

Małgorzata Pietras  
Automatyka i Robotyka  
235794

dr inż. Łukasz Jeleń  
środa 7:30

## Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Projekt 3  
data oddania: 05.06.2019

# 1 Wstęp

Projekt trzeci polegał na stworzeniu gry wykorzystującej algorytm minimax. W sprawozdaniu przedstawiony jest opis jego działania na podstawie gry w kółko i krzyżyk. Użytkownik ma do wyboru dowolny rozmiar planszy. Gra toczy się pomiędzy użytkownikiem a komputerem.

## 2 Opis gry i stosowanych technik SI

Gra w kółko i krzyżyk tradycyjnie rozgrywana jest na planszy 3x3 pól. W omawianej implementacji możliwy jest wybór planszy o dowolnym rozmiarze. W grze biorą udział dwie osoby (użytkownik i komputer), jedna ma krzyżyki, a druga kółka. Celem gry jest ułożenie na planszy symboli tak, aby zajęły całą linię poziomo, pionowo, bądź na ukos. Pierwszy gracz, który tego dokona, zostaje zwycięzcą; jeżeli plansza zostanie zapelniona, następuje remis.

Schemat gry przebiega następująco: stworzenie planszy, generowanie ruchu człowieka i zastosowanie algorytmów do wygenerowania ruchu komputera. Następnie sprawdzane są warunki zwycięstwa i jeżeli gra toczy się dalej - ponownie ruch wykonuje człowiek. Gra kończy się zazwyczaj przegraną człowieka lub remisem.

Algorytm minimax bazuje na teorii gier von Neumanna, Morgensterna z 1944. Algorytm ten służy do wyznaczania najbardziej korzystnego ruchu dla komputera, ponieważ jest decyzją kompletną dla drzew skończonych. Nazwa algorytmu pochodzi od przyjętej w nim strategii postępowania: symulowanie ruchów gracza i przeciwnika, przy czym: znajdując się na poziomie przeciwnika wybieramy ruch najgorszy dla gracza, a znajdując się na poziomie gracza wybieramy ruch dla niego najlepszy. Analizując drzewo gry w ten sposób, nie da się dojść do liści drzewa, tj. do sytuacji w której gra jest rozegrana. W użytym algorytmie dla przypisywane są następujące wartości punktów: 10 jeżeli wygrywa człowiek, -10 jeżeli komputer i 0 w przypadku remisu. Złożoność czasowa algorytmu to  $O(b^m)$ , a pamięciowa  $O(bm)$ . Z tego powodu dla dużych plansz czas trwania algorytmu staje się bardzo długi i należy zastosować ograniczenia głębokości przeszukiwania.

W celu usprawnienia pracy programu dla dużych plansz zastosowana została modyfikacja algorytmu minimax w postaci cięć alfa i beta. W podejściu tym węzły nie mające wpływu na wartość przypisaną do przodków są eliminowane z analizy. Cięcie alfa wybiera najlepszą wartość dla wartości maksymalnej dla kolejnych kroków. Cięcie alfa wybiera najlepszą wartość dla wartości maksymalnej dla kolejnych kroków. Cięcie beta wybiera najlepszą wartość dla wartości minimalnej dla kolejnych kroków. Złożoność obliczeniowa algorytmu minimax z zastosowaniem cięć alfa i beta to  $O(b^{m/2})$

### 3 Podsumowanie i wnioski

- Algorytm minimax gwarantuje zwycięstwo komputera, lub w najlepszym przypadku remis.
- Możliwe jest wymuszenie przegranej komputera poprzez ograniczenie przeszukiwania algorytmu.
- Dla planszy 3x3 program działa bardzo szybko.
- Dla większych plansz (od 4x4) praca programu staje się bardzo długa, nawet po próbie zaimplementowania cięć alfa-beta.
- Aby usprawnić działanie programu dla dużych plansz należy ograniczyć przeszukiwanie drzewa.
- Złożoność obliczeniowa algorytmu minimax z cięciami alfa i beta najprawdopodobniej zgadza się z oczekiwaną  $O(b^{m/2})$ .

### 4 Literatura

1. Grębosz J., Symfonia C++ standard
2. Goodrich M., Data Structures and Algorithms in C++
3. Cormen T., Wprowadzenie do algorytmów
4. Szelc M., Przegląd metod sztucznej inteligencji służących do projektowania i realizacji gier strategicznych na przykładzie wybranej gry strategicznej
5. 30.05.2019 <https://codereview.stackexchange.com/questions/185501/simple-tic-tac-toe-with-minimax-algorithm-follow-up>
6. 30.05.2019 <https://eduinf.waw.pl/inf/prg.php>