

Politechnika Wrocławska
Wydział Elektroniki
Kierunek Automatyka i Robotyka
Sprawozdanie z laboratorium

Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Projekt 3: Gra kółko i krzyżyk

05.06.2020

Link do projektu: <https://github.com/bartech99/PAMSI3>

Prowadząca: mgr inż. Marta Emirsajłow
Termin: piątek, 7:30

Autor: Bartosz Wójcik, 249010

1. Wprowadzenie

Celem projektu było stworzenie gry z wykorzystaniem algorytmu sztucznej inteligencji. Tematem niniejszego projektu jest kółko i krzyżyk, z zastosowaniem algorytmu minimax. Gracz ma możliwość wyboru, czy chce grać przeciwko innemu użytkownikowi, czy przeciwko komputerowi. Ponadto, użytkownik ma możliwość wybrania rozmiaru planszy oraz ilości znaków skreślonych w rzędzie, aby wygrać rundę.

2. Opis zastosowanych rozwiązań

Program został napisany obiektowo, z wykorzystaniem polimorfizmu. Dana jest jedna klasa bazowa – Player, na bazie której powstają kolejne – Person oraz Computer. Klasa Person zawiera wszystkie metody niezbędne do wykonania ruchu przez użytkownika, natomiast Computer – wszystko, co związane z metodami sztucznej inteligencji. Obie klasy płynnie współpracują z klasą Board – przechowującą planszę jako dwuwymiarową tablicę i metody wykonujące operacje na tej tablicy, a także z klasą Menu – odpowiedzialną za bezpieczne pobieranie potrzebnych wartości od gracza oraz wyświetlaniem komunikatów.

Funkcja główna programu wykorzystuje zalety polimorfizmu. Po odebraniu od użytkownika informacji o trybie gry, alokuje miejsce dla potrzebnych klas – jedno dla klasy Computer, drugie dla Person, lub dwa razy dla Person. Następnie tworzy dwa wskaźniki polimorficzne i obu przypisuje po jednym graczem.

Po skończonej rundzie, gracz może od razu rozpocząć kolejną, na planszy o tych samych lub innych wymiarach i w tym samym lub innym trybie gry. Tę funkcję udało się zaimplementować dzięki dynamicznemu tworzeniu obiektów.

3. Opis algorytmu minimax

Zastosowany algorytm minimax jest często wykorzystywany przy projektowaniu gier dwuosobowych. Zajmuje się on wyznajdowaniem najlepszego możliwego ruchu. Z powodu długiego czasu działania, wynikającego z rekurencyjnej natury tego algorytmu, zaimplementowano metodę cięć alfa-beta. Przed uruchomieniem samego algorytmu, program analizuje niezbędną ilość rekurencyjnych wywołań przy użyciu metody Depth. Algorytm wykonuje się do momentu znalezienia korzystniejszego ruchu, niż przy poprzednim wywołaniu.

4. Podsumowanie i wnioski

- Im większy rozmiar generowanej planszy, tym więcej czasu potrzebuje algorytm.
- Zastosowanie cięć alfa-beta i ograniczenie głębokości rekurencji pozwala na zoptymalizowanie programu pod względem czasu przeprowadzania operacji.
- Dzięki zastosowaniu metod sztucznej inteligencji, użytkownik ma małą szansę na wygraną w pojedynku z komputerem, gdy ilość znaków w rzędzie jest taka sama jak rozmiar planszy.

5. Bibliografia

- <https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-3-tic-tac-toe-ai-finding-optimal-move/>
- <https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-4-alpha-beta-pruning/>