

Akademia Górniczo Hutnicza  
Wydział Informatyki  
Kierunek Informatyka- Data Science

Kraków, 13.02.2024

Języki i biblioteki analizy danych

Projekt zaliczeniowy

**„Kółko i krzyżyk z ciążeniem”**

Wojciech Burzak



## 1. Opis projektu:

Celem projektu było stworzenie interaktywnej gry „Kółko i krzyżyk z ciężeniem”.

Gra umożliwia rozgrywkę z komputerem. Program oferuje interfejs graficzny, umożliwiający wygodną rozgrywkę. Silnik ruchu komputera zaprojektowany jest tak aby podejmował mądre decyzje, minimalizując losowość ruchów.

## 2. Opis realizacji

Projekt „Kółko i krzyżyk z ciężeniem”, został zrealizowany zgodnie z koncepcją, posiada funkcjonalności zgodne z założeniami. Niezrealizowana została jedynie część projektu dzięki której możliwe byłoby sieciowe połączenie się z innymi komputerami w celu rozegrania turnieju i sprawdzeniu który algorytm działa najlepiej.

## 3. Wykorzystane biblioteki

- **Pygame** – biblioteka Pygame została użyta do stworzenia interaktywnego interfejsu graficznego gry. Obsługuje rysowanie planszy, wyświetlanie komunikatów, obsługę zdarzeń (takich jak kliknięcia myszą) oraz zarządzanie ekranem i jego odświeżaniem
- **NumPy** – biblioteka NumPy została wykorzystana do reprezentacji planszy gry jako dwuwymiarowej tablicy oraz do wykonywania operacji na tej tablicy, takich jak sprawdzanie możliwych ruchów, ocenianie pozycji na planszy oraz implementacja algorytmu Minimax
- **Sys** - biblioteka sys została wykorzystana do obsługi zdarzeń związanych z wyjściem z gry, takich jak zamknięcie okna gry
- **Math** - biblioteka math została wykorzystana w projekcie do zaokrąglania wartości liczbowych, np. podczas obliczania pozycji myszy w kolumnach planszy

#### 4. Przebieg gry

##### Inicjalizacja gry:

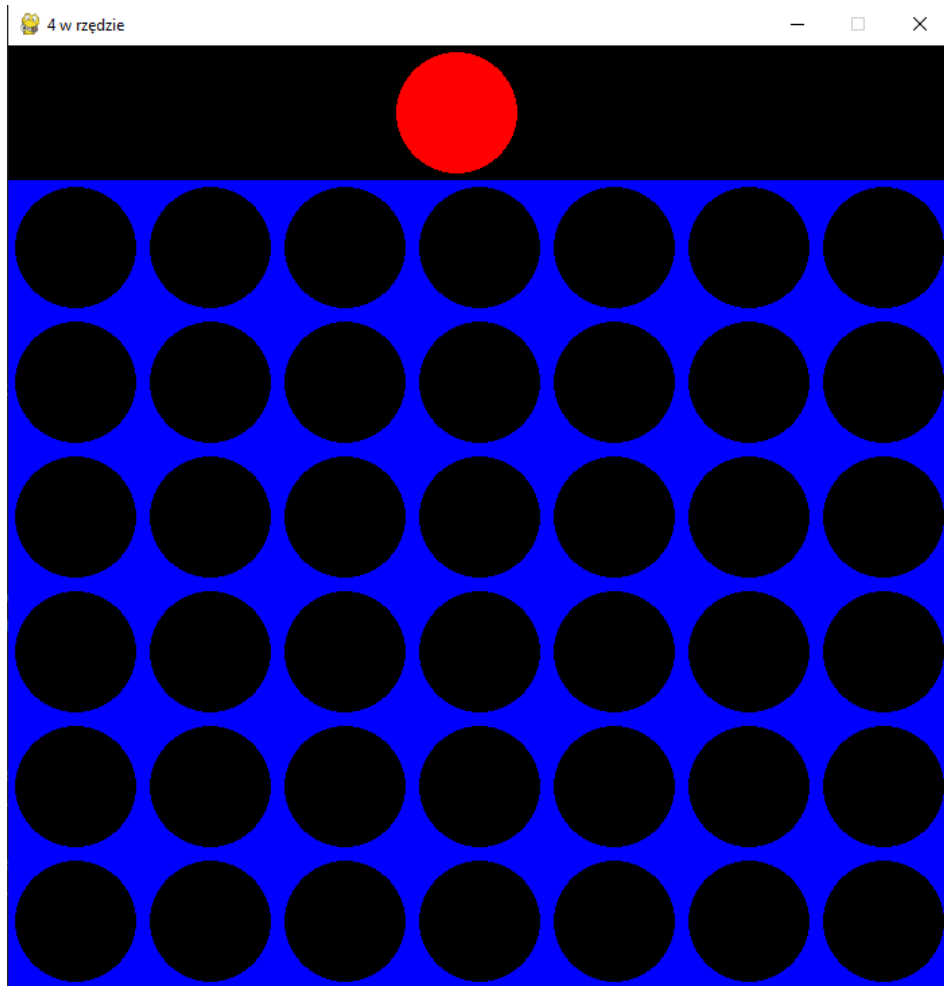
Rozpoczynając grę, użytkownik jest proszony o wybór rozpoczynającego rozgrywkę



Po kliknięciu w lewą część użytkownik rozpoczyna rozgrywkę (analogicznie komputer).

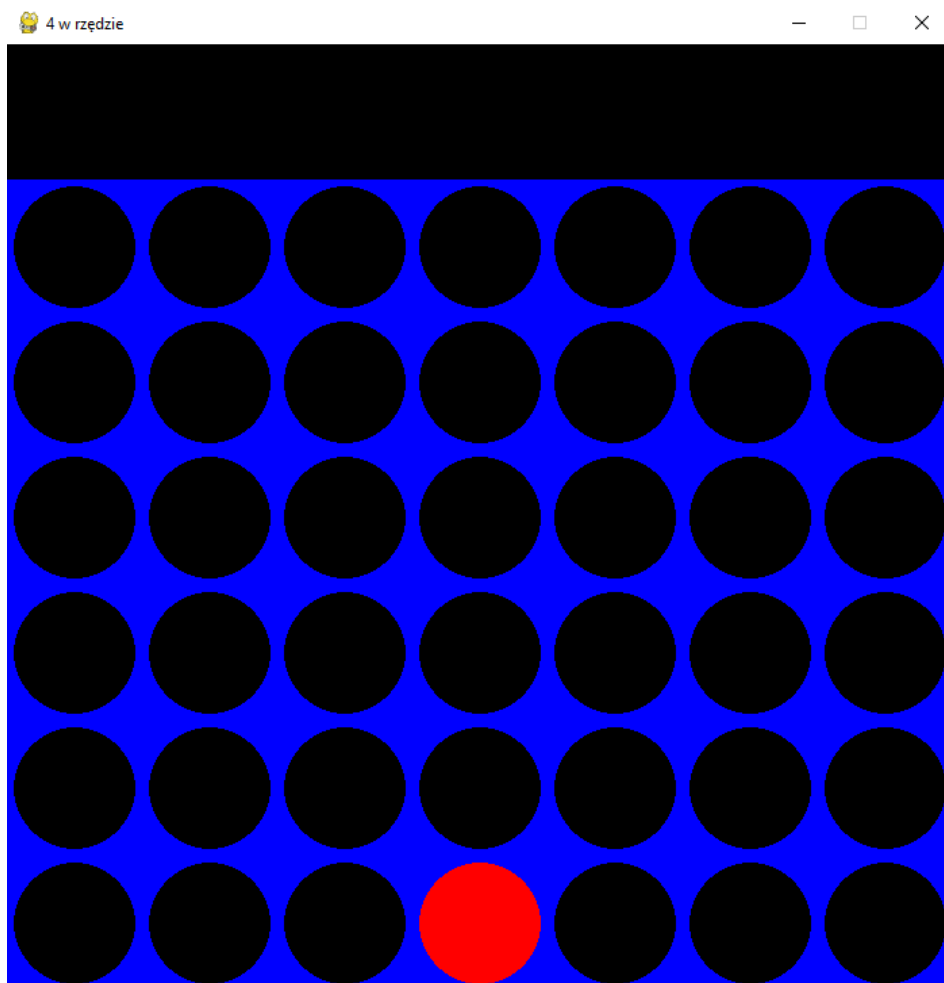
### Rysowanie planszy:

Plansza jest rysowana przy użyciu biblioteki Pygame i składa się z siatki kwadratów oraz kółek reprezentujących żetony. Gracz gra czerwonymi żetonami- komputer żółtymi.



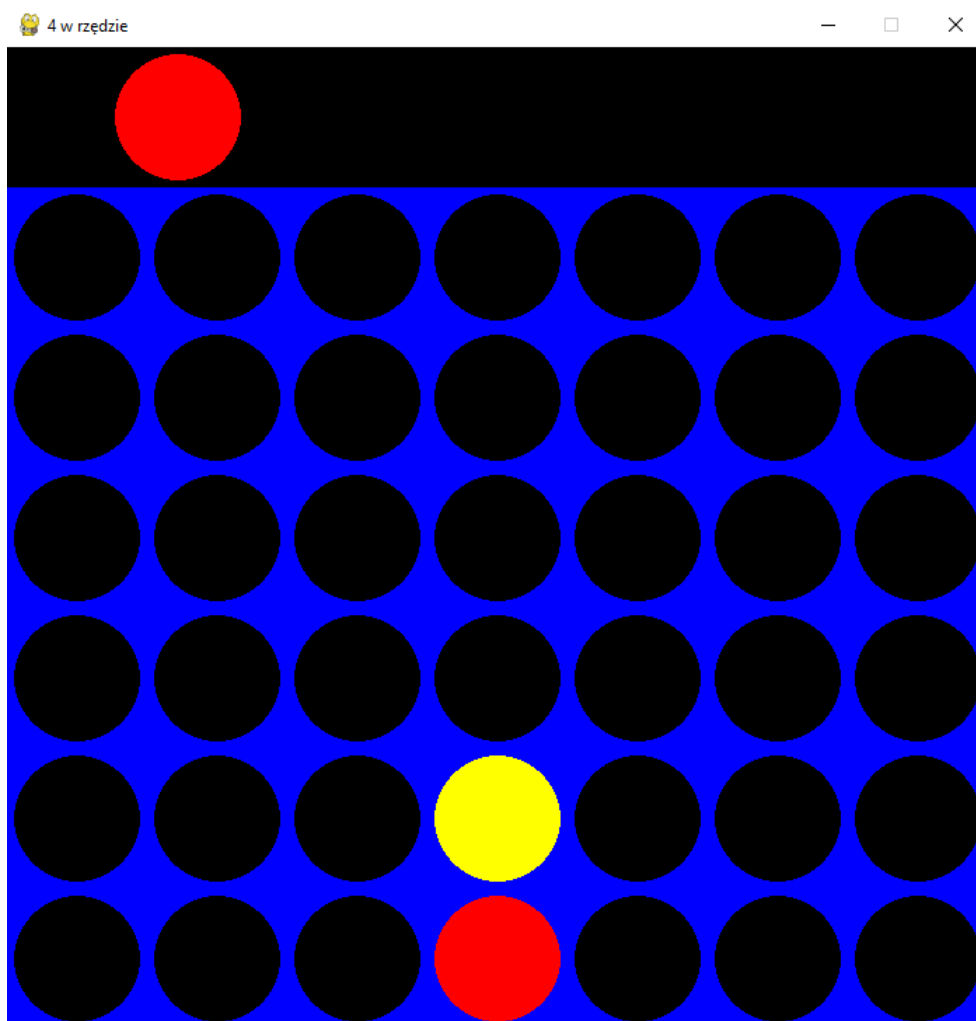
### Ruch gracza:

Gracz wykonuje ruch, klikając na wybraną kolumnę. Po wykonaniu ruchu plansza jest aktualizowana i sprawdzane jest, czy gracz wygrał.



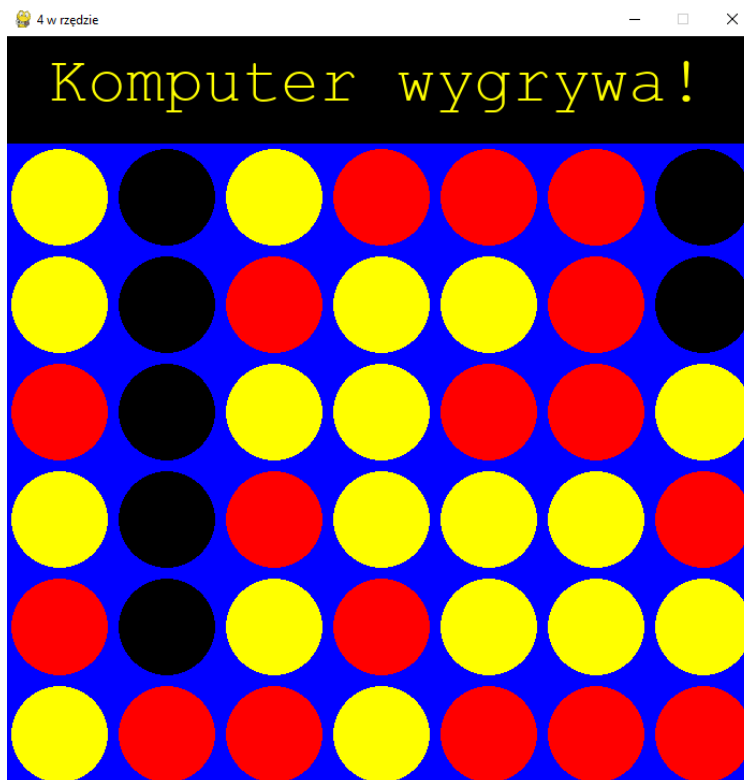
### Ruch komputera:

Komputer wykonuje ruch przy użyciu algorytmu Minimax z wykorzystaniem alfa beta obcięcia.



### Zakończenie gry:

Gra kończy się, gdy jeden z graczy ułoży cztery żetony w rzędzie lub plansza zostanie wypełniona. Wyświetlany jest komunikat o zwycięzcy i program się zamyka.



## 5. Algorytm Minimax z Obcięciem Alfa-Beta

Algorytm Minimax jest klasyczną metodą używaną w teorii gier i sztucznej inteligencji do określania optymalnych ruchów w grach dwuosobowych o sumie zerowej, takich jak Kółko i krzyżyk z ciężeniem. W kontekście tej gry, algorytm przeszukuje drzewo gry, aby znaleźć najlepszy możliwy ruch dla komputera, biorąc pod uwagę potencjalne ruchy przeciwnika. Algorytm wykonuje to przez symulowanie wszystkich możliwych sekwencji ruchów do określonej głębokości i ocenianie stanów planszy za pomocą funkcji heurystycznej.

### Przebieg algorytmu

Algorytm rozpoczyna z maksymalizującym graczem (komputer) i ustawia wartości alfa (najgorszy możliwy wynik dla maksymalizującego gracza) oraz beta (najgorszy możliwy wynik dla minimalizującego gracza) na nieskończoność odpowiednio minus i plus.

Algorytm rekursywnie przeszukuje drzewo gry, zmieniając graczy z maksymalizującego na minimalizującego i odwrotnie na każdym poziomie drzewa. Dla każdego węzła (stanu gry) algorytm dokonuje oceny za pomocą funkcji heurystycznej, jeśli osiągnięto liść drzewa gry lub głębokość przeszukiwania.

Aby zoptymalizować proces przeszukiwania, algorytm wykorzystuje obcięcie alfa-beta. Jeśli w dowolnym momencie przeszukiwania określony ruch prowadzi do gorszego wyniku niż już znaleziony (dla maksymalizującego gracza jest to niższy niż wartość alfa, a dla minimalizującego niższy niż wartość beta), dalsze przeszukiwanie tego ruchu jest przerywane. Pozwala to znacznie zredukować liczbę węzłów, które muszą być eksplorowane.

Po przeszukaniu drzewa gry lub osiągnięciu zadanego limitu głębokości, algorytm wybiera ruch, który maksymalizuje szanse komputera na wygraną, biorąc pod uwagę potencjalne odpowiedzi przeciwnika.

### Funkcje oceny

`score_position(board, piece)` – funkcja ocenia stan planszy z perspektywy danego gracza. Używa heurystyki, takie jak ilość potencjalnych czwórek, które gracz może utworzyć, preferencje dla środkowych kolumn i blokowanie ruchów przeciwnika.

`evaluate_window(window, piece)` - funkcja ocenia segmenty planszy (okna czterech kolejnych pól), przyznając punkty za potencjalne możliwości wygranych i blokowania ruchów przeciwnika.



## **6. Podsumowanie i wnioski**

Z założeń projektu nie udało zrealizować się komunikacji sieciowej. Najlepszym kierunkiem dla rozwoju programu byłoby dodanie tego elementu. Możliwym ulepszeniem byłoby także ulepszenie interfejsu graficznego- stworzenie menu opcji oraz lepszej komunikacji z użytkownikiem. Największym wyzwaniem okazało się stworzenie grającej na wysokim poziomie funkcji odpowiedzialnych za ruch komputera z użyciem algorytmu Min-Max.