

# **Łukasz Szarejko**

## **WSI\_21Z\_LAB3**

### **Treść zadania:**

Tematem trzecich ćwiczeń są dwuosobowe gry deterministyczne. Państwa zadaniem będzie napisanie programu / skryptu, który buduje drzewo zadanej gry a następnie gra sam ze sobą wykorzystując do tego algorytm minimax. Należałoby sprawdzić dwa przypadki:

- jeden z graczy gra w sposób losowy (tzn. nie używa naszego algorytmu) a drugi stara się optymalizować swoje ruchy (random vs minimax)
- obaj gracze podejmują optymalne decyzje (minimax vs minimax)

W raporcie należałoby przedstawić przykład wykonania programu wraz z odpowiadającymi stanami drzewa gry i wyborami algorytmu minimax. Zastosowanie przycinania alfa-beta nie jest częścią zadania, ale może okazać się potrzebne zależnie od otrzymanego typu gry. Proszę wykazać w raporcie wygraną każdej ze stron, czyli że gra nie jest ustawiona.

Python 3.9.7

Użyte biblioteki:

Pygame – wersja 2.0.3

Numpy – wersja 1.21.3

W swojej implementacji użyłem przycinania alfa beta aby program wykonywał się szybciej.

Gra została napisana tak, aby użytkownik widział jej działanie, czyli z interfejsem graficznym. Przy implementacji pominąłem animacje spadania żetonu, gdyż nie to było najważniejsze w zadaniu. Aby lepiej śledzić ruchy komputera, zalecane jest wpisanie w kodzie wykonującym ruchy (funkcja play w klasie Game) linijkę `sleep(0.4)` (Są one zakomentowane w celach przeprowadzania eksperymentów). Aby zmienić tryb gracza, przy jego konstrukcji należy podać atrybut 'minmax' lub 'easy'. Wartość 'easy' jest wartością domyślną. Gdy podamy wartość 'minmax', do konstruktora należy wtedy podać wartość głębokości drzewa  $\geq 1$ . Aby gra ruszyła należy użyć na obiekcie klasy Game funkcji `play()`, która zwróci gracza, który wygrał.

## Drzewo vs losowy dla różnych głębokości drzewa od 1 do 10

W każdym z poniższych doświadczeń, gra jest puszczana 25 razy.

Głębokość drzewa	Wygrane Drzewa	Wygrane Losowego	Remisy
1	23	0	2
2	18	5	2
3	20	2	3
4	19	4	2
5	18	4	3
6	17	6	2
7	21	2	2
8	19	3	3
9	20	4	1
10	15	10	0

Analizując wyniki, ciężko jest stwierdzić, dlaczego drzewo o większej głębokości (np. 10) ma tak słabszą skuteczność. Wydaje mi się, że jest to spowodowane jakiegoś rodzaju przeuczeniem i nie braniu pod uwagę tego, że drugi gracz wykonuje ruchy losowe a nie najlepsze z możliwych.

## Drzewo vs Drzewo

Jak wynikało z poprzedniego doświadczenia, zaimplementowany przeze mnie algorytm działa dość dobrze dla głębokości drzewa równej 7 lub 3.

W pojedynku drzewo vs drzewo użyję zatem tych głębokości, a gra będzie się wykonywała 100 razy.

Wyniki eksperymentu:

Głębokość drzewa	Wygrana gracza zaczynającego	Remis	Wygrana drugiego gracza
3	59	6	35
7	52	5	43

## Czy gra jest sprawiedliwa

Z powyższego doświadczenia wynika, że gra raczej nie jest sprawiedliwa, chociaż obaj gracze stosują algorytm to gracz rozpoczynający przeważnie wygrywa. W celu upewnienia się, czy to nie jest błędne działanie mojego algorytmu, sprawdziłem na stronach internetowych, że rzeczywiście gra jest ustawiona (pierwszy gracz grając optymalnie zawsze wygra), czyli nie możemy nazwać ją sprawiedliwą.

Niestety, zaimplementowany przeze mnie algorytm nie jest idealny. Często możemy zauważyć, że nie blokuje on ruchów zwycięskich przeciwnika, a dąży jedynie do swojego

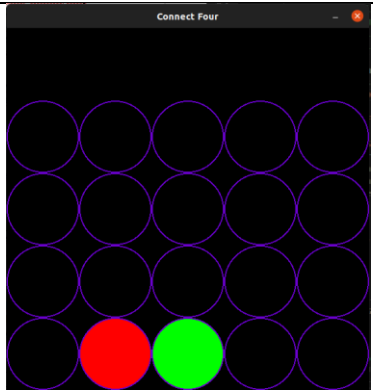
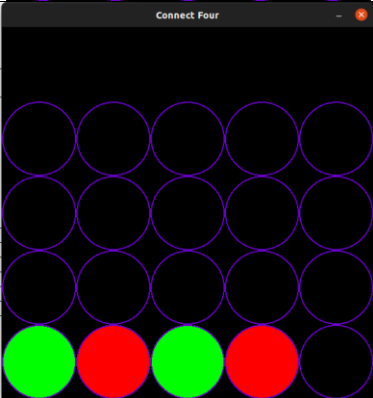
zwycięstwa. Stąd wynikają wyniki w tabeli powyżej. Nie jestem w stanie aktualnie dojść do rozwiązania tego problemu.

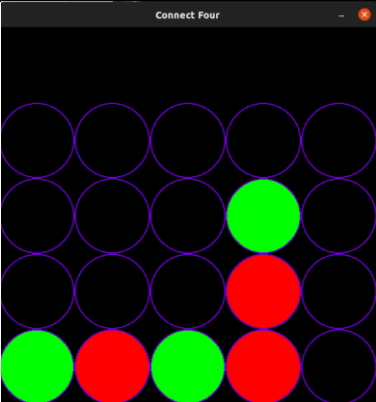
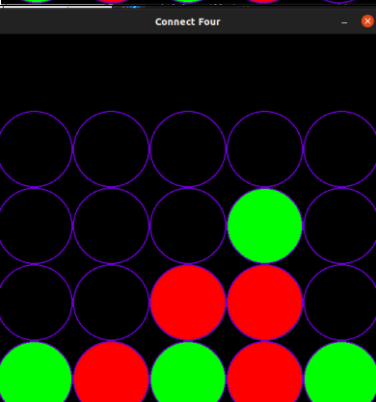
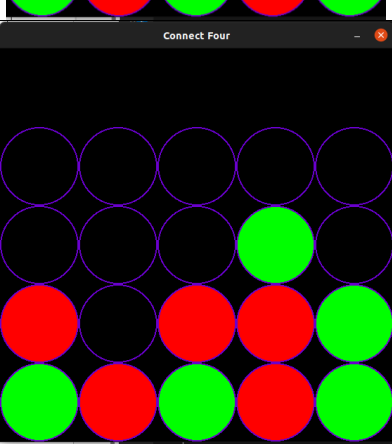
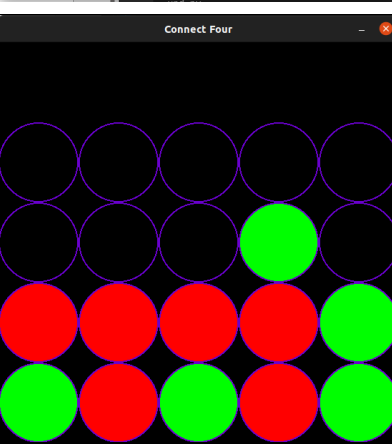
### **Analiza działania algorytmu na części przykładowej rozgrywki**

Gracz rozpoczynający to gracz czerwony, w każdym kroku widzieć będziemy też ruch drugiego gracza, gdyż dopiero wtedy zatrzymuję grę do analizy.

Warto też zaznaczyć w tym miejscu jak wyliczam heurystykę: 10pkt gdy gracz max ma 2 żetony na 4 polach w pionie, poziomie, diagonalnie oraz 2 wolne pola, 100 gdy 3 swoje i jedno wolne, 10000 gdy 4 swoje żetony. Gracz min jest liczony analogicznie, tylko, że z wartościami ujemnymi.

W tym eksperymencie, użyłem gracza minmax z głębokością równą 7.

Wyliczona wartość algorytmu minimax	Podgląd planszy	
10		
30		

100			
100			
10000			
10000 Gracz rozpoczynający wygrał			

Komentarz:

Jak możemy zauważyć, od samego początku gracz rozpoczynający jest lepiej rozstawiony, a w ciągu gry punkty obliczone przez algorytm rosną, do momentu, gdy jest już pewny wygranej. Następuje wtedy wykonanie ostatniego ruchu kończącego grę. Jest to zgodne z przewidywaniami, gdyż gra jest ustawiona, czyli przy wykonywaniu poprawnych ruchów przez gracza rozpoczynającego, dąży do jego zwycięstwa.