie, ponieważ ułatwiło to zadanie prawidłowego skalowania się pół widocznych w menu graficznym. Maksymalny rozmiar pola ze względu na ograniczoną szerokość menu graficznego wynosi 50x50 (pola są ledwo widoczne, jednak rozgrywka jest możliwa).

3.2 Klasa sędziego

Obiekt tej klasy przechowuje większość informacji na temat trwającej rozgrywki oraz pośredniczy w wykonywaniu ruchów i kontrolowaniu czy gra została rozstrzygnięta.

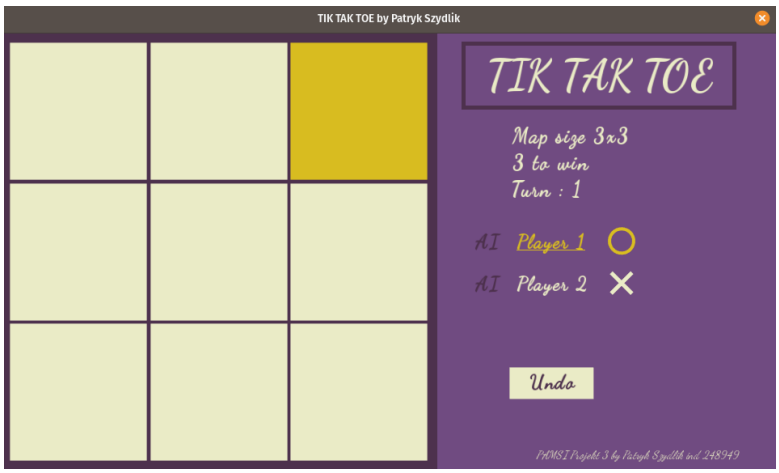
3.3 Klasa AI

Obiekt tej klasy opierając się na strategii MinMax przydziela poszczególnym ruchom na planszy punktacje według , której następnie dokonuje wyboru jaki ruch wykonać. Algorytm minmax jest rekurencyjny dlatego została ograniczona głębokość możliwej rekursji (czyli ilość ruchów do przodu, które przewidywało AI) w celu zapewnienia płynnej rozgrywki.
Specjalnie napisana w tym celu funkcja getOptimalDepth, oblicza ona optymalny poziom rekursji dla obecnej sytuacji na planszy, która zapewni przewidywanie największej liczby ruchów przy jednoczesnym ograniczeniu czasu ruchu AI.

4 Obsługa menu graficznego

4.1 Stawianie znaku

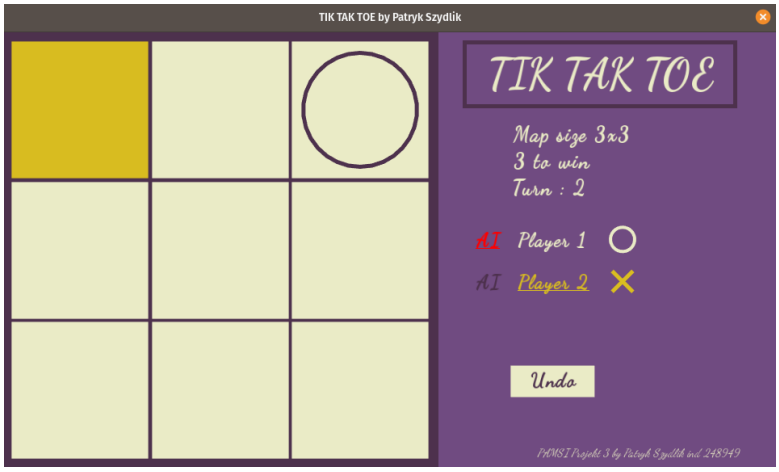
Po najechaniu myszką na wolne pole, zostanie ono podświetlone na złoto.// Jeśli trwa obecnie tura gracza to może on kliknąć na wolne pole aby postawić tam swój znak. Podświetlana treść po prawej stronie pokazuje, którego gracza tura właśnie trwa.



Rysunek 1: Podświetlenie pola

4.2 Aktywowanie AI

Poprzez najechanie i kliknięcie w napis AI przy wybranym graczu zostanie uruchomiona dla niego strategia MinMax, co będzie uwidocznione przez zmianę koloru napisu AI na czerwony. Istnieje możliwość uruchomienia gry AI vs AI (są one jednak mało interesujące).



Rysunek 2: Uruchomienie przeciwnika AI, stawianie znaku, zmiana gracza

4.3 Cofanie ruchu

Istnieje możliwość cofania ruchu w grze. Dokonuje się tego poprzez naciśniecie na przyciski ”Undo”. W przypadku gry Player vs AI cofnięcie ruchu przez gracza cofa również ruch AI, który nastąpił po turze gracza.

5 Wnioski

Strategia MinMax ze względu na wielokrotnie wywoływaną rekursję ma bardzo dużą złożoność obliczeniową. W samym programie ilość wywołań rekurencyjnych funkcji MinMax zależy od ustalonej głębokości rekursji oraz od ilości wolnych pól do wyboru, a ogólnie jest opisana wzorem :

$$\frac{(N)!}{(N-D)!}$$

Gdzie N to ilość wolnych pól na planszy i dla pierwszych tur gry wynosi ona s^2, gdzie s to rozmiar pola gry, więc zależy kwadratowo od tego parametru. Natomiast parametr D to głębokość rekursji.
Jak można wywnioskować z samego wzoru ilość wywołań funkcji MinMax ze względu na zmiane głębokości rekursji rośnie w sposób zbliżony aczkolwiek delikatnie wolniejszy od wykładniczego, ponieważ jest ona dokładnie równa iloczynowi kolejno malejących D liczb , dlatego też wymagane było ograniczenie głębokości rekursji w programie przez zastosowanie funkcji, która szacowała ilość dozwolonych wywołań funkcji MinMax przez program, co umożliwiło ograniczenie czasu "zastanawiania się" gracza AI.

6 Kod programu

Zdalne repozytorium z kodem programu dostępne jest pod linkiem:

https://bitbucket.org/patryk_sz/pamsi/src/master/Project_3/

7 źródła

Podczas wykonywania projektu poza materiałami wykładowcy wspierałem się nastepującymi stronami:

1. Wiki - MinMax
2. Brilliant - MinMax
3. MinMax - explained