

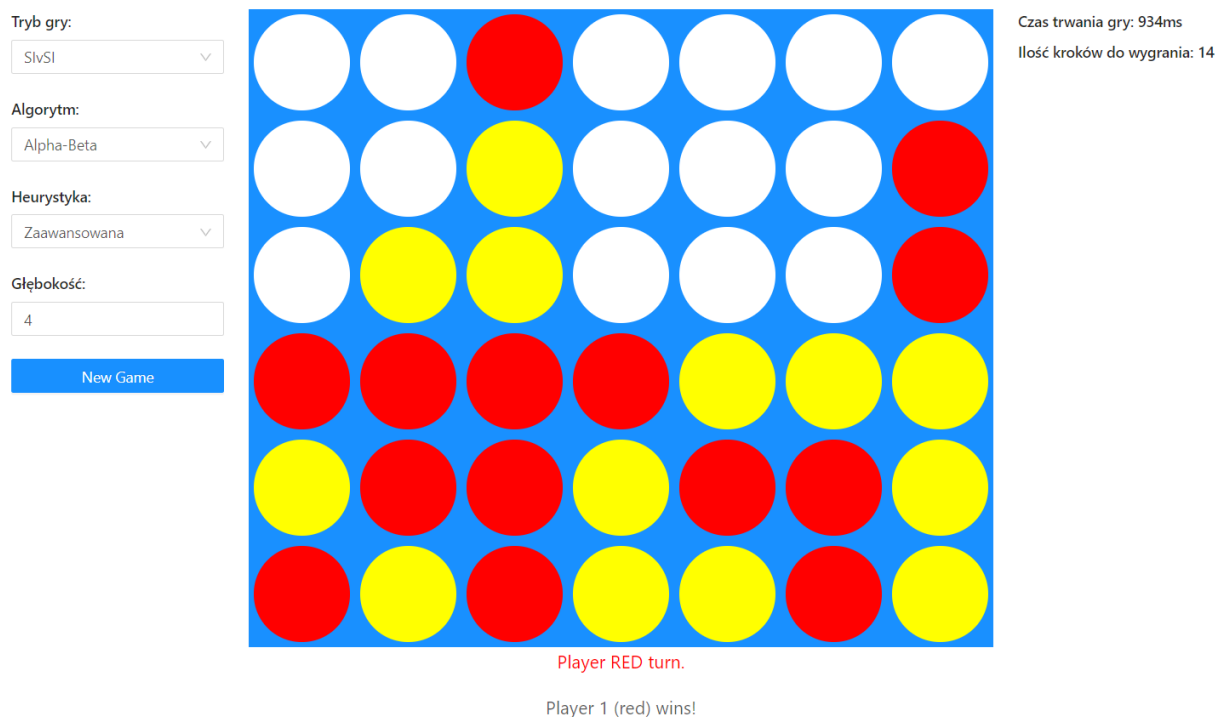
1. Wstęp

W ramach rozbudowania aplikacji tworzonej w ramach trzeciej listy laboratoryjnej zaimplementowana została druga, bardziej zaawansowana heurystyka. Spis działających funkcjonalności zawiera wszystkie wymagane funkcjonalności dla tej listy laboratoryjnej, to jest:

- Algorytm min-max
- Algorytm min-max z trybem cięć alpha-beta
- Trzy tryby rozgrywki:
 1. Gracz kontra gracz
 2. Gracz kontra komputer
 3. Komputer kontra komputer
- Interfejs graficzny aplikacji pozwalający na zmianę parametrów w trakcie działania gry, w tym trybu rozgrywki, algorytmu, głębokości i heurystyki
- Dwie heurystyki

Zarówno min-max jak i alpha-beta działają teraz znacznie szybciej niż ostatnio przez wprowadzenie odpowiednich poprawek.

Rozbudowany o nowe funkcjonalności interfejs graficzny wygląda teraz następująco:



2. Opis działania heurystyk

Zaimplementowana została nowa heurystyka, znacznie bardziej złożona od poprzedniej. Pierwotna, prosta heurystyka sprawdzała tylko ruchy wygrywające, przegrywające i remis. Nowa heurystyka również sprawdza ruchy wygrywające, ale w przypadku gdy głębokość (depth) przyjmie wartość 0, algorytm zwraca wynik zależny od specjalnej funkcji oceny planszy. Funkcja bierze pod uwagę ilość pozycji zajmowanych przez gracza w środkowej kolumnie, oraz sprawdza wszystkie kombinacje (nazywane dalej „okienkami”) ułożenia czterech pozycji kolejno w poziomie, pionie i na obu skosach. Każde „okienko” dostaje własną ocenę, zależnie od ilości pionków gracza, pustych oraz pionków przeciwnika. Pseudokod pokazujący działanie funkcji oceny:

```
Jeżeli playerFieldsCount == 4:
```

```
    score += 100
```

```
Jeżeli playerFieldsCount == 3 && emptyFieldsCount == 1:
```

```
    score += 5
```

```
Jeżeli playerFieldsCount == 2 && emptyFieldsCount == 2:
```

```
    score += 2
```

```
Jeżeli opponentFieldsCount == 3 && emptyFieldsCount == 1:
```

```
    score -= 4
```

3. Porównanie czasów działania dla prostej heurystyki

Na początku porównamy algorytm min-max z i bez cięć alpha-beta dla prostej heurystyki i głębokości = 6.

MinMax:

Tryb gry:

SlvSI

Algorytm:

MinMax

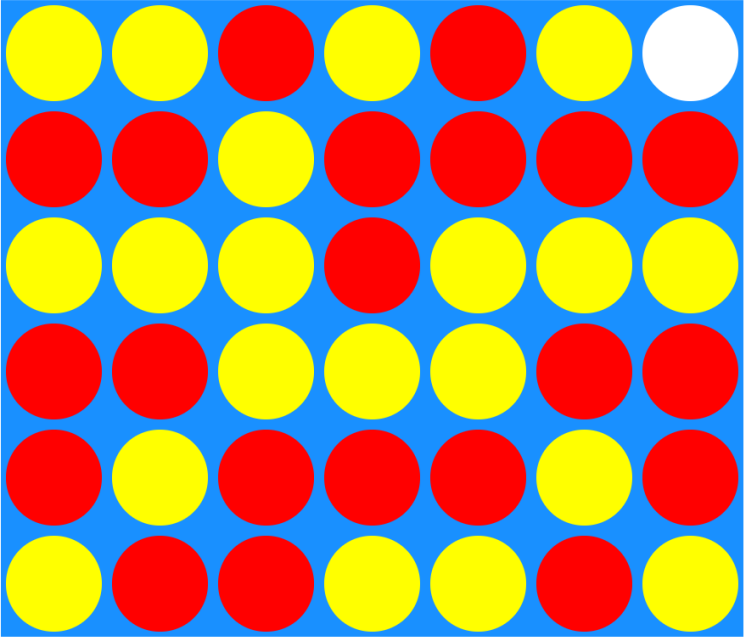
Heurystyka:

Prosta

Głębokość:

6

New Game



A 7x7 Connect Four board. Columns 1-6 are full. Column 7 has one empty space at the top. Row 1: Yellow, Yellow, Red, Yellow, Red, Yellow, White. Row 2: Red, Red, Yellow, Red, Red, Red, Red. Row 3: Yellow, Yellow, Yellow, Red, Yellow, Yellow, Yellow. Row 4: Red, Red, Yellow, Yellow, Yellow, Red, Red. Row 5: Red, Yellow, Red, Red, Red, Yellow, Red. Row 6: Yellow, Red, Red, Yellow, Yellow, Red, Yellow.

Czas trwania gry: 13234ms

Ilość kroków do wygrania: 21

Player RED turn.

Player 1 (red) wins!

Czas trwania gry: 13234ms

Ilość kroków do wygrania: 21

MinMax z trybem cięć alpha-beta:

Tryb gry:

Algorytm:

Heurystyka:

Głębokość:

[New Game](#)

Player YELLOW turn.

Player 2 (yellow) wins!

Czas trwania gry: 1004ms
Ilość kroków do wygrania: 17

Czas trwania gry: 1004ms

Ilość kroków do wygrania: 17

Zgodnie z oczekiwaniami tryb cięć alpha-beta bardzo przyspiesza działanie algorytmu, bo ponad 10 krotnie.

Następnie porównamy wpływ głębokości na czas wykonywania obliczeń zmieniając jej wartość z 6 na 7, dla algorytmów kolejno MinMax i Alpha-Beta. Dalsze głębokości nie zostaną pokazane jako że zwykły MinMax jest na tyle wolny, że na wyniki trzeba czekać bardzo długo.

MinMax dla głębokości 7:

Tryb gry:

SlvSI ▾

Algorytm:

MinMax ▾

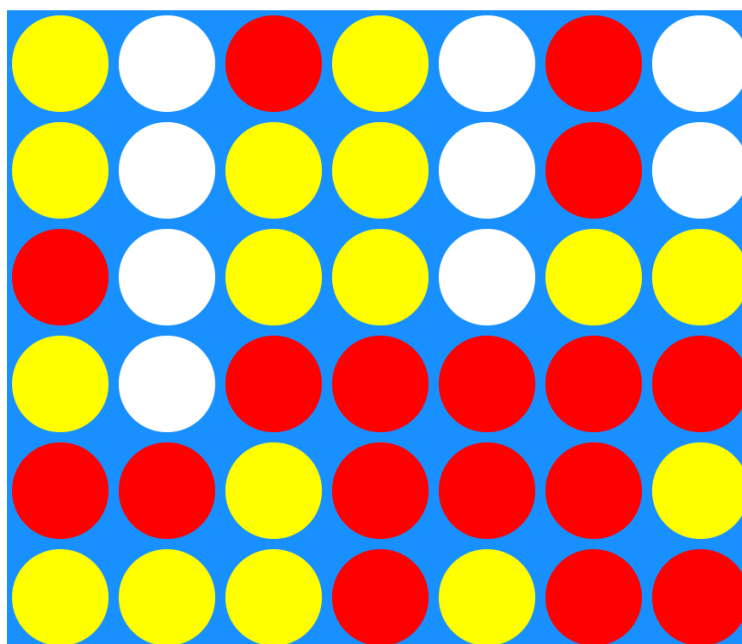
Heurystyka:

Prosta ▾

Głębokość:

7

New Game



Czas trwania gry: 94561ms

Ilość kroków do wygrania: 17

Player RED turn.

Player 1 (red) wins!

Czas trwania gry: 94561ms

Ilość kroków do wygrania: 17

MinMax z trybem cięć alpha-beta dla głębokości 7:

Tryb gry:

Algorytm:

Heurystyka:

Głębokość:

Player RED turn.

Player 1 (red) wins!

Czas trwania gry: 977ms

Ilość kroków do wygrania: 17

Tu przewaga zwiększa się coraz bardziej, jako że zwiększenie głębokości powyżej 6 nie daje żadnych widocznych korzyści w kwestii samego działania algorytmu, a znacznie wydłuża czas działania algorytmu. Alpha-Beta poprawnie ucina zbędne gałęzie przez co czas wykonania nie zmienił się względem wywołania dla głębokości 6 i tym razem jest już nie 10, a prawie 100 razy szybszy od algorytmu MinMax. Przewiduje się więc że różnica ta będzie się jeszcze bardziej zwiększać wraz ze zwiększaniem głębokości, głębokości wyższe niż 7 nie zostaną jednak ujęte w sprawozdaniu jako że czas wymagany do ich obliczeń jest zbyt duży.

MinMax z trybem cięć alpha-beta dla głębokości 10:

Tryb gry:

Algorytm:

Heurystyka:

Głębokość:

[New Game](#)

Player YELLOW turn.

Player 2 (yellow) wins!

Czas trwania gry: 61241ms

Ilość kroków do wygrania: 15

Czas trwania gry: 61241ms

Ilość kroków do wygrania: 15

By pokazać jak dużo zmienia tryb cięć alpha-beta sprawdzono działanie algorytmu dla głębokości 10. Zachowując wcześniejszą tendencję przyrostu czasu wykonywania MinMax powinien dla tej głębokości wykonywać się 11 godzin. Tryb cięć alpha-beta powoduje, że obliczenia wykonują się w akceptowalnym czasie, nie tracąc jednocześnie na jakości działania algorytmu.

4. Porównanie czasów działania dla zaawansowanej heurystyki

Następnie wszystkie poprzednie badania zostaną powtórzone dla nowo zaimplementowanej, bardziej złożonej heurystyki.

MinMax dla głębokości 6:

Tryb gry:

Algorytm:

Heurystyka:

Głębokość:

Czas trwania gry: 267574ms

Ilość kroków do wygrania: 21

Czas trwania gry: 267574ms

Ilość kroków do wygrania: 21

Nowa heurystyka jest zauważalnie wolniejsza, chociaż z testów funkcjonalnych przeprowadzonych w trybie „Gracz kontra komputer” okazała się ona być znacznie lepsza – ani razu nie udało się z nią wygrać, co było jednak wykonalne w przypadku prostszej heurystyki.

MinMax z trybem cięć alpha-beta dla głębokości 6:

Tryb gry:

SilvSI

Algorytm:

Alpha-Beta

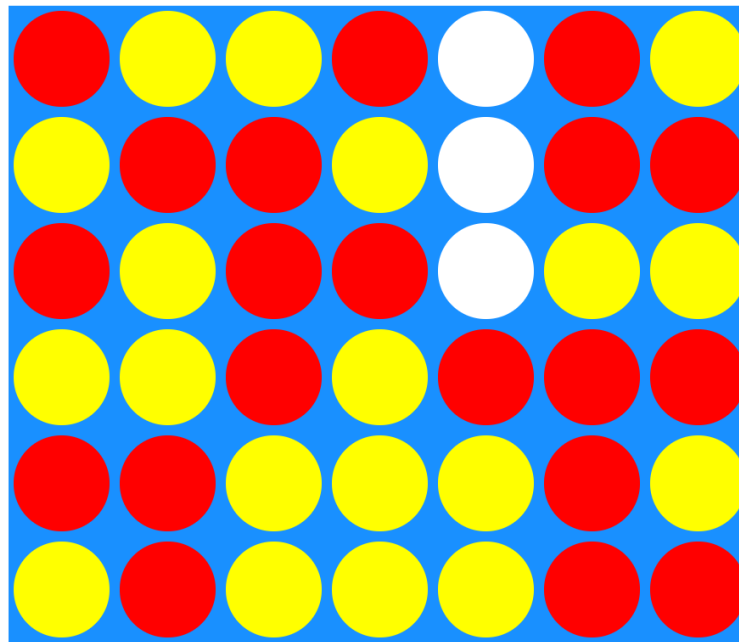
Heurystyka:

Zaawansowana

Głębokość:

6

New Game



Czas trwania gry: 4813ms
Ilość kroków do wygrania: 20

Player RED turn.

Player 1 (red) wins!

Czas trwania gry: 4813ms

Ilość kroków do wygrania: 20

Tryb cięć alpha-beta znów okazuje się być nieporównywalnie szybszy od zwykłego MinMaxa, tym razem dając wynik około 55 razy lepszy.

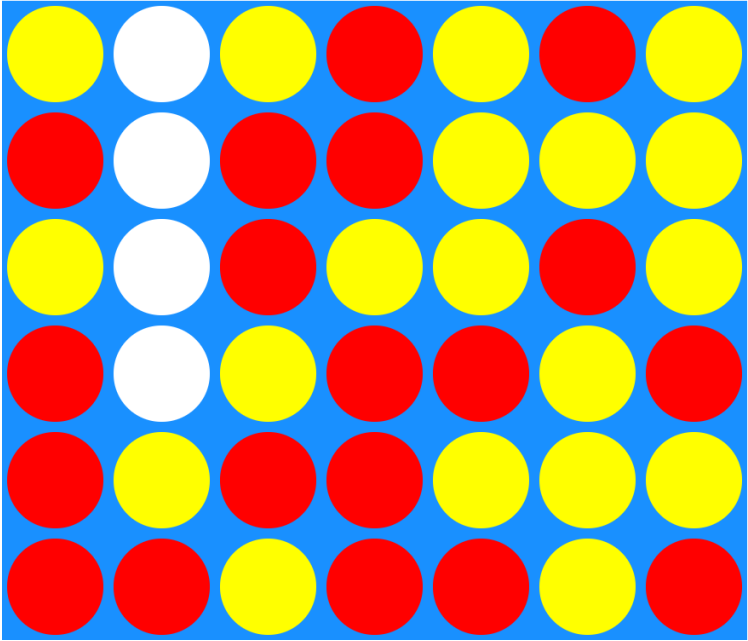
MinMax dla głębokości 5:

Tryb gry:

Algorytm:

Heurystyka:

Głębokość:



Player YELLOW turn.

Player 2 (yellow) wins!

Czas trwania gry: 37666ms

Ilość kroków do wygrania: 19

Czas trwania gry: 37666ms

Ilość kroków do wygrania: 19

Warto zaznaczyć również, że heurystyka zaawansowana zachowuje się bardzo dobrze również dla głębokości 5 i to właśnie dla tej głębokości przeprowadzane były wcześniejsze testy funkcjonalne w trybie „Gracz kontra komputer”.

MinMax z trybem cięć alpha-beta dla głębokości 5:

Tryb gry:

Algorytm:

Heurystyka:

Głębokość:

[New Game](#)

Player RED turn.

Player 1 (red) wins!

Czas trwania gry: 6117ms

Ilość kroków do wygrania: 19

W tym przypadku odnotowana została najmniejsza zmiana na korzyść alpha-beta jak dotąd, bo okazał się on być tylko 6 razy szybszy.

Przez to jak długo wykonuje się ta heurystyka nie będzie ona badana dla głębokości większych od 6 w zwykłym trybie MinMax.

MinMax z trybem cięć alpha-beta dla głębokości 10:

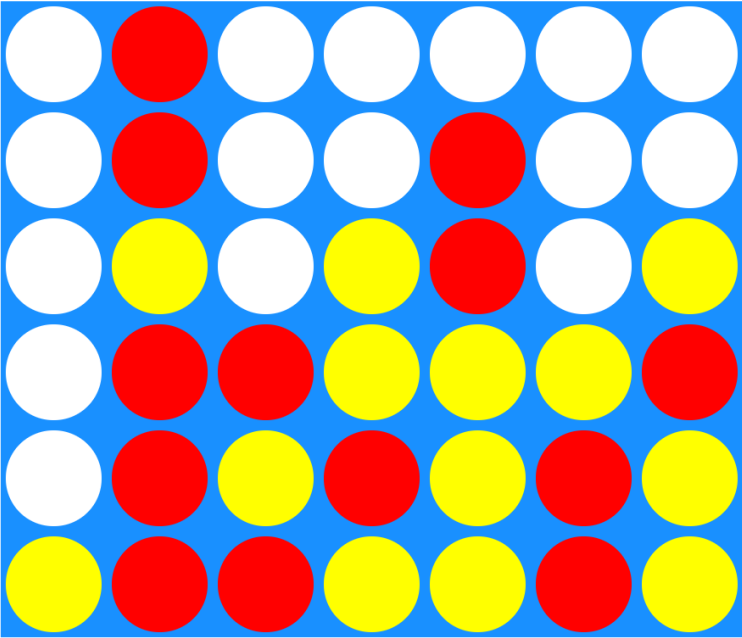
Tryb gry:

Algorytm:

Heurystyka:

Głębokość:

New Game



Player YELLOW turn.

Player 2 (yellow) wins!

Czas trwania gry: 802479ms

Ilość kroków do wygrania: 13

Analogicznie do wywołania alpha-beta i głębokości 10 dla heurystyki prostej, znów alpha-beta pozwala uzyskać wyniki w akceptowalnym czasie, bo około 13 minut.

5. Wnioski

Tryb cięć alpha-beta znacznie skraca czas obliczeń nie pogarszając jakości działania algorytmu.

W celu poprawy działania algorytmu i zwiększenia poziomu trudności należy skupić się na heurystyce. Bardziej złożona heurystyka okazała się być wolniejsza, jednak nie udało się z nią ani razu wygrać w testach funkcjonalnych w trybie „Gracz kontra komputer”. Różne heurystyki są więc dobrym sposobem na implementację różnorodnych poziomów trudności gry.

Innym sposobem dostosowania poziomu trudności gry jest zmiana głębokości na którą schodzi algorytm MinMax, jako że powoduje to zmianę ilości przewidzianych przez sztuczną inteligencję ruchów, co może albo zmniejszyć albo zwiększyć trudność rozgrywki.