

# Sprawozdanie – Gra (kółko i krzyżyk)

Kinga Długosz

Indeks: 249002

Prowadzący: Marta Emirsajłow

## 1. Wprowadzenie

W programie zaimplementowano grę w kółko i krzyżyk z wykorzystaniem algorytmu MinMax z alfa-beta cięciami. Gracz posiada możliwość definiowania rozmiaru pola (kwadratowego) wraz z ilością znaków w rzędzie. W grze została również dodana możliwość wyboru poziomu trudności.

## 2. Opis tworzonej gry

Plansza do gry została zbudowana przy wykorzystaniu tablicy dynamicznej, w której przechowujemy informacje o stanie gry. Rozmiar tablicy jest wybierany w zależności od wielkości planszy na jakiej chcemy grać. Program został wyposażony w funkcję, która sprawdza, czy nastąpiła wygrana, w zależności od wyboru ilości znaków w rzędzie. Cała gra jest sterowana przez funkcję o nazwie „rozgrywka”, która zarządza kolejnością ruchów gracza i komputera. Program sprawdza, czy gracz podał odpowiedni numer pola oraz czy dane pole nie jest już zajęte. Gdy dane pole jest wolne zostaje wstawiony symbol „X”, a następnie wywoływana zostaje funkcja szukająca najbardziej optymalnego ruchu dla komputera. Gra kończy się w przypadku wygranej komputera lub gracza, oraz gdy jest remis. Wybór ruchu dla komputera jest dokonywany za pomocą algorytmu MinMax z alfa-beta cięciami.

MinMax jest algorytmem stosowanym w prostych grach logicznych do wyznaczania optymalnych ruchów. Jest algorytmem rekurencyjnym, który wywołuje sam siebie do analizy kolejnych ruchów w grze. Algorytm dokonuje oceny stanu gry na danym poziomie. Stawia dla danego gracza wszystkie możliwe ruchy, a następnie wywołuje sam siebie rekurencyjnie z wyższym poziomem dla drugiego gracza. Algorytm zapamiętuje te ruchy, które maksymalizują dla danego gracza wartość stanu gry otrzymaną z wyższego poziomu. Jeśli graczem dla bieżącego poziomu jest A, to szuka on ruchów dających wartość większą od -10 (co jest wygraną dla gracza B). Jeśli graczem jest B, to szuka on ruchów dających wartość mniejszą od 10 (co jest wygraną dla gracza A). Po wyznaczeniu najlepszego ruchu zwracana jest jego wartość. Jeśli poziom jest równy zero, to znaleziony ruch jest wykonywany. W przypadku, gdy mamy możliwość osiągnięcia wygranej lub porażki, na kilka różnych sposobów wybór ruchów jest oceniany na podstawie poziomu głębokości (liczba ruchów do końca gry) w przypadku wygranej szukamy najszybszych ruchów (najmniej kombinacji), natomiast gdy przegrywamy, algorytm będzie starał się grać jak najdłużej to będzie możliwe.

W celu przyspieszenia algorytmu MinMax program został wyposażony w cięcia alfa-beta, redukujące liczbę węzłów, przeszukujących przez algorytm MinMax. Warunkiem stopu jest znalezienie przynajmniej jednego rozwiązania czyniącego obecnie badaną opcję ruchu gorszą od poprzednio zbadanych opcji. Wybranie takiej opcji ruchu nie przyniosłoby korzyści graczowi ruszającemu się, dlatego też nie ma potrzeby przeszukiwać dalej gałęzi drzewa tej opcji. Ta technika pozwala zaoszczędzić czas poszukiwania bez zmiany wyniku działania algorytmu.

### 3. Omówienie przebiegu eksperymentów

Gra została testowana dla różnych poziomów głębokości przeszukiwań. Program w każdym przypadku próbował osiągnąć jak najlepszy wynik, lecz gdy mamy poziom przeszukiwań na poziomie 2-3 ruchy do przodu, z łatwością jesteśmy w stanie pokonać SI, lecz gdy wybierzemy poziom przeszukiwań na 7 ruchów wygrana nie jest już możliwa, lecz tylko remis (istnieje wyjątek dla plansz większych niż 3x3 przy liczbie znaków w rzędzie potrzebnych do wygranej mniejszej bądź równej 3). Algorytm dąży do osiągnięcia jak najszybszej wygranej, lecz gdy wie, że już na pewno przegrywa to wtedy tak wykonuje swoje ruchy, aby gra trwała jak najdłużej.

Przykład gry na poziomie przeszukiwania 7 ruchów:

<pre> ---+---+---       ---+---+---       Gracz wybiera ruch : 5 ---+---+---   X   ---+---+---       Komputer wybiera ruch : 1 ===== O       ---+---+---   X   ---+---+---       Gracz wybiera ruch : 3 O     X ---+---+---   X   ---+---+---       Komputer wybiera ruch : 7 ===== O     X ---+---+---   X   ---+---+--- O       </pre>	<p>1</p> <p>2</p>	<pre> Gracz wybiera ruch : 4 O     X ---+---+--- X   X   ---+---+--- O       Komputer wybiera ruch : 6 ===== O     X ---+---+--- X   X   O ---+---+--- O       Gracz wybiera ruch : 8 O     X ---+---+--- X   X   O ---+---+--- O   X   Komputer wybiera ruch : 2 ===== O   O   X ---+---+--- X   X   O ---+---+--- O   X   Gracz wybiera ruch : 9 O   O   X ---+---+--- X   X   O ---+---+--- O   X   X REMIS!!! Jeszcze raz? (t -tak) </pre>	<p>3</p> <p>4</p> <p>5</p>
--	-------------------	--	----------------------------

## 4. Wnioski

Algorytm MinMax jest algorytmem, który w grach typu kółko i krzyżyk jest w stanie przewidzieć, ruchy gracza, i na ich podstawie wybrać dla siebie jak najbardziej optymalne rozwiązanie. Niestety szukanie najlepszego rozwiązania wiąże się z czasem potrzebnym do jego znalezienia. W celu przyspieszenia gry wprowadza się głębokość przeszukiwań, do jakiej program ma wywoływać się rekurencyjnie. Gdy wybierzemy za mały poziom przeszukiwań bez większego problemu jesteśmy w stanie wygrać z komputerem, lecz już na poziomie około 6-7 ruchów wygrana nie jest już możliwa. Kolejnym znacznym przyspieszeniem programu, które pozwala na zwiększenie poziomu przeszukiwań są cięcia alfa-beta. Po zastosowaniu ich w programie od razu można było zauważyć, że czas szukania najbardziej korzystnego rozwiązania dla komputera znacznie się skrócił.

## 5. Literatura:

- [https://pl.wikipedia.org/wiki/Algorytm\\_alfa-beta](https://pl.wikipedia.org/wiki/Algorytm_alfa-beta)
- [http://wazniak.mimuw.edu.pl/index.php?title=Sztuczna\\_inteligencja/SI\\_Modu%C5%82\\_8\\_-\\_Gry\\_dwuosobowe](http://wazniak.mimuw.edu.pl/index.php?title=Sztuczna_inteligencja/SI_Modu%C5%82_8_-_Gry_dwuosobowe)
- [http://eduinf.waw.pl/inf/utills/001\\_2008/0415.php](http://eduinf.waw.pl/inf/utills/001_2008/0415.php)
- <http://www.algorytm.org/praktyka/kolko-krzyzyk.html>
- [https://pl.wikipedia.org/wiki/Algorytm\\_min-max](https://pl.wikipedia.org/wiki/Algorytm_min-max)