


Sprawozdanie z gry			
 <p>Wrocław University of Science and Technology</p>	<b>Student:</b>  <i>Jakub Król</i> 226269	<b>Data Laboratoriów:</b> 31.05.2017r. g.16:30  <b>Wykonano:</b> 20.06.2017r.	<b>Prowadzący:</b> <i>Dr inż. Krzysztof</i> <i>Halawa</i>
		<b>Grupa laboratoryjna:</b> E02-18p	<b>Ocena:</b>

## Wstęp:

Zadaniem było zaimplementowanie **algorytmu minmax**. Algorytm ten zastosowano w grze **Tick Tack Toe** czyli **kółko i krzyżyk**. Autor w dalszej części dokumentu będzie prezentował program w **języku C/C++**. Pomimo jednego pliku, dla łatwiejszej kompilacji, stworzono plik makefile.

W programach są używane wskaźniki oraz zmienne dynamiczne, aby sprawdzić czy nie następują wycieki pamięci posłużono się programem valgrind z opcją leak-check=full.

Program nie korzysta z STL zgodnie z zaleceniami wykładowcy.

# Program

6. Kółko i krzyżyk (5) + 0.5 za wersję graficzną  
Gracz powinien posiadać możliwość definiowania rozmiaru pola (kwadratowego) wraz z ilością znaków w rzędzie.

## Opis:

Wszelkie algorytmy zawarte w programie są przystosowane do rozmiarów  $n \times n$ , jednak z powodu mało wydajnej maszyny do testowania – ograniczono się do rozmiarów w zakresie od 3 do 9 jednocześnie ograniczając „głębokość” obliczeń.

Wygrana z AI na planszy 3x3 jest niemożliwa, można odprowadzić tylko do remisu

Gra jest zabezpieczona przed złośliwym użytkownikiem.

Nie zastosowano trybu graficznego, cała implementacja jest w konsoli.

## Kod:

[https://github.com/Rexluu/PAMSI/tree/Final\\_versions/game](https://github.com/Rexluu/PAMSI/tree/Final_versions/game)

## Testy:

### - Plansza 3x3

Wprowadz rozmiar planszy: 3  
Wprowadz ilosc znakow wygrywajacych: 3

	A.	B.	C.
0.			
1.			
2.			

Twój ruch [np. A0]: B1

	A.	B.	C.
0.	0		
1.		X	
2.			

Twój ruch [np. A0]: C0

	A.	B.	C.
0.	0		X
1.		X	
2.	0		

Twój ruch [np. A0]: A1

	A.	B.	C.
0.	0		X
1.	X	X	0
2.	0		

Twój ruch [np. A0]: B2

	A.	B.	C.
0.	0	0	X
1.	X	X	0
2.	0	X	

Twój ruch [np. A0]: C2

	A.	B.	C.
0.	0	0	X
1.	X	X	0
2.	0	X	X

R-E-M-I-S !

- Plansza 3x3, przegrana

- - - - -  
 Kołko i krzyżyk  
 - - - - -

.....

Twoj ruch [np. A0]: A0

.....

Twoj ruch [np. A0]: B0

.....

Twoj ruch [np. A0]: A1

.....

Niestety, przegrales!

- Plansza 4x4, wygrywaja 4

- - - - -  
Kolko i krzyzyk  
- - - - -

Wprowadz rozmiar planszy: 4

Wprowadz ilosc znakow wygrywajacych: 4

- - - - -

	A.	B.	C.	D.
0.				
1.				
2.				
3.				

Twój ruch [np. A0]: A3

- - - - -

	A.	B.	C.	D.
0.	0			
1.				
2.				
3.	X			

Twój ruch [np. A0]: B3

- - - - -

	A.	B.	C.	D.
0.	0	0		
1.				
2.				
3.	X	X		

Twój ruch [np. A0]: C3

- - - - -

	A.	B.	C.	D.
0.	0	0		
1.				
2.				
3.	X	X	X	0

Twój ruch [np. A0]: ■

- Plansza 9x9, wygrywają 3

- - - - -

	A.	B.	C.	D.	E.	F.	G.	H.	I.
0.	0	0							
1.						X			
2.						X			
3.						X			
4.									
5.									
6.									
7.									
8.									

Gratulacje, wygrałeś!

Powody wygranej są dwa:

- Ograniczenie głębokości liczenia
- Przy takich ustawieniach po prostu co by nie zrobił komputer i tak można wygrać,

## Zabezpieczenia:

```
- - - - -
      Kolko i krzyzyk
- - - - -
```

Wprowadz rozmiar planszy: -1  
Wprowadz ilosc znakow wygrywajacych: 0

Rozmiar planszy musi zawierać się w przedziale 3-9!

```
- - - - -
      Kolko i krzyzyk
- - - - -
```

Wprowadz rozmiar planszy: 2  
Wprowadz ilosc znakow wygrywajacych: 3

Rozmiar planszy musi zawierać się w przedziale 3-9!

```
- - - - -
      Kolko i krzyzyk
- - - - -
```

Wprowadz rozmiar planszy: 3  
Wprowadz ilosc znakow wygrywajacych: 4

Liczba znakow wygrywajacych nie moze byc wieksza od rozmiaru planszy i musi byc wieksza od 2!

```
- - - - -
      Kolko i krzyzyk
- - - - -
```

Wprowadz rozmiar planszy: 4  
Wprowadz ilosc znakow wygrywajacych: -1

Liczba znakow wygrywajacych nie moze byc wieksza od rozmiaru planszy i musi byc wieksza od 2!

```
- - - - -
```

```
- - - - -
      A.  B.  C.
0. | X |   |   |
1. |   | 0 |   |
2. |   |   |   |
```

Twój ruch [np. A0]: A3  
Podano zły ruch!

Twój ruch [np. A0]: A0  
Nie mozesz wykonac tego ruchu!

Twój ruch [np. A0]: B1  
Nie mozesz wykonac tego ruchu!



## Valgrind:

- - - - -

	A.	B.	C.
0.	X	X	0
1.	X	0	
2.	0		

Niestety, przegrales!

==7995==

==7995== HEAP SUMMARY:

==7995== in use at exit: 0 bytes in 0 blocks

==7995== total heap usage: 16 allocs, 16 frees, 240 bytes allocated

==7995==

==7995== All heap blocks were freed -- no leaks are possible

==7995==

==7995== For counts of detected and suppressed errors, rerun with: -v

==7995== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)