



Projekt z przedmiotu Programowanie współbieżne i rozproszone

Katarzyna Pyrczak, Michał Kowalczyk

Informatyka, III rok, rok akademicki 2020/2021

Temat: Symulacja gry planszowej "Kółko i krzyżyk"

Spis treści

1	Dane studentów i data oddania	2
2	Cel projektu	2
3	Diagram kooperacji klas	3
4	Opis i schemat struktury zadaniowej programu	4
5	Użyte niestandardowe pakiety	4
6	Metody rozwiązania problemu	4
7	Instrukcja obsługi	4
8	Testy	6
9	Możliwe rozszerzenia programu	7
10	Ograniczenia programu	7

1 Dane studentów i data oddania

Katarzyna Pyrczak

Numer albumu: 309148

Grupa: 2b

Michał Kowalczyk

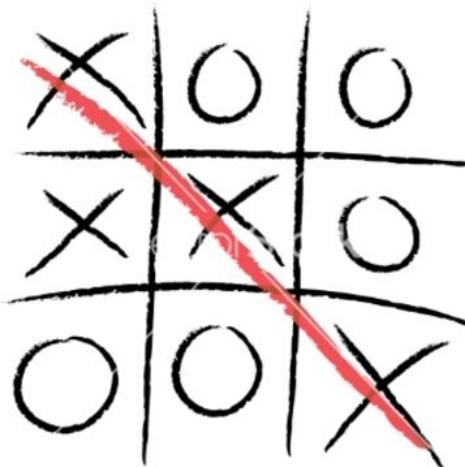
Numer albumu: 297875

Grupa: 2a

Data oddania: 02.02.2021

2 Cel projektu

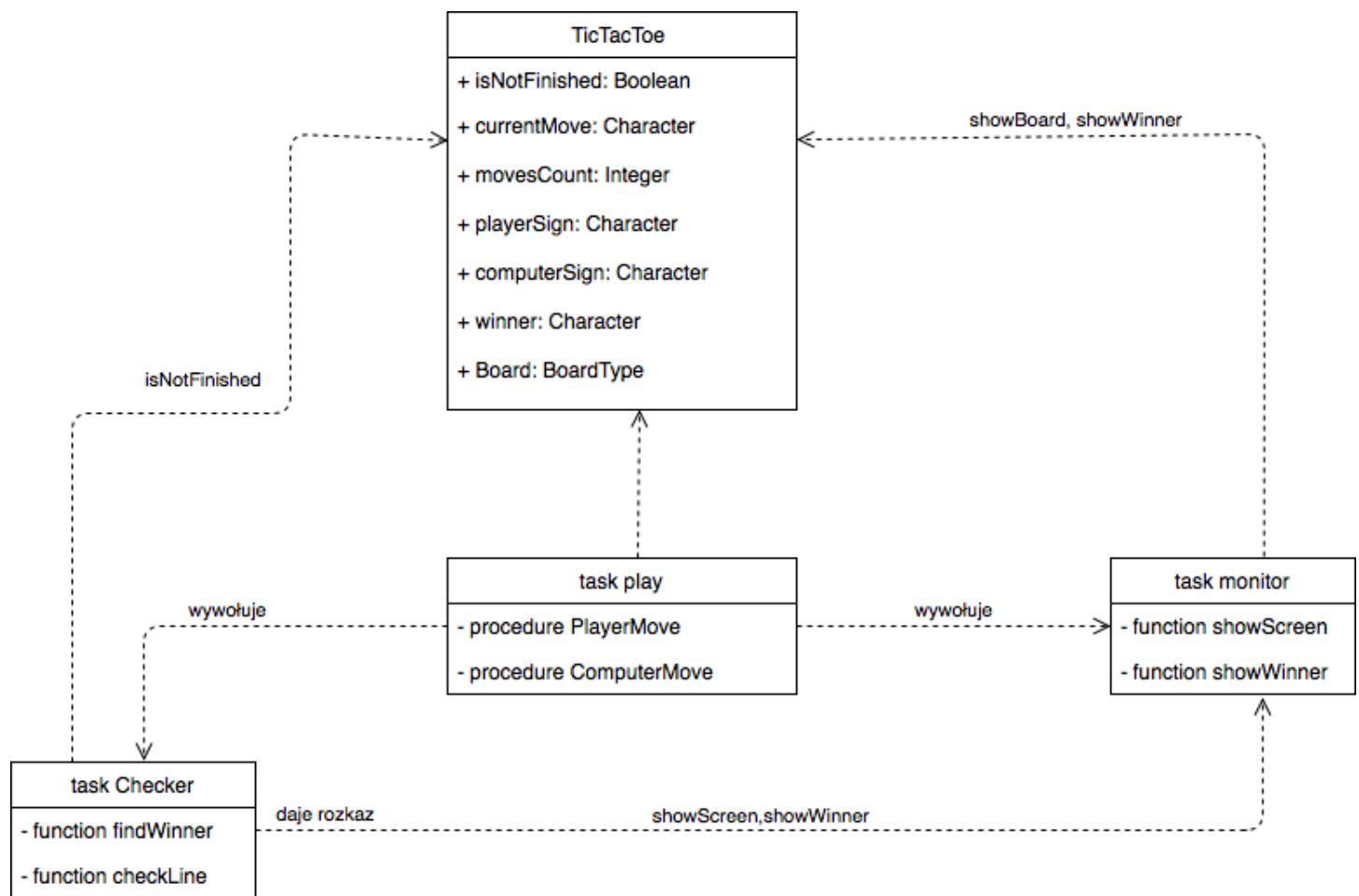
Celem projektu było zainspirowanie popularnej gry planszowej "kółko i krzyżyk". W takiej grze udział bierze dwóch graczy. Jednemu z nich zostaje przypisany symbol kółka "o", a drugiemu krzyżyka "x". Gracze na zmianę dodają w wolne miejsce planszy swój symbol i dążą do tego, aby ich znak wypełnił cały wiersz, kolumnę bądź przekątną. Komu uda się to jako pierwszemu, ten wygrywa. Przeciwnicy mogą również działać na swoją niekorzyść, tj. blokować drugiej osobie szybsze osiągnięcie celu.



Rysunek 1: Schemat gry, źródło: <https://board-games-galore.fandom.com/wiki/Tic-tac-toe>

W grze takiej wymagana jest komunikacja między graczami, co można zobrazować w języku Ada za pomocą komunikacji pomiędzy poszczególnymi taskami (zadaniami).

3 Diagram kooperacji klas



4 Opis i schemat struktury zadaniowej programu

Struktura programu została podzielona pomiędzy zadania, procedury oraz funkcje. Działanie programu polega na tym, że na zmianę - gracz oraz "komputer" wybierają nowe pozycje dla swojego symbolu. Task monitor odpowiada za wyświetlanie aktualnej planszy oraz zwycięzcy, o ile taki w danym momencie jest. Za sprawdzanie, czy któryś z graczy wygrał odpowiada z kolei task Checker. Korzysta on z funkcji findWinner, która za pomocą funkcji checkLine sprawdza, czy któryś z przeciwników zapisał cały wiersz, kolumnę bądź przekątną planszy. Gra trwa dopóki zmienne movesCount nie osiągnie swojej maksymalnej wartości, co znaczy, że już wszystkie pola na planszy są zapelnione. Task play za zadanie ma wywoływanie na przemian procedur PlayerMove oraz ComputerMove, które jak sama nazwa wskazuje, odpowiadają kolejnym ruchom gracza i komputera. Wybranie nowych pozycji powoduje, że plansza Board się zmienia, co musi zostać przedstawione przez task monitor.

5 Użyte niestandardowe pakiety

Do niestandardowych pakietów użytych przez nas można zaliczyć użycie ANSI escape sequences.

6 Metody rozwiązania problemu

Rozwiązaniem koncepcji działania gry planszowej jest podział działań takich jak pobieranie nowej pozycji dla symbolu od gracza, sprawdzanie, czy któryś z graczy został już zwyciężą, wyświetlanie zawartości planszy na części realizowane przez osobne taski, które się ze sobą komunikują.

7 Instrukcja obsługi

Nasz projekt jest kompatybilny ze środowiskiem Linux. Na początku należy program skompilować poprzez wpisanie w terminalu **\$ gnatmake main.adb**. Po skompilowaniu należy uruchomić główną pętle programu - na Linuxie poprzez komendę **\$./main**, a na Windowsie **\$./main.exe**

Po uruchomieniu programu użytkownik zostaje poproszony o wybranie symbolu, którym chce grać. Program spodziewa się na wejściu symbolu o typie Character.

```
Put_Line("Choose your sign, press 'x' or 'o' ");
Get(playerSign);
```

```
micha1@DESKTOP-9CR4LVE:/mnt/c/Users/micha/Desktop/studia/PWiR/PWiR/projekt$ ./main
Tic-Tac-Toe!

Choose your sign, press 'x' or 'o'
```

W sytuacji gdy użytkownik wybierze inny symbol niż 'o' bądź 'x', zostanie poproszony o to ponownie.

```
while playerSign /= 'x' and playerSign /= 'o' loop
  Put_Line("Wrong character, press 'x' or 'o' ");
  Get(playerSign);
```

```
micha1@DESKTOP-9CR4LVE:/mnt/c/Users/micha/Desktop/Tic-Tac-Toe!
Choose your sign, press 'x' or 'o'
'b
Wrong character, press 'x' or 'o'
l
Wrong character, press 'x' or 'o'
```

Następnie gracz jest proszony o wybór pozycji dla nowowstawianego elementu. Zostaje poproszony o podanie numeru wiersza oraz kolumny. Program spodziewa się na wejściu typu Character.

```
begin
    Put_Line("Enter the line number: ");
    Get(Tmp);

    Row := char2int(Tmp);
    Put_Line("Enter the column number: ");
    Get(Tmp);
    Column := char2int(Tmp);
```

```
| |
+---+
| |
+---+
| |
+---+
Enter the line number:
2
Enter the column number:
```

Również w tym przypadku - jeżeli input miał zły typ, wychodził poza zakres lub gdy dana pozycja była już zajęta, gracz zostaje ponownie poproszny o podanie numeru wiersza i kolumny.

```
while isNotPossibleChoice loop
    if Row<1 or Row>3 or Column<1 or Column>3 then
        Put_Line("Enter the correct line number: ");
        Get(Tmp);
        Row := char2int(Tmp);
        Put_Line("Enter the correct column number: ");
        Get(Tmp);
        Column := char2int(Tmp);
    elsif Board(Row,Column) /= ' ' then ---other condition because otherwise program was exited
        Put_Line("Enter the correct line number: ");
        Get(Tmp);
        Row := char2int(Tmp);
        Put_Line("Enter the correct column number: ");
        Get(Tmp);
        Column := char2int(Tmp);
```

Gdy gracz wybierze miejsce dla swojego pionka i to miejsce jest puste, jest tam umieszczany jego symbol. Następnie komputer losowo wybiera nowe położenie dla swojego symbolu, również pod warunkiem, że jest ono puste i w kolejnej rundzie ruch znowu należy do naszego gracza.

```
o |   |
---+---+
| x |
---+---+
|   |
Enter the line number:
1
Enter the column number:
1
Enter the correct line number:
-
```

8 Testy

W zależności od podejmowanych decyzji grę można wygrać, przegrać lub zremisować.

– wygrana

```
o | x | o
---+---+
| x |
---+---+
| x |
Congratulations. You win!!!
michal@DESKTOP-9CR4LVE:/mnt/c/Users
```

– przegrana

```
x | x | o
---+---+-
o | o | o
---+---+-
x | x |
You lost!!!
michal@DESKTOP-9CR4LVE:/mn
```

– remis

```
x | o | x
---+---+-
o | x | x
---+---+-
o | x | o
It's draw!
michal@DESKTOP-9CR4LVE:/mn
```

9 Możliwe rozszerzenia programu

Ewentulne rozszerzenie programu mogłyby zmierzać w kierunku:

- ustalania rozmiaru planszy przez gracza
- dodania najbardziej optymalnych strategii, którymi powinien kierować się komputer przy wyborze nowego pola dla pionka

10 Ograniczenia programu

Użyte w programie ANSI escape sequences mogą nie być wspierane przez środowisko Windows.