

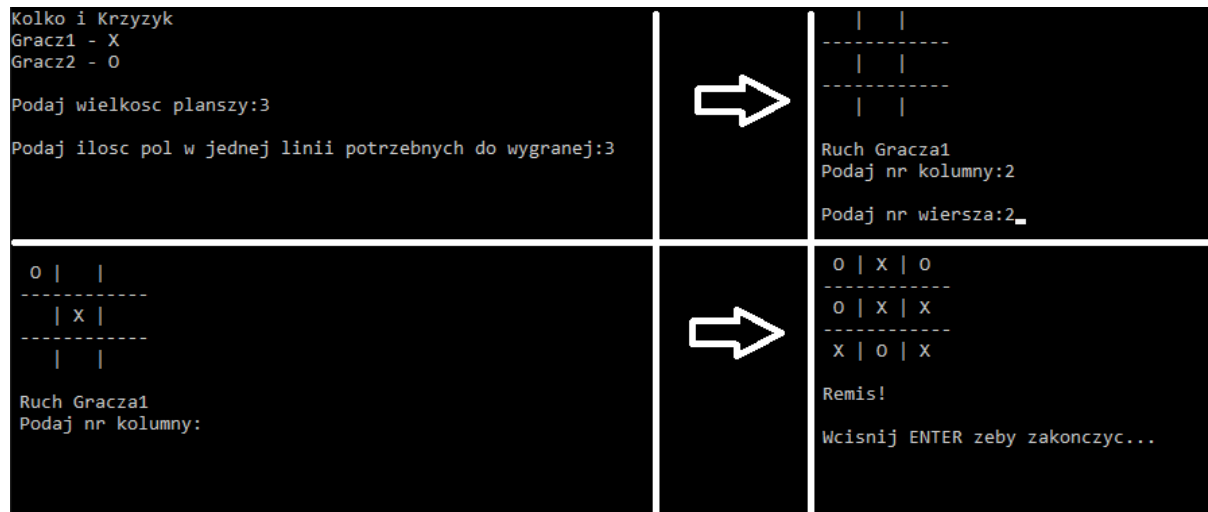
Prowadzący: Dr inż. Łukasz Jeleń	Projektowanie Algorytmów i Metody Sztucznej Inteligencji	Termin zajęć: Śr. 7:30
Wojciech Gołębiowski 241477	<b><u>„Kółko i Krzyżyk”</u></b>	Data oddania sprawozdania: 4.06.2019 r.

## 1. Wprowadzenie

Kółko i krzyżyk gra strategiczna oparta na strategii minmax rozgrywana przez dwóch graczy. Gracze obejmują pola na przemian dążąc do objęcia określonej liczby pól w jednej linii, przy jednoczesnym uniemożliwieniu tego samego przeciwnikowi. Pole może być objęte przez jednego gracza i nie zmienia swego właściciela przez cały przebieg gry.

## 2. Opis tworzonej gry wraz z wytłumaczeniem stosowanych technik SI.

Głównym elementem gry jest dynamiczna tablica dwuwymiarowa której rozmiary są definiowane przez użytkownika w konstruktorze klasy. Składa się ona z pól typu int które przyjmują wartości 0,-1,1 dla odpowiednio: pustego pola, pola zajętego przez Gracza1 i pola zajętego przez Gracza2 (czyli w tym przypadku przez algorytm). Grę rozpoczyna użytkownik ustalając rozmiar planszy oraz ilość pól w linii potrzebnych do wygrania gry, następnie wykonuje ruch wybierając pole na planszy poprzez wpisanie nr kolumny i numeru wiersza w którym znajduje się to pole. Następny ruch należy do algorytmu który wybiera pole wyliczając najlepsze miejsce do wybrania na podstawie metody minmax i alfa beta cięć. Kolejny ruch należy do gracza. Gra kończy się jeżeli gracz lub algorytm uzyskają ustaloną na początku gry ilość pól w jednej linii.



Rysunek 1 Przykładowy przebieg gry

Algorytm żeby sprawdzić który ruch jest najlepszy w danym przypadku używa funkcji MinMax która bierze pod uwagę wszystkie możliwe przypadki rozwinięcia się gry zakładając, że przeciwnik też gra optymalnie i ucinając rozwinięcia gry które nie muszą już być przeszukiwane, ponieważ już istnieje lepszy ruch. Funkcja MinMax zwraca wartość w zależności od przypadku, wybiera on wygraną która potrzebuje jak najmniej ruchów do zakończenia gry i zwraca wartość 10 - ilość ruchów potrzebnych do wygrania (głębokość drzewa), dla remisu zwraca 0, a dla przypadku w którym wygrywa przeciwnik -10 plus ilość ruchów do zakończenia gry.

### **3. Podsumowanie i wnioski**

- Przez to, że algorytm nie ma ograniczenia co do głębokości przeszukiwania drzewa możliwych zakończeń gry dla większych plansz czas odpowiedzi algorytmu na ruch gracza jest wydłużony.
- Jeżeli i gracz i algorytm grają optymalnie to zawsze będzie remis!

### **4. Bibliografia**

- [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
- [pl.wikipedia.org](https://pl.wikipedia.org) (stan na 2.06.19)